

МІЖРЕГІОНАЛЬНА  
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МАУП

**Л. С. Ямпольський, Б. П. Ткач,  
О. І. Лісовиченко**

**СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ  
В ПЛАНУВАННІ, МОДЕЛЮВАННІ  
ТА УПРАВЛІННІ**

*Затверджено  
Міністерством освіти і науки України  
як підручник  
для студентів вищих навчальних закладів*

МАУП

Київ  
ДП «Видавничий дім «Персонал»  
2011

ББК 32.813я73  
Я57

Рецензенти: *А. О. Мельник*, д-р техн. наук, проф.  
*О. Г. Руденко*, д-р техн. наук, проф.  
*В. О. Кузьменко*, д-р техн. наук, проф.

*Схвалено Вченою радою Міжрегіональної Академії управління персоналом  
(протокол № 2 від 24.02.2009)*

*Затверджено Міністерством освіти і науки України  
(лист № 1/11-3212 від 16.04.2010)*

*Авторські права на Комплекс зареєстровані Державним департаментом  
інтелектуальної власності Міністерства освіти і науки України  
(Свідоцтво № 32608 від 01.04.2010)*

**Ямпольський Л. С.**

Я57 Системи штучного інтелекту в плануванні, моделюванні та управлінні :  
підруч. для студ. вищ. навч. закл. / Л. С. Ямпольський, Б. П. Ткач, О. І. Лісови-  
ченко. — К. : ДП «Вид. дім «Персонал», 2011. — 544 с. : іл. — Бібліогр. : с. 523–  
532.

ISBN 978-966-608-961-1

Підручник є продовженням циклу розробленої авторами вперше в Україні навчальної літе-  
ратури нового покоління з широким впровадженням сучасних інформаційних технологій в елек-  
тронній версії, виконаній в PDF-форматі, призначеній для Internet-розповсюдження, оснащеної  
тестуючими та пошуковими системами і лабораторними практикумами, і супроводжується мультимедійною анімацією складних для сприйняття процесів функціонування технологічних та ор-  
ганізаційно-технічних систем.

Для студентів, викладачів, аспірантів і науковців, які займаються проблемою штучного інте-  
лекту.

**ББК 32.813я73**

© Л. С. Ямпольський, Б. П. Ткач, О. І. Лісовиченко, 2011

© Міжрегіональна Академія управління персоналом  
(МАУП), 2011

© ДП «Видавничий дім «Персонал», 2011

ISBN 978-966-608-961-1

*У світі є два типи людей: ті, хто все робить,  
і ті, хто отримує за це визнання.  
Якщо можете, спробуйте стати людиною першого типу —  
тут набагато менша конкуренція.*

Дуайт Морроу

## ПЕРЕДМОВА ЗАГАЛЬНА

*Для справжнього вченого важливіше творити,  
аніж бути зрозумілим.*

Генрі Дейвід Торо

**Актуальність створення гнучких комп'ютерно-інтегрованих комплексів навчання.** Еволюція автоматизованих виробничих, організаційно-технічних, технологічних, економічних та інших систем веде до створення сучасних Комп'ютерно-Інтегрованих Комплексів Засобів Дистанційно-Віртуального Навчання (КІКЗ ДВН) (рис. 1П) з використанням комп'ютерних технологій відображення і подання інформації процесу навчання. На сьогодні майже не існує розроблених КІКЗ ДВН, які б задовольняли вимоги подання матеріалу з урахуванням індивідуальних особливостей студентів у вивченні матеріалу, наявності систем контролю знань з метою підвищення креативного мислення, а також систем розширеного пошуку інформації з можливістю її отримання не тільки за ключовими словами.

**Означення ПЗ.1.** *Дистанційно-віртуальне навчання* — це навчання, яке виконується з використанням технологічних режимів клієнт-сервер (на віддалених ресурсах Internet-середовища) і клієнт-клієнт (на необхідних ресурсах клієнтської машини) та *організоване* таким чином, що користувач може розглядати КІКЗ ДВН як єдине велике операційне середовище, що суттєво спрощує процедуру проходження етапів навчання.

*Метою створення КІКЗ ДВН* є забезпечення користувача дружнім інтерфейсом і багатофункціональним середовищем навчально-методично-наукової інформації про певну проблемну галузь. Дружній інтерфейс підвищує інтерес користувача до використання КІКЗ ДВН у процесі навчання за рахунок впровадження мультимедійних та інтернет-технологій, а також удосконаленого середовища подання матеріалу і пошуку інформації. Багатофункціональне середовище дає можливість вивчати теоретичний матеріал, виконувати лабораторно-дослідні практикуми ієрархічного за структурою та пріоритетністю матеріалу і проектно-розрахункові роботи, здійснювати самоконтроль користувача та атестацію з боку викладача.

Реалізація КІКЗ ДВН у такому обсязі можливостей потребує семантично узгоджених інструментальних засобів, на основі яких функціонують і синхронізуються окремі складові.

**Проблемно-орієнтований КІКЗ ДВН.** При створенні КІКЗ ДВН в основу було покладено його реалізацію за окремими напрямками знань, об'єднаними навчальним планом у відповідні напрямки підготовки студентів (наприклад, проектно-технологічний, схемотехнічний тощо).

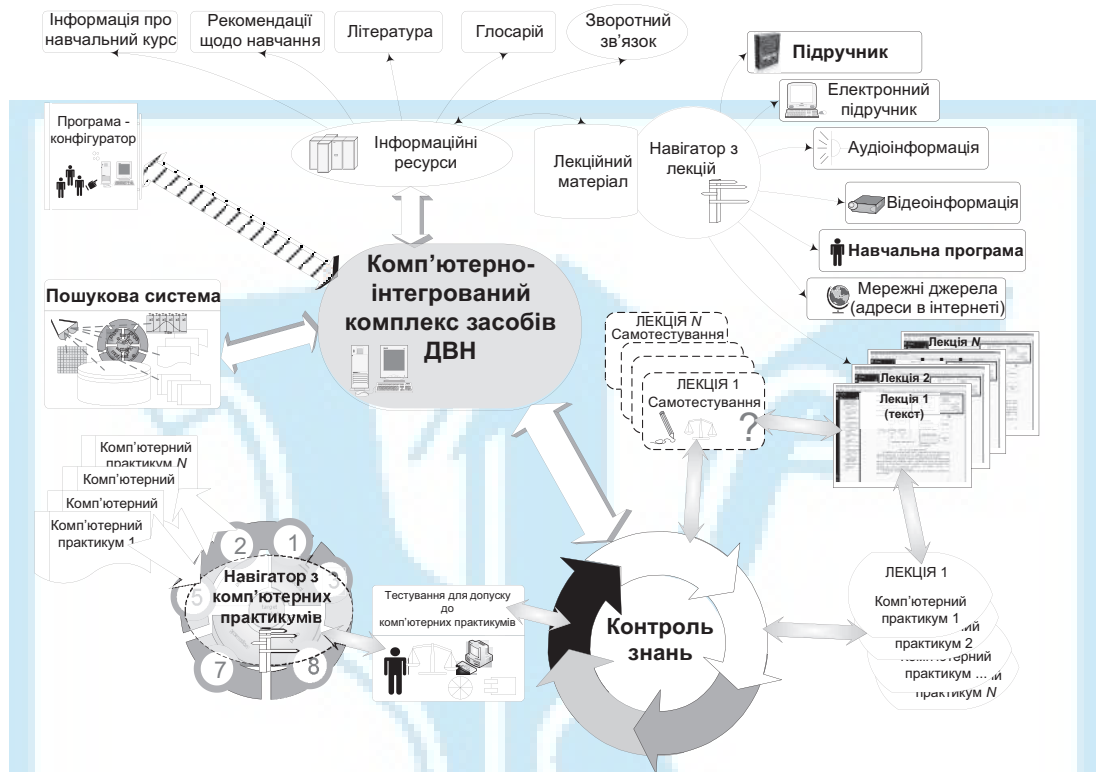


Рис. 1П. Структура КІКЗ ДВН

У підручнику запропоновано підхід до складу та змісту, який за допомогою КІКЗ ДВН (рис. 2П) забезпечує:

- **по-перше**, використання матеріалу у навчальному процесі одразу для декількох спеціальностей;
- **по-друге**, подання всього матеріалу про властивості, підходи щодо створення і застосування систем штучного інтелекту (СШІ) та їхніх інтелектуалізованих версій у логічній послідовності його викладення і взаємозв'язку складових підручника без повторень і дублювання навчального матеріалу;
- **по-третє**, подання матеріалу про СШІ таким чином, щоб кожна його складова мала самостійне значення і давала змогу забезпечувати навчальний процес інших споріднених спеціальностей навчально-методичним матеріалом, коли окрема частина підручника (його складові й окремі її частини) відповідає вимогам методичного і змістового забезпечення конкретного теоретичного курсу з відповідної навчальної дисципліни;
- **по-четверте**, поєднання на звичайному (паперовому) та електронному носіях, причому перший призначений для лекційного викладення курсу з поданням ключових вузлових аспектів проблемної галузі, тоді як електронна версія охоплює ширше коло питань щодо навчальної дисципліни з детальним її аналізом і призначена для студентів з урахуванням їх самостійної роботи з відповідною навчально-методичною літературою;

- **по-н'яте**, супроводження електронної версії підручника мультимедійною анімацією складних для сприймання студентом деяких елементів функціонування систем, поданих рисунком у статичному вигляді з текстовою формою опису принципу дії; кольорове анімаційне «оживлення» статичних зображень елементів гнучких комп'ютерно-інтегрованих систем (ГКІС) та їх інтелектуалізованих версій дає змогу користувачеві дослідити всі необхідні для сприймання деталі функціонування різноманітних за складністю, фізичною основою та послідовністю дії компонентів як під час їх експлуатації, так і в процесі проектування, моделювання та управління з використанням технологій штучного інтелекту;
- **по-шосте**, оснащення електронної версії підручника підсистемами контролю залишкових знань і тестування (входять окремими складовими в КІКЗ ДВН), у результаті користування якими студенти мають можливість виконати самоаналіз глибини засвоєння навчального матеріалу.

Потрібно зазначити, що створення підручника з такими властивостями для проектно-технологічного та спеціального напрямку підготовки студентів означених та інших споріднених через базу (бакалаврську) підготовку спеціальностей є актуальним внаслідок незабезпечення україномовними підручниками, тим більше – їх інтелектуалізованими електронними версіями. Крім того, у вітчизняній практиці немає підручників з комплексним поданням дисциплін професійно-орієнтованої та спеціальної підготовки, а також підручників, електронні версії яких оснащені мультимедійним анімаційним супроводженням і тестуючими системами\*.



### ВЛАСТИВОСТІ КОМП'ЮТЕРНО-ІНТЕГРОВАНОГО КОМПЛЕКСУ ЗАСОБІВ ДВН

- Паперовий і розширений електронний варіанти
- Сучасні технології подання навчального матеріалу
- Мультимедійне подання графічного матеріалу
- Інтерактивний інтерфейс у процесі дистанційного навчання
- Ефективна самонавчаюча пошукова система
- Вкладені лабораторні практикуми
- Автоматизоване оцінювання знань студента викладачем
- Самоконтроль залишкових знань вкладеними тестуючими системами

Рис. 2П. Властивості КІКЗ ДВН

\* Першим в Україні мультимедійним виданням був комплекс підручників: Гнучкі комп'ютеризовані системи: проектування, моделювання і управління: Підручник / Л. С. Ямпольський, П. П. Мельничук, Б. Б. Самооткін, М. М. Поліщук, М. М. Ткач, К. Б. Остапченко, О. І. Лісовиченко. – Житомир: ЖДТУ, 2005. – 680 с.

У підручнику описано основи створення інтелектуалізованих комп'ютерно-інтегрованих комплексів (ІКІК) широкого призначення, системне забезпечення формування інтелектуальних систем, а також методи і принципи функціонування складових ІКІК.

Подано термінологію та основні поняття, що застосовуються в системах із штучним інтелектом, розглянуто поширені моделі подання знань у сучасних інтелектуалізованих системах і дано їх порівняльну характеристику. Наведено приклади реалізації умовних інтелектуальних систем підтримки рішень щодо моделювання та управління технологічними та організаційно-технічними системами. Розкрито особливості формування логічних виведень і прийняття рішень в умовах лінгвістичної невизначеності при гіперпросторовому моделюванні процесів у складних і різних за своїми фізичними, структурними та функціональними властивостями об'єктах управління. Викладено основи створення комбінованих і гібридних інтелектуальних систем.

Підручник проілюстрований численними прикладами реальних систем інтелектуальної підтримки при розв'язанні задач розпізнавання образів, планування підготовчої стадії виробництва, зокрема діяльності інженерно-конструкторських і технологічних підрозділів, а також управління робототехнічними комплексами (РТК).

**Вибір принципів реалізації системи тестування.** Існує досить багато систем тестування (СТ), які використовують різноманітні алгоритми проходження тесту. В КІКЗ ДВН було запропоновано дві тестові підсистеми: самоконтролю користувача, яка безпосередньо вбудована в КІКЗ ДВН, та перевірки знань викладачем як окрема складова.

В СТ враховано такі функціональні вимоги:

- наявність ефективних засобів розробки тестів та адміністрування;
- наочність і простота процесу самоперевірки залишкових знань;
- інформативність;
- стабільність і безвідмовність роботи.

Оскільки підручник створювався для забезпечення можливості і дистанційного навчання, то основним засобом подання матеріалів є формати PDF та HTML. Враховуючи цю особливість, СТ інтегрована в КІКЗ ДВН і може завантажуватися як з робочого місця користувача, так і через Інтернет.

**Підсистема самоконтролю** складається з модуля підготовки тесту, який використовується для можливості формування ефективної бази даних (БД) тестових запитань з варіантами відповідей, та html-файлів з елементами JavaScript для відображення в головному фреймі підручника тестових запитань до відповідних його розділів, які використовуються для подання тесту та відображення результату користувачеві.

З існуючих найпоширеніших реалізацій модуля підготовки тесту таких систем за технологіями клієнт-сервер (БД зберігається на віддаленому ресурсі) та клієнт-клієнт (вся необхідна інформація розміщується на клієнтській машині) в КІКЗ ДВН застосовано другий варіант, який дає змогу завантажувати систему на локальному комп'ютері, не приєднаному до мережі.

Наочність і простота забезпечуються гнучкою структурою і різновидами форм побудови тестів, а інформативність підтримується контекстними підказками під час проходження тесту.

**Підсистема перевірки знань** створена за технологією клієнт-сервер з огляду на вимогу компактності та легкого настроювання на функціонування. Означені вимоги реалізуються за рахунок створення модуля підсистеми, який функціонує в режимах адміністратора та звичайного користувача з наданням таких можливостей:

- проходження тесту студентом за обраною тематикою;
- адміністративні дії викладача, пов'язані з веденням БД тестів і студентів;
- отримання звітів з рейтинговими оцінками за результатами тестувань;
- активний моніторинг процесу проходження тестування.

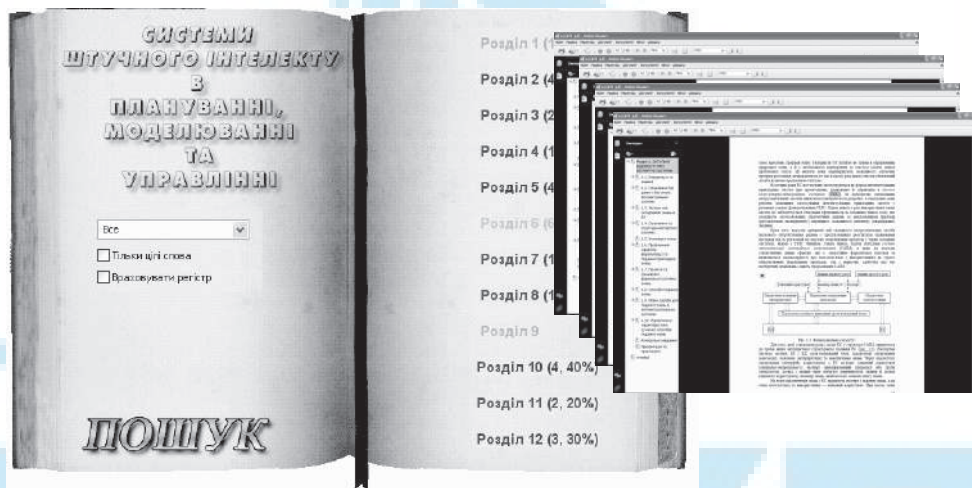
Особливістю реалізації підсистеми є можливість відновлення працездатності й оновлення інформації проходження тесту в разі виникнення нештатних ситуацій у функціонуванні (зависання і раптове відключення комп'ютера), що досягається такою реалізацією структури БД, в яку оперативно вноситься інформація про кожну відповідь користувача.

**Особливості реалізації пошукової системи.** Система пошуку інформації (СПІ) (рис. 3П) для цього КІКЗ ДВН розроблялася з урахуванням можливості реалізації таких задач:

- адаптації СПІ до будь-якого локального ресурсу;
- автоматичного оновлення БД виявлених ресурсів;
- формування критеріїв пошуку з урахуванням різних видів (текст, малюнки, анімація, відео, таблиці тощо) подання інформації.

Оскільки ефективність будь-якої СПІ визначається швидкістю пошуку та якістю фільтрації даних з урахуванням їх видів, у КІКЗ ДВН використовується структурованість за видами інформації та її індексація за ключовими словами в кожному

### Пошукова система для комп'ютерно-інтегрованого комплексу засобів ДВН



- Легка адаптація під автоматизоване учбове місце різних напрямків
- Автоматизоване оновлення бази даних ключових слів
- Пошук інформації різноманітного типу
- Обробка слів з різним мовним вкладенням
- Зручне подання результату пошуку
- Комбіноване використання різних параметрів пошуку

Рис. 3П. Властивості пошукової системи для КІКЗ ДВН

з видів. Крім того, результати в цій реалізації СПІ дають змогу подавати знайдені ресурси згідно з оцінкою входження пошукової інформації в кожний знайдений ресурс у спадному порядку.

Додаткові можливості розширеного пошуку забезпечуються використанням БД синонімів ключових слів, що дає змогу вживати як запити скорочені ключові слова та абрєвіатури.

Отже, результатом розробки одного з перших у нашій державі системних проєктів КІКЗ ДВН є створення зручного за інтерфейсом і багатофункціонального за призначенням робочого місця студента і викладача, яке, враховуючи реалізовані властивості, дає можливість використовувати його як базову оболонку навчаючої системи для різних напрямків підготовки в системі освіти.

Комп'ютерний набір підручника у PDF-форматі, створення мультимедійного анімаційного супроводу графічного матеріалу та демонстраційних прикладів презентацій, лабораторних практикумів, а також тестуючих систем виконано під керівництвом Л. С. Ямпольського та О. І. Лісовиченка магістрантами А. М. Дзінько та Р. І. Дзіньком за участі студентів 4–5 курсів напряму підготовки 0605 «Системна інженерія» Національного технічного університету України «КПІ», яким автори висловлюють щире подяку.

Публікації книги видавничою базою МАУП автори зобов'язані рекомендаціям Вченої ради цієї академії. Автори дякують професорам А. О. Мельнику, О. Г. Руденку, В. О. Кузьменку за копiтку наукове рецензування підручника і висловлені цiнні зауваження та поради, які були враховані при його доопрацюванні і сприяли підвищенню наукового рівня і якості тексту.



## СПИСОК СКОРОЧЕНЬ

*Пишіть коротко і неясно.*

Наполеон 1 Бонапарт

AgC	— агрегатний стан
AIM	— автоматизована імітаційна модель
АлгІМ	— алгоритм імітаційної моделі
АП	— асинхронний протокол
АРМ	— автоматизоване робоче місце
АРТС	— автоматизована розподілена транспортна система
АС	— агрегатний стан
АСк	— автоматизований склад
АТСС	— автоматизована транспортно-складська система
АФ	— активаційна функція
БД	— база даних
БЗ	— база знань
БП	— база правил
ВП	— виробничий процес
ВС	— виробнича система
ГАВ	— гнучке автоматизоване виробництво
ГВМ	— гнучкий виробничий модуль
ГВС	— гнучка виробнича система
ГІММК	— гнучка інтелектуалізована мультимодульна конфігурація
ГІС	— гібридна інтелектуальна система
ГКІС	— гнучка комп'ютерно-інтегрована система
ГПР	— граф потреб ресурсів
ГРЗ	— граф розподілу заявок
ГСП	— гнучка сітка Петрі
ДЗ	— джерело знань
ДО	— дошка оголошень
ДПС	— дискретно-подійна система
ЕКМ	— елементарна координатна модель
ЕкО	— експертне оцінювання
ЕО	— елементарний об'єкт
ЕОМ	— електронно-обчислювальна машина
ЕОТ	— електронно-обчислювальна техніка
ЕРК	— електроробочар
ЕРС	— електрорушійна сила
ЕС	— експертна система
ЗП	— захватний пристрій
ІВК	— інженерно-виробничий комплекс
ІВС	— інтегрована виробнича система
ІК	— інтерфейс користувача
ІКІК	— інтелектуалізований комп'ютерно-інтегрований комплекс
ІМ	— імітаційна модель
ІНС	— інтелектуалізована нейросітка

ІНСП	— ієрархічна навантажена сітка Петрі
ІНФС	— інтегрована нейро-фаззі-система
ІПСП	— ієрархічно-пофарбована сітка Петрі
ІР	— інтелектуалізований робот
ІРСТБ	— ієрархічно-розфарбована сітка з типових блоків
ІРСП	— ієрархічно-розфарбована сітка Петрі
ІСП	— інгібіторна сітка Петрі
ІС	— інтелектуальна система
ІСРЧ	— інтелектуальна система реального часу
ЙУЕ	— ймовірність успіху експерименту
КБ	— конструкторське бюро
КГІС	— комбіновано-гібридна інтелектуальна система
КД	— конструкторська документація
КЗОР	— коефіцієнт завантаження обробного ресурсу
КП	— керуючий процес
КПВ	— конструкторська підготовка виробництва
КФВ	— код функціональної відповідності
ЛЗ	— лінгвістична змінна
ЛСП	— логічна сітка Петрі
МГУА	— метод групового урахування аргументів
МЗЗ	— модуль здобуття знань
МКС	— маркер кольорів спрацьовування
МЛВ	— механізм логічних виведень
МММС	— математична модель маніпуляційної системи
МН	— міжопераційний нагромаджувач
МНК	— модуль накопичення
МПЕ	— модуль планування експериментів
МПЗ	— модель подання знань
МПП	— модуль порад і пояснень
МС	— маніпуляційна система
МСА	— метод скінченних автоматів
МСЕ	— метод скінченних елементів
МСП	— маркована сітка Петрі
МСПР	— маніпуляційна система промислового робота
НАГ	— надлишковий абстрактний граф
НКК	— нечітка керуюча компонента
НЛСП	— нечітка логічна сітка Петрі
НС	— нейронна сітка
НСП-модель	— навантажена СП-модель
НФС	— нейро-фаззі-система
ОбР	— обробний ресурс
ОВ	— об'єкт виробництва
ОІВ	— об'єкт інтелектуального вибору
ОК	— об'єкт керування
ОКП	— оперативно-календарне планування
ОМ	— об'єкт моделювання
ООПГ	— об'єктно-орієнтований принцип групування

## Список скорочень

ОР	–	об'єкт роботизації
ОУ	–	об'єкт управління
ПА	–	процесна алгебра
ПАР	–	просторово-агрегатна розподіленість
ПЗ	–	програмне забезпечення
ПР	–	просторово-ієрархічна розподіленість
ПЛСП	–	проста логічна сітка Петрі
ПМ	–	пошуковий модуль
ПМДН	–	пошуковий модуль динамічного навантаження
ПНФ	–	попередньо нормальна форма
ПОО	–	пошуковий образ об'єкта
ПП	–	правило продукції
ППОІ	–	пристрій первинної обробки інформації
ППР	–	просторово-подійна розподіленість
ППФ	–	правильно побудована формула
ПР	–	промисловий робот
ПРП	–	протокол розподілу потужностей
ПС	–	продукційна система
ПСВР	–	показник ступеня використання ресурсів
ПУПО	–	підсистема управління преселективною обробкою
ПЧГ	–	подійно-часовий граф
РК	–	розподілене керування
РСП	–	розфарбована сітка Петрі
РТК	–	робототехнічний комплекс
РТО	–	роботизована технологічна одиниця
РТП	–	роботизована технологічна позиція
РТС	–	робототехнічна система
РШ	–	робот-штабелер
САІМ	–	система автоматизованого імітаційного моделювання
САПР	–	система автоматизованого проектування
СБЗ	–	система баз знань
СВ	–	складальний вузол
СЗ	–	система знань
СК	–	сіткова конструкція
СкА	–	скінченний автомат
СкП	–	складальна позиція
СМАК	–	сіткова модель алгоритму керування
СМО	–	система масового обслуговування
СНСП	–	самонавантажувальна сітка Петрі
СП	–	сітка Петрі
СПЗ	–	система подання знань
СПІ	–	система пошуку інформації
СПЦ	–	сукупність перетинних циклів
СС	–	семантична сітка
СТ	–	система тестування
СТБ	–	система типових блоків
СТЗ	–	система технічного зору

СУБД	— система управління БД
СУКІС	— семантично-узгоджене комплексно-інтегроване середовище
СУС	— система упорядкування середовища
СШІ	— система штучного інтелекту
ТГВМ	— тестовий гнучкий виробничий модуль
ТЕП	— типовий елемент процесу
ТМ	— технологічний маршрут
ТМО	— теорія масового обслуговування
ТО	— технологічне обладнання
ТП	— технологічний процес
ТП ГКІС	— технологічний процес ГКІС
ТрМ	— транспортна мережа
ТрП	— транспортний пристрій
ТРСП	— типова розфарбована СП
ТС	— транспортний світлофор
ТСБ	— типовий сітковий блок
ТСС	— транспортно-складська система
ФК	— фаззі-контролер
ФН	— функція належності
ФНН	— функція нечіткої належності
ФС	— формальна система
ФСЗ	— формальна система знань
ФТМ	— формалізована технологічна мова
ЦС	— централізований склад
ЧСП	— часова сітка Петрі
ЧПУ	— числове програмне управління
ШІ	— штучний інтелект
ШНС	— штучна нейронна сітка
ШР	— штатний режим



## ПЕРЕДМОВА

*Досвід, певна річ, збільшує нашу мудрість,  
проте він не зменшує нашої глупоти.*

Бернард Шоу

Останніми десятиріччями на сторінках періодичної та монографічної літератури, присвяченій створенню нових інформаційних технологій, які призначені для полегшення діяльності людини як у її повсякденному побуті, так і у виробничій сфері, дедалі більше місце займають питання, пов'язані зі створенням штучного інтелекту (ШІ). Цей науковий напрям у першому наближенні характеризується розробкою математичних, а також лінгвістичних (у широкому розумінні цих слів) методів і технічних засобів, які імітують функції людського мозку.

Можна навести три основні аспекти, що сприяли розвитку зазначеного напрямку. Насамперед це *гносеологічний аспект*, який полягає в тенденції людини до пізнання самої себе, властивостей і можливостей свого інтелекту. Тут найважливішим для людини є створення математичної моделі самоусвідомлення для розв'язання задач самовдосконалення, якнайповнішого використання можливостей, закладених у людину природою. Тому відбувається оцінювання того, що стимулює людину до пізнання і є джерелом її прагнення до створення ШІ.

Другим є *інтенціональний застосовний аспект*, коли знання й уявлення можливостей людини дають змогу визначити сферу та міру їх використання. Тому в цьому аспекті ШІ спрямований на розвиток і доповнення обмежених можливостей людини (наприклад, людина краще розв'язує задачі одновимірного розпізнавання, тоді як електронна обчислювальна машина (ЕОМ) потенційно може розв'язувати задачі у вимірностях і постановках, недосяжних для людини, і не тільки з позицій швидкодії та обсягів обчислень, як у разі звичайного застосування ЕОМ, а вже й з позицій інтелектуалізації задач).

Третім є *екстенціональний застосовний аспект*, коли ЕОМ розв'язує задачу нехай і не краще за людину, але дає змогу поширити її досвід і знання для розв'язання значно більшої кількості задач, ніж це змогла б зробити реальна (обмежена) кількість експертів у цій галузі знань, яких для цього можна залучити. Це й є звичайною галуззю застосування експертних систем (ЕС), тобто тим рівнем інтелектуалізації, якого на сьогодні досягнуто у вирішенні проблеми створення ШІ в широкому використанні у виробничій діяльності людини.

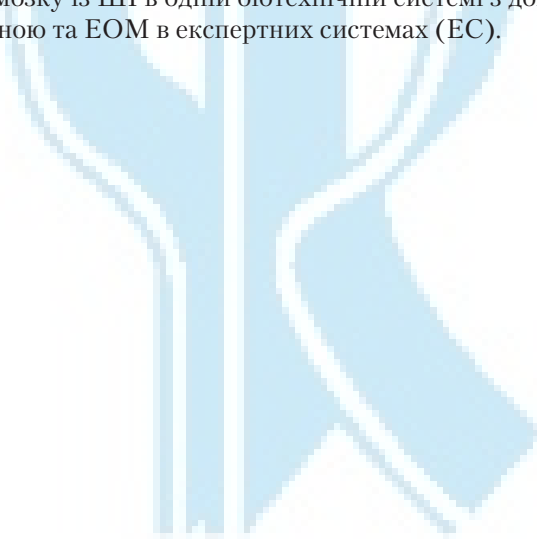
Все згадане супроводжується наявністю об'єктивної соціальної потреби у впровадженні елементів ШІ у виробництво. Справді, існує необхідність створення інтелектуальних технічних систем, які могли б сприймати зовнішню інформацію, орієнтуватися, переміщуватися у просторі, приймати рішення і, головне, *виконувати цілеспрямовані дії в неупорядкованому середовищі* (космосі, океані, пустелі, під землею тощо). Крім того, відчувається нагальна необхідність наблизити користувача до ЕОМ, забезпечивши між ними двосторонній обмін інформацією — діалог природною мовою.

Зрештою, є проблема розв'язування задач у складних, погано структурованих галузях, які важко піддаються формалізації. До таких галузей, зокрема, можна зарахувати процеси, що відбуваються в гнучких автоматизованих виробництвах (ГАВ) в усіх складових їх життєвого циклу — від науково-технічних прогнозувань щодо

сталості номенклатури виробів до їх виготовлення та вирішення проблем ефективного збуту готової продукції.

Відтепер є реальний шлях до розв'язання всіх цих найскладніших виробничих задач — це узагальнення та формалізація знань, якими володіють спеціалісти-експерти цього профілю, перетворення їх у машинну форму подання із залученням ЕОМ для формування обґрунтованих висновків. У подальшому можна сподіватись на процес самонавчання й удосконалення набутих знань самою машиною.

Отже, про ШІ ведуться досить гострі дискусії на сторінках періодичних видань, причому основне питання їх полягає в тому, чи можна створити технічну систему — штучний мозок, який повністю відтворював би функції природного людського мозку з його емоціями, творчістю, інтуїцією тощо. Це питання має технічний, психологічний, філософський і соціальний аспекти, і навряд чи можна сподіватись на однозначну відповідь на нього найближчими роками. Але немає жодного сумніву в тому, що проблематика ШІ вже давно перейшла із суто наукових досліджень у практичне русло, де одним з найпріоритетніших напрямів є суміщення можливостей людського мозку із ШІ в одній біотехнічній системі з доцільним розподілом функцій між людиною та ЕОМ в експертних системах (ЕС).



## ВСТУП

*Все має бути викладено так просто, як тільки можливо,  
але не простіше.*

Альберт Ейнштейн

*Інтелектуал — людина, яка потребує більше слів,  
аніж необхідно для того, щоб сказати більше, аніж вона знає.*

Дуайт Ейзенхауер

Виникнення ЕС в останні десятиріччя є важливим етапом розвитку ШІ [38–40; 119; 133]. Ці системи знайшли широке застосування в таких галузях, як медицина, геологія, юриспруденція, обчислювальна техніка, інформаційний пошук. За нашого часу ЕС починають інтенсивно розробляти у сфері промислового виробництва [91]. Але перш ніж з'ясувати причини, які зумовили увагу до ЕС, а також до проблематики їх створення, спробуємо дізнатися, що таке «інтелект». Зауважимо, що визначеного, чіткого поняття «інтелектуалізація», оцінки його рівня, якщо йдеться про заміну природного інтелекту людини «штучним інтелектом», який за певних умов використовуватиметься замість нього, досі не існує. Тому найчастіше ступінь інтелектуальності визначають на інтуїтивному рівні, наприклад, *методом експертних оцінок* (із залученням до оцінки репрезентативної групи експертів) і *методом тестування* (шляхом розв'язування тих чи інших тестових завдань). Отже, враховуючи сказане, скористаємось визначенням, яке, мабуть, теж не є повним, але достатнім для практичного використання.

**Означення В1.** *Інтелектом* називатимемо здатність мозку розв'язувати (інтелектуальні) задачі шляхом набуття інформації від зовнішнього середовища, перевірки її на достовірність і тлумачення як знань, упорядкування, накопичення та цілеспрямованого перетворення останніх у процесі навчання, що ґрунтується на досвіді й адаптації мозку до різноманітних обставин.

Отже, характерними рисами інтелекту, які виявляються в процесі розв'язування різних задач, є його здатність до навчання, узагальнення, накопичення досвіду (знань і навичок) та адаптації до змінюваних умов у процесі розв'язування конкретної задачі. Тому мозок, наділений інтелектом, є універсальним засобом розв'язування широкого кола задач (включаючи і неформалізовані), для яких немає стандартних, заздалегідь відомих методів розв'язання.

Тип задачі визначає наперед алгоритм її розв'язання. В чому полягає специфіка алгоритмів розв'язання інтелектуальних задач?

Розглянемо як приклад алгоритм розв'язування інтелектуальної задачі на доведення. Процес її розв'язування — результат інтелектуальної діяльності — зводиться до доведення, тобто до виконання послідовності логічних операцій або кроків, які починаються з умови (передумов) задачі й закінчуються теоремним висновком. При цьому кожний крок приводить до деякого нового положення, одержаного з відповідним чином дібраних окремих умов, відомих фактів (аксіом) або з раніше доведених (виведених) положень (лем). Аналогічно можна уявити собі й інші математичні задачі.

Таким чином, щоб розв'язати будь-яку нетривіальну задачу, необхідно скласти добре скоординовану, узгоджену схему операцій (логічних, математичних або прикладних), які починаються з умови (передумов) задачі й закінчуються висновком

(метою), послідовно ведучи від даних до невідомого, від об'єкта, який дуже добре відомий, до об'єктів, яких треба досягти.

Ураховуючи зазначене вище, а також ґрунтуючись на наведеному вище визначенні інтелекту, дамо визначення терміна «штучний інтелект» [114; 120].

**Означення В2.** У загальному розумінні *штучний інтелект* — це сукупність автоматичних методів і засобів цілеспрямованої переробки інформації (знань) відповідно до набутого в процесі навчання й адаптації досвіду при розв'язанні всіляких інтелектуальних задач.

Початок поширення цього терміна можна датувати кінцем 70-х — початком 80-х років ХХ ст., хоча такими задачами займалися фахівці з ШІ починаючи з 50-х років. Однак у той період загальноприйнята передумова полягала в необхідності створення насамперед дуже сильного механізму виведення (розв'язувача) задач за вторинності обсягу та якості знань, тобто за наявності обмеженого обсягу фактів та інформації за допомогою потужного механізму виведення передбачалося одержати якісні результати (розв'язок або знання). Зокрема, американські математики Н. Ньюел і Х. О. Саймон побудували *загальний розв'язувач задач* як універсальний засіб такого роду, де у вигляді таблиць «стани — дії» передбачалася можливість переходу з одного стану, що характеризується набором фактів (сукупністю інформації), в цільовий стан (рис. В.1, а). При пошуку розв'язку використовувалися також і вкладені евристики, які не відігравали, проте, основної ролі.

Цей підхід сприяв розвитку фундаментальних формальних основ ШІ, але не супроводжувався комерційним поширенням останніх, оскільки не було необхідної умови — соціального замовлення. Крім того, не виконувалась і достатня умова — не було тоді і потужних засобів технічної підтримки систем із ШІ на основі комунікаційних сіток (рис. В.1, б).

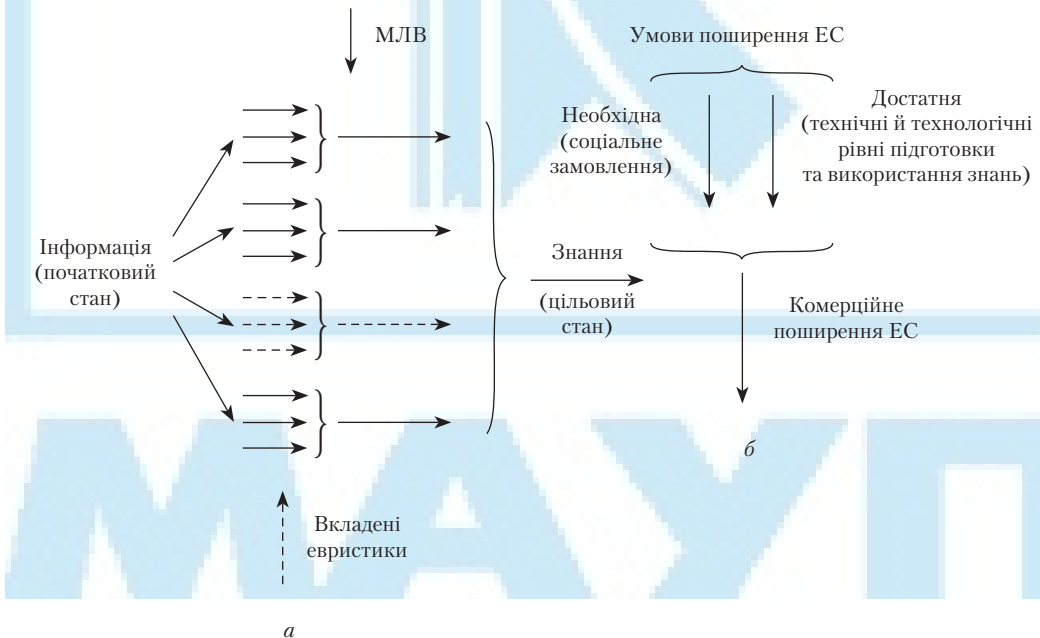


Рис. В.1. Схема загального розв'язування задач (а) та умови поширення ЕС (б)



У ході послідовного розвитку досліджень з ШІ відбувся їх розподіл на дві самостійні вітки. Цей розподіл зберігається й досі, що пов'язано з існуванням двох точок зору на питання про те, яким чином будувати системи з ШІ. Прихильники *однієї точки зору* переконані в тому, що найважливішим є результат, тобто добрий збіг поведінки штучно створених і природних інтелектуальних систем, а що стосується внутрішніх механізмів формування поведінки, то розробник ШІ зовсім не повинен копіювати чи навіть брати до уваги особливості природних, живих аналогів.

*Друга точка зору* полягає в тому, що власне вивчення механізмів природного мислення й аналіз даних про способи формування розумової поведінки людини можуть створити основу для побудови систем з ШІ, причому побудова ця має здійснюватись насамперед як моделювання, відтворення технічними засобами принципів і конкретних особливостей функціонування біологічних об'єктів.

Таким чином, *перша вітка досліджень* пов'язана з розглядом продукту інтелектуальної діяльності людини, вивченням його структури (з поділом всіляких проявів інтелектуальної діяльності — розв'язуванням задач, доведенням теорем, іграми тощо) і прагненням відтворити цей продукт засобами сучасної техніки, тобто ЕОМ. Інакше кажучи, ця вітка досліджень пов'язана з *феноменологічним та імітаційним моделюванням*. Якщо вдасться запрограмувати ЕОМ так, щоб вона успішно розв'язала конкретну задачу, то відповідний вид інтелектуальної діяльності можна вважати автоматизованим. Ясно, що успіхи цього напрямку досліджень з ШІ тісно пов'язані з розвитком ЕОМ і майстерністю програмування, тобто з комплексом науково-технічних досліджень — комп'ютерними науками. Цю вітку досліджень також часто називають *машинним інтелектом*.

*Друга вітка досліджень з ШІ* пов'язана з розглядом даних про нейрофізіологічні та психологічні механізми інтелектуальної діяльності, а в ширшому значенні — з розглядом розумової поведінки людини. Розробники прагнуть відтворити ці механізми за допомогою технічних пристроїв, щоб поведінка їх добре збігалася з поведінкою людини в певних, наперед заданих межах. Отже, ця вітка досліджень пов'язана насамперед зі *структурним моделюванням*. При позитивному вирішенні цієї проблеми вважають, що відповідний вид людської діяльності автоматизований. Ця вітка досліджень з ШІ дістала назву *штучного розуму*, а характерним тут є прагнення до відтворення ширшого, ніж у машинному інтелекті, спектра проявів розумової діяльності людини.

У спрощеному вигляді структуру основних напрямків, які існують у ШІ, зображено на рис. В.2.

Еволюція методології систем із ШІ є яскравим прикладом діалектичного розвитку. Тепер знову на порядку денному, але на якісно новому рівні, стоїть задача реалізації індуктивного підходу до побудови інтелектуальних систем, коли на основі наявних статистичних даних і фактів системою самостійно формуються гіпотези та виявляються закономірності взаємозв'язків з використанням як строго обґрунтованих формальних процедур, так і евристик, одержаних завдяки експертним оцінкам і сформованих у самій системі.

Таким чином, *ШІ — це складова частина інформатики, в якій створюються наукові й технічні передумови для розв'язання за допомогою систем обробки інформації задач, які до цього були пов'язані головним чином з людськими здібностями* (розуміння мови; розпізнавання образів; навчання, судження, вирішення проблем). Тепер інтелектуальні прикладні системи застосовуються в гнучкій автоматизації у вигляді ЕС для проектування, планування, управління, діагностування та навчання.

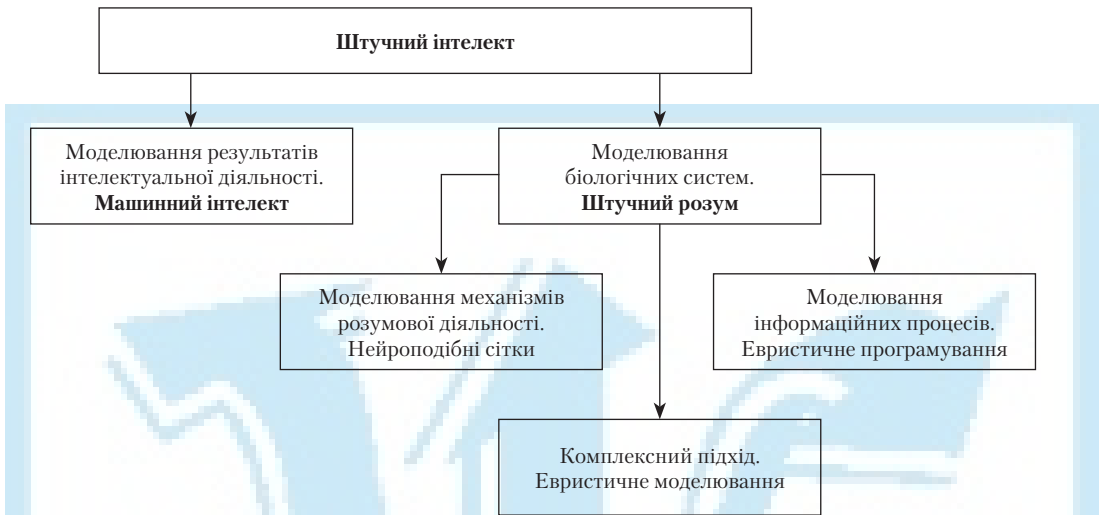


Рис. В.2. Основні напрямки досліджень у галузі штучного інтелекту

Можна виділити чотири основних напрями досліджень у галузі ШІ [89]:

1. **Моделювання на ЕОМ** окремих функцій творчих процесів (ігрові задачі, автоматичне доведення теорем, автоматичний синтез програм і алгоритмів тощо). Цей напрям почав розвиватися раніше від інших і породив термін ШІ.

2. **Зовнішня інтелектуалізація ЕОМ** (фундаментальні та прикладні дослідження, що належать до комплексного діалогового інтерфейсу). Це найбільш бурхливо розвинений і найбільш важливий напрям досліджень із ШІ. Інтелектуальний інтерфейс стрімко підвищує ефективність автоматизованих систем планування, технологічної підготовки виробництва, проектування, проведення наукових досліджень і оперативного управління виробництвом, а також автоматизованих систем керування завдяки тому, що інтелектуальний інтерфейс інтенсифікує роботу кінцевого користувача. Не виходячи за межі мови своєї предметної галузі, фахівці мають можливість: а) здійснювати зі свого робочого місця *пошук у базах даних/знань* (БД/БЗ) необхідної документальної та фактографічної інформації з доступом до бібліотечних мереж і мереж розподілених БД; б) *розв'язувати проектні, планові й управлінські задачі* за їх постановкою (описом) і вихідними даними незалежно від складності математичних моделей цих задач і контролювати в діалоговому режимі всі стадії обчислювального процесу; в) *використовуючи накопичені в ЕОМ знання про предметну галузь, здійснювати* розпізнавання та діагностику процесів у складних системах, приймати рішення, формулювати плани дій, висувати та перевіряти гіпотези, виявляти закономірності в результатах спостережень, робити логічний висновок. Перша можливість реалізується інтелектуальними пошуковими системами; друга — інтелектуальними пакетами прикладних програм і розрахунково-логічними системами (які є подальшим розвитком інтелектуальних пакетів прикладних програм); третя — ЕС, які почали інтенсивно поширюватись у галузях знань, що важко формалізуються (особливе значення мають гібридні ЕС, які є об'єднанням традиційних ЕС з розрахунково-логічними).

3. **Внутрішня інтелектуалізація ЕОМ**, пов'язана з вирішенням проблеми побудови ЕОМ нових поколінь, оскільки для задач ШІ важливими є ЕОМ і методи обробки символічної інформації, що потребує витрат майже 85–90 % ресурсів сучасних машин. У зв'язку з цим використання ЕОМ традиційної архітектури, навіть орієнтованих на розпаралелювання числових даних, виявляється малоефективним при розв'язуванні задач ШІ. Це спричинює появу машин БД і машин БЗ, Лісп-, Пролог-, Рефал-машин, лінгвістичних процесорів тощо.

4. **Створення інтелектуальних робіт**, що потребує розробки як спеціалізованих ЕОМ, так і цілого комплексу електромеханічних, енергетичних та інформаційних систем: сенсорів, двигунів, джерел енергії для локальних мехатронних компонентів. Як і в системах із ШІ, інтелектуальні роботи орієнтовані на знання, що формуються на основі потоків інформації про зовнішнє середовище та надходять до бортових ЕОМ (наприклад, рухомих роботів). Оскільки знання, що надходять до ЕОМ робота, ситуативні, потрібна обробка їх у реальному масштабі часу. Остання обставина потребує швидкодії ЕОМ у межах 10 млрд операцій за секунду (переважно для символічної обробки інформації).

Найпоширенішими проблемами інтелектуалізації є: *розпізнавання образів, мислення та обчислювальні задачі*.

Щодо **проблеми розпізнавання**, то в робототехніці вирізняють декілька відмінних процесів, які узагальнюються цим поняттям: *по-перше*, мають на меті класичну постановку *задачі ідентифікації*, коли необхідно вирізнити певний конкретний об'єкт серед подібних до нього (наприклад, впізнати серед інших виробничих об'єктів заготовки, деталі, напівфабрикати, оснастку, обробляючи ресурси тощо). Ця найтипівіша в робототехніці задача, на відміну від загальноживаного в теорії управління визначення й ототожнення структури деякої системи за вхідними та вихідними інформативними показниками, полягає саме в тому, що об'єкт роботизації (ОР), який спостерігається, вирізняється з-поміж інших за деякими характерними його властивостями; *по-друге*, наступною постає *задача класифікації*, коли виникає необхідність визначити належність ОР до одного із задалегідь відомих його класів (наприклад, належність до деталей форми тіла обертання, плоскої, коробчастої або одно- чи багатоступінчастої, з однією чи декількома осями, площинами симетрії тощо); *по-третє*, дуже поширеною і важливою для підготовки і здійснення роботизованого виробництва є *задача орієнтації*, коли визначається поточне положення ОР у просторі або відносно інших об'єктів виробництва і надалі формується алгоритм переведення ОР з поточного в наперед визначене і необхідне для ефективного проходження технологічного процесу. До речі, іншим різновидом цієї задачі є навігаційна, за якої автономні транспортні роботи — робокари здійснюють просторову орієнтацію в процесі транспортування ОР.

**Проблему моделювання мислення** найчастіше пов'язують з формулюванням наслідків з фактів, які безпосередньо спостерігаються або визначаються відомими, або, іншими словами, йдеться про *механізми логічних виведень* (МЛВ), які часто формуються на основі підсвідомого *інтуїтивного мислення* (природа якого ще недостатньо вивчена і дуже важко формалізується), а також *дедуктивної логіки* — формалізується законами логіки на основі конструкцій, які дістали назву **силогізмів** (передбачають виведення часткових наслідків із загальних правил). Проілюструємо останнє прикладом з робототехніки.

**Твердження 1.** *Усі промислові роботи призначені для виконання або обслуговування виробничих процесів.*

**Твердження 2.** Робот «PUMA» обслуговує і виконує складні (механоскладальні, зварювальні тощо) технологічні операції.

**Висновок:** Робот «PUMA» – промисловий робот.

Якщо обидва твердження у силі істинні та задовольняють загальні формальні вимоги, тоді й висновок є істинним незалежно від змісту тверджень, що входять до силіогізму. При порушенні цих формальних вимог продукуються логічно помилкові висновки, наприклад:

**Твердження 1.** Усі промислові роботи є переналаджуваними машинами, які виконують або обслуговують технологічні виробничі операції.

**Твердження 2.** Верстат мод. 16K20T1 є переналаджуваною машиною, яка виконує технологічні виробничі операції.

**Висновок:** Верстат мод. 16K20T1 – промисловий робот.

Як бачимо, висновок хибний.

Не вдаючись у детальний аналіз труднощів, які виникають у процесі формалізації та автоматизації логічних виведень, наведемо їхні спільні риси – **велику розмірність і погану формалізованість**, зокрема, надзвичайно актуальною є проблема **нечіткої логіки**. Реальні програми, що здійснюють аналогічні виведення, заведено називати **експертними системами**.

Нарешті, надзвичайно важливою розумовою компонентою є спроможність спілкуватися з природним або штучно створеним інтелектом. Рівень інтелекту в обох випадках можна визначити на основі **тестування**, так, як це відбувається у звичайному діалозі, скажімо, під час складання заліків або іспитів студентами викладачеві. Така перевірка набутих студентами знань тотожна своєрідному тестуванню. Проте якщо створюється навчальна тестуюча програма, найчастіше в такій системі студентові надається можливість використовувати ствердну або заперечну форму відповіді (за поширеною аналогією «так – ні») або ж вибирати правильну відповідь серед поданих у «вікні» декількох, у тому числі й неправильних відповідей. Такий підхід дає змогу принаймні визначити рівень знань студента на понятійному рівні, але недостатній для формування оцінки розуміння і засвоєння глибинних основ навчального матеріалу. Останнім часом набули поширення програми з використанням ідеї так званого **фатичного діалогу**, в основі якого лежить формальне перепарафразування почутого **без розуміння співрозмовника**, але системи, які здатні тільки на підтримку такого діалогу, не можна вважати інтелектуальними через те, що **для такого розуміння необхідно мати суму знань про предмет розмови**. Отже, при створенні псевдоінтелектуальних програм, здатних підтримувати фатичний діалог з людиною або іншими аналогічними програмами, вдаються до побудови алгоритмів зіставлення зі зразками. Як зразки для навчання системи можуть використовуватися вибірки з тематичних діалогів репрезентативної групи людей (наприклад, групи студентів) із застосуванням варіантів порівнянь: а) **з повним збігом**; б) **із замінювачами**; в) **з наданням значень змінним у процесі зіставлення**; г) **із зіставленням більше ніж з одним зразком** [21]. Прикладом такої псевдоінтелектуальної системи може бути діалогова система «Аліса».

Варто зазначити, що жодна з існуючих систем штучного інтелекту, які застосовуються у різних галузях сьогодні, не може реалізувати функції інтелектуальної системи в достатньому обсязі, тому правильніше говорити про них як про **інтелектуалізовані системи** з певними рисами, які наближають їх до інтелектуальних.

Як уже зазначалося, останніми роками особлива увага приділялася створенню ЕС, призначених для розв'язання задач експертного оцінювання ситуацій в різних

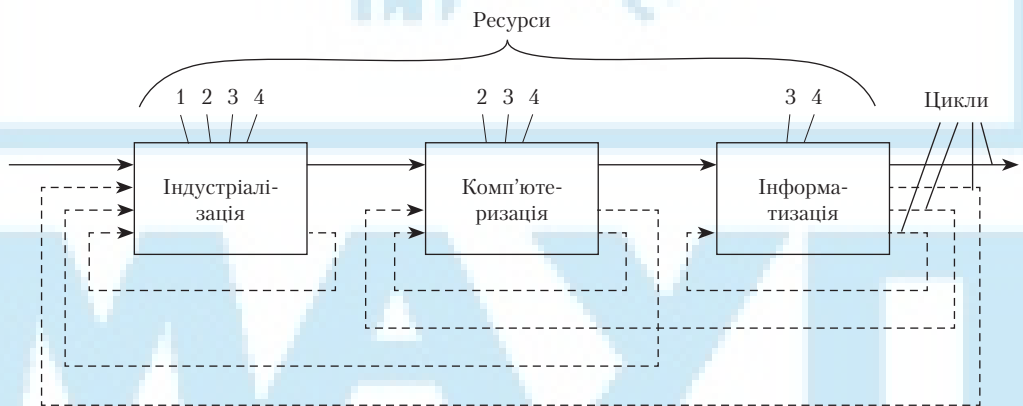
предметних галузях. Щоб зрозуміти місце й роль ЕС у сучасному суспільному розвитку, треба зробити невеликий екскурс в історію еволюції науково-технічного прогресу за останні, скажімо, 50 років. У ході суспільного розвитку цей період найповніше характеризують три стадії — індустріалізація, комп'ютеризація та інформатизація (рис. В.3).

Стадію *індустріалізації* (пріоритети в якій віддавались паливно-енергетичним ресурсам, матеріалознавству, потім — розвитку технології та інтелекту) пройшли всі промислово розвинені країни. Як результат були закладені підвалини для розвитку й удосконалення елементної бази засобів обчислювальної техніки та створення на цій основі ЕОМ.

Стадію *комп'ютеризації* (тут пріоритети віддавались розвитку матеріалознавства, технології виробництва ЕОМ та інтелектуальним ресурсами) пройшли найбільш розвинені країни Заходу, а також окремі країни третього світу, держави ж Східної Європи (так само й Україна) нещодавно теж вступили в цю фазу розвитку, але відставання їх від США та Японії за обсягом і якістю парку ЕОМ (особливо суперЕОМ і персональних ЕОМ), розвитком комунікаційних систем сягає загрозливої дистанції.

У загальному розумінні *інформатизація*, коли пріоритети віддаються інтелекту й інформаційним технологіям, характеризує перехід суспільства в новий якісний стан — інформаційне суспільство. Останнє вирізняється бурхливим розвитком космонавтики, ядерної енергетики та робототехніки з якісно новим технологічним забезпеченням, яке ґрунтується на комп'ютеризованих засобах праці й інформаційних технологіях і потребує принципово нового інформаційного середовища — сукупності ЕОМ, систем комп'ютерних комунікацій, БД і БЗ, а також програмних комплексів. Послідовне здійснення інформатизації робить реальними ЕС, системи підготовки та підтримки рішень, системи з ШІ.

При побудові ЕС постає запитання: які знання вони повинні відображати та в якій формі? Структура знань, що залежить від сфери їх застосування, містить різні факти з предметної галузі, взаємозв'язки між ними, правила дій та ін. Вона має містити також знання, що стосуються способу включення знань в ЕС. Складність і подібність структур знань спричинили появу кількох різних способів їх подання,



**Рис. В.3. Еволюційні стадії суспільного розвитку**

(1 — паливно-енергетичні; 2 — матеріали; 3 — технологічні; 4 — інтелектуальні)

з яких слід виділити логічну модель, фреймову, продукційні системи, семантичні (СС) та нейронні (НС) сітки, а також сітки Петрі (СП). Кожен із цих способів має свої переваги та недоліки і тяжіє до певної структури знань. Останніми роками дедалі частіше використовуються моделі подання знань, які об'єднують згадані вище способи.

Особливість *систем подання знань* (СПЗ) полягає в моделюванні діяльності людини, яка часто здійснюється в неформальному вигляді. Якщо, скажімо, методи розв'язування обчислювальних задач ґрунтуються на чітких алгоритмах, побудованих з використанням поняття збігу, то в основі моделей подання знань лежить інформація, одержана від експертів, яка часто має якісний і до того ж суперечливий характер. Однак, виходячи зі специфіки функціонування ЕОМ, така інформація повинна бути зведена до однозначного формального вигляду, що здійснюється завдяки ідеям багатозначної логіки, теорії нечітких множин та аналогічних математичних моделей.

За нашого часу моделі подання знань є предметом досліджень і розробок вузького кола фахівців (насамперед програмістів і математиків), тоді як потреба в таких моделях відчувається практично в усіх предметних галузях. Це пояснюється необхідністю побудови ЕС, структура яких нерозривно пов'язана з формами подання знань, які, у свою чергу, визначаються особливостями предметної галузі. Такий стан справ посилюється станом публікацій з цієї теми, кількість яких щодо методів подання та використання знань не перевищує в кращому випадку десятків, а літератури з цього питання, придатної для навчального процесу, практично немає. Ця книга є одним з перших україномовних видань з проблем\* ШІ, що створювало методологічні та термінологічні труднощі при її поданні як підручника. В ньому наведено короткий огляд деяких відомих ЕС, який має в основному ілюстративний і навчальний характер.

Важливу роль відіграють також приклади, що пояснюють структури моделей, правила виведення та ін. Це робить книгу дуже корисною не тільки для студентів, а й для користувачів, які не є фахівцями в галузі ЕС, але зацікавлені в їх використанні.

Водночас розглядаються й численні приклади реалізації систем планування, моделювання та управління, в яких механізм прийняття рішень ґрунтується на засадах штучного інтелекту на основі наведених вище способів подання знань: логічних моделях, фреймових і продукційних системах, СС і НС, на принципах функціонування класичних СП та їх розширень. Так, у структурі підручника значна роль відводиться новітнім модифікаціям класичного апарату СС і шляхам створення сучасних моделей складних просторово-розподілених дискретно-подійних систем технологічного та організаційно-технічного призначення на цій основі.

У підручнику наведено детальний аналіз основних топологій НС, в тому числі розглядаються можливості створення нейроконтролерів та використання найрофаззі-технологій для розв'язання задач розпізнавання образів, контролю та керування технологічними процесами в умовах невизначеності, в тому числі у випадках, які характеризуються надзвичайними умовами функціонування обробляючих ресурсів у складних комп'ютерно-інтегрованих системах, де необхідна обробка не-

\* Першим україномовним підручником з проблем штучного інтелекту є видання: *Ямпольський Л. С., Лавров О. А.* Штучний інтелект у плануванні та управлінні виробництвом: Підручник. — К.: Вища шк., 1995.

чіткої інформації, відображення суттєво нелінійних багатопараметричних об'єктів, які важко формалізуються.

Нарешті, вперше в навчальній літературі наведено матеріал, який стосується процесних алгебр (ПА), які не призначені для опису чи подання будь-чого на певному рівні ієрархічного моделювання у ГКІС. Використання цього апарату забезпечує композиційність у формально-математичному, а не тільки в інтуїтивному змісті (тобто надається можливість застосовувати математичні операції на рівні опису системи із застосуванням знаків, символів тощо, перетворювати вирази за певними правилами, як це робиться в алгебрі взагалі, але в цьому випадку кожний вираз визначає систему, а точніше процес, що нею реалізується). На відміну від традиційної теорії автоматів, ПА більшою мірою спроможні подавати паралелізм, комунікаційність тощо. Що ж до СП, то порівняно з ними ПА краще забезпечують зручний синтаксис, проте гірше подають паралелізм. Можна стверджувати, що ПА спрямовані своїм змістом на моделювання часу та синхронізацію і різняться своїми семантичними та синтаксичними характеристиками.

У підручнику не ставиться мета розглянути всі існуючі ПА з їх відмінностями, перевагами та недоліками, а робиться спроба подати деяку стандартну ПА для виявлення основних характеристик цього формального апарату при застосуванні до моделювання взаємодії компонентів комп'ютерно-інтегрованих виробничих систем. З означеної специфіки задачі, що розглядається, та характеристик апаратів моделювання ми вважали за необхідне зупинитися на ПА через те, що вона, по-перше, є найменше дослідженою з позицій застосування до задач інтелектуалізованого моделювання процесів у ГКІС, а по-друге, властивості цього апарату є привабливими для подання важливих аспектів функціонування складних систем і, таким чином, являють собою своєрідну модель подання знань про технологічне середовище.

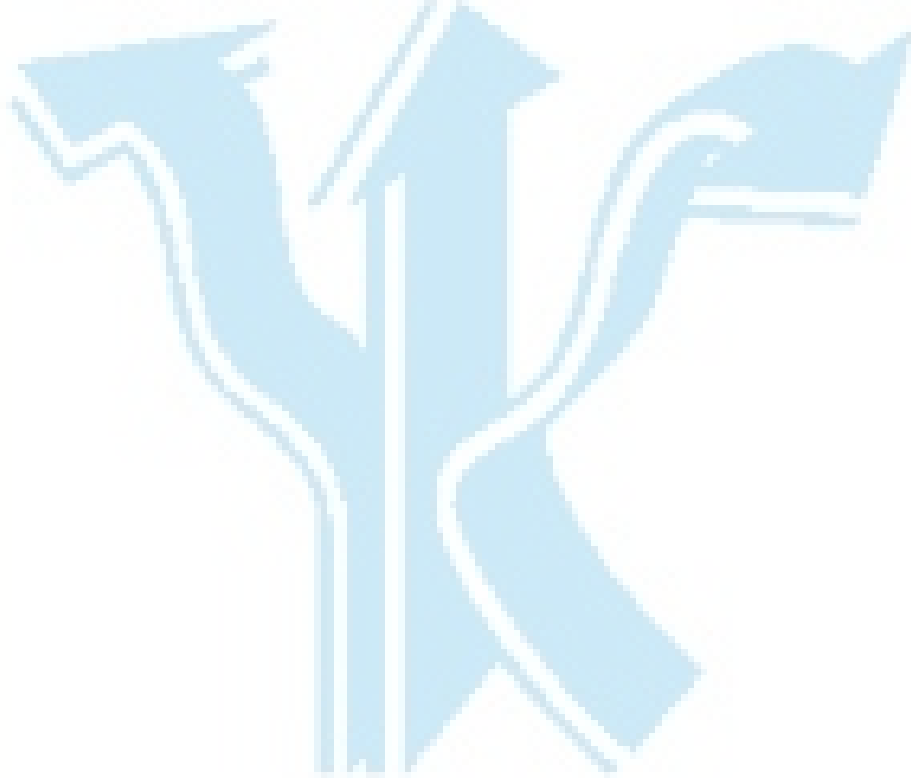
Незважаючи на збільшення кількості прикладів практичного застосування нових інформаційних технологій, напрямок, пов'язаний із створенням інтелектуальних (і навіть інтелектуалізованих) систем все ж таки належить більшою мірою майбутньому, аніж сьогоденню. Більшість створених і створюваних систем мають поки що експериментальний характер і на цьому етапі далекі від досконалості. Нові інформаційні технології, як і інші напрямки науки і техніки, які живлять розробки в галузі штучного інтелекту, перебувають у стадії становлення та інтенсивного розвитку досліджень (в тому числі фундаментальних) як щодо центральної проблеми – подання знань і методів їх використання, так і щодо принципів, методів, технологій побудови інтелектуальних систем, розробки їх архітектури, технології застосування, створення і перевірки спеціальних мов програмування і специфікацій, орієнтованих на розв'язання задач з проблематики штучного інтелекту.

Зрозуміло, що зміст книги не є настільки повним, щоб починати самостійне проектування інтелектуалізованих прикладних систем, але він достатній для того, щоб ознайомитися з досягненнями в галузі їх розробки і ввійти в коло проблем, які супроводжують цей процес, вибрати самостійно тип моделі знань стосовно конкретної предметної галузі й почати роботу з реалізації ЕС разом з фахівцем у галузі системотехніки.

Студенти спеціальностей напрямів підготовки «Системна інженерія», «Автоматизація та комп'ютерно-інтегровані технології», «Комп'ютерні науки», «Прикладна математика» можуть ознайомитися також з об'єктно-орієнтованою спрямованістю наведених у підручнику прикладів, що безпосередньо стосуються їх влас-

ної кваліфікації, а саме — проблемою інформатизації в робототехніці, в гнучких комп'ютерно-інтегрованих виробництвах, розпізнавання образів у широкому розумінні цієї задачі у різних аспектах її застосування.

Зазначимо також, що студенти інших споріднених спеціальностей, для яких читаються дисципліни з проблематики штучного інтелекту, мають змогу здобути базові знання з принципів побудови систем із ШІ, а також розширити коло своїх професійних інтересів наведеними прикладами практичного застосування інтелектуальних систем у різних предметних галузях і різного призначення.





## ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЕКСПЕРТНІ СИСТЕМИ

Можемо стільки, скільки знаємо.

Ф. Бекон

### 1.1. Інформація та знання

Цей підручник доцільно розпочати з термінологічних пояснень, оскільки в тій галузі, якої він стосується, термінологія поки що не усталилася, й багато термінів тлумачаться неоднозначно. На вибір і тлумачення термінів впливають традиції та смак фахівців з таких, що перетинаються, і близьких до неї галузей, наприклад, інформатика, знання і дані, математична логіка, системне програмування, пакети прикладних програм, БД і БЗ. Тому спробуємо порівняти деякі тлумачення й коротко пояснити, в якому розумінні вживаються окремі терміни в підручнику.

Як уже зазначалося, міжнародна спільнота вступила в стадію інформатизації, що ґрунтується на двох принципово відмінних складових — інформації та знаннях, які є базисними підвалинами ЕС. За умов істинності, достовірності відображення фактів інформація вважається первинним знанням (рис. 1.1). У свою чергу, утворене завдяки законам логіки вторинне знання (в подальшому називається *виведенням*, або *висновком*) також є інформацією.

Проте *інформація* як відображення дійсності в принципі *не може бути утвореною* (можуть лише змінюватися спосіб і форма її відображення), тоді як *знання* завдяки законам логіки та застосовуваним засобам (зокрема природному та/або штучному інтелекту) саме *утворюється*. Інформація підвладна старінню, може

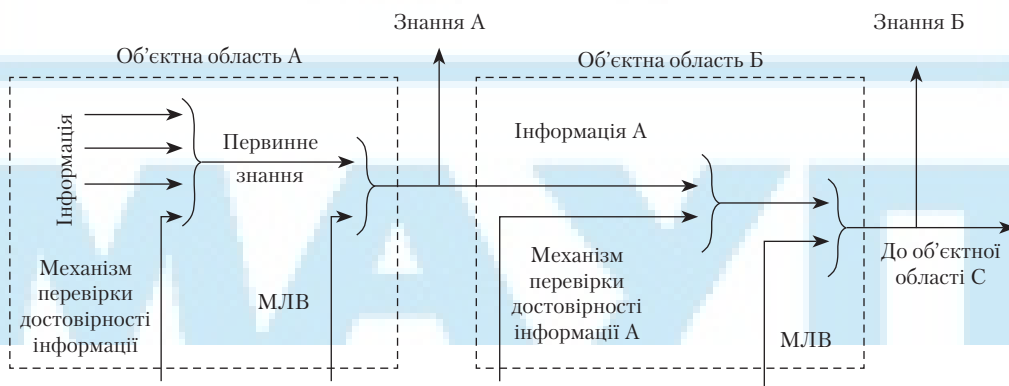


Рис. 1.1. Уявлення про взаємозв'язки інформації та знань

бути суб'єктивною, недостатньою (або дезінформацією). Знання ж як істинне відображення дійсності в мисленні людини завжди об'єктивне й достовірне. Ці взаємозв'язки інформації та знання й лежать в основі підходу до створення ЕС.

У [122] знання визначається як «особлива форма духовного засвоєння результатів пізнання, процесу відображення дійсності, яка характеризується усвідомленням їх істинності». З точки зору ШІ в інженерії знань це означення необхідно пов'язати з виведенням, і тоді прийнятнішим є таке означення:

**Означення 1.1.** Знання — це формалізована перевірена інформація, на яку посилаються чи використовують у процесі виведення.

Якщо ця процедура виконується програмним шляхом, то знання — це обов'язково інформація, подана в певній формі. Якщо ж розглядати знання з точки зору розв'язування задач у деякій предметній галузі, то їх зручно поділити на дві великі категорії — *факти та евристику*.

Перша категорія — факти — вказує зазвичай на добре відомі в цій предметній галузі обставини, тому знання цієї категорії іноді називають тестовими, маючи на увазі їх достатню висвітленість у спеціальній літературі чи підручниках. Друга категорія знань — евристика — ґрунтується на власному досвіді фахівця цієї предметної галузі — експерта, нагромадженому завдяки багаторічній практиці. В ЕС саме ця категорія знань відіграє вирішальну роль у підвищенні ефективності системи, оскільки до цієї категорії входять такі знання, що потребують високого інтелекту для здійснення аналізу інформації, яка надходить від об'єкта досліджень, а саме: способи зосередження, способи вилучення некорисних ідей, способи використання нечіткої інформації та ін.

Крім цих двох категорій знань, вирізняють ще *метазнання* (знання про знання) та *правила* (знання про прийняття рішень).

Отже, під фактами розуміють знання типу «А це А», і вони є характерними для БД і сіткових моделей. Метазнання вказують на знання стосовно способів використання знань, а також знання стосовно властивостей знань (що необхідно для управління БЗ) і виведення, ототожнення, навчання тощо. Зрештою, під правилами розуміють знання типу «ЯКЩО — ТО».

Таким чином, знання, якими заповнюється ЕС, містять знання **першого роду** — *декларативні* (загальновідомі факти, явища, закономірності-істини, визнані в певній предметній галузі й зафіксовані в книгах, статтях, довідниках тощо чи утворені іншими інтелектуальними системами знань завдяки виведенням після обробки інформації) та знання **другого роду** — *процедурні* (емпіричні правила, інтуїтивні міркування й факти, що, як правило, не публікуються, але дають змогу досвідченому експерту навіть в умовах неповноти їх і суперечливості ефективно приймати рішення).

Успіх реалізації ЕС тим більший, чим вища питома вага знань першого роду. Якщо переважають знання другого роду, то деяким розв'язанням труднощів у ситуаціях, коли експерт не може чітко сформулювати правила прийняття рішень або не хоче передавати комусь свої знання, є використання підсистеми накопичення знань. При цьому в автоматичному чи напівавтоматичному режимі формуються емпіричні залежності з неповних даних, і підсистема накопичення знань повинна зі знань другого роду побудувати знання першого роду, створивши теорію, а потім вивести достовірні дані. Знання зазвичай мають класифікаційну ієрархічну структуру вигляду «джерело інформації — книга — навчальна література — підручник — інформатика». Кожен з елементів такої структури

має різні зв'язки з елементами інших ієрархічних структур, тому існує потреба подати всі знання у вигляді сіток. Цей підхід характерний не тільки для декларативних, а й для процедурних знань.

Зазначимо, що знання можуть бути *неповними* (коли для доведення або спростування певного твердження не вистачає інформації), *недостовірними* (наприклад, через випадковість впливу чинників, через об'єктивну невизначеність), *нечіткими* (через неточну визначеність понять, якими подаються знання: наприклад, «поруч», «далеко», «важкий», «великий» тощо).

Для створення можливості використання знань необхідно вирішити проблему здобуття знань, або в ширшому розумінні — проблему навчання. Саме з цим пов'язане здобування інформації з навколишнього середовища та її накопичення в пам'яті з метою дальшого використання.

Зазначена проблема має три взаємопов'язані аспекти: функції того, хто навчається; властивості середовища — джерела інформації; взаємодія того, хто навчається, із середовищем. Як наслідок, при навчанні ставляться принаймні дві мети: здобування інформації із середовища та підвищення її рівня й структурування. Доки не вивчена вся інформація, вона не може стати знанням, і дані нижчого рівня, наприклад, початкові дані, можна використати на цьому рівні. З підвищенням рівня даних універсальність їх у плані застосування збільшується. Друга мета забезпечується різними засобами.

Виходячи з розглянутих вище тлумачень інформації та знання можна сформулювати концепцію інформатизації.

**Означення 1.2.** *Інформатизація* — це процес максимізації використання та відтворення знань як невичерпного ресурсу розвитку при мінімізації непоповненого ресурсу — часу для відтворення й комунікації знань.

Це означення інформатизації, з одного боку, підтверджує еволюційність процесу суспільного розвитку, оскільки відтворення, зберігання та комунікація знань за мінімальними ресурсами часу забезпечуються саме стадією комп'ютеризації, а з іншого — дає змогу визначити рівень науково-технічного прогресу суспільства за ступенем його інформатизації.

Сучасний етап розвитку інформатики характеризується еволюцією моделей даних у напрямі переходу від традиційних знань (реляційна, ієрархічна, мережна) до моделей знань [58].

## 1.2. Означення баз даних і баз знань інтелектуальної системи

Спочатку визначимо зміст термінів «база знань» (БЗ) і «база даних» (БД).

**Означення 1.3.** *База знань (БЗ) інтелектуальної системи (ІС)* — це сукупність усіх знань, які зберігаються в пам'яті системи.

Якщо надалі говорити про БЗ як про систему знань про проблемне середовище (тобто частину об'єктивного світу, що відображає це середовище у вигляді цілісного явища з усіма взаємозв'язками, які існують між елементами і динамікою їх зміни), то тоді БЗ виступає стосовно інших засобів систем управління як система декларативних знань (та асоційованих з ними процедур) з єдиними принципами подання, спільною мовою опису, загальною схемою маніпулювання цими знаннями.

Взагалі ж БЗ є кон'юнкцією більш простих тверджень, або *концептуальних одиниць* — фактів або правил виведення, кожна з яких відповідає окремому реченню природної мови. Концептуальна одиниця може бути досить складною і тоді її можна роз-

глядати як кон'юнкцію елементарних тверджень — *бінарних фактів*, кожний з яких описується *бінарним предикатом*, тобто предикатом, який залежить від двох змінних.

**Приклад 1.1.** Як приклад розглянемо твердження: «Робот ЕРК1 доставив до складальної позиції СП2 сотовий (С) та шахтний (Ш) нагромаджувальні пристрої, наповнені комплектуючими складального вузла СВ2 — відповідно, деталями класів А та В». Якщо подати це твердження у вигляді кон'юнкції бінарних фактів, матимемо запис:

$\epsilon$  (ЕРК\_1, Робот)  
 $\epsilon$  (СП\_2, Складальна\_позиція)  
 $\epsilon$  (С, Нагромаджувальний\_пристрій)  
 $\epsilon$  (Ш, Нагромаджувальний\_пристрій)  
 $\epsilon$  (Клас «А», Клас\_деталей)  
 $\epsilon$  (Клас «В», Клас\_деталей)  
 $\epsilon$  (Класи\_деталей, комплектуючі)  
 $\epsilon$  (СВ\_2, Складальний\_вузол)  
Нагромаджувальні\_пристрої (ЕРК\_1, СП\_2),

де предикати *Складальна\_позиція*, *Нагромаджувальний\_пристрій*, *Клас\_деталей*, *Складальний\_вузол* введені для задавання зв'язку між різними інформаційними одиницями.

Бінарним предикатам і бінарним фактам безпосередньо відповідає тріада «об'єкт — атрибут — значення», де *об'єктом* є, як правило, назва деякої інформаційної одиниці, *атрибутом* — назва певної ознаки, а *значенням* — конкретне значення цієї ознаки об'єкта.

Якщо концептуальна одиниця є фактом, її можна описати певним *предикатом* — логічною функцією, яка залежить від тієї чи іншої кількості змінних і може набувати одне з двох можливих значень: *істинність* або *хибність*. Істинність тверджень перевіряється шляхом їх *інтерпретації* — підстановкою конкретних значень аргументів.

Наприклад, маємо три твердження:

**Твердження 1.1.** «Маніпуляційна система» та «Система управління» утворюють «Промисловий робот».

**Твердження 1.2.** «Орієнтуючі пристрої» та «Нагромаджувальні пристрої» утворюють «Систему упорядкування середовища».

**Твердження 1.3.** «Помислові роботи» та «Автомагістральні шляхи» утворюють «Робототехнічну систему», які всі відповідають одному предикатові:

**Та** ( $a, b, c$ ).

Як бачимо, перші дві інтерпретації змінних  $a, b, c$  конкретними значеннями породжують істинні твердження, а третя — хибне.

Вирізняють екстенціональну та інтенціональну частини у БЗ.

**Означення 1.4.** *Екстенціональна частина БЗ* — це сукупність усіх явних фактів ( $F$ ); *інтенціональна частина БЗ* — це сукупність усіх правил виведення ( $R$ ) та процедур ( $P$ ), за допомогою яких з існуючих фактів можна вивести нові твердження. Перша містить факти, які запам'ятовуються явно, а друга — правила, які дають змогу отримувати нові факти. Так, процедури  $P$  визначають, яким саме чином слід застосовувати правила  $R$ , і якщо таких процедур (якими насамперед є складові спеціалізованих мов штучного інтелекту *Лісп*, *Пролог*, *Пленер*) недостатньо, користувач має подбати про їх написання. Отже, йдеться про поняття *інтенціонального відношення*, яким описується БЗ.

**Означення 1.5.** *Інтенціональне відношення* — це правило або сукупність правил, що упорядковують кожний запис у БЗ.

Наявність розвинутої інтенціональної частини є однією з основних рис, які відрізняють структуру ІС від звичайної (пасивної) системи накопичення знань. Пошук в інтенціональній БЗ складається з двох етапів:

- пошук необхідного факту в екстенціональній частині;
- дедуктивне виведення факту на основі правил інтенціональної частини.

**Приклад 1.2.** Як приклад розглянемо умовну БЗ для  $n$  об'єктів роботизації (ОР), що містить дані про те, які технологічні операції пройшов кожний з ОР. При цьому відомо, що жодний з  $n$  ОР не може пройти операцію нарізання внутрішньої різьби без операції свердлення отвору під неї. Нехай у БД зберігаються дані про об'єкти роботизації Деталь 1, Деталь 2, Деталь ..., Деталь  $n$ , які пройшли обидві операції. Тоді інтенціональна БД (або вже БЗ) матиме такий вигляд:

**Екстенціональна частина**

ОР	Технологічна операція
Деталь 1	Нарізання внутр. різьби
Деталь ...	Нарізання внутр. різьби
Деталь $n$	Нарізання внутр. різьби

**Правило:** *ЯКЩО* пройшов ( $X$ , Нарізання внутр. різьби), *ТО* пройшов ( $X$ , Свердлення отвору під різьбу).

Узагальнюючи сказане, сформулюємо таке: *знаннями ІС є триада*  $\langle F, R, P \rangle$ .

Тепер чіткіше можна подати уявлення про терміни «система БД» та «система БЗ».

Почнемо з терміна «система БЗ», який не є загальноприйнятим, і існують інші близькі за значенням і більш поширені його аналоги. Він утворився за аналогією з терміном «система БД», який є калькою англomовного терміна «*data base system*».

**Означення 1.6.** *Система БД (СБД)* — це як інструментальна система, що забезпечує створення та використання БД, так і система, що забезпечує функціонування конкретної прикладної БД або кількох БД, тобто прикладна система.

У першому значенні у вітчизняній літературі зазвичай вживають термін «система управління БД» (СУБД). Ключовим поняттям стосовно СБД, яке відображає їх специфіку, є поняття БД, а стосовно систем БЗ (СБЗ) — поняття БЗ.

**Означення 1.7.** *Системою БЗ* називається система, що забезпечує створення й використання БЗ.

Її розглядають як інструментальну систему, що називається також *системою управління БЗ* (СУБЗ), або як прикладну систему з конкретною використовуваною БЗ.

Однак в англomовній літературі замість терміна «*knowledge base system*» частіше вживається термін «*knowledge based system*» (**система, що ґрунтується на знаннях**), і ми вважаємо за доцільне як технічний вживати саме останній термін, який дає, до речі, попереднє скорочення — СБЗ.

### 1.3. Зв'язки між складовими знань у базах даних

Основними зв'язками між об'єктами (складовими знань) є узагальнення (абстрагування) та агрегація (деталізація, або уточнення). *Агрегація* задає побудову об'єкта та його складових. Наприклад, агрегація дає змогу визначити в БЗ «Про-

мисловий робот» (ПР) як об'єкт, що складається з типових механізмів — перетворювальних, передавальних, захватних, гальмуючих, напрямних, орієнтуючих, виконавчих тощо. У свою чергу, скажімо, агрегація напрямних механізмів ПР дає змогу за відповідною БЗ агрегувати їх виходячи з умов взаємодії рухомих елементів — на напрямні змішаного тертя, сухого тертя, гідростатичні, гідродинамічні, кочення (з роликками, голками, кульками як тілами кочення); нарешті, знов за відповідною БЗ, подальшою агрегацією можна виявити складові, скажімо, напрямних кочення — власне напрямні, тіла кочення, сепаратори, елементи натягу тощо. За такого роду відношеннями закріпився спеціальний термін **HAS\_PART**, тобто «*ціле — частина*». Обернене відношення («Є частиною») позначається **IS\_PART**.

Узагальнення (відношення «Є») задає ієрархію класів (*екземпляр — клас — підклас*). З цим відношенням «Є» пов'язаний і широковживаний механізм логічного виведення за успадкуванням (або виведення за наслідуванням): у разі виконання деякої умови для всього класу вона виконується і для кожного представника цього класу, і для всіх підкласів цього класу — підкласи успадковують властивості класів, а ті, у свою чергу, — властивості надкласів (якщо інше не задано явним чином). Проте якщо відношення «*клас — підклас*» є транзитивним (для цього відношення виконуються всі властивості часткового порядку), то відношення «*екземпляр — клас*» не є транзитивним, хоча екземпляри й успадковують властивості своїх класів. Наприклад, якщо «*Захватний пристрій*» є екземпляром класу «*Типові механізми ПР*», а «*Лещата*» є екземпляром класу «*Захватний пристрій*», то «*Лещата*» мали б бути, але в жодному разі не є екземпляром класу «*Типові механізми ПР*», бо за призначенням і застосуванням не є такими. Отже, в цьому випадку йдеться про наявність винятків, коли характерні риси класу успадковуються всіма підкласами, окрім деяких. Тоді дедуктивна система знань, побудована на таких суперечливих твердженнях, з одного боку, «*Лещата*» є «*Захватним пристроєм*», а з іншого — не є «*Типовим механізмом ПР*», втрачає монотонність.

**Твердження 1.4.** Система дедуктивного виведення є монотонною тоді і тільки тоді, коли виконується властивість: якщо з набору тверджень  $(t_1, \dots, t_m)$  випливає твердження  $r$ , то  $r$  випливає і з набору тверджень  $(t_1, \dots, t_m, q)$ . Іншими словами, в монотонній логіці додавання нових фактів і правил не повинно впливати на істинність висновків, які могли бути отримані без них.

У разі порушення монотонності при додаванні винятків до наявної БЗ, що призводить до суперечностей, корисними є спеціальні *немонотонні логіки* із застосуванням різних пріоритетних прийомів, коли, наприклад, *підклас успадковує відповідну властивість лише від найближчого попередника, тобто від класу, найближчого до нього в ієрархії класів*.

Нарешті, слід розрізняти поняття «*клас*» і «*множина*». Перше має відкритий, інтенціональний характер, тоді як друге («*множина*») — закритий, екстенціональний і передбачає певну сукупність об'єктів взагалі, не обов'язково з однотипними властивостями (на відміну від «*класу*», де набір властивостей притаманний його екземплярам). Слід також зауважити, що складні системи можуть бути описані канонічною формою у вигляді композиції двох ієрархій: **класів** — задається відношенням узагальнення та **об'єктів** — задається відношенням агрегації.

#### 1.4. Означення та структура експертної системи

До терміна «система баз знань» (СБЗ) близьким за змістом є термін «експертні системи», в якому акцент робиться на знання експертів, тобто спеціалістів певної галузі. В літературі можна зустріти кілька іноді досить багатослівних й екзотичних визначень ЕС, але суть їх полягає в тому, що ЕС — це система, яка забезпечує створення й використання за допомогою комп'ютерів БЗ експертів.

Головною причиною переходу від досліджень, які стосуються використання логічних виведень, до досліджень щодо використання знань є бажання практичного застосування систем із ШІ. Отже, ЕС — це перший крок на шляху реалізації досягнень у галузі ШІ. Вони відкривають широкі можливості й за нашого часу вже практично використовуються в промисловості.

Експертні системи є інтелектуалізованими інформаційними системами, які за процедурами  $P$  та правилами  $R$  на основі наявних знань  $F$  здійснюють дедуктивне виведення нових знань. Існують різні визначення ЕС, але в основі всіх лежить таке їх означення [21]:

**Означення 1.8.** *Експертні системи (ЕС)* — це інтелектуалізовані програмні засоби, здатні в ході діалогу з людиною одержувати, накопичувати та коригувати знання із заданої предметної галузі, виводити нові знання, розв'язувати на основі цих знань практичні задачі та пояснювати хід їх розв'язування.

Експертні системи акумулюють знання експертів — провідних фахівців у певній предметній галузі. В основі роботи ЕС лежить дедуктивне виведення нових тверджень з існуючих. Типове застосування ЕС — консультації для фахівців середньої кваліфікації і нефахівців у тій галузі, для якої вона розроблена. Тому ЕС повинна мати функції, що давали б змогу розв'язувати задачі, правильне розв'язання яких за відсутності експерта (спеціаліста конкретної галузі) є неможливим.

У розробці ЕС зазвичай беруть участь фахівці двох категорій: *експерт*, що є висококваліфікованим фахівцем у конкретній предметній галузі і знання якого треба передати ЕС; *інженер знань*, який формалізує знання експерта і приводить їх до вигляду, придатного для занесення до БЗ.

Структурну схему ЕС показано на рис. 1.2. Вона містить:

- механізм подання знань у конкретній предметній галузі та управління ними — БЗ;
- механізм, який на основі знань, що зберігаються в БЗ, робить логічні виведення — *механізм логічних виведень* (МЛВ);
- інтерфейс для правильного передавання відповідей користувачу — *інтерфейс користувача* (ІК);
- механізм здобуття знань від експерта, підтримки БЗ і доповнення її за необхідності — *модуль здобуття знань* (МЗЗ);
- механізм, який подає різні коментарі до висновку й пояснює мотиви останнього, — *модуль порад і пояснень* (МПП).

Останній модуль характерний для ЕС і відповідає на запитання, як і чому кінцевий користувач за допомогою ЕС прийняв те чи інше рішення (таким чином, МПП виконує роль підсистеми аналізу й інтерпретації рішень).

У реальних ЕС функції названих механізмів можуть бути відповідним чином підсилені або розширені. Модуль порад і пояснень є важливим не тільки для користувача системи, а й для експерта, що подає знання в конкретній предметній га-

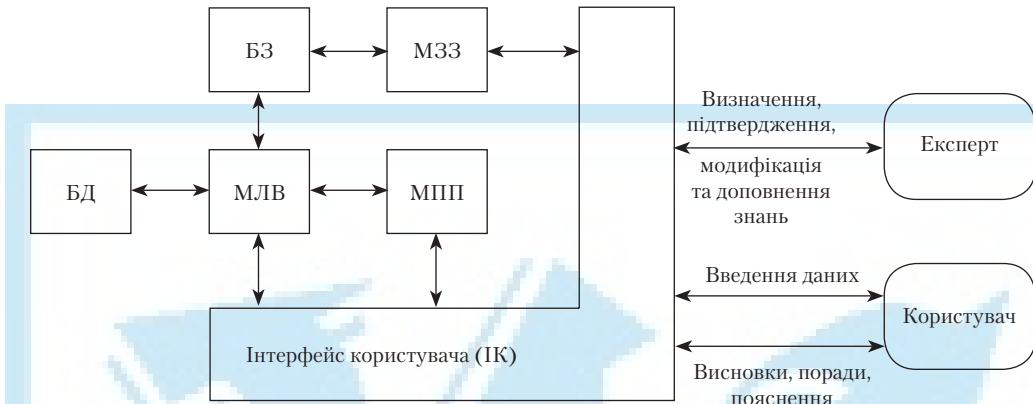


Рис. 1.2. Структурна схема ЕС

дузі, оскільки дає змогу експертові визначати, як працює система і як використовуються подані ним відомості.

Взагалі при створенні чи виборі ЕС слід урахувувати такі особливості їх використання:

1. Всупереч дуже поширеному погляду найдорожчою частиною ЕС є не програмне забезпечення, а сам експерт, його підготовка. Тому основне зусилля має бути спрямоване на правильний вибір експерта з певної предметної галузі знань.

2. Експертна система повинна містити не тільки експертні (як правило, поверхові) оцінки фахівців, а й елементи фундаментальних знань, оскільки експерт часто приймає рішення на основі досвіду й інтуїції, тобто деяких логічних і статистичних оцінок, не аналізуючи (а часто, й не знаючи) глибинних фізичних основ явищ.

3. Потрібно передбачити механізм використання локальних оптимумів, розташованих на межах (або просто віддалених) зон глобальних оптимумів, тобто тих зон, де може виявлятися ефект саморегулювання системи. Цей підхід ще не дуже вивчений, але є приклади виходу на такі зони та утримання в них систем на незначних часових проміжках, коли забезпечується ефективність, яка в кілька разів перевищує звичайні показники функціонування ЕС. Такі дії можуть бути не завбачені експертами через їх нетрадиційність і нетривіальність (наприклад, це стосується сукупності параметрів технологічних процесів у гнучких виробництвах, які забезпечують зони локальних екстремумів).

4. Повинна враховуватися проблема взаємодії оператора й ЕС. Щоб користуватися ЕС, оператор має довіряти їй, а для цього ЕС повинна забезпечувати високий рівень експертних оцінок й оптимальність прийнятих рішень (рекомендацій), а також високу надійність, для чого вводяться резервування 2 з 3, 5 із 7 тощо. При цьому вихід з ладу (відмова) до 40–50 % елементів не призводить до втрати функціональної працездатності ЕС.

Модуль порад і пояснень здатний здобути знання, якими несвідомо володіє фахівець. Наприклад, якщо використовується реальна сукупність тестових даних, то виведення їх стає незрозумілим, фахівець обходить «гострі кути». Проте МПП працює так, щоб допомогти пояснити причину неправильного виведення даних. Це



дає змогу зробити усвідомленими знання, якими до того фахівець володів несвідомо. Зрештою виникає ситуація «я знаю, що я знаю», завдяки чому знання фахівця можна використати ефективніше.

Експертна система працює в двох режимах: *здобуття знань і розв'язування задач*. У режимі здобуття знань у спілкуванні з ЕС бере участь експерт через посередництво інженера-програміста. В цьому режимі експерт наповнює систему знаннями (правилами), які дають змогу їй у режимі розв'язування самостійно розв'язувати задачі. Режиму здобуття знань при традиційному підході до розробки програм відповідають етапи алгоритмізації, програмування та відпрацювання, що виконуються програмістом. Але для ЕС характерним є те, що програму розробляє не програміст, а фахівець, який не володіє програмуванням.

У режимі розв'язування задач у спілкуванні з ЕС бере участь користувач, якого цікавлять результат і (або) спосіб одержання рішення.

Залежно від призначення ЕС користувач може або не бути фахівцем цієї проблемної галузі (в цьому разі він, не вмючи одержати сам відповідь, звертається до ЕС за порадою), або бути фахівцем (тоді він може й сам одержати результат, але звертається до ЕС з метою прискорення процесу одержання результату чи з метою покласти на ЕС виконання трудомісткої роботи).

У режимі здобуття знань експерт вводить у систему продукції (правила) про галузь експертизи, що подаються природною мовою. Об'єднання продукцій, які вводяться заново, з БЗ здійснюється компонентом здобуття знань. Аби перекоонатися в тому, що процес відпрацювання задачі завершено, експерт надає системі тестові приклади. Якщо результат, одержаний системою, експерта не задовольняє, то за допомогою інтерфейсу, пов'язаного з поясненнями, експерт одержує відомості про те, як цей результат був створений. Після закінчення налагодження ЕС стає придатною користувачеві для експлуатації.

У режимі розв'язування задач дані про задачу користувача після обробки їх лінгвістичним процесором надходять у БД.

Процес здобуття знань можна звести до послідовності виконання таких дій: 1) визначається необхідність модифікації (розширення знань); 2) у разі модифікації здобуваються нові знання, інакше процес здобуття знань закінчується; 3) нові знання перетворюються у форму, «зрозумілу» для ЕС; 4) знання системи модифікуються — здійснюється перехід до першої дії. У виконанні перелічених дій можуть брати участь експерт, інженер знань (програміст) і ЕС. Залежно від того, хто розв'язує задачу, виокремлюють різні покоління ЕС.

У перших системах із ШІ взаємодію з ними здійснював тільки програміст, до завдання якого входило опанування за допомогою експерта предметної галузі, після чого при розробленні системи він міг виступати і як експерт, і як програміст. Недостатнє знання предметної галузі не давало змоги програмісту гарантувати повноту та несуперечливість здобутих знань. Крім того, неминучі модифікації системи призводили (через відсутність поділу системи на БЗ і МЛВ) до неможливості зберегти вже досягнуту несуперечливість знань.

Подальші розробки систем із ШІ ґрунтувались на відокремленні знань від програм й оформленні перших у вигляді простих інформаційних систем, які дістали назву БЗ. У цьому разі експерт взаємодіє із системою або безпосередньо, або через інженера знань. Перевага цього підходу порівняно із згаданою моделлю полягає в тому, що БЗ спрощує модифікацію знань. При цьому дві перші дії в процесі здобуття знань виконує експерт за допомогою інженера знань, третю — програміст, а

четверту — ЕС. Суттєвим недоліком такого підходу є його висока трудомісткість, оскільки автоматизованою є тільки одна дія в процесі здобуття знань.

Нарешті, експерт може взаємодіяти з ЕС через інтелектуальний редактор без посередництва програміста. Модель інтелектуального редактора повинна бути забезпечена розвиненими діалоговими засобами та значними знаннями про структуру БЗ (тобто метазнаннями). При використанні інтелектуального редактора дві перші дії в процесі здобуття знань виконує експерт, а дві останні — ЕС.

На сьогодні найдосконалішими є ЕС, які здобувають знання аналогічно тому, як це робить людина-експерт. У цьому разі індуктивна програма аналізує дані, що містять відомості про деяку предметну галузь, автоматично формуючи значущі відношення й правила, які описують предметну галузь. Припускається, що в БЗ у явному вигляді зберігаються конкретні факти про проблемну галузь, а завданням індуктивної програми є вироблення значущих узагальнень. Основною перевагою цього підходу є автоматизація всіх перелічених вище чотирьох дій у процесі здобуття знань. У галузі створення ЕС, що використовують індуктивні програми, ще тільки робляться перші кроки і поки що не існує ЕС, які здобувають знання виключно від індуктивних програм. Є лише експериментальні програми, за допомогою яких можна здійснювати індуктивні узагальнення.

Наступне покоління ЕС пов'язують зі здобуттям знань безпосередньо з текстів, написаних природною мовою, тобто йдеться про розуміння текстів, схем, креслень, графіків тощо. Складність тут полягає не тільки в опрацюванні природної мови, а й у необхідності відтворення за текстом моделі деякої проблемної галузі. Ці вимоги поки що перевищують можливості існуючих програм розуміння, незважаючи на те, що в цьому разі аналіз текстів обмежений досить вузькою предметною галуззю.

Останніми роками ЕС дедалі частіше застосовуються як форма інтелектуальних прикладних систем при проектуванні, плануванні й управлінні в *гнучких комп'ютерно-інтегрованих системах* (ГКІС). За допомогою спеціальних інструментальних засобів знижуються витрати на їх розробку, а спеціальні мови уможливають застосування інтелектуальних прикладних систем у реальних умовах функціонування ГКІС. Однак навіть у разі використання таких систем не забезпечується очікувана їх ефективність головним чином тому, що складність застосовуваних стратегічних рішень за аналізованими фактами (результатами експерименту) перевищує можливості інтелекту (кваліфікації) людини.

Крім того, відсутні адекватні цій складності інструментальні засоби наукового обґрунтування рішень з продукуванням результатів оцінювання наслідків від їх реалізацій на основі моделювання процесів у таких складних системах, якими є ГКІС. Постає, таким чином, завдання побудови *систем автоматизованого імітаційного моделювання* (САІМ), в яких на основі наявних статистичних даних (фактів) самостійно формулюються гіпотези та виявляються закономірності про взаємозв'язки з використанням як суворо обґрунтованих формальних процедур, так і евристик, одержаних під час експертних оцінювань і навіть сформованих САІМ.

Для того щоб усвідомити роль і місце ЕС у структурі САІМ, звернемося до трохи іншої інтерпретації структурного подання ЕС (рис. 1.3). Експертна система містить БЗ і БД, розв'язувальний блок, підсистеми спілкування (взаємодії), пояснень (інтерпретації) та накопичення знань. Через підсистему спілкування (інтерфейс користувача) з ЕС пов'язані: кінцевий користувач (фахівець-непрограміст); експерт (кваліфікований фахівець або група фахівців, досвід і знання яких набагато перевищують знання й досвід кінцевого користувача); інженер знань, який володіє мовами опису знань.

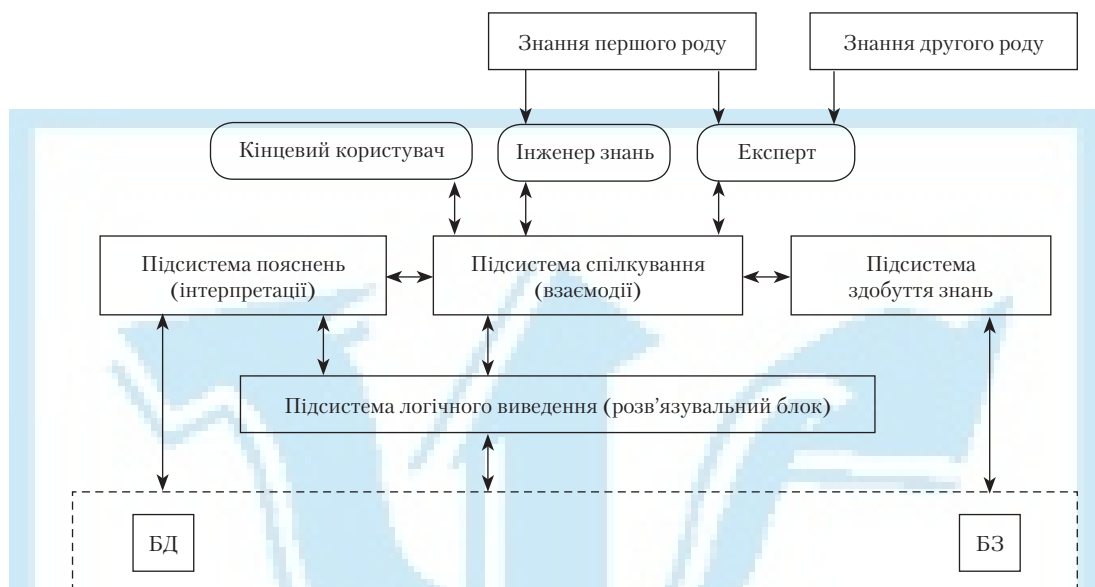


Рис. 1.3. Функціональна схема ЕС

На етапі накопичення знань з ЕС працюють експерт та інженер знань, а на етапі експлуатації та використання — кінцевий користувач. При цьому мова подання знань, яка використовується для розроблення ЕС, називається *мовою розроблення* ЕС, а система програмного забезпечення, що забезпечує згадані вище функції ЕС, — *інструментом для розроблення* ЕС.

### 1.5. Інженерія знань

Численні дослідження в галузі ШІ дали змогу укріпитися в думці, що необхідними для вирішення проблеми здобуття знань і здійснення виведення є знання експертів. Оскільки знання останніх нагромаджуються в пам'яті ЕОМ і використовуються для вирішення прикладних проблем, системи, що реалізують цей принцип, дістали назву *експертних*, а професор Є. Фейгенбаум (один із створювачів ЕС) запропонував для цієї галузі інформатики назву «інженерія знань», яка тепер має ширший зміст.

*Інженерія знань* — це сукупність технічних способів, спрямованих на створення систем, які призначені для вирішення проблем з використанням знань. Знання ж, як зазначено вище, є суттю інформації з обмеженою семантикою. Проте з позиції прикладних аспектів необхідно, щоб знання мали таку форму, для якої була б певною мірою властива свобода досягнення поставленої мети.

Питання про те, якою мірою припустима ця свобода, або, точніше, яким умовам повинні відповідати знання з їх описовими можливостями включно, має неоднакову відповідь залежно від галузі застосування знань. Зазвичай у сфері технічного застосування використовується різноманітне середовище їх подання, і, крім мовного опису, воно містить рисунки, математичні формули тощо. Мовний опис, який потрібен у прикладних галузях інформації (включаючи мову в широкому її розумінні й графі-

ку), називається *мовою подання знань*. Для використання такої інформації у вигляді знань потрібні інтелектуальні функції, що часто перевищують функціональні можливості сучасних комп'ютерів. Все це (подання знань і їх використання, які розглядаються стосовно конкретної прикладної галузі) є *предметом інженерії знань*.

Хоча тут йдеться про ЕС як про щось єдине, вони у своєму конкретному втіленні мають найрізноманітнішу форму. Проте для всіх ЕС властива загальна риса: обробка інформації на ЕОМ виконується у формі, цілком відмінній від традиційної форми її обробки за Тьюрингом фон Нейманом. У загальному випадку ЕС є людино-машинними системами, побудованими з таких основних елементів, як БЗ з використанням вибраної форми подання знань, і МЛВ, що забезпечує маніпулювання цими знаннями при вирішенні прикладних проблем: він пов'язує знання воедино, а потім виводить з їх послідовності висновок.

Форма виведення інформації залежить від способу подання знань. У СБЗ, у тому числі й ЕС, подання знань є фундаментальним поняттям, а рішення про вибір способу цього подання досить суттєво впливає на будь-яку їх складову частину. Можна навіть сказати, що можливості СБЗ визначаються поданням у них знань. І навпаки, щоб система обробки знань відповідала певним прикладним призначенням, необхідно створити відповідне подання знань.

Оскільки подання знань є засобом опису знань людини, бажано, щоб описові можливості цього засобу були якомога вищими. Але, з іншого боку, якщо форма подання знань стає надмірно складною, то ускладнюється й механізм виведення інформації. При цьому не тільки утруднюється проектування ЕС, а й виникає небезпека втрати достовірності виконуваних нею дій. Тому в остаточному підсумку проектування подання знань передбачає вироблення всіх цих умов (обмежень), а потім вибір рішення на основі деякого компромісу між ними.

Ознайомимося з деякими характерними проблемами подання знань, які використовуються в сучасних ЕС.

## 1.6. Проблемний характер формалізації та подання прикладних знань

Насамперед необхідно формалізувати відповідну галузь прикладного знання. Це складне завдання, оскільки розв'язується воно вручну і потребує спільної роботи фахівців-прикладників і математиків. Для проведення формалізації необхідно вибрати або побудувати концептуальну схему моделі системи знань (СЗ). Розробка методології всіх цих операцій і становить зміст першої проблеми — *формалізації знань*.

Друга проблема — *подання знань* — пов'язана з розробкою формального апарату для опису способів їх фіксації в пам'яті ЕОМ.

Розробка теорії обчислень та інших перетворень, що здійснюються в побудованих моделях, становить третю проблему — *використання знань*.

І нарешті четверта, технологічна проблема, вирішенням якої займаються системні програмісти, — це *розробка засобів програмної підтримки моделей*, тобто створення баз знань і систем управління ними.

Основна увага в ШІ надається другій і третій з перерахованих проблем, причому провідна роль відводиться проблемі подання знань. На практиці її розробляють разом з питаннями побудови концептуальних схем моделей знань, і багато хто вважає, що саме ця проблема є основною для сучасного ШІ, бо саме вона має забезпечувати такі характеристики СЗ, як *надійність, гнучкість і компактність* (рис. 1.4).



тем диференціальних рівнянь можуть описувати процеси в об'єктах найрізноманітнішої природи. Той же універсализм буде і при використанні оптимізаційних моделей лінійного чи нелінійного програмування тощо.

Якщо математичну модель подати без коментарів, то неможливо сказати, який конкретно об'єкт або процес нею описується. Можна тільки зробити висновок про те, з якими класами об'єктів цю модель можна порівняти. Семантика відома лише спеціалістам, які формалізують процеси в тому чи іншому об'єкті. При цьому коментарі, що розкривають конкретні знання про об'єкт, а отже, й семантика формально-математичних моделей перебувають поза ЕОМ.

Становлення нової інформаційної технології зумовлено тим, що в теорії ШІ розроблено логіко-лінгвістичні моделі, які дають змогу формалізувати конкретні змістові знання про об'єкти керування (ОК) та процеси, що в них відбуваються, тобто ввести в ЕОМ логіко-лінгвістичні моделі поряд з математичними. Такі моделі – семантичні сітки (СС), продукційні системи, фрейми – іноді об'єднуються поняттям «програмно-апаратні засоби в системах із ШІ». Саме цим моделям зобов'язані своєю появою БЗ.

Таким чином, нова інформаційна технологія відрізняється від існуючої насамперед тим, що за допомогою спеціальних формалізмів (логіко-лінгвістичних моделей) декларативні та процедурні знання подаються в електронній формі, завдяки чому розв'язування задач за допомогою ЕОМ відбувається більш ефективно.

Етап перекладу умов задачі (ситуації) з природної мови на формалізовану присутній в кожній науковій дисципліні. Фахівцям, які займаються дослідженням операцій, постійно доводиться описувати формальною мовою складні ситуації. Цей процес називається *моделюванням*.

Система знань, подана формальною мовою, є *формальною системою*. Вона об'єднує суто абстрактні об'єкти (не пов'язані із зовнішнім світом), визначаючи правила оперування множиною символів у виключно синтаксичному трактуванні без урахування змістового осмислення (або семантики).

**Означення 1.9.** *Формальна система знань (ФСЗ) буде визначеною, якщо:*

- 1) задано скінченний *алфавіт* (скінченну множину символів);
- 2) визначено процедуру побудови *формул* (або слів);
- 3) виділено деяку множину формул, які називаються *аксіомами*;
- 4) задано скінченну множину *правил виведення*, що дають змогу одержувати з деякої скінченної множини формул іншу множину формул, тобто  $U_1 i U_2 i \dots i U_p \rightarrow \rightarrow W_1 i W_1 i \dots i W_n$ , де  $U_i, W_i$  – формули формальної системи, а стрілка  $\rightarrow$  читається як «спричинює» або «впливає».

Алфавіт, який заздалегідь вважається скінченним, іноді називається словником.

Спосіб подання формул (п. 2 означення 1.9 формальної системи знань) визначає конкретну синтаксичну конструкцію формул (*граматику формул*) – правильно побудовані послідовності символів. На відміну від цього в п. 4 означення 1.9 задаються дозволені правила виведення формул одна з одною.

**Означення 1.10.** *Формальне доведення* (або просто доведення) визначається як скінченна послідовність формул  $M_1, M_2, \dots, M_r$  така, що кожна формула  $M_i$  або є аксіомою, або за допомогою одного з правил виведення є вивідною з попередньої формули  $M_j$ , де  $j < i$ .

**Означення 1.11.** *Формула T називається теоремою, якщо існує доведення, за яким вона є останньою, тобто  $M_r \equiv T$ ; то ж будь-яка аксіома – це теорема.*

Якщо  $T$  є теоремою, то цей факт коротко записується так:  $\vdash T$ .

Вирізняють два типи правил виведення:

1) такі, що застосовуються до формул, які розглядаються як одне ціле (в цьому разі їх називають *продукціями*, або *продукційними правилами*);

2) такі, що можуть застосовуватись до будь-якої окремої частини формули, до того ж самі ці частини є формулами, які входять до складу ФСЗ (у цьому разі кажуть про *правила переписування*).

Так, правило виведення

$$x < y \text{ і } y < z \mapsto x < z$$

застосовується до формули як до цілого — це продукція (з двома антецедентами, або посиленнями).

На відміну від попереднього правило  $x - x \mapsto 0$  має сенс при будь-якому вигляді  $x$ , що входить у правило як підвираз. Це вже правило переписування, і для позначення слова «спричинює» в даному разі використовується стрілка  $\mapsto$ .

І продукція, і переписування мають тільки один напрямок виведення — зліва направо.

*Підстановка* полягає в заміщенні всіх входжень будь-якої змінної формулою з ФСЗ, що не містить цієї змінної. Правило підстановки придатне для використання в будь-якій ФСЗ. Незаміщені символи називаються *константами*, а заміщені — *операторами*.

Оскільки ФСЗ завжди є моделлю якоїсь реальності (конкретної чи математичної), запроваджується поняття *інтерпретації*.

**Означення 1.12.** *Інтерпретація* — це поширення вихідних положень будь-якої ФСЗ на реальний світ із наданням змістовності кожному символу цієї системи і встановленням взаємно однозначної відповідності між її символами та реальними об'єктами.

**Висновок 1.1.** *Теореми ФСЗ, будучи раз інтерпретованими, стають після цього твердженнями, і можна робити висновки щодо їх істинності чи помилковості.*

Корисність і зручність ФСЗ полягає в їх абстрагуванні від конкретної реальності, завдяки чому одна й та сама ФСЗ може бути моделлю різних конкретних ситуацій. Але при цьому необхідно, щоб для цієї ФСЗ завжди існувала принаймні одна інтерпретація, в якій кожна теорема цієї системи була б істинною.

## 1.8. Способи подання знань

Однією з ключових проблем створення ШІ є проблема подання і використання знань. В галузі ШІ поняття про знання сформувалося в ході досліджень із створення принципів і техніки роботи з великими обсягами даних і з побудови даних — БД. Ефективність БД багато в чому залежить від того, яким саме способом організуються, структуруються дані в пам'яті ЕОМ. До певного часу основну роль у цьому відігравали формальні характеристики даних: належність їх до деякої табличної рубрики, входження до однієї тематичної групи тощо.

Проте ефективність БД можна істотно підвищити, якщо зв'язувати знання, що зберігаються, не за рахунок форм тих чи інших документів (таблиць, списків), а за рахунок тих відношень, які існують між фактами в об'єкті управління або в природному середовищі. І відношення ці мають бути не випадковими, ситуативними, а відображати істотні зв'язки об'єкта, його природу, тобто виникла необхідність відоб-

ражати в БД знання про об'єкт. Такі БД стали називати *інтелектуальними базами даних* або *базами (системами) знань*. Ідеологія створення СЗ в основному пов'язана з формалізацією семантичної пам'яті, точніше, деяких її моделей, розроблених у когнітивній психології, а саме — семантичних сіток, фреймів тощо (насамперед сіток Петрі, які набули величезного значення для подання і перетворення знань про дискретно-подійні процеси, а також нейросіток у задачах класифікації, кластеризації, розпізнавання образів та нейро-фаззі-технологіях). Водночас розробляються СЗ на основі засобів логіко-символічних перетворень, що вивчаються в ШІ вже тривалий час і мають розвинений відповідний математичний апарат.

Форма подання знань суттєво впливає на характеристики та властивості системи, оскільки для маніпулювання будь-якими знаннями реального світу за допомогою ЕОМ необхідно здійснювати їх моделювання. При цьому враховують такі фактори, як однорідність подання та простота розуміння знань. Зазвичай при розгляді нескладних задач зупиняються на деякому середньому (компромісному) поданні знань. Розв'язання ж складних і великих задач потребує структурування й модульного подання знань.

Кожна СЗ є математичною моделлю деякої галузі прикладного, неформалізованого знання. Система понять і відношень такої моделі відображає систему понять і відношень прикладного знання, а залежність, яка існує в моделі, апроксимує відповідну залежність у ньому.

Створення СЗ передбачає розв'язання таких взаємопов'язаних проблем:

- формалізація відповідної галузі прикладних знань. Це важке завдання, оскільки розв'язується воно вручну, потребує спільної роботи фахівців-прикладників і математиків і полягає у виборі або *побудові концептуальної схеми моделі*. Розробка методології всіх цих операцій і становить зміст першої проблеми — *формалізації*;
- друга проблема — *подання знань* — пов'язана з розробкою формального апарату для опису способів їх фіксації в пам'яті ЕОМ;
- розробка теорії обчислень та інших перетворень, що здійснюються в побудованих моделях, становить третю проблему — *використання знань*;
- нарешті, четверта, технологічна проблема, розв'язання якої займаються системні програмісти, — це проблема *розробки засобів програмної підтримки моделей*, тобто створення БЗ і систем управління ними.

Основна увага в ШІ приділяється другій і третій з перерахованих проблем, причому провідна роль відводиться проблемі подання знань. На практиці її розв'язують разом з питаннями побудови концептуальних схем моделей подання знань (МПЗ), і багато хто вважає, що саме ця проблема і є основною для сучасного ШІ.

Можна виділити два напрями, що склалися в дослідженнях ШІ, які спираються на принципи організації людської пам'яті (і моделювання виведення на основі цих принципів), а також на логічне дослідження. Типовими моделями подання знань у **першому напрямі** є моделі, що ґрунтуються на використанні *продукцій, фреймів і семантичних сіток*. Результатом досліджень **другого напрямку** стало створення моделі подання знань на основі *логіки предикатів першого порядку*. При цьому серйозна увага приділялась поданню знань і генерації виведення з використанням теоретичної послідовної системи, в основному математичної формалізації та логічної повноти. Навпаки, перший напрям ґрунтується на розумінні процесу усвідомлення чогось людиною, що потребує скоріше виразності, аніж математичної чіткості.



У подальшому ці типові моделі подання знань були доповнені спеціальними для конкретних випадків засобами, але класифікація моделей при цьому залишилася незмінною. Мова, що використовується для розроблення систем на основі цих моделей, називається *мовою подання знань*.

Наступні розділи книги дають змогу повніше ознайомитися з особливостями типових МПЗ. Тут же обмежимося лише загальною їх характеристикою.

**Логічна модель** використовується для подання знань у системі логіки предикатів першого порядку та виведення висновків за допомогою силогізмів. У найпростішому випадку застосовуються логічні моделі подання фактів за допомогою предикатів з використанням атомарних формул:

ОРІЄНТАЦІЯ (РОБОТ, ДЕТАЛЬ): робот орієнтує деталь;

ПРОМИСЛОВИЙ РОБОТ (МП-9С): МП-9С — промисловий робот (ПР).

Логічна модель може мати вигляд *правильно побудованої формули* (ППФ), що включає квантори існування ( $\exists$ ) і загальності ( $\forall$ ): ( $\exists x$ ) [МАНІПУЛЯТОР ( $x$ )  $\vee$  БАГАТОЛАНКОВИЙ ( $x$ )]: деякий маніпулятор має багатоланкову структуру;

( $\forall x$ ) [ЗАХВАТ РОБОТА ( $x$ )  $\rightarrow$  ФУНКЦІЯ ( $x$ ), ЗАХОПЛЕННЯ Й УТРИМАННЯ]: всі захвати ПР призначені для захоплення й утримання об'єктів роботизації.

Відмітними рисами логічних МПЗ (в тому числі й наведених вище) є єдність теоретичного обґрунтування та можливість реалізації системи формально точних означень і виведень.

У *моделі правил* знання подаються сукупністю правил вигляду «ЯКЩО — ТО». Системи баз знань (СБЗ), які ґрунтуються на цій моделі, називаються *продукційними*. Вони містять два діаметрально протилежні типи — з прямими (діагностичними) та зворотними (задачі проектування) виведеннями. Так, у продукційних системах із *зворотними виведеннями* за допомогою правил будується дерево І/АБО, що пов'язує в одне ціле факти та висновки, причому оцінка цього дерева на основі фактів, які є у БД, зводиться до виведення.

Виведення можуть бути і двоспрямованими, коли спочатку оцінюється невеликий за обсягом масив одержаних даних і вибирається гіпотеза (за прикладом прямих виведень), а потім запитуються дані, необхідні для прийняття рішень про придатність цієї гіпотези. На основі цих виведень можна реалізувати більш потужну та гнучку продукційну систему зі зворотним виведенням.

Продукційні системи з *прямими виведеннями* містять три компоненти: 1) базу правил, що складається з набору продукцій (правил виведення); 2) БД, яка містить множину фактів; 3) інтерпретатор для одержання виведення на основі цих знань. Дві перші бази утворюють БЗ, а інтерпретатор відповідає МЛВ у вигляді циклу «розуміння — виконання», причому в кожному циклі виконується частина вибраного виведення оновлює БД. Як наслідок, зміст БД перетворюється від першорядного до цільового, тобто цільова функція (система) синтезується в БД.

Незважаючи на простий цикл вибору та виконання (або оцінки) правил, необхідність періодичного порівнювання (отожнювання) результатів прямого виведення в продукційних системах із зразком у БД призводить до того, що зі збільшенням кількості правил суттєво уповільнюється швидкість виведення. Отже, такі системи не здатні розв'язувати великомасштабні задачі.

Інакше кажучи, збільшення обсягу знань, необхідність розв'язання складних задач, виконання гнучких виведень або підвищення швидкості виведення потребують структурування знань, тобто шляхом попереднього групування відповідних правил у деякому стані процесу виведення можна обмежити діапазон їх вибору. Структу-

рування БД є ефективним і в тих випадках, коли об'єкти, для яких використовуються ці правила, мають ієрархічну структуру.

Дедалі частіше МПЗ, які ґрунтуються на застосуванні зазначених правил, використовуються в комбінації з **фреймовими моделями** — систематизованими у вигляді єдиної теорії психологічними моделями пам'яті людини та її усвідомленням.

Фрейм — це парадигма для подання знань з метою використання цих знань комп'ютером. Вперше фреймова модель була представлена М. Мінським як спроба побудувати фреймову мережу, або парадигму, з метою досягнення більшого ефекту розуміння [71; 39]. З одного боку, Мінський намагався сконструювати базу даних, що містить енциклопедичні знання, а з другого — він хотів створити найбільш наочну описову базу, яка містила б інформацію в структурованій і впорядкованій формі. Ця структура дала б змогу вводити інформацію в комп'ютер в більш гнучкій формі, маючи доступ до того розділу, який потрібен у поточний момент. Мінський розробив таку схему, в якій інформація міститься в спеціальних чарунках — так званих фреймах, які з'єднані в мережу або систему фреймів. Новий фрейм активізується з настанням нової ситуації. Відмітною його рисою є те, що він одночасно містить великий обсяг знань і водночас є достатньо гнучким для того, щоб його використати як окремий елемент БД, тобто одночасно підпорядковується як системному, так і ієрархічному принципу побудови знань.

Термін «фрейм» був найбільш популярний у середині сімдесятих років, коли існувало багато його тлумачень, відмінних від інтерпретації Мінського. Фрейми використовуються в системах штучного інтелекту (наприклад, в ЕС) як одна з поширених форм подання знань.

Важливим моментом у цій теорії є поняття *фрейму* — спеціальної структури даних для подання деякого концептуального об'єкта. Інформація про фрейм міститься в *слоті* — складовій фрейму. Всі фрейми взаємозв'язані й утворюють єдину фреймову систему, в якій органічно об'єднані декларативні та процедурні знання. Приклад фреймової системи, що описує склад ПР, показано на рис. 1.5.

Теорія фреймів стала поштовхом до розроблення кількох поширених тепер мов подання знань (FPL, FMS, KRL, KEE, KRINE, LOOPS та ін.) — середовища для розробок і досліджень у галузі інженерії знань. Цими мовами часто користуються фахівці цієї галузі. Зокрема, мовою FMS фреймова модель розглядається як ієрархічна структура даних із модульним поданням знань у вигляді певних форматів, які називаються фреймами. Кожний з них описує один концептуальний об'єкт, а конкретні властивості останнього та факту, що його стосується, відображають слоти — структурні елементи цього фрейму.

Оскільки для концептуального подання об'єктів властива ієрархічність, цілісний образ знань будується у вигляді однієї фреймової системи, що має ієрархічну структуру. В слот можна підставляти різні дані; специфічною процедурою виведення у фреймі є приєднувальна процедура, що використовується як слот. Мова подання знань, які базуються на фреймовій моделі, особливо ефективна для структурного опису складних понять і розв'язання задач, в яких відповідно до ситуації бажано застосовувати різні способи виведення. Слід додати, що фреймову систему без механізму приєднувальних процедур (отже, і механізму пересилання повідомлень) часто використовують як БД продукційної системи.

Більш наочними є мови, що спираються на **сіткову модель подання знань**. В основі такої моделі лежить ідея про те, що будь-які знання можна відобразити у вигляді сукупності об'єктів (понять) і зв'язків (взаємовідношень) між ними.

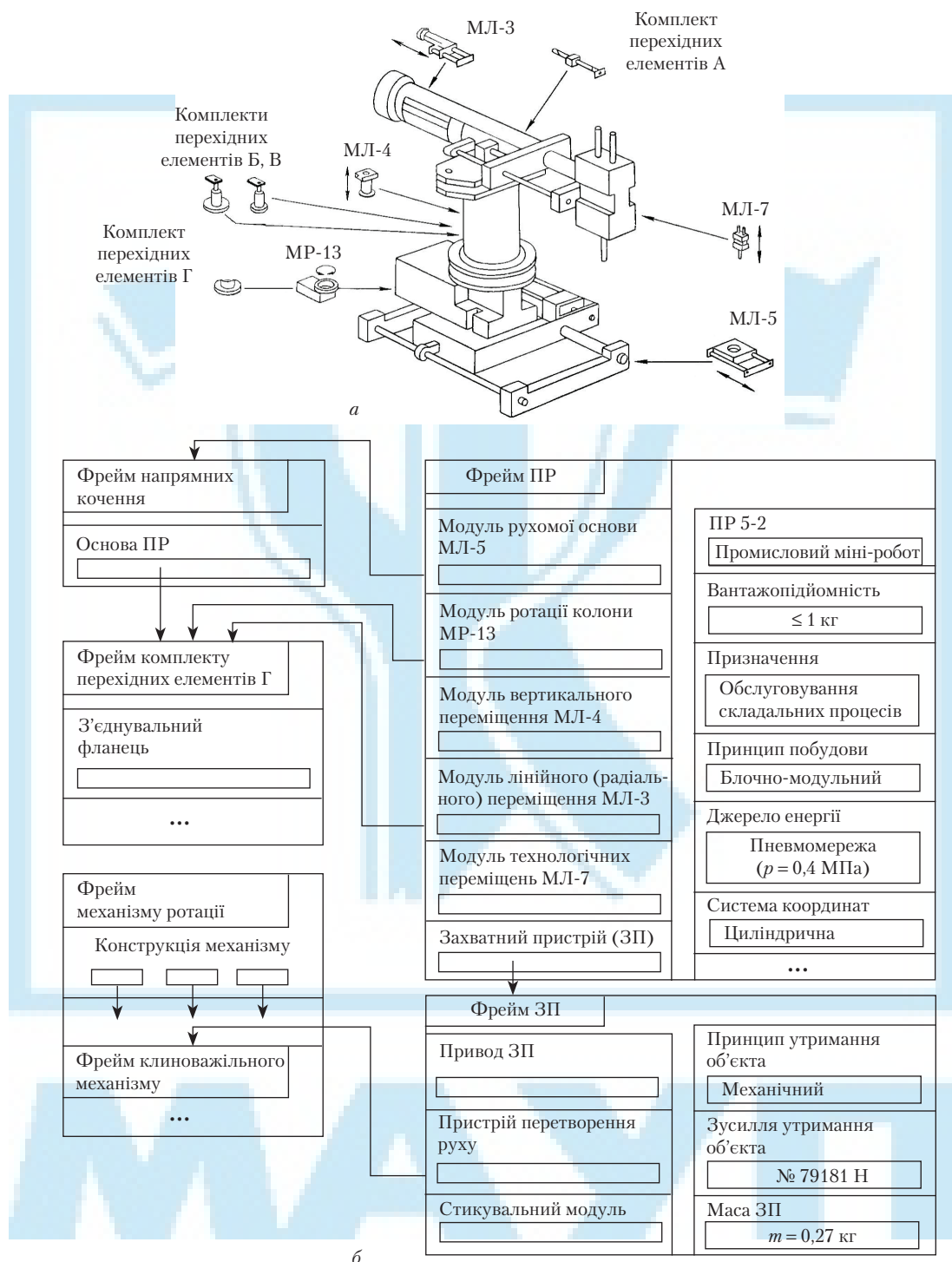


Рис. 1.5. Структура ПР типу ПР 5-2 (а) та приклад фреймової системи, що описує його склад (б)

Розглянемо, наприклад, текст, що містить деякі декларативні знання: «Зліва від верстата розташований приймальний бункер. Відстань до нього дорівнює двом метрам. Праворуч від верстата – бункер готової продукції. Він розташований поряд з верстатом. Робот переміщається паралельно верстату і бункерам на відстані 1 метр».

На рис. 1.6 показано сіткове подання сукупності знань, зафіксованих у цьому об'єкті, у вигляді **моделі семантичної сітки**. Поняття і об'єкти, що зустрічаються в тексті, зображені у вигляді сітки, а взаємовідношення – у вигляді дуг, що зв'язують відповідні вершини.

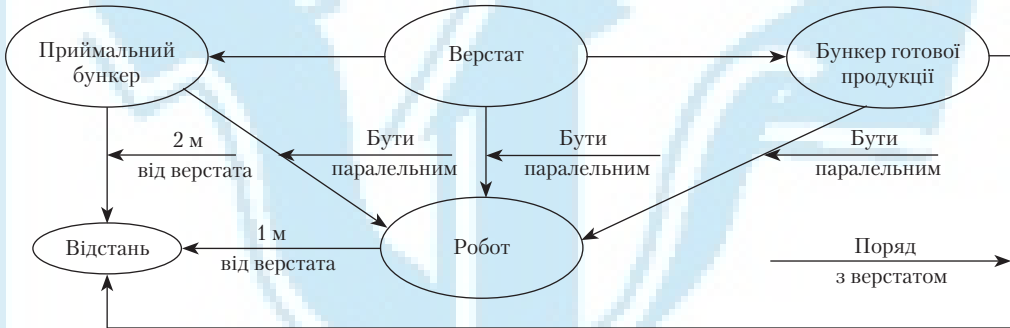


Рис. 1.6. Приклад сіткового подання сукупності знань

На рис. 1.7 зображено фрагмент семантичної сітки, що ілюструє речення «ПР типу ТУР-10 протягом часу від  $t_1$  до  $t_2$  виконував операцію маніпулювання об'єктом типу ступінчастого вала». На рисунку є дуги  $a$ ,  $b$ , «Виконавець», «Об'єкт», «Початок» і «Кінець», причому дуги  $a$  та  $b$  означають відповідно «підмножину» й «елемент» і належать до ієрархічних понять.

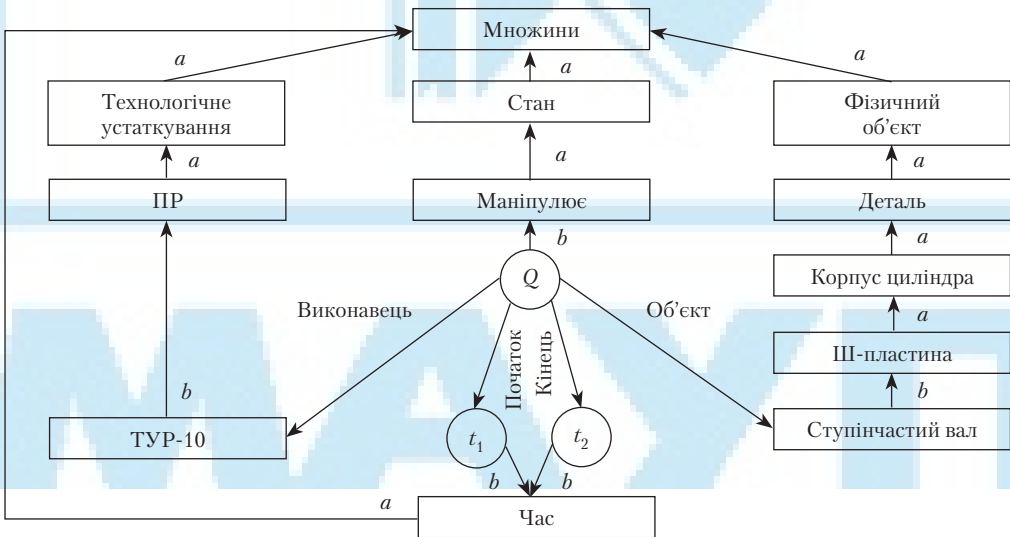


Рис. 1.7. Фрагмент семантичної сітки

Нейронні сітки, або **штучні нейронні сітки**, відображають розвиток моделей, які з'явилися завдяки спробам імітування механізму мислення людини, що вже з огляду на цю обставину є формальними МПЗ. Для певних класів задач, що стосуються ШІ (насамперед розпізнавання, інтелектуального управління динамікою складних механічних систем тощо), нейросіткові моделі часто забезпечують більш ефективні рішення, ніж традиційні символічні підходи.

Нейронна сітка є сукупністю нейронів і зв'язків між ними. Кожний нейрон виконує відносно просту функцію, але, будучи об'єднаними численними зв'язками в єдину структуру, всі вони разом утворюють систему, здатну розв'язувати складні задачі.

Вирізняють два типи нейроподібних сіток. У *сітках першого типу* вузлами є формальні елементи, що описують окремі нейрони. В *сітках другого типу* вузлами є формальні елементи, що відповідають не окремим нейронам, а особливим їх сукупностям – нейронним ансамблям. Під *нейронним ансамблем* розуміють таку сукупність взаємозв'язаних нейронів, яка збуджується повністю при збудженні окремої її частини. Багато дослідників вважають, що саме нейронний ансамбль, а не окремий нейрон і є функціональною одиницею мозку як системи, що забезпечує складну діяльність людини як живої істоти.

Як об'єкт модельного опису нейронний ансамбль відрізняється від окремого нейрона двома основними особливостями. Одна з них полягає в тому, що початкове збудження ансамблю змінюється неперервно, а не за законом «так – ні». Нейронний ансамбль можна описати як нелінійний перетворювач аналогової інформації, який задається набором певних статичних і динамічних характеристик.

Інша особливість нейронного ансамблю полягає в тому, що він може бути поставлений у відповідність до деякої змістової одиниці – поняття, образу тощо, тобто елементу, який бере участь у процесі розумової діяльності людини.

Отже, нейроподібна сітка, вузли якої відповідають ансамблям, стає сіткою із семантикою, або семантичною сіткою (СС). Ця обставина значно змінює підхід до проблем синтезу сіток і змістової інтерпретації процесів, що в них відбуваються.

У найзагальніших рисах результатом функціонування нейронної сітки є сигнал, який ідентифікує належність вхідного зображення до одного з кількох класів. Завдяки властивості навчатися сітка «зберігає» ці зображення у своїй «пам'яті» (тобто сітка здійснює категоризацію зображень).

Елементарна складова сітки – *нейрон* – має кілька входів і один вихід. Структуру нейрона зображено на рис. 1.8, а (це Adeline – *adaptive linear element* [105; 201]).

Елементи вхідного вектора множаться на вагові коефіцієнти  $W_1, W_2, \dots, W_n$ ; зважені значення підсумовуються й надходять до входу порогового елемента, двійковий вихід якого є виходом нейрона. У загальному випадку для відтворення нелінійностей у нейроні можуть застосовуватися не тільки ступінчасті (рис. 1.8, б), а й похилі функції (рис. 1.8, в).

Однією з найважливіших особливостей нейронної сітки є її здатність до навчання. В процесі навчання нейрона відбувається піднастроювання вагових коефіцієнтів таким чином, щоб він правильно класифікував максимальну кількість вхідних зображень.

Звичайно, навчання нейронної сітки здійснюється значно складніше, ніж навчання одного нейрона. На практиці використовують різні евристичні підходи, що

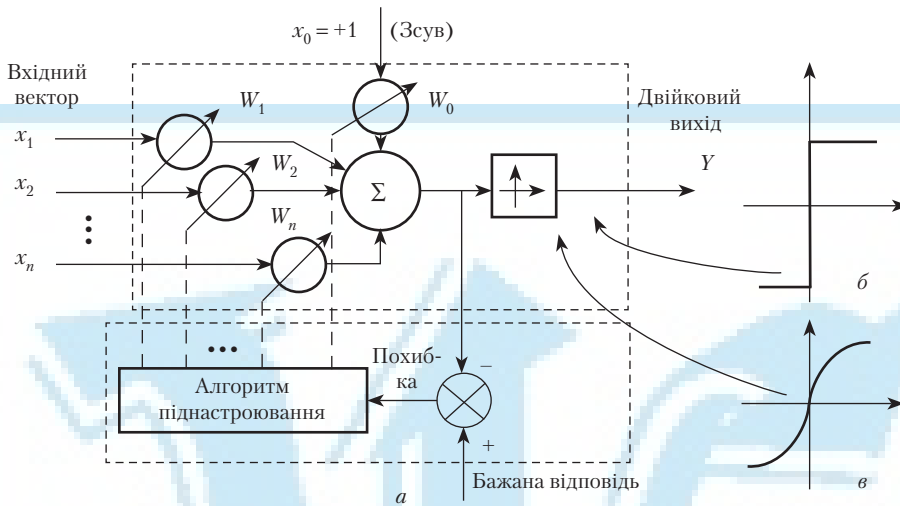


Рис. 1.8. Структура нейрона (а) та варіанти нелінійностей у ньому (б, в)

забезпечують наближене розв'язання задачі навчання нейронної сітки за прийнятний час. Одним з найвідоміших є метод *зворотного поширення*, який узагальнює на багатoshарові нейронні сітки підхід, що лежить в основі методу найменших квадратів. У методі зворотного поширення на основі мінімізації середньоквадратичних похибок між бажаним і фактичним виходами векторів розв'язку задачі послідовно піднастроюються ваги зв'язків.

Результати досліджень у галузі моделювання нейронних сіток значно розширили клас задач, що розв'язуються за допомогою нейроподібних сіток. Тепер до цього класу належать комбінаторні, оптимізаційні та інші задачі. Завдяки успіхам мікроелектроніки підготовлено технологічну базу для створення обчислювальних пристроїв, які здатні здійснювати паралельну обробку інформації. Ці два фактори зумовили появу нейрокомп'ютерів — ЕОМ, архітектура яких найкраще пристосована до розв'язання задач моделювання нейронних сіток. За нашого часу нейрокомп'ютери створюються у вигляді компактних приставок до персональних ЕОМ, суттєво підвищуючи їхні функціональні можливості.

Ефективним засобом подання та моделювання дискретних процесів є **сітки Петрі** (СП). Основні їхні властивості полягають у можливості відображення *паралелізму, асинхронності, ієрархічності* об'єктів, що подаються, простішими засобами. Тому використання сіток Петрі для дослідження ієрархічних дискретних систем, зокрема ГКІС, є найкращим.

Сітка Петрі — *причинно-наслідкова модель зображення подій*, що виникають у процесі роботи дискретної системи. *Процеси* подаються як множина подій та умов. *Події* — це дії, послідовність настання яких керується станами системи. *Стан* системи визначається сукупністю умов, серед яких виділяють: *передумови* — пов'язані з фактом настання події; *постумови* — пов'язані з фактом здійснення події.

Таким чином, СП зображається сукупністю пов'язаних подій та умов, що виникають в описуваній системі.

Для задавання СП найчастіше використовують такі способи, як теоретико-множинне визначення, графічне зображення та матричне подання.

1. Теоретико-множинне визначення СП.

Формально сітка Петрі  $N$  може бути задана у вигляді п'яти елементів:

$$N = (P, T, F, H, \mu_0),$$

де  $P = \{p_1, p_2, \dots, p_n\}$  – скінченна непорожня множина позицій (станів, місць);  $T = \{t_1, t_2, \dots, t_m\}$  – скінченна непорожня множина переходів (подій);  $P \cap T = \emptyset$ ;  $F: P \times T \rightarrow \{0, 1, 2, \dots\}$  – функція, яка, визначивши передумови здійснення подій, призначає для кожного переходу вхідну множину позицій  ${}^*t_j = \{p_i \mid F(p_i, t_j) \neq 0\}$ ;  $H: T \times P \rightarrow \{0, 1, 2, \dots\}$  – функція, яка, визначивши постумови, призначає для кожного переходу вихідну множину позицій  $t_j^* = \{p_i \mid H(t_j, p_i) \neq 0\}$ ;  $\mu_0: P \rightarrow \{0, 1, 2, \dots\}$  – початкове маркування – кількість маркерів у позиції  $p_i$ .

Необхідно зауважити, що перші чотири елементи визначають структуру системи, а останній елемент – динаміку поведінки системи, її початковий стан. Динаміка сітки пов'язана з рухом маркерів по позиціях у результаті спрацювань переходів, внаслідок чого маркуються нові позиції  $\mu(p_i)$ .

2. Графічне зображення СП.

Графічно СП – це дводольний орієнтований мультиграф, де: *дводольність* означає наявність двох типів вершин (позицій та переходів); *орієнтованість* означає, що всі дуги мають певний напрямок; *мультиграф* – дуги можуть мати кратність (вона позначається значенням над дугою або кількістю дуг).

Графічне зображення пов'язане з теоретико-множинним визначенням таким чином (рис. 1.9): позиції  $P$  відображаються колами; переходи  $t$  – рисками; функції  $F$  і  $H$  – орієнтованими дугами, кількість або кратність яких визначається значенням функцій; маркування  $\mu$  сітки відображається кількістю маркерів у позиціях.

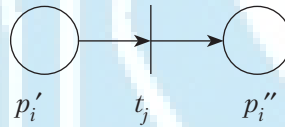


Рис. 1.9. Елементи сітки Петрі

3. Матричне подання СП.

Матричне подання – це аналітичний спосіб подання СП. У цьому випадку функції інцидентів та початкове маркування зображаються у вигляді матриць розміром  $[n \times m]$ :

$$F = [F_{ij} \mid i = \overline{1, n}; j = \overline{1, m}]; \quad H = [H_{ij} \mid i = \overline{1, n}; j = \overline{1, m}]$$

та вектором-стовпцем розміром  $[n \times 1]$ :

$$\mu_0 = [\mu_{0i} \mid i = \overline{1, n}],$$

де  $F_{ij} = F(p_i, t_j)$ ;  $H_{ij} = H(t_j, p_i)$ ;  $\mu_{0i} = \mu_0(p_i)$ .

4. Правила роботи СП.

Робота (функціонування) СП – послідовність спрацювання переходів, внаслідок яких відбувається зміна маркувань позицій.

Перехід  $t_i$  може спрацювати, якщо він збуджений. Перехід  $t_i$  вважається збудженим, якщо виконується така умова:

$$\forall p_i \in {}^*t_j : \mu(p_i) \geq F(p_i, t_j),$$

яка виконується до умови здійснення модельованої цим переходом події – у кожній вхідній позиції переходу кількість маркерів не менша кратності дуг, що їх з'єднують (рис. 1.10).

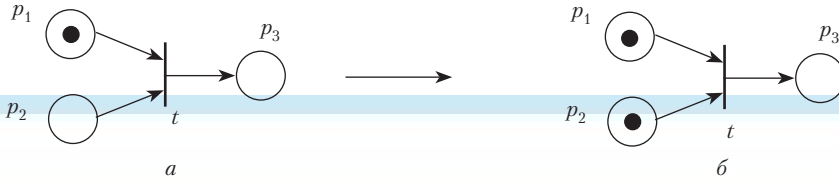


Рис. 1.10. Виконання умови збудження переходу:

*a* – умова не виконана; *б* – умова виконана

Тоді умова спрацювання збудженого переходу  $t_j$  має вигляд:

$$\forall p_i \in {}^*t_j \cup t_j : \mu'(p_i) = \mu(p_i) + H(t_j, p_i) - F(p_i, t_j),$$

тобто при спрацюванні збудженого переходу маркування  $\mu$  замінюється маркуванням  $\mu'$  за правилом: з вхідних позицій переходу забирається певна кількість маркерів, яка визначається функцією  $F(p_i, t_j)$ , а вихідні позиції переходу отримують іншу кількість маркерів, яка визначається функцією  $H(t_j, p_i)$  (рис. 1.11).

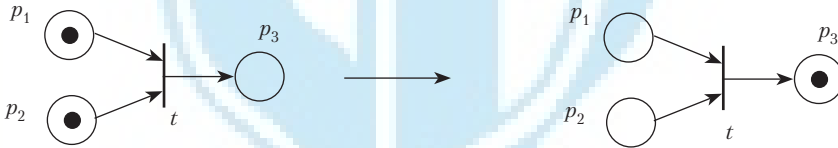


Рис. 1.11. Правило спрацювання збудженого переходу

Необхідно зазначити, що в разі використання матричного способу подання СП умова збудження переходу  $t_j$  має вигляд:

$$\mu \geq F \times U,$$

а умова спрацювання:

$$\mu' = \mu + (H - F) \times U,$$

де  $U = [U_j \mid j = \overline{1, m}]$  – вектор-стовпець розміром  $[m \times 1]$ , у якого всі елементи дорівнюють 0, крім  $U_j = 1$ .

У будь-якому стані СП може існувати декілька одночасно збуджених переходів. Але послідовність їх спрацювання не встановлена і може бути будь-якою без одночасного спрацювання переходів. Тому в СП визначають декілька прийнятних послідовностей спрацювань переходів, які породжують послідовності маркувань, що виникають. Це відображає паралелізм та недетермінізм СП. Таким чином, з функціонуванням сітки Петрі пов'язують дві послідовності: спрацювуваних переходів та виникаючих (досяжних) маркувань. Ці послідовності є взаємозв'язаними.

Два маркування  $\mu$  і  $\mu'$  вважаються безпосередньо досяжними, якщо у функціонуванні сітки існує перехід  $t_j$ , спрацювання якого переводить сітку з  $\mu$  у  $\mu'$ , тобто  $\mu \xrightarrow{t_j} \mu'$  (рис. 1.12).

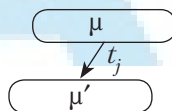


Рис. 1.12. Безпосередньо досяжне маркування



Два маркування  $\mu$  і  $\mu'$  вважаються досяжними, якщо у функціонуванні сітки існує послідовність переходів  $G = (t_{j_1}, t_{j_2}, \dots, t_{j_k})$ , яка переводить сітку з  $\mu$  у  $\mu'$ , а саме  $\mu \rightarrow \mu_1 \rightarrow \mu_2 \rightarrow \dots \rightarrow \mu_k \rightarrow \mu'$  (рис. 1.13).

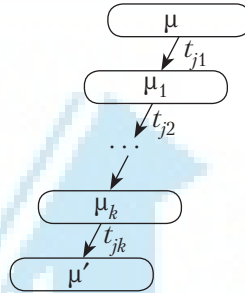


Рис. 1.13. Послідовність досяжних маркувань

Таким чином, формально функціонування сітки подається: мовою СП  $L(N)$  — множиною послідовностей спрацьовуючих переходів; множиною досяжності  $R(N)$  — множиною маркувань, досяжних з початкового маркування.

Ці послідовності об'єднуються у рамках єдиної моделі зображення роботи сітки — графа досяжності — орієнтованого графа, вершинами якого є маркування з множини  $R(N)$ , а дугами — спрацьовуючі переходи з  $L(N)$  (рис. 1.14, а). Граф з відсутніми циклами є *деревом досяжності* (рис. 1.14, б).

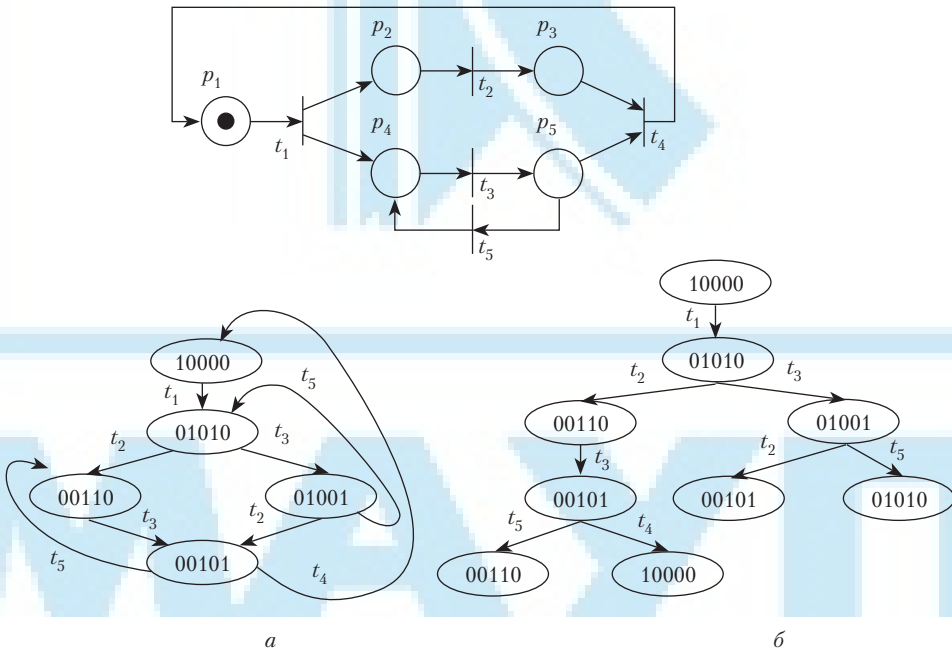


Рис. 1.14. Граф (а) і дерево (б) досяжності сітки Петрі

## 1.9. Мовні засоби для подання знань в інтелектуалізованих системах

**Загальна характеристика мов подання знань.** Не вдаючись у специфіку окремих мовних засобів, відзначимо одну загальну особливість застосування їх у системах подання та використання знань. Ця особливість ілюструється місцезнаходженням мов подання знань у процесі розроблення систем ШІ (СШІ). Уявімо собі узагальнено три методи *A*, *B* та *C*, які застосовуються при розробленні прикладних СШІ (рис. 1.15).

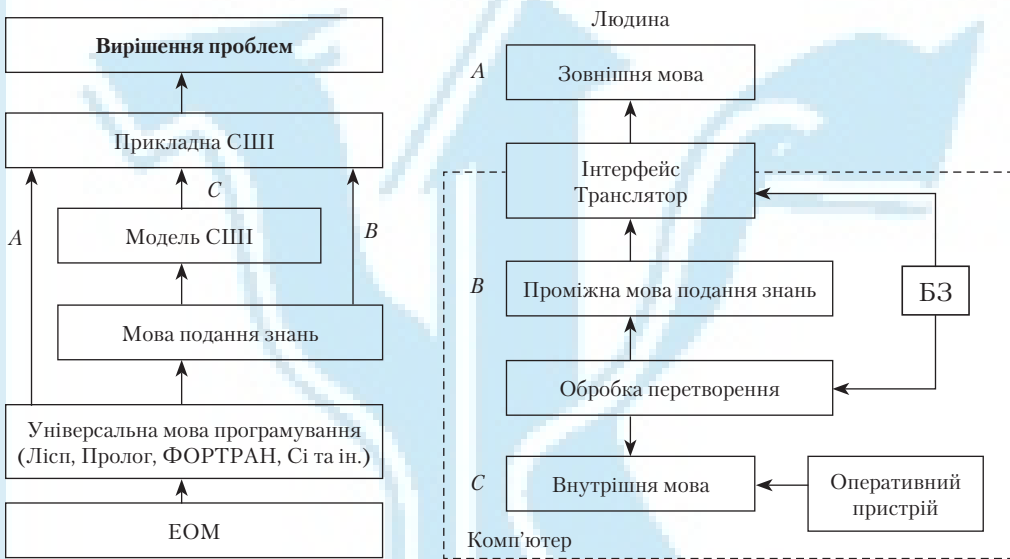


Рис. 1.15. Місце мовних засобів у процедурах подання знань

У методі прямого розроблення *A* використовується універсальна мова програмування і передбачається спочатку створення механізму управління БЗ із наданням системі специфічних функцій конкретної проблемної галузі, тобто реалізації спеціалізованої СШІ. Цей метод потребує значного часу на його реалізацію.

Метод *B* ґрунтується на використанні мови подання знань або мови розроблення СШІ (зокрема, він застосовується при створенні СШІ на основі продукційних систем). Цей метод вирізняється простотою і за наявності оболонки (toy system) потребує на розроблення СШІ порівняно небагато часу. Проте метод *B* характеризується деякою невідповідністю методам подання знань і управління виведенням, внаслідок чого він не став універсальним.

Класичним методом із застосуванням мови подання знань є метод *C*. Фреймова МПЗ характеризується значною гнучкістю: вважається можливою комбінація декларативних і процедурних знань в одній одиниці подання знань — фреймі; можливими є ієрархічна побудова БЗ згідно зі ступенем абстракції поняття, а також реалізація будь-якої системи виведення з використанням об'єктно-орієнтованого методу управління виведенням — обміну повідомленнями.

Однак при застосуванні фреймової моделі виникає ряд проблем. Насамперед треба відповідно до прикладної задачі побудувати модель СШІ з використанням багаторівневої ієрархічної структури. Другою не менш важливою проблемою, пов'я-

заною з приєднувальною процедурою, є вибір фреймової структури, оскільки мови цього рівня орієнтовані на фахівців (тобто є середовищем для досліджень і розробок у галузі ШІ та інженерії знань).

Інакше кажучи, мови фреймового типу мають характер універсальних мов програмування вищого рівня, ніж Лісп, Пролог. Системи ж обробки знань здебільшого бувають написані мовами ШІ типу Лісп, Пролог або Пленер, і нерідко для того, щоб підвищити швидкість виконання програм і використовувати не найновіший комп'ютер, їх описують мовами Сі та ФОРТРАН. Отже, залишається суперечність між необхідним рівнем інтелекту для опису фреймової структури та прагненням обмежитись при цьому рамками простих методів подання знань.

**Мовні засоби для розроблення експертних систем.** Створення ЕС — складна і трудомістка процедура, причому обсяг витрачених зусиль суттєво залежить від умінь створювачів орієнтуватись у досить широкому на сьогодні арсеналі інструментальних засобів.

Як зазначалося (див. п. 1.9), програмні засоби розроблення та реалізації систем із ШІ можна поділити на дві групи: 1) процедурні мови; 2) декларативні мови. Всі без винятку мови, орієнтовані на традиційні ЕОМ, є *мовами першої групи*. А мови *другої групи* виникли у зв'язку з обробкою знань.

Перетворення мов — це не що інше, як переклад, і перекладач (у даному разі комп'ютер) має бути дуже добре обізнаним з обома мовами (або хоча б повинен розуміти їх). Мова, одержана як наслідок перетворення, є машино-орієнтованою мовою і тому з нею проблем не виникає, але є необхідність забезпечити можливість обробки ЕОМ ще однієї мови, а саме тієї, яку людина використовує для опису задач. Для цього вводиться нова машино-орієнтована мова, що відрізняється семантикою від традиційних таких мов. При цьому необхідно, щоб машина розуміла зовнішнє подання проблеми (зовнішню мову), яке використовується людиною (на рис. 1.15 нова машино-орієнтована мова позначена блоком *B* і є мовою опису знань).

Для зручності висловлювання мова (блок *A*), що є мовою, якою людина (перебуваючи поза комп'ютером) описує знання та проблему для комп'ютера, називається *зовнішньою*; мова подання знань (блок *B*) — *проміжною*, а процедурна мова (блок *C*), одержана при перетворенні проміжної мови для подальшої обробки на ЕОМ, — *внутрішньою*.

**Процедурні мови** відображають мовний напрям розвитку сучасних ЕОМ; вони дають змогу описувати алгоритми за допомогою процедур, а потім — зміст проблеми. В цих мовах відсутня безпосередня відповідність між об'єктами, які описуються процедурами, та концепціями реального світу. Крім того, є посилення на те, що відповідність між об'єктами та явищами реального світу і процедурним їх описом не має зв'язку із семантикою самої мови. Приймається, що опис відповідності реального світу та світу комп'ютера беззастережно встановлюється людиною.

Семантика **декларативних мов** визначається шляхом установаження відповідності концептів описуваного світу або ж їх взаємозв'язків при описі на мовному рівні. Типовою декларативною мовою є природна мова. Декларативними також вважаються мови ШІ кількох типів, наприклад СС.

Для того щоб чіткіше уявити перетворення різних мов та особливості, якими супроводжуються такі перетворення, можна ввести поняття *рівнів подання інформації*. Рівнем тут вважається логічний рівень подання інформації, який семантично еквівалентний рівню в ієрархії високорівневих мов. Однак відстань між рівнями, про які тут говориться, більша, ніж у високорівневих мовах.

Окрім того, що рівень є логічним поняттям подання інформації, він відповідає певним апаратним засобам, які потрібні для його реалізації. Наприклад, сучасні ЕОМ з архітектурою Тьюринга фон Неймана мають щонайменше чотири рівні, які відповідно називаються «висловлювання – бітовий рівень», «знакова інформація – регістровий рівень», «одиничні дії – рівень команд», «процедурне подання – рівень програми».

Кожний рівень описується парою «логічне подання інформації – внутрішньо-машинне подання». На всіх рівнях реалізується міжрівневий механізм, який перетворює інформацію більш високого рівня в інформацію нижчого рівня. Як правило, чим вищий рівень подання, тим ясніше людина розуміє його інформацію, а чим нижчий рівень, тим легшою є її обробка ЕОМ. Таким чином, рівень, на якому можлива взаємодія людини та машини, залежить від проектування механізму перетворення відповідного рівня в подання нижчого рівня.

Декларативне подання утворює ще один, зовсім відмінний від розглянутих рівень. По-перше, воно ґрунтується на методах символної обробки інформації; по-друге, остання подається через одержаний дедуктивно опис проблеми; по-третє, подання знань і виведення повністю розміщується на цьому рівні і вважається, що на ньому зосереджена вся обробка інформації. Але в системах з такою простою структурою неможливо виконувати алгебраїчні операції над дійсними числами та операції булевої алгебри, і тому їх важко суміщувати зі звичайними ЕОМ.

Для усунення зазначеного недоліку експертні системи слід комбінувати зі структурами, в яких рівень подання знань накладається на програмний рівень. Така структура використовується в нових механізмах перетворення інформації, розроблених з метою зниження рівня подання знань до рівня процедурного їх подання.

Сучасні мови подання знань можна зарахувати до найнижчого підрівня декларативного рівня, тобто сподівання на те, що в недалекому майбутньому з'являться мови більш високого рівня, цілком обґрунтоване. Для вибору найбільш придатної мови спочатку треба з'ясувати умови, яким повинна відповідати декларативна мова. Щоб забезпечити можливість ієрархічного подання знань на кожному з рівнів, насамперед необхідно створити механізм їх обробки, який перетворює верхнє подання знань у наступне, причому цей механізм повинен бути побудований на основі інформації та механізму обробки знань нижнього рівня, з яким межує цей рівень.

Як уже згадувалося, системи з ШІ є деяким програмним комплексом або технічним засобом (наприклад, таким є інтелектуальний робот, в якого апаратно «зашито» програму), що дає змогу розв'язувати виробничі задачі на рівні людини-оператора (експерта). Але очевидно, що будь-яку програму можна написати машиноорієнтованою мовою або універсальною мовою високого рівня (ПЛ/1, СІ, БЕЙСІК, Алгол, Ада, ФОРТРАН, Паскаль та ін.).

При цьому постає цілком слушне запитання: навіщо розглядати спеціалізовані засоби, для вивчення яких потрібен певний час, якщо універсальною мовою високого рівня володіє практично будь-який програміст? Відповідь на це запитання дає практика: процес програмування системи з ШІ на спеціалізованих засобах (мовах високого рівня) потребує в два-три рази менше часу, аніж на універсальних. Однак завжди слід пам'ятати, що параметри ефективності (обсяг пам'яті та швидкодія) систем із ШІ, реалізованих на основі спеціалізованих засобів, здебільшого нижчі, ніж при реалізації цих систем машинною мовою (асемблером).

Сьогодні багато фахівців погоджуються, що об'єднання засобів ШІ з традиційними засобами обчислювальної техніки є ключем до успішного розроблення застосувань ЕС. Для цього необхідно, щоб ЕС без зайвих ускладнень можна було зв'язувати в одне ціле з прикладними програмами, що виконуються на універсальних ЕОМ і високопродуктивних персональних комп'ютерах.

### 1.10. Порівняльна характеристика сучасних способів подання знань

Наведені вище способи подання знань, на перший погляд, не пов'язані між собою. Це, звичайно, неправильно і відображає лише той факт, що всі вони розроблялися за різними принципами. Проте при цьому загальною залишалася єдина мета — подання знань і їх використання при вирішенні різних проблем. Тому можна сказати, що взаємна відмінність способів подання знань полягає скоріше в зовнішньому їх вигляді, а суть не змінюється (хоча, природно, рівень системного розвитку для відповідного способу подання знань буде свій).

Якщо задатися питанням, якою є мета, що її повинна досягти СПЗ, то відповідь на нього не завжди буде зрозумілою. Раніше інженерією знань ставилося трохи абстрактне завдання, що полягало в успішному застосуванні знань. У більш чіткому вигляді задача обробки знань полягає в підтримці інтелектуальної діяльності людини. Для ефективного розв'язання такої задачі необхідно вибрати такий спосіб подання знань, який був би адекватним значущості мети в кожному конкретному випадку.

Інакше кажучи, технологія здобуття, подання й обробки знань із подальшим їх виведенням повинна, з одного боку, задовольняти поставлену користувачем мету, а з другого — не призводити до надмірних для заданих обмежень складнощів під час реалізації системи. Для цього з'ясуємо деякі очевидні особливості основних способів подання знань.

**Продукційні правила.** Подання знань за допомогою продукційних правил дуже просте, а виведення їх, які виконуються на основі формалізму виду «ЯКЩО — ТО», наочні, доступні й аналогічні всім тим силогізмам, з якими ми зустрічаємось у численних життєвих ситуаціях. До того ж яскраво виражена модульність цих правил дає змогу створювати нові знання, не вдаючись у зміст інших.

Нарешті, забезпечується можливість побудови виведень в однорідній формі, завдяки чому створювані системи стають простими та легко зрозумілими.

З іншого боку, ця їх простота є наслідком небагатофарбового подання всіх описів, спрямована вона на прості, однорідні за властивостями задачі і призводить до різкого падіння ефективності вирішення таких проблем, які складаються з кількох різнорідних задач. Але навіть і в цьому разі не вдається побудувати систему для управління знаннями як єдиним цілим, і тому цей процес здійснюється людиною.

З підвищенням складності проблеми стає складнішим і управління процесами перетворення знань. Зокрема, при накопиченні або щільній обробці знань за тривалий період може виникнути ситуація, коли (чи то через участь людини, яка обробляє або накопичує знання, чи то через тривалий час) з'являються знання, що суперечать раніше здобутим знанням.

**Семантичні сітки.** Це узагальнена назва методів опису об'єктів з використанням сіток і вона ж є назвою одного із способів подання знань.

Характерною особливістю СС є наочність знань як системи, кожне окреме знання розглядається як деяке відношення між сутностями та поняттями і формально,

як і в продукційних системах, визначені заздалегідь, і вже існуючі всередині системи знання можна нарощувати незалежно від зберігання їх модульності. Разом з тим усі знання, що охоплюють однакові сутності та поняття, можна зобразити у вигляді відношень між різноманітними вузлами, які описують ці сутності, що й дає змогу говорити про легкість розуміння такого подання знань.

На основі СС здійснюються виведення, але для цього потрібні спеціальні алгоритми. В продукційних системах виведення визначаються для обмеженого формалізму «ЯКЩО – ТО», тому алгоритми також формалізовані, але, незважаючи на свою простоту, вони досить точні. Оскільки СС є узагальненою назвою СПЗ, а також враховуючи те, що форма подання знань сітками не обумовлюється, для кожного конкретного формалізму визначаються свої власні правила виведення, завдяки чому підсилюється елемент довільності, внесений людиною. Виведення ж, досить ретельно не перевірені, ховають у собі загрозу створення суперечностей між результатами виведення та знаннями, на яких вони ґрунтуються.

**Логіка предикатів.** Мови подання знань логічного типу широко використовувалися на ранніх стадіях розвитку інтелектуальних систем через те, що ця логіка відзначається високим рівнем модульності знань й водночас дає змогу одержати єдину систему їх подання, в якій логічно роз'яснюються властивості знань як одного цілого. Отже, за допомогою логіки предикатів можна, визначаючи довільно знання, з'ясувати, чи існують суперечності між новими та існуючими знаннями. Але невдовзі мови подання знань логічного типу були усунуті (або в усякому разі сильно посунуті) мовами інших типів через те, що логіка предикатів завдяки зберіганняю властивості цілісності має такі недоліки, як надмірний рівень формалізації подання знань, труднощі їх прочитання, досить низька продуктивність обробки. А оскільки в логіці предикатів усі відношення описуються предикатами, при їх комп'ютерній обробці не можуть повною мірою виявитись переваги, які має структура даних, що також погіршує ефективність обробки початкових знань.

**Фреймові системи.** Фреймова модель подання знань характеризується значною гнучкістю: вважається можливою комбінація декларативних і процедурних знань в одній одиниці подання знань – фреймі; можливими є ієрархічна побудова баз знань згідно зі ступенем абстракції поняття, а також реалізація будь-якої системи виведення з використанням об'єктно-орієнтованого методу управління виведенням – обміну повідомленнями. Як і продукційні, фреймові системи визначають форму подання знань і відрізняються від інших СПЗ тим, що надають користувачеві більший ступінь свободи. Такі системи можуть використовуватися не тільки для опису знань, а й для написання алгоритмів їх виведення. З точки зору користувача, який дуже добре розуміється на обробці знань, це дає певну перевагу, але для звичайного користувача є великим навантаженням.

Можна сказати, що фреймові системи як поширення традиційних процедурних систем дають змогу по-різному будувати їх. Однак та обставина, що у фреймових системах знання задаються, по суті, процедурами (фреймами, які, у свою чергу, складаються зі слотів), порівняно з іншими методами процес здобуття знань ускладнюється, а можливості динамічної адаптації фреймової системи до змін зовнішнього середовища збіднюються. Та й відношення між фреймами в складних проблемах стають складнішими. Отже, однією з труднощів подання знань і мови фреймів є відсутність формальної семантики. Це ускладнює порівняння властивостей подання знань різних мов фреймів, а також повне логічне пояснення мови фреймів.

Нарешті, зв'язок слотів з процедурами, а також те, що вони передають один одному управління завдяки обміну повідомленнями, означає, що структура фреймової системи описує структуру управління процедурою обробки знань. Внаслідок цього фреймова система зводиться до форми подання одиниць керуючих структур і одиниць структур даних у загальних фреймах, через що дуже часто вдаються до досить надуманих прийомів. При цьому втрачається легкість інтуїтивного розуміння процесів подання та виведення знань, що є особливістю цієї системи.

Такі обмеження припустимі для порівняно нескладних проблем, але якщо вони ускладнюються, то опис і управління у фреймовій системі стають важчими, ніж у традиційних процедурних системах подання знань, і тоді про фреймову систему як про універсальну систему обробки даних говорити складно.

Отже, для більшості способів подання знань, які сьогодні використовуються, характерним є різке ускладнення опису проблеми при збільшенні її масштабів, і цей недолік важко усунути, не позбавившись одночасно переваг відповідних методів. Єдиний виняток – логіка предикатів, хоча й вона не позбавлена цього недоліку. Але якщо в інших способах при ускладненні проблеми фактично взагалі зникає можливість управління знаннями (оскільки одержаний при цьому результат не є гарантією його достовірності), то логіка предикатів дає змогу гарантувати одержання надійних результатів навіть при зниженні ефективності подання знань. Це означає, що логіка предикатів дає єдину СПЗ, за допомогою якої можна вирішити проблему, крок за кроком задаючи початкову інформацію.

**Схожість способів подання знань.** Хоча всі описані вище способи подання знань і відрізняються, на перший погляд, якісно, але в них є багато чого спільного і відмінність їх має концептуальний характер. Справді, подання знань за допомогою логіки предикатів дуже близьке до продукційних правил, оскільки, інтерпретуючи правило продукцій як речення логіки предикатів, без особливих труднощів можна замінити його предикатами. За таких дій вираз «ОРІЄНТАЦІЯ (РОБОТ, ДЕТАЛЬ)», описаний предикатною формулою, відповідає відношенню «РОБОТ – ОРІЄНТУЄ – ДЕТАЛЬ» у СС.

У загальному випадку кожен дугу СС можна задати предикатами як відношення між сутностями, що описують вузли на кінцях дуги. Можливість опису деякого подання знань різними методами означає, що останні є рівнозначними. Проте ця попарна рівнозначність відношень не означає, що такі способи подання знань, як продукційні правила, логіка предикатів, СС, дають змогу дістати рівнозначні СПЗ. Це пояснюється тією обставиною, що подання знань однією формулою не можна передати іншою формулою, а також тим, що відрізняються самі ідеї і теоретичні системи, які становлять основу цих формул, що й впливає на методи виведення знань.

Тому найкраще здійснити декомпозицію множини знань і підготувати джерела знань (ДЗ) для кожної окремої проблеми. Аналогія цієї ідеї – декомпозиція СС.

Такий підхід є звичайним і для фреймової структури. Фрейми, що розглядаються як множина знань із сильним взаємозв'язком, відповідають ДЗ. Суттєва відмінність між фреймами та моделлю дошки оголошень (ДО), що реалізується за правилами продукцій, а також розподіленими СС полягає у початкових умовах: і в сітках, і в ДО управління великою кількістю ДЗ виконує сама система, тоді як у фреймовій системі цей процес із самого початку здійснюється самим користувачем за допомогою механізму обміну повідомленнями.

Водночас реально приклади успішного функціонування ДО обмежуються випадками дуже незначної взаємодії окремих проблем і простого управління ДЗ. У решті випадків управління ДЗ стає складним, причому його алгоритми дуже часто навіть важко чітко собі уявити.

Згадана тенденція означає, що велике поширення систем обробки знань приводить у підсумку до процедурного опису проблем. І хоча ця тенденція справді спостерігається, проявляти з цього приводу песимізм не варто. Це скоріше свідчить про те, що окремі компоненти систем, які з'явилися донині, побудовані на основі порівняно простих концепцій. Подальшими напрямками вирішення зазначеної тенденції є: модульність знань; розвиток систем великого масштабу, а також систем з різноманітними ДЗ. Для реалізації цих напрямів потрібна нова методологія, деякі аспекти якої розглядаються нижче (див. розд. 2 і 6).

### Контрольні завдання

1. Прокоментувати гносеологічний, інтенціональний та екстенціональний аспекти, що сприяли розвитку робіт в напрямку створення ШІ.
2. Дати означення інтелекту й охарактеризувати складові ШІ.
3. Навести необхідну та достатню умови успішного розвитку та впровадження систем із ШІ.
4. Охарактеризувати вітки досліджень, пов'язаних зі створенням машинного інтелекту і штучного розуму.
5. Розкрити зміст еволюції методології систем із ШІ.
6. Навести і прокоментувати чотири основні напрями досліджень у галузі ШІ.
7. Дати детальне пояснення найпоширеніших проблем інтелектуалізації: розпізнавання, мислення, обчислювальних задач.
8. Проаналізувати стадії індустріалізації, комп'ютеризації та інформатизації як еволюційний процес науково-технічного прогресу.
9. Дати уявлення про взаємозв'язок інформації та знань і розкрити зміст основних категорій знань: фактів, евристик, метазнань і правил.
10. Охарактеризувати роль декларативних і процедурних знань у складі ЕС.
11. Навести означення й охарактеризувати зміст термінів «база знань» і «база даних» і розкрити екстенціональну та інтенціональну складові БЗ.
12. Дати означення і розкрити функції основних складових ЕС.
13. Навести і прокоментувати основні режими використання ЕС.
14. Навести й охарактеризувати основні ознаки інженерії знань.
15. Розкрити проблемний характер формалізації, подання й використання знань.
16. Дати означення і розкрити зміст поняття формальної системи знань.
17. Пояснити, в чому полягає зміст формального доведення.
18. Розкрити зміст основних типів правил виведення — продукцій та переписувань.
19. Дати тлумачення інтерпретації і твердження і розкрити взаємозв'язок між ними.
20. Перелічити і дати загальну порівняльну характеристику основних способів подання знань.
21. Дати загальну характеристику мов подання знань і мовних засобів для розроблення ЕС.



## ТИПОВІ МОДЕЛІ ПОДАННЯ ЗНАТЬ У СУЧАСНИХ ЕКСПЕРТНИХ СИСТЕМАХ

---

---

*Нема нічого висловленого, що було б висловлено вперше.*

Теренцій

### 2.1. Логічні моделі подання знань

Апарат математичної логіки ґрунтується на спеціальних математичних моделях, які називаються формальними системами. Така система визначається у такий спосіб.

**Означення 2.1.** *Логічна модель* – це формальна система, що визначається як  $S = (E, R, A, V)$ , де  $E$  – алфавіт (множина) базових елементів;  $R$  – множина синтаксичних правил, які дають змогу будувати з елементів множини  $E$  синтаксично правильні висловлювання (формули);  $A$  – множина аксіом;  $V$  – множина правил виведення, що дають змогу з деякого набору висловлювань будувати іншу множину.

Формальні системи при цьому являють собою деяке логічне числення, і всі знання про предметну сферу описуються у вигляді формул цього числення або правил виведення. Опис у вигляді формул дає можливість подавати декларативні знання, а правила виведення – процедурні знання.

В основі математичної логіки лежать *числення висловлювань* і *числення предикатів*. Загалом числення (логіка) предикатів є класичним механізмом подання знань у дослідженнях ШІ, причому ця логіка є поширенням логіки висловлювань, оскільки основним об'єктом тут є змінне висловлювання (предикат), істинність і помилковість якого залежать від значень його змінних. Зупинимось коротко на основних положеннях, а також правилах виведення логіки висловлювань і логіки предикатів першого порядку.

У математичній логіці висловлюванням називається таке речення, яке є істинним (I) або помилковим (II).

**Числення (логіка) висловлювань.** У логіці висловлювань символи  $P, Q, D$  та інші, що використовуються для позначення висловлювань, називаються *атомарними формулами* і являють собою **алфавіт логічної СПЗ** (перша умова відповідності формальній системі). З атомарних висловлювань за допомогою логічних відношень, або *сентенціональних зв'язків*  $\neg$  (заперечення, або «ні»; іноді позначається рискою над символом  $\bar{P}$ ),  $\wedge$  (кон'юнкція, або «і»; позначається також & або  $\cdot$ ),  $\vee$  (диз'юнкція, або «або»),  $\vdash$  (твердження, що означає « $\vdash B$ », тобто «за  $B$ »),  $\in$  (знак належності),  $\subset$  (знак включення, або охоплення),  $\supset$  (імплікація, або висновок; позначається ще  $\rightarrow$  і означає «якщо ..., то ...»),  $\leftrightarrow$  (рівнозначність, що означає «тоді і тільки тоді»),  $\sim$  (еквіваленція, що означає « $P$ , якщо і тільки якщо  $Q$ »), будуються складові висловлювань, тобто оператори і правила їх застосування і є **механізмом створення складних утворень** – складних висловлювань (друга умова

відповідності формальній системі). Вираз у вигляді висловлювання або складового висловлювання називається *правильно побудованою формулою* (ППФ). Тут значимо тільки таке: *правильно побудована формула* буде тотожно істинною тоді і тільки тоді, коли внесення в опис (формальне подання певних знань з конкретної ситуації) заперечення цієї ППФ перетворить цей опис на непослідовний. Іншими словами, опис буде непослідовним, якщо в ньому можна вивести суперечність вигляду:  $A \ \& \ \neg A$ .

Нехай  $G$  – задана пропозиційна (пропозиція – висловлювання) формула, а  $B_1, B_2, \dots, B_n$  – її атомарні формули. Інтерпретацією формули  $G$  є таке приписування істинних значень атомарним формулам  $B_1, B_2, \dots, B_n$ , за якого кожній формулі  $B_i$  приписано або І, або ІІ (тобто або істину, або помилку, але не обидві їх разом) – саме в цьому полягає сутність *третьої умови відповідності* логічної СПЗ формальній системі, тобто *аксіоматичність тверджень* (або висловлювань).

Для перевірки істинності складних утворень використовують таблиці відповідностей – *таблиці істинності*, які пов'язують сентенціональними зв'язками («ні» – заперечення, «і» – кон'юнкція, «або» – диз'юнкція, «якщо ..., то» – імплікація, «якщо і тільки якщо» – еквіваленція) складові, що входять у висловлювання.

	$P$		0	0	1	1	
	$Q$		0	1	0	1	
$P$	0	1	$P \wedge Q$	0	0	0	1
$\bar{P}$	1	0	$P \vee Q$	0	1	1	1
	$P \rightarrow Q$		1	1	0	1	
	$P \sim Q$		1	0	0	1	

Якщо формула є істинною в усіх можливих її інтерпретаціях, то вона є *загальноозначуваною формулою* (тавтологією) і її позначають знаком  $\blacksquare$  (або  $\models$ ). Іншими словами, якщо формула є тотожно істинною, тобто такою, яка набуває значення 1 при будь-яких значеннях її компонентів, то вона є тавтологією.

**Приклад 2.1.** Прикладом тавтології може бути висловлювання:

«Якщо впровадити нову роботизовану технологію ( $P$ ), то якість продукції поліпшиться ( $Q$ ). При поліпшенні якості продукції ( $Q$ ) її збут збільшиться ( $R$ ). Нова технологія впроваджена ( $P$ ). Отже, збут продукції збільшився ( $R$ )». Таке висловлювання подається формулою

$$(P \rightarrow Q) \wedge (Q \rightarrow R) \wedge P \rightarrow R. \quad (2.1)$$

Якщо ж формула помилкова при всіх її інтерпретаціях, то кажуть, що вона є *суперечливою*, і її позначають знаком  $\square$ . Суперечлива формула є нездійснюваною.

Сказане означає: якщо хоча б один з членів диз'юнктивної нормальної форми дорівнюватиме 1, то відповідна їй формула є тавтологією. Так, для нашого прикладу 2.1 маємо:

$$\begin{aligned} (P \rightarrow Q)(Q \rightarrow R)P \rightarrow R &= (\bar{P} \vee Q)(\bar{Q} \vee R)P \rightarrow R = (\bar{P}\bar{Q} \vee \bar{P}R \vee QR)P \rightarrow R = \\ &= PQR \rightarrow R = \overline{PQR} \vee R = \bar{P} \vee \bar{Q} \vee \bar{R} \vee R = \bar{P} \vee \bar{Q} \vee (\bar{R} \vee R) = \bar{P} \vee \bar{Q} \vee 1 = 1. \end{aligned}$$

Якщо ж хоча б один член кон'юнктивної нормальної форми дорівнюватиме 0, то відповідна їй формула є суперечливою.

Кажуть, що дві формули еквівалентні або що формула  $P$  еквівалентна формулі  $Q$  (тобто  $P = Q$ ), коли істинні значення  $P$  і  $Q$  збігаються при кожній інтерпретації

формул  $P$  і  $Q$ . Існує багато еквівалентних формул, які називаються законами і які використовуються при перетвореннях формул з однієї форми в іншу, особливо в нормальну форму.

Отже, в межах логічної моделі істинному твердженню відповідає **теорема** – правильно побудована формула, яка може бути виведена з аксіом шляхом скінченної кількості застосувань правил виведення.

**Твердження 2.1.** *Будь-яка формула може бути перетворена в нормальну форму за допомогою еквівалентних перетворень – саме це й створює можливість реалізації четвертої умови відповідності логічної СПЗ формальній системі, тобто наявності механізму логічного виведення.*

Різні підстановки у тавтологію, незалежно від їх конкретного змісту, завжди є істинними реченнями через одну тільки свою логічну структуру. Інакше кажучи, тавтології можна розглядати як деякі логічно істинні схеми міркувань або тверджень. Тому вони відіграють роль законів (теорем) логіки висловлювань, які претендують на встановлення методів побудови правильних умовиводів.

**Означення 2.2.** *Логічне виведення в межах заданої формальної системи полягає в утворенні з деякої сукупності початкових даних ППФ нових формул, які є тавтологіями.*

Перше правило майже очевидне, а друге безпосередньо випливає із закону Modus ponens (2.11).

Існує множина тавтологій, а отже, й законів (теорем) логіки висловлювань.

Наведемо деякі з еквівалентних перетворень.

$$P \leftrightarrow Q = (P \leftrightarrow Q) \wedge (Q \leftrightarrow P); \quad (2.2)$$

$$P \rightarrow Q = \neg P \vee Q; \quad (2.3)$$

$$\left. \begin{aligned} P \vee (Q \wedge H) &= (P \vee Q) \wedge (P \vee H) \\ P \wedge (Q \vee H) &= (P \wedge Q) \vee (P \wedge H) \end{aligned} \right\} \text{— дистрибутивні закони;} \quad (2.4)$$

$$\neg(\neg P) = P \text{— закон подвійного заперечення;} \quad (2.5)$$

$$\left. \begin{aligned} \neg(P \vee Q) &= \neg P \wedge \neg Q \\ \neg(P \wedge Q) &= \neg P \vee \neg Q \end{aligned} \right\} \text{— закони де Моргана;} \quad (2.6)$$

$$\left. \begin{aligned} P \vee \neg P &= 1 \\ P \wedge \neg P &= 0 \end{aligned} \right\} \text{— закон вилучення третього;} \quad (2.7)$$

$$\neg(P \wedge \neg P) = \square \text{— закон суперечності;} \quad (2.8)$$

$$P \vee \square = P; \quad P \wedge \blacksquare = P; \quad (2.9)$$

$$P \vee \blacksquare = \blacksquare; \quad P \wedge \square = \square. \quad (2.10)$$

$$(B \rightarrow D) \wedge B \mapsto D, \quad (2.11)$$

де знак  $\mapsto$  означає вивідність – Modus ponens, або закон відокремлення;

$$(B \mapsto D) \wedge \neg B \mapsto \neg D \text{— Modus tollens;} \quad (2.12)$$

$$(B \rightarrow D) \rightarrow (\neg D \rightarrow \neg B) \text{— контрапозиції;} \quad (2.13)$$

$$(B \rightarrow D) \wedge (D \rightarrow P) \rightarrow (B \rightarrow P) \text{— силіогізму} \quad (2.14)$$

та ін.

Кожний із законів логіки висловлювань відображує у символічній формі деяку схему доведення. Наприклад, згідно із законом відокремлення Modus ponens (2.11),

якщо істинним є те, що деяке висловлювання  $B$  імплікує висловлювання  $D$  та, окрім того,  $B$  є істинним, то істинним є й  $D$ . У свою чергу, закон Modus tollens (2.12) застосовується при доведенні від супротивного: бажаючи довести твердження  $B$ , припускається, що  $B$  є помилковим (хибним), і показується, що  $B$  імплікує деяке висловлювання  $D$ , про яке відомо, що воно є помилковим ( $\neg D$  є істинним). Із цього робиться висновок, що  $B$  є істинним. Так, у наведеному прикладі 2.1 формула (2.1) не є тавтологією, якщо вона набуває значення 0 хоча б на одному з наборів  $P \rightarrow Q$ ,  $Q \rightarrow R$  і  $P$  значень змінних. Цю обставину використовують для розпізнавання тавтологій скороченим методом «зворотного міркування», який ще дістав назву «методу резолюцій» і який полягає у пошуку таких змінних, за яких формула виявляється помилковою (хибною). Так, наведена формула (2.1) може набути значення 0, якщо і тільки якщо  $R$  є помилковим, а  $(P \rightarrow Q) \wedge (Q \rightarrow R) \wedge P$  є істинним. При цьому мають бути істинними всі складові  $P \rightarrow Q$ ,  $Q \rightarrow R$  і  $P$ . Тоді при істинному  $P$  формула  $P \rightarrow Q$  буде істинною тільки тоді, якщо істинним є  $Q$ . У свою чергу, при істинному  $Q$  формула  $Q \rightarrow R$  буде істинною тільки при істинному  $R$ . Таким чином, розглянута формула (2.1) може бути помилковою, якщо і тільки якщо  $R$  буде одночасно і істинним, і помилковим, що неможливо через закон суперечності. Отже, формула (2.1) – тавтологія.

Щоб з'ясувати, чи є формула (2.1) тавтологією, можна скористатися відповідною їй таблицею істинності:

$P$	0	0	0	0	1	1	1	1
$Q$	0	0	1	1	0	0	1	1
$R$	0	1	0	1	0	1	0	1
$P \rightarrow Q$	1	1	1	1	0	0	1	1
$Q \rightarrow R$	1	1	0	1	1	1	0	1
$(P \rightarrow Q)(Q \rightarrow R)P$	0	0	0	0	0	0	0	1
$(P \rightarrow Q)(Q \rightarrow R)P \rightarrow R$	1	1	1	1	1	1	1	1

Отже, формальна теорія виведень ставить своїм головним завданням утворення з деякої сукупності вихідних тавтологій нових формул, які також є тавтологіями, і це завдання розв'язується за допомогою **правил виведення**:

- якщо  $A$  тавтологія, то замінюючи в ній літеру  $X$  всюди, де вона входить, довільною формулою  $B$ , отримуємо також тавтологію (*правило підставлення*);
- якщо  $A$  і  $A \rightarrow B$  суть тавтології, то  $B$  – також тавтологія (*правило виведення*).

**Означення 2.3.** Формула  $Q$  є вивідною, якщо вона може бути виведена із скінченної сукупності початкових формул  $P_1, P_2, \dots, P_n$  шляхом скінченної кількості кроків застосування правил виведення  $P_1 \rightarrow (P_2 \rightarrow (\dots (P_n \rightarrow Q) \dots))$ .

Для скорочення багаторазового використання основних правил виведення застосовуються спеціальні правила.

Нехай задано формули  $P_1, P_2, \dots, P_n$  і формулу  $Q$ . Кажуть, що  $Q$  є *логічним висновком формул*  $P_1, P_2, \dots, P_n$  (позначається  $(P_1, P_2, \dots, P_n) \Rightarrow Q$ ) тоді і тільки тоді, коли для будь-якої інтерпретації, в якій формули  $P_1, P_2, \dots, P_n$  є істинними,  $Q$  – також істинна формула. Тут  $P_1, P_2, \dots, P_n$  називаються *аксіомами* (*постулатами* чи *посиланнями*) формули  $Q$ .

**Означення 2.4.** Теоремою у дедуктивних системах пошуку є твердження про необхідність доведення того, що деяка формула логічно випливає з інших формул.

Проблема пошуку розв'язків зводиться до проблеми *доведення теореми*, тобто до побудови міркувань, які встановлюють, що деяка формула логічно випливає з інших формул. При автоматизації логічного міркування в межах певної системи з ШІ постає проблема опису задачі або проблемної галузі на основі деякої прикладної системи математичної логіки та реалізації процедур пошуку розв'язків у процесі виведення.

У логіці висловлювань доведено дві теореми, на яких ґрунтується коротший і простіший спосіб виведення.

**Теорема дедуції.** *Нехай задано формули  $P_1, P_2, \dots, P_n$  і формулу  $Q$ . Тоді  $Q$  є логічним висновком  $P_1, P_2, \dots, P_n$  тоді і тільки тоді, коли  $(P_1 \wedge P_2 \wedge \dots \wedge P_n) \rightarrow Q$  є вивідною формулою.*

**Теорема про суперечність.** *Нехай задано формули  $P_1, P_2, \dots, P_n$  і формулу  $Q$ . Тоді  $Q$  є логічним висновком  $P_1, P_2, \dots, P_n$  тоді і тільки тоді, коли  $(P_1 \wedge P_2 \wedge \dots \wedge P_n) \rightarrow \neg Q$  є суперечливою формулою.*

**Пряме та зворотне виведення.** Найбільш поширеними є два типи виведень: пряме і зворотне. Ідея **прямого виведення** ґрунтується на таких кроках. На першому кроці підбирається правило виведення, умова якого може бути утворена з формул вихідної бази знань  $BZ_B$  із застосуванням за необхідності *уніфікації* (тобто знаходження *необхідної підстановки*). Зазвичай починається усе з формул, які подають знання про початковий стан середовища. Далі на наступних кроках за правилом виведення отримують нові істинні формули, які є уніфікованим висновком і які розміщують у базу  $BZ_B$ , в результаті чого остання перетворюється в базу знань  $BZ_1$ . З неї виводяться нові формули, і все повторюється доти, доки на черговому кроці уніфікований висновок не збіжиться із цільовою формулою. Таким чином, при прямому виведенні на кожному кроці база знань поповнюється виведеними істинними формулами.

При **зворотному виведенні** все робиться навпаки. На першому кроці підбирається правило виведення, наслідком якого є цільова формула, а умова може бути утворена з формул вихідної бази знань  $BZ_B$  із застосуванням за необхідності уніфікації. Уніфіковані формули, які входять в умову, беруться за нові цільові формули (їх називають *підцільовими формулами*), і все повторюється спочатку, але вже для кожної з них. І так доти, доки всі уніфіковані підцільові формули не виявляться аксіомами, які входять у вихідну  $BZ_B$ .

**Приклад 2.2.** Розглянемо *постановку задачі* прямого і зворотного виведення на простих прикладах середовища із кубиків. Кожний з кубиків може лежати або безпосередньо на столі, або на одному з кубиків, утворюючи з кубиками, які лежать під ним, стовпець. Нехай початковий стан середовища з кубиків такий: кубику  $C$  та  $B$  лежать на столі, кубик  $A$  лежить на кубику  $C$ , кубик  $D$  на кубику  $B$  (рис. 2.1, а).

Початковий стан середовища кубиків можна подати такими атомарними формулами:

- |     |                |
|-----|----------------|
| (1) | $na(A,C)$ ;    |
| (2) | $na(C,Стіл)$ ; |
| (3) | $na(D,B)$ ;    |
| (4) | $na(B,Стіл)$ ; |
| (5) | $вільний(A)$ ; |
| (6) | $вільний(D)$ ; |

причому атомарні формули  $na(A,C)$ ;  $na(C,Стіл)$ ;  $na(D,B)$ ;  $na(B,Стіл)$  описують положення кубиків відагентно один відносно одного, тоді як атомарні формули  $вільний(A)$ ;  $вільний(D)$  вказують, що на кубиках  $A$  та  $B$  нічого не лежить.

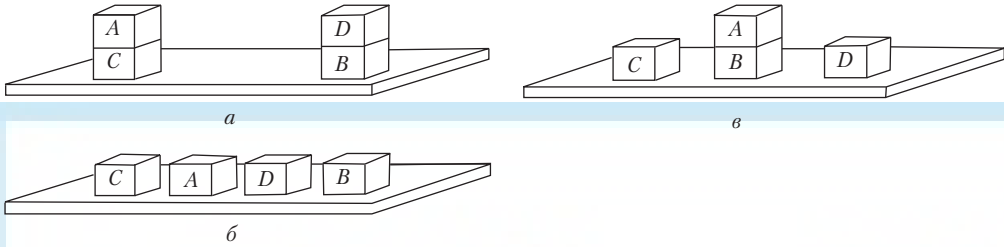


Рис. 2.1. Середовище кубиків: *a* – початковий стан (вирази (1)–(6));  
*б* – розташування згідно з проміжною функцією прямого виведення (вирази (16)–(19));  
*в* – розташування згідно з цільовою функцією прямого виведення (вираз (13))

Далі запишемо формулами **умови виконання дій**. Дії, які можна здійснювати в середовищі кубиків, – це переміщувати один з вільних кубиків на стіл або на інший кубик, якщо він є вільним від кубиків зверху. Для позначення цих дій введемо атомарні формули *перемістити*( $x, y$ ) та *перемістити*( $x, \text{Стіл}$ ). Тоді для нашого прикладу достатньо двох формул, які визначають умови виконання дій:

- (7)  $na(x, y) \wedge \text{вільний}(x) \supset \text{перемістити}(x, \text{Стіл});$   
 (8)  $na(x, \text{Стіл}) \wedge \text{вільний}(x) \wedge \text{вільний}(z) \supset \text{перемістити}(x, z).$

Тепер запишемо формулами **умови переходів станів середовища**:

- (9)  $na(x, y) \wedge \text{вільний}(x) \wedge \text{перемістити}(x, \text{Стіл}) \supset na(x, \text{Стіл});$   
 (10)  $na(x, y) \wedge \text{вільний}(x) \wedge \text{перемістити}(x, \text{Стіл}) \supset \text{вільний}(y);$   
 (11)  $na(x, y) \wedge \text{вільний}(x) \wedge \text{вільний}(z) \wedge \text{перемістити}(x, z) \supset na(x, z);$   
 (12)  $na(x, y) \wedge \text{вільний}(x) \wedge \text{вільний}(z) \wedge \text{перемістити}(x, z) \supset \text{вільний}(y).$

Відповідно до рис. 2.1, *в*, **цільову умову** визначає формула

- (13)  $na(A, B) \wedge na(B, \text{Стіл}) \wedge na(C, \text{Стіл}) \wedge na(D, \text{Стіл}).$

**Пряме виведення.** Почнемо з формули (9), уніфікуючи яку, отримаємо формули (14), (15):

- (14)  $na(A, C) \wedge \text{вільний}(A) \supset \text{перемістити}(A, \text{Стіл});$   
 (15)  $na(D, B) \wedge \text{вільний}(D) \supset \text{перемістити}(D, \text{Стіл}).$

На основі правила Modus ponens (2.11) та формул (14), (15) робимо висновок щодо можливості виконання дій з переміщення кубиків *A* і *D* на стіл: істинні атоми «*перемістити*( $A, \text{Стіл}$ )», «*перемістити*( $D, \text{Стіл}$ )». Ці атомарні формули розміщують у вихідній базі знань БЗ<sub>в</sub>. Далі можна скористатися формулами (9), (10), уніфікуючи які, отримуємо формули (16), (17), (18), (19):

- (16)  $na(A, C) \wedge \text{вільний}(A) \wedge \text{перемістити}(A, \text{Стіл}) \supset na(A, \text{Стіл});$   
 (17)  $na(A, C) \wedge \text{вільний}(A) \wedge \text{перемістити}(A, \text{Стіл}) \supset \text{вільний}(C);$   
 (18)  $na(D, B) \wedge \text{вільний}(D) \wedge \text{перемістити}(D, \text{Стіл}) \supset na(D, \text{Стіл});$   
 (19)  $na(D, B) \wedge \text{вільний}(D) \wedge \text{перемістити}(D, \text{Стіл}) \supset \text{вільний}(B).$

В результаті кубики *A* і *D* опинилися на столі, тоді як кубики *C* і *B* нікуди не переміщалися і, згідно з початковим станом, залишилися на столі (це відповідає проміжній ситуації, наведеної на рис. 2.1, б). Використовуючи уніфіковану формулу (8), отримуємо формулу (20):

- (20)  $na(A, \text{Стіл}) \wedge \text{вільний}(A) \wedge \text{вільний}(B) \supset \text{перемістити}(A, B).$

Нарешті, скориставшись формулою (9), отримаємо формулу (21):

- (21)  $na(A, \text{Стіл}) \wedge \text{вільний}(A) \wedge \text{вільний}(B) \wedge \text{перемістити}(A, B) \supset na(A, B).$

В результаті маємо істинні атомарні формули  $na(A, B)$ ,  $na(B, \text{Стіл})$ ,  $na(C, \text{Стіл})$ ,  $na(D, \text{Стіл})$ . Згідно з правилом виведення кон'юнкції робимо висновок про істинність цільової формули  $na(A, B) \wedge na(B, \text{Стіл}) \wedge na(C, \text{Стіл}) \wedge na(D, \text{Стіл})$ . На цьому пряме виведення завершено. У процесі прямого виведення база знань збільшується, бо до неї заносяться знов виведені формули, причому багато з них можуть бути «зайвими» і непотрібними для виведення цільової формули.

**Зворотне виведення.** Зворотне виведення розпочинається, наприклад, з цільової формули  $na(A, B) \wedge na(B, Стіл) \wedge na(C, Стіл) \wedge na(D, Стіл)$  (рис. 2.1, в) і продовжується у зворотному від розглянутого вище напрямку. При цьому «зайві» (непотрібні) формули не виводяться, що робить його у деяких випадках більш ефективним, особливо якщо для виведення проміжних формул досить незначної частини бази знань.

На прикладі 2.2 було досліджено тільки принципи прямого та зворотного виведення. Можливі також їх комбінації. Проте для всіх випадків вихідною для виведення є деяка початкова (вихідна) БЗ<sub>в</sub>.

Наприкінці слід тільки навести такі означення:

**Означення 2.5.** Числення є повним, якщо загальнозначущі формули є вивідними в цьому численні і тільки вони. У протилежному разі числення називається неповним.

**Означення 2.6.** Повною стратегією називається така стратегія, яка забезпечує виведення усіх загальнозначущих формул цього числення і тільки їх.

**Означення 2.7.** Числення називається несуперечним, якщо не існує формули, яка виводиться в цьому численні разом із своїм запереченням. У протилежному разі числення називається суперечним.

**Числення (логіка) предикатів.** У логіці предикатів першого порядку вводять ще три логічні поняття: *терми*, *предикати* та *квантори*. Розглянемо поняття атомарної формули в логіці предикатів, яке є ширшим від аналогічного поняття в логіці висловлювань, оскільки в останньому випадку атомарна формула розглядається як одне ціле, а її структура та склад не аналізуються. Тому реалізація багатьох міркувань у логіці висловлювань неможлива.

Для побудови атомарних формул у логіці предикатів застосовуються такі типи символів:

- індивідуальні символи (імена об'єктів), або константи (малі літери  $a, b, c, \dots$ );
- символи предметних змінних (малі літери  $x, y, z$ ). Конкретні значення предметних змінних стають предметними сталими (константами);
- функціональні символи (малі літери  $f, g, h, \dots$  або змістовні слова з цих літер);
- предикатні символи (великі літери  $P, Q, D, \dots$  або змістовні слова з цих літер).

Будь-яка функція або предикатний символ складаються з термів.

**Означення 2.8.** Терм — це клас логічних понять, утворених константами та предметними змінними.

Якщо  $f$  є  $n$ -місним функціональним символом і  $t_1, \dots, t_n$  — терми, то  $f(t_1, \dots, t_n)$  — також терм.

**Означення 2.9.** Предикат — це пропозиційна функція предметних змінних, заміна яких їх конкретними значеннями (предметними сталими) перетворює цю функцію в осмислене (І або ІІ) висловлювання.

**Означення 2.10.** Атомарною формулою логіки першого порядку є предикат  $P(t_1, \dots, t_n)$ , де  $P$  —  $n$ -місний предикатний символ, а  $t_1, \dots, t_n$  — терми.

Додатково використовуються два спеціальних символи — квантори  $\forall$  (загальності) і  $\exists$  (існування), щоб схарактеризувати змінні. Областю дії квантора, що входить у формулу, є формула, до якої він застосовується. У формулах, які складаються більше ніж з однієї предикатної літери, використовуються фігурні дужки для позначення області дії кванторів. Так,  $\forall x\{\dots\}$  означає, що будь-яка змінна  $x$  у цих дужках стосується квантора  $\forall$ .

Одна й та сама змінна може мати вільні та зв'язані входження до однієї й тієї самої формули. Входження змінної  $x$  у формулу називається *зв'язаним*, коли  $x$  або є змінною квантора, що входить у цю формулу, або на  $x$  діє квантор, який входить у цю формулу. В протилежному разі входження змінної  $x$  у цю формулу називається *вільним*. Наприклад, у формулі  $P(x, y) \rightarrow \forall xP(x)$  перше входження змінної  $x$  є вільним, а друге і третє — зв'язаними; змінна  $y$  є вільною, оскільки єдине входження  $y$  є вільним. Отже, змінні називаються вільними (зв'язаними) в такій формулі, якщо існують вільні (зв'язані) їх входження в цю формулу.

Формула логіки предикатів  $P(x)$  є вільною для змінної  $y$ , якщо в  $P$  відсутні вільні входження  $x$ , які підпадають під дію кванторів  $\forall y$  або  $\exists y$ . Підставити змінну  $y$  замість змінної  $x$  у предикатну формулу  $P$  — означає замінити кожне вільне входження змінної  $x$  у  $P$  входженням  $y$ . Наприклад, якщо формула  $P(x)$  має вигляд  $H(x, z) \rightarrow \forall yQ(y)$ , то вона є вільною для  $y$ ; але якщо формулу  $P(x)$  записано у вигляді  $H(x) \vee \forall yQ(x, y)$ , то вона не є вільною для  $y$ .

**Означення 2.11.** *Правильно побудованою формулою (ППФ) логіки предикатів є вираз, який будується на основі атомарних ППФ, логічних відношень і кванторів.*

Іншими словами, якщо атомарна формула є ППФ (наприклад,  $P$  і  $Q$  — ППФ), то  $(\neg P)$ ,  $(P \vee Q)$ ,  $(P \wedge Q)$ ,  $(P \rightarrow Q)$ ,  $(P \leftrightarrow Q)$  також є ППФ. Якщо  $P$  — ППФ, а  $x$  — змінна в  $P$ , то  $\forall xP$  і  $\exists xP$  теж є ППФ.

У логіці предикатів як нормальна виступає *попередньо нормальна форма* (ПНФ), причому метою перетворення початкових формул у ПНФ є спрощення цієї процедури.

**Означення 2.12.** *Формула  $P$  матиме ПНФ, якщо вона записується як  $\Psi_1x_1, \dots, \Psi_nx_n (M)$ , де кожне значення  $\Psi_ix_i$ ,  $i = 1, n$ , яке називається префіксом, є  $\forall x_i$  або  $\exists x_i$ ;  $M$  — формула (або матриця), що не містить кванторів.*

Отже, якщо **алфавіт** та **синтаксичні правила побудови формул** у логіці предикатів першого порядку визначено *означеннями 2.8–2.12*, то множина **аксіом** визначається так:

- кожна аксіома класичного числення висловлювань *трансформується* в аксіому класичного числення предикатів. Ця трансформація полягає тільки в тому, що всі її підформули і вона сама розглядаються тепер як формули логіки предикатів першого порядку;
- до цієї множини аксіом, які є аналогом аксіом класичного числення висловлювань, додаються дві наступні:

$$\forall(x)\alpha(x) \supset \alpha(y); \alpha(y) \supset \exists(x)\alpha(x).$$

Загальнозначущість цих аксіом легко перевіряється за допомогою таблиць істинності.

*Множиною правил класичного числення предикатів є Modus ponens  $\alpha, \alpha \supset \beta \vdash \beta$ , в якому  $\alpha$  і  $\beta$  є формулами логіки предикатів першого порядку, а також два правила введення кванторів*

$$\alpha \supset \beta \vdash \alpha \supset \forall(x)\beta(x); \alpha \supset \beta \vdash \exists(x)\alpha(x) \supset \beta.$$

Класичне числення предикатів першого порядку не є єдиним. Існує безліч інших числень, які побудовані на основі класичного, але використовують, окрім правил класичного числення предикатів, й інші правила.

Розглянемо основні *правила логіки предикатів*.

1. *Правило узагальнення* (правило зв'язування квантором спільності). Нехай  $Q$  — формула, що містить вільні входження змінної  $x$ ;  $P(x)$  — формула. Тоді, якщо  $Q \rightarrow P(x)$  є вивідною формулою, то формула  $Q \rightarrow \forall xP(x)$  також буде вивідною.



2. *Правило зв'язування квантором існування.* Нехай  $Q$  — формула, що не містить вільних входжень змінної  $x$ ;  $P(x)$  — формула. Тоді, якщо  $P(x) \rightarrow Q$  є вивідною формулою, то формула  $\exists xP(x) \rightarrow Q$  також буде вивідною.

3. *Правило універсальної конкретизації.* Нехай  $P(x)$  — формула, вільна для  $y$ . Тоді з  $\forall xP(x)$  можна вивести  $P(y)$  підставленням у  $P(x)$  замість змінної  $x$  змінної  $y$ , тобто якщо предикат  $P$  виконується для всіх значень змінної  $x$ , то він виконуватиметься й для будь-якого значення змінної  $y$ .

4. *Правило спеціалізації.* Це правило застосовується для визначення конкретного значення квантора загальності, тобто якщо деякому класу об'єктів притаманна якась властивість, то будь-який об'єкт цього класу також матиме її:  $\forall xP(x), a \vdash xP(a)$ .

5. *Правило конкретизації для квантора існування.* Це правило дає змогу перейти від формули  $\exists xP(x)$  до формули  $P(a)$ . Нехай  $a$  — такий певний елемент, що коли формула  $\exists xP(x)$  є істинною, то формула  $P(a)$  також буде істинною. Тоді  $\exists xP(x) \vdash P(a)$ .

Ці правила поряд з правилами підстановки, вміщення (Modus ponens), теоремою дедукції, а також законами еквівалентних перетворень для логіки предикатів мають характер *універсальної істинності*. Вони можуть застосовуватись або для встановлення істинності тверджень у цілому, або для виведення чи доведення логічних висновків.

**Приклад 2.3.** Розглянемо як приклад подання знання на предикатах: «Коли температура в печі досягає 120 градусів і минуло менше 30 хв з моменту ввімкнення печі, тиск не може перевищувати критичного значення. Якщо з моменту ввімкнення печі минуло більше ніж 30 хв, то необхідно відкрити вентиль № 2». Логічна модель подання цього знання має вигляд:

$$P(p = 120) \wedge T(t < 30) \rightarrow (D < D_{\text{кр}});$$

$$P(p = 120) \wedge T(t > 30) \Rightarrow F(\text{№ 2}).$$

У цьому записі використано такі позначення:

$P(p = 120)$  — предикат стає істинним, коли температура досягає 120 градусів;

$T(t < 30)$  — предикат, що залишається істинним протягом 30 хв від початку процесу;

$T(t > 30)$  — предикат, що стає істинним після закінчення 30 хв з початку процесу;

$(D < D_{\text{кр}})$  — твердження про те, що тиск нижчий від критичного;

$F(\text{№ 2})$  — команда відкрити вентиль № 2.

Крім того, в цих записах використано типові логічні зв'язки кон'юнкції ( $\wedge$ ), імплікації ( $\rightarrow$ ) і логічної наступності ( $\Rightarrow$ ).

Перший рядок у записі являє собою декларативні знання, а другий — процедурні.

**Метод резолюцій.** Розглянемо універсальний метод логічного виведення — *метод резолюцій* [92], згідно з яким складний процес логічного виведення зводиться до послідовного виконання дуже простих операцій, кожна з яких може бути легко запрограмована. Ідея цього методу полягає в перевірці наявності в множині  $S$  порожнього (помилкового) диз'юнкта  $\square$ .

Якщо множина  $S$  містить диз'юнкт  $\square$ , то вона невиконувана, а якщо не містить, то виводяться нові диз'юнкти доти, доки не буде знайдено диз'юнкт  $\square$ .

Метод резолюцій є правилом виведення, при використанні якого породжуються нові диз'юнкти з множини  $S$ . Правило резолюції формулюється так: якщо в будь-яких двох диз'юнктах  $C_1$  і  $C_2$  існує контрарна пара літер ( $L$  і  $\neg L$ ), то, викресливши їх, можна побудувати диз'юнкцію з тих частин диз'юнктів  $C_1$  і  $C_2$ , які залишилися.

Інакше кажучи, для того, аби в логічних виведеннях можна було використовувати метод резолюцій, необхідно, щоб дві логічні фрази **резолюувалися між собою**. Це можливо, якщо одна з фраз має позитивний, а друга — негативний літерал з одним і тим самим позначенням предиката та однаковою кількістю аргументів і якщо аргументи обох літералів можуть бути уніфіковані (погоджені).

Далі, нехай ідеться про атомарні формули, які не містять змінних, але можуть набувати значень «істинність» або «хибність» у висловлюваннях. Тоді кажуть, що два висловлювання містять **контрарну пару** атомарних формул (просту контрарну пару), якщо одна з них включає деяку атомарну формулу без заперечення, а інша — ту саму формулу під знаком заперечення. За таких умов справедливим є таке твердження:

**Твердження 2.2.** Два висловлювання, які містять контрарну пару, можуть бути резолювані одне іншим, і **результатом резолюції (резолювентою) є диз'юнкція літералів**, які залишаються в обох висловлюваннях після викреслювання контрарної пари.

Тоді загальне правило резолюцій формулюється так [21]:

**Правило резолюції.** Два висловлювання, які містять контрарну пару, можуть бути резолювані одне іншим, якщо літерали, що входять до контрарної пари, можна уніфікувати, тобто якщо для них обох існує найбільш загальний уніфікатор  $\eta$ . Результатом резолюції є **диз'юнкція літералів**, які залишаються в обох фразах після викреслювання контрарної пари, причому для цих літералів повинен бути застосований уніфікатор  $\eta$ .

Важливим положенням методу резолюцій є **властивість повноти** (див. означення 2.5), з якої випливає:

**Означення 2.13.** Множина  $S$  диз'юнктив є **невиконуваною** тоді і тільки тоді, коли існує виведення порожнього диз'юнкта з  $S$ .

Поширюючи цей метод на логіку предикатів, для утворення та визначення контрарних пар використовують підстановку й уніфікацію, про які вже згадувалося при розгляді методів прямого та зворотного виведення. Тут розкриємо зміст **узагальненого правила резолюції**, яке пояснюється такими **уніфікацією** та **підстановкою**:

$$\begin{aligned} & \lambda_1^{\mu_1}(x_1) \vee \dots \vee \lambda_{f-1}^{\mu_{f-1}}(x_{f-1}) \vee \lambda_f^{\mu_f}(x_f) \vee \lambda_{f+1}^{\mu_{f+1}}(x_{f+1}) \vee \dots \vee \lambda_m^{\mu_m}(x_m); \\ & \eta_1^{\tau_1}(z_1) \vee \dots \vee \eta_{k-1}^{\tau_{k-1}}(z_{k-1}) \vee \eta_k^{\tau_k}(z_k) \vee \eta_{k+1}^{\tau_{k+1}}(z_{k+1}) \vee \dots \vee \eta_n^{\tau_n}(z_n); \\ & \forall (f, k) \left( \lambda_f = \eta_k; \mu_f \neq \tau_k; \quad \text{Уніфікація} \left( \lambda_f^{\mu_f}(x_f); \eta_k^{\tau_k}(z_k) \right) = \chi \right). \end{aligned}$$

Підстановка  $(\chi(\lambda_1^{\mu_1}(x_1) \vee \dots \vee \lambda_{f-1}^{\mu_{f-1}}(x_{f-1}) \vee \lambda_f^{\mu_f}(x_f) \vee \lambda_{f+1}^{\mu_{f+1}}(x_{f+1}) \vee \dots \vee \lambda_m^{\mu_m}(x_m) \vee \eta_1^{\tau_1}(z_1) \vee \dots \vee \eta_{k-1}^{\tau_{k-1}}(z_{k-1}) \vee \eta_k^{\tau_k}(z_k) \vee \eta_{k+1}^{\tau_{k+1}}(z_{k+1}) \vee \dots \vee \eta_n^{\tau_n}(z_n)))$ .

Тут  $\lambda_f^{\mu_f}(x_f)$ ,  $f = 1, \dots, m$  означає літерал такий, що  $\lambda_f^{\mu_f}(x_f) = \lambda_f(x_f)$ , якщо  $\mu_f = 1$ , та  $\lambda_f^{\mu_f}(x_f) = \neg \lambda_f(x_f)$ , якщо  $\mu_f = 0$ . Такий самий зміст має символ  $\tau_k$  в літералі  $\eta_k^{\tau_k}(z_k)$ ,  $k = 1, \dots, n$ . Отже, «Підстановка  $(\chi(\Psi))$ » означає формулу, яка отримується з формули  $\Psi$  в результаті підстановки  $\chi$  в усі її атомарні формули. Сутність узагальненого правила Modus ponens можна виразити так. Якщо існує істинна формула

$$\lambda_1^{\mu_1}(x_1) \vee \dots \vee \lambda_{f-1}^{\mu_{f-1}}(x_{f-1}) \vee \lambda_f^{\mu_f}(x_f) \vee \lambda_{f+1}^{\mu_{f+1}}(x_{f+1}) \vee \dots \vee \lambda_m^{\mu_m}(x_m),$$

яка є диз'юнкцією літералів  $\lambda_f^{\mu_f}(x_f)$ ,  $f = 1, \dots, m$ , та істинна формула

$$\eta_1^{\tau_1}(z_1) \vee \dots \vee \eta_{k-1}^{\tau_{k-1}}(z_{k-1}) \vee \eta_k^{\tau_k}(z_k) \vee \eta_{k+1}^{\tau_{k+1}}(z_{k+1}) \vee \dots \vee \eta_n^{\tau_n}(z_n),$$

яка теж є диз'юнкцією літералів  $\eta_k^{\tau_k}(z_k)$ ,  $k = 1, \dots, n$ , і в цих формулах предикатні символи  $(\lambda_f = \eta_k)$ ;  $\mu_f \neq \tau_k$  літералів  $\lambda_f^{\mu_f}(x_f)$  та  $\eta_k^{\tau_k}(z_k)$  однакові, і для атомів  $\lambda_f^{\mu_f}(x_f)$  та  $\eta_k^{\tau_k}(z_k)$  існує уніфікація  $\chi$ , яка робить їх однаковими (*Уніфікація*  $(\lambda_f(x_f); \eta_k(z_k)) = \chi$ ), то можна вивести істинну формулу шляхом підстановки  $\chi$  в диз'юнкцію вихідних формул після видалення в них літералів  $\lambda_f^{\mu_f}(x_f)$  та  $\eta_k^{\tau_k}(z_k)$ . Виведена таким шляхом істинна формула й називається **резольвентою**. Як і у випадку резолюції для логіки висловлювань (2.11), узагальнена резолюція може бути виражена із застосуванням імплікації. Зауважимо, що вихідні формули в узагальненій резолюції та резольвент є диз'юнкцією літералів, які зазвичай називають **клаузами**. Отже, для користування резолюцією для виведень необхідна клаузальна форма подання знань.

Повернімося тепер до прикладу 2.1, а з кубиками і повторимо пряме виведення тієї ж цільової формули (13)

$$(13) \quad na(A,B) \wedge na(B,Стіл) \wedge na(C,Стіл) \wedge na(D,Стіл),$$

поданої рис. 2.1, в, але використовуючи узагальнену резолюцію.

**Приклад 2.4.** Звернімо увагу на те, що формули (1)–(6) вже є клаузами. Всі інші формули за допомогою простих перетворень на основі закону  $\lambda \vee \eta \equiv \neg \lambda \supset \eta$  легко перетворюються в клаузи. В результаті замість формул (7)–(12) отримаємо:

формули, які визначають **умови виконання дій**

$$(22) \quad \neg na(x,y) \vee \neg вільний(x) \vee перемістити(x,Стіл);$$

$$(23) \quad \neg na(x,Стіл) \wedge \neg вільний(x) \wedge \neg вільний(z) \vee \neg перемістити(x,z);$$

формули, які визначають **умови переходів станів середовища**

$$(24) \quad \neg na(x,y) \vee \neg вільний(x) \vee \neg перемістити(x,Стіл) \vee na(x,Стіл);$$

$$(25) \quad \neg na(x,y) \vee \neg вільний(x) \vee \neg перемістити(x,Стіл) \vee вільний(y);$$

$$(26) \quad \neg na(x,y) \vee \neg вільний(x) \vee \neg вільний(z) \vee \neg перемістити(x,z) \vee na(x,z);$$

$$(27) \quad \neg na(x,y) \vee \neg вільний(x) \vee \neg вільний(z) \wedge \neg перемістити(x,z) \vee вільний(y);$$

формулу **цільової функції**

$$(28) \quad na(A,B) \wedge na(B,Стіл) \wedge na(C,Стіл) \wedge na(D,Стіл).$$

Цільова функція, відповідно до рис. 2.1, б, є кон'юнкцією літералів і при прямому виведенні, як і в разі використання узагальненого правила Modus ponens (приклад 2.2), буде виводитися для кожного літерала, що входять у функцію (28), окремо.

**Пряме виведення за допомогою узагальненої резолюції.** Використовуючи уніфіковані клаузи (1), (22), які після уніфікації перетворюються в клаузи (29), (30), отримаємо резольвенту (31):

$$(29) \quad na(A,C);$$

$$(30) \quad \neg na(A,C) \vee \neg вільний(A) \vee \neg перемістити(A,Стіл);$$

$$(31) \quad \neg вільний(A) \vee перемістити(A,Стіл).$$

Використовуючи клаузи (5) і (31), отримаємо резольвенту (32):

$$(32) \quad \neg вільний(A) \vee перемістити(A,Стіл).$$

Перепишемо клаузу (25) після уніфікації як уніфіковану клаузу (33):

$$(33) \quad \neg na(A,y) \vee \neg вільний(A) \vee \neg перемістити(A,Стіл) \vee вільний(y).$$

Тепер, використовуючи резольвенту (32) та уніфіковану клаузу (33), отримаємо резольвенту (34):

$$(34) \quad \neg na(A,y) \vee \neg вільний(A) \vee вільний(y).$$

Далі, використовуючи клаузи (1), (5) та уніфіковану резолювенту (34), отримаємо резолювенту (35):

(35) *вільний*(C).

Аналогічно можна отримати такі резолювенти:

(36) *на*(A,Стіл);

(37) *перемістити*(D,Стіл);

(38) *вільний*(B);

(39) *на*(D,Стіл).

Таким чином, було доведено, що кубики A і D можна перемістити на стіл, у результаті чого кубики B і C стають вільними.

Уніфікуємо клаузу (23), перетворивши її у клаузу (40):

(40)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(A) \vee \neg \text{вільний}(B) \vee \text{перемістити}(A, B)$ ,

і тепер, використовуючи клаузи (5), (6), (36) і (40), отримаємо резолювенту (41):

(41) *перемістити*(A,B).

Уніфікуємо клаузу (26), перетворивши її у клаузу (42):

(42)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(A) \vee \neg \text{вільний}(B) \vee \neg \text{перемістити}(A, B) \vee \text{на}(A, B)$ .

Нарешті, скориставшись клаузами (5), (6), (36), (41), а також уніфікованою клаузою (42), отримаємо резолювенту (43):

(43) *на*(A,B).

Таким чином, знову маємо істинні атомарні формули

$\text{на}(A, B), \text{на}(B, \text{Стіл}), \text{на}(C, \text{Стіл}), \text{на}(D, \text{Стіл})$ ,

і, згідно з правилом введення кон'юнкцій, робимо висновок щодо істинності цільової формули

$\text{на}(A, B) \wedge \text{на}(B, \text{Стіл}) \wedge \text{на}(C, \text{Стіл}) \wedge \text{на}(D, \text{Стіл})$ ,

що й відповідає наведеному на рис. 2.1, б цільовому положенню кубиків.

На цьому пряме виведення із застосуванням узагальненого правила резолюцій завершено. Перевага використання узагальненого правила резолюцій для прямого виведення полягає у спроможності виведення значної кількості формул порівняно з виведенням на основі узагальненого правила Modus ponens. Проте в процесі виведення база знань, як і раніше, збільшується за рахунок занесення до неї нових виведених формул. Деякі з цих формул можуть виявитися непотрібними для виведення цільової формули.

(44)  $\neg \text{на}(A, B)$ ;

(45)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(A) \vee \neg \text{вільний}(B) \vee \neg \text{перемістити}(A, B) \vee \text{на}(A, B)$ ;

(46)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(A) \vee \neg \text{вільний}(B) \vee \neg \text{перемістити}(A, B)$ ;

(47)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(A) \vee \neg \text{вільний}(B) \vee \neg \text{перемістити}(A, B)$ ;

(48)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(A) \vee \neg \text{вільний}(B) \vee \text{перемістити}(A, B)$ ;

(49)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(A) \vee \neg \text{вільний}(B)$ ;

(50)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(A) \vee \neg \text{вільний}(B)$ ;

(51)  $\neg \text{вільний}(A)$ ;

(52)  $\neg \text{на}(A, \text{Стіл}) \vee \neg \text{вільний}(B)$ ;

(53)  $\neg \text{на}(A, y) \vee \neg \text{вільний}(A) \vee \neg \text{перемістити}(A, \text{Стіл}) \vee \text{на}(A, \text{Стіл})$ ;

(54)  $\neg \text{на}(A, y) \vee \neg \text{вільний}(A) \vee \neg \text{перемістити}(A, \text{Стіл}) \vee \neg \text{вільний}(B)$ ;

(55)  $\neg \text{на}(A, y) \vee \neg \text{вільний}(A) \vee \neg \text{перемістити}(A, \text{Стіл}) \vee \neg \text{вільний}(B)$ ;

(56) *вільний*(A);

(57)  $\neg \text{на}(A, y) \vee \neg \text{перемістити}(A, \text{Стіл}) \vee \neg \text{вільний}(B)$ ;

(58)  $\neg \text{на}(D, B) \vee \neg \text{вільний}(D) \vee \neg \text{перемістити}(D, \text{Стіл}) \vee \text{вільний}(B)$ ;

(59)  $\neg \text{на}(D, B) \vee \neg \text{вільний}(D) \vee \text{перемістити}(D, \text{Стіл})$ ;

(60)  $\neg \text{на}(D, B) \vee \neg \text{вільний}(D) \vee \text{вільний}(B)$ ;

- (61)  $\neg na(D,B) \vee \neg \underline{вільний}(D) \vee вільний(B)$ ;
- (62)  $\underline{вільний}(D)$ ;
- (63)  $\neg na(D,B) \vee вільний(B)$ ;
- (64)  $\neg na(D,B) \vee вільний(B)$ ;
- (65)  $na(D,B)$ ;
- (66)  $вільний(B)$ ;
- (67)  $вільний(B)$ ;
- (68)  $\neg вільний(B)$ ;
- (69)  $Nil$ .

Що й треба було довести.

Нарешті, закінчимо розділ розглядом ще одного прикладу — планування поведінки промислового робота, жорстко закріпленого на рухомій платформі.

**Приклад 2.5.** Отже, нехай роботу відомо, що його маніпулятор жорстко закріплений на рухомій платформі, яку встановлено в цеху. Щоб відповісти на запитання, де розміщується маніпулятор, маємо два сформульованих факти, які можна записати у вигляді двох ППФ:  $B_1 \leftrightarrow x P(\text{платформа}, x) \rightarrow P(\text{маніпулятор}, x)$ , а також  $B_2 \leftrightarrow \leftrightarrow x P(\text{платформа}, \text{цех})$ , де двомісному предикату  $P(y,z)$  надано очевидну інтерпретацію: « $y$  розміщується в  $z$ »  $P(y,z)$ .

На запитання, де розміщується маніпулятор, робот може дати відповідь, якщо спочатку доведе, що ППФ

$$B \leftrightarrow x P(\text{маніпулятор}, x)$$

є вивідною із системи ППФ, після чого те значення  $x$  (константа), яке справді існує, й буде відповіддю на поставлене запитання.

Процес виведення, що ґрунтується на методі резолюцій, можна відобразити деревом виведення (рис. 2.2), з якого також можна дістати відповідь на запитання, де розміщується маніпулятор. Це здійснюється так. Спочатку до заперечення теореми додається її заперечення, тобто сама теорема. Як наслідок маємо тавтологію (тобто ППФ, що є тотожно істинною для всіх інтерпретацій) у вигляді  $\neg P(\text{маніпулятор}, x) \vee \vee P(\text{маніпулятор}, x)$ . Потім відповідно до структури дерева виведення, зображеної на рис. 2.2, знову формуються резольвенти доти, доки в корені дерева не утвориться деяка ППФ, що відіграє роль відповіді.

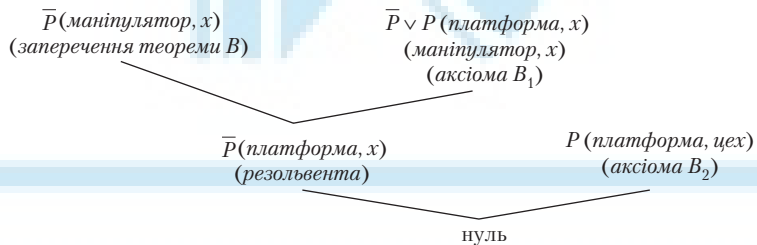


Рис. 2.2. Дерево виведення

У розглянутому прикладі є одна резольвента  $\neg P(\text{платформа}, x) \vee P(\text{маніпулятор}, x)$ , а в корені дерева — ППФ  $P(\text{платформа}, \text{цех})$ , яка містить відповідь на запитання, де розміщується маніпулятор, і форма відповіді є наближеною до форми теореми-запитання. Відмінність полягає в тому, що в теоремі-запитанні є змінна, яка пов'язана квантором існування, а у відповідній ППФ — константа (відповідний терм).

Таким чином, описана система логічного виведення, що ґрунтується на методі резолюцій, є ефективним засобом для автоматичного пошуку доведень (логічних висновків) і одержання відповіді в термінах обчислення предикатів.

Застосування логіки предикатів має ряд переваг.

1. Вона забезпечує чіткий, відпрацьований формалізм її подання, який добре інтерпретується природною мовою і дає змогу легко оперувати результатами аналізу.

2. Завдяки застосуванню в логічних формулах формальних правил одержання виведень забезпечується гарантія правильності висновків, якщо їх вдається вивести, а також надається механізм для виявлення причин невдачі, коли висновки вивести неможливо.

3. Логіка предикатів є системою, застосування якої дає змогу вирішувати складну проблему, крок за кроком задаючи початкову інформацію.

4. Логіка предикатів дає змогу гарантувати одержання надійних результатів при зниженні ефективності подання знань. Остання властивість використовується при розв'язуванні задачі підтримки несуперечливості фактів у БД.

З іншого боку, логіці предикатів внаслідок зберігання властивості цілісності притаманні такі недоліки, як надмірний рівень формалізації подання знань, складність їх прочитання, не дуже висока продуктивність обробки. Але найгірше те, що в логіці предикатів усі відношення описуються предикатами, і тому при комп'ютерній обробці ті переваги, які мають структури даних, повною мірою проявитися не можуть.

## 2.2. Продукційні системи

В основі правил (або продукцій) лежить конструкція вигляду «ЯКЩО (умова), ТО (наслідок)». Умовна частина її називається антецедентом, а заключна — консеквентом. У загальному вигляді продукцію можна подати як

$$(n); S; U; B \Rightarrow C; P, \quad (2.15)$$

де  $n$  — ім'я (ідентифікатор) продукції;  $S$  — характеристика сфери застосування продукції (область компетенції);  $U$  — умова застосування;  $B \Rightarrow C$  — ядро (власне конструкція «ЯКЩО, ТО»);  $P$  — постумови продукції (які виконуються після її реалізації) — всі ці атрибути продукційної системи утворюють її **алфавіт** (першу умову відповідності формальній системі).

Як і в логічних СПЗ, у продукційних системах для формування **розгалужених утворень** в антецедентній (лівій, умовній) та консеквентній (правій, постумовній) частинах використовуються логічні оператори і квантори — це реалізує **другу умову відповідності** формальній системі.

**Третя умова відповідності** продукційної СПЗ формальній системі — **аксіоматичність тверджень** — полягає в істинності або хибності атомарних формул, які входять у ліву (або праву) частину правила, значення яких занесено в БЗ системи.

Нарешті, база правил і метаправил являє собою **механізм логічного виведення** в продукційних СПЗ — **четверта складова відповідності** останніх формальній системі.

Таким чином, частина правил «ЯКЩО» має назву посилання, а «ТО» — виведення, або дії. Антецедент зазвичай подається кон'юнкцією простіших умов  $b_1 \wedge b_2 \wedge \dots \wedge b_n$ ; консеквент може складатися також з кількох компонентів, причому це можуть бути не тільки факти, твердження, а й операції виклику різних спеціальних процедур. В останньому випадку правило відповідає структурі: «ЯКЩО(умова), ТО(дія)».

Наприклад, «ЯКЩО  $B_1, B_2, \dots, B_n$ , ТО  $C$ ». Такий запис означає, що «коли всі умови від  $B_1$  до  $B_n$  є істиною, то  $C$  також є істиною» або ж «коли всі умови від  $B_1$  до  $B_n$  стають істиною, то слід виконувати дію  $C$ ».

Простота й наочність цього способу зумовили застосування його в багатьох ЕС. Системи обробки знань, в яких використовується подання знань продукційними правилами, називаються *продукційними*.

Продукційна система передбачає наявність БЗ (яка складається з правил), БД (що зберігає інформацію, пов'язану з поточною задачею) і МЛВ. Факти, що є умовами та наслідками правил, розміщуються на тимчасове зберігання в БД (яка, таким чином, набуває функції *робочої пам'яті*).

Набір правил, антецеденти яких на поточний момент є істинними, утворює *конфліктну множину* (або конфліктний набір) актуалізованих продукцій (або фронт готових продукцій). Вибір правил з конфліктної множини для виконання є однією з найважливіших функцій МЛВ. Методи, пов'язані з організацією вибору правил виведення або подання знань, можна поділити на дві взаємодоповнюючі групи: спрямовані на обмеження конфліктних наборів; ті, що забезпечують власне розв'язання конфліктів.

*Обмеження конфліктних наборів* досягається:

- початковим групуванням правил за сферою їх застосування;
- використанням можливостей «класної дошки» (спеціального робочого поля пам'яті) для встановлення умов застосування продукцій;
- управлінням за іменами (шляхом встановлення на них спеціальної процедури, що здійснює звуження конфліктного набору);
- використанням метаправил, які обмежують початковий набір, тощо.

Для *розв'язування конфліктів* можна використовувати різні стратегії, суть яких полягає в:

- початковому призначенні пріоритетів продукціям (виходячи з апіорних оцінок їх важливості);
- переважному виборі продукції, що має найвищий показник частоти використання;
- призначенні пріоритетів даним (умовам), коли при виборі продукцій перевага віддається тим, які активізовані умовами з найвищим пріоритетом;
- переважному виборі продукції, що має найкоротшу умовну частину;
- упорядкуванні продукцій згідно з часом їх актуалізації, тобто з урахуванням тривалості перебування їх у конфліктному наборі;
- випадковому виборі (застосовується, як правило, тоді, коли основні стратегії однозначного рішення не забезпечують).

Механізм, реалізований як засіб виведень у продукційних системах, здійснює функцію пошуку в БЗ, послідовно виконуючи операції над знаннями для одержання висновків. Існують два способи одержання останніх — прямиий і зворотний. При *прямих виведеннях* здійснюється просування до поставленої мети з послідовним застосуванням правил до даних (фактів), які беруться за відправну точку. При цьому вибирається один з елементів даних, які є в БД, і якщо при порівнянні цей елемент з посилкою правила узгоджується, то виводиться відповідний висновок, який заноситься в БД, або виконується дія, що визначається вибраним правилом, і відповідно змінюється вміст БД. Часто ці виведення називаються такими, що керуються даними, або *висхідними виведеннями*, коли послідовно виводяться нові результати, починаючи з уже відомих даних.

Виведення, при яких процес відбувається в напрямку від поставленої мети до відправної точки, є *зворотними* (їх називають також низхідними, або орієнтованими на мету). Процес одержання *низхідних виведень* починається з поставленої мети.

Якщо ця мета з консеквентом (висновком) правила узгоджується, то посилка правила приймається за підмету (гіпотезу), і цей процес повторюється доти, доки підмета (гіпотеза) не збіжиться з одержаними даними.

Системи, до яких ставляться вимоги високої універсальності, повинні забезпечувати обидва способи виведення. В таких системах спочатку оцінюється невеликий обсяг одержаних даних і вибирається гіпотеза (за прямим виведенням), а потім запитуються необхідні дані для прийняття рішення про використання цієї гіпотези. На основі таких виведень можна реалізувати потужну й гнучку продукційну систему.

**Приклад 2.6.** Розглянемо спрощений приклад ПС зі зворотним виведенням. Літерами позначимо елементи БД, які вважатимемо істинними, коли вони містяться в ній.

БД:  $A, F$ .

База правил:

- Правило 1:  $A \& B \& C \rightarrow D$ .
- Правило 2:  $D \& F \rightarrow G$ .
- Правило 3:  $C \& F \rightarrow G$ .
- Правило 4:  $B \rightarrow C$ .
- Правило 5:  $F \rightarrow B$ .
- Правило 6:  $L \rightarrow J$ .
- Правило 7:  $G \rightarrow H$ .

Припустимо, що мета полягає у виведенні істинності  $H$ . Насамперед перевіряється, чи міститься  $H$  у БД. Оскільки в даному разі це не так, система намагається вивести істинність  $H$ , використовуючи правила, які містять  $H$  у правій частині. Таким є правило 7. Тепер система намагається вивести істинність  $G$ , тому що, згідно з правилом 7, істинність  $G$  спричинює істинність  $H$ . Знову перевіряється БД: у ній немає  $G$ ; отже, організовується пошук правила, яке містить  $G$  в правій частині. Таких правил кілька (наприклад, правила 2 і 3).

При виборі стратегії «розв'язання конфлікту» скористаємось упорядкуванням правил за пріоритетами, а саме, коли правилу з найменшим номером відповідає найвищий пріоритет. Вибираємо правило 2; тому метою тепер стає виведення істинності  $D$  та  $F$ . Для цього досить сказати, що  $A$  — істина (оскільки міститься в БД),  $B$  — істина (згідно з правилом 5),  $C$  — істина (згідно з правилом 4).

Отже, істинність  $D$  та  $F$  доведено, і з правила 2 випливає істинність  $G$ , а з нею — й істинність  $H$  (правило 7). Мети досягнуто, а елементи, істинність яких доведено, додаються до БД — це елементи  $H, G, D, C, B$ .

Відзначимо переваги та недоліки продукційних систем. До переваг слід зарахувати: простоту створення й розуміння окремих правил; простоту виконання й можливість модифікації; простоту МЛВ. Основні недоліки: неясність взаємних відношень правил; складність оцінювання цілісного образу знань; вкрай низька ефективність їх обробки; відмінність від людської структури знань; відсутність гнучкості в логічному виведенні. Тому, якщо об'єктом дослідження є невеликі задачі, виявляються тільки найсильніші сторони продукційних систем. У разі збільшення обсягу знань, необхідності розв'язання складних задач, виконання гнучких виведень або підвищення їх швидкості ефективним є структурування БД.

**Приклад 2.7.** Нарешті розглянемо ще один приклад продукційної моделі — роботувану дільницю, яку оснащено роботами-маніпуляторами (РМ), міжопераційними нагромаджувачами готових виробів (МН), транспортним роботом — електророботом (ЕРК) і централізованим складом виробів (ЦС) (рис. 2.3).



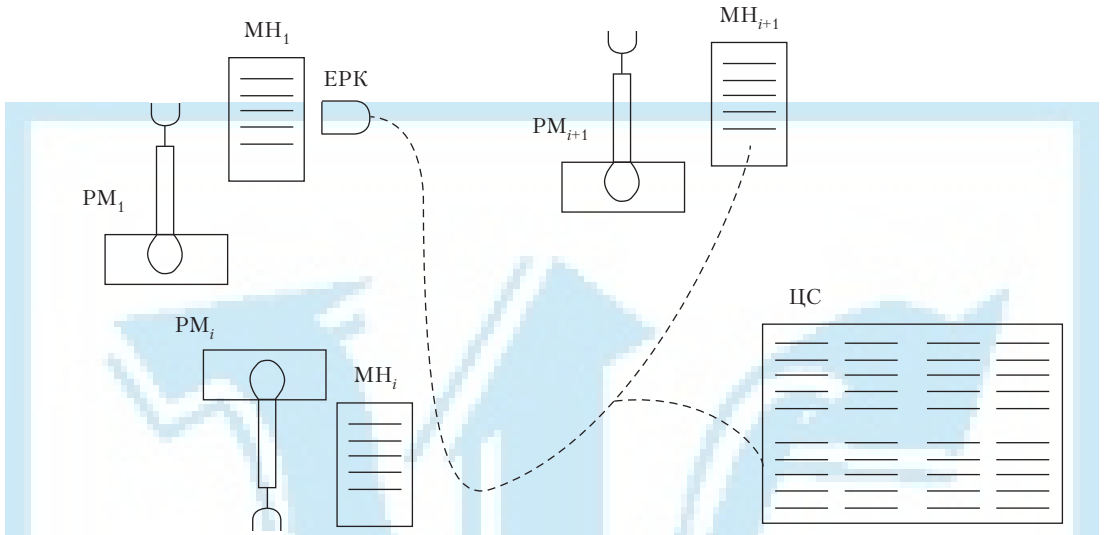


Рис. 2.3. Схема роботизованої ділянки

Завдання ЕРК полягає в послідовному об'їзді МН, збиранні виробів (якщо вони є в МН) і транспортуванні їх на ЦС. Якщо ЕРК заповнений, то передусім слід заїхати до ЦС і там розвантажитись, а потім продовжити об'їзд МН.

Схему алгоритму такого спрощеного варіанта диспетчеризування матеріальних потоків у роботизованій ділянці (отже, і керування ЕРК) показано на рис. 2.4.

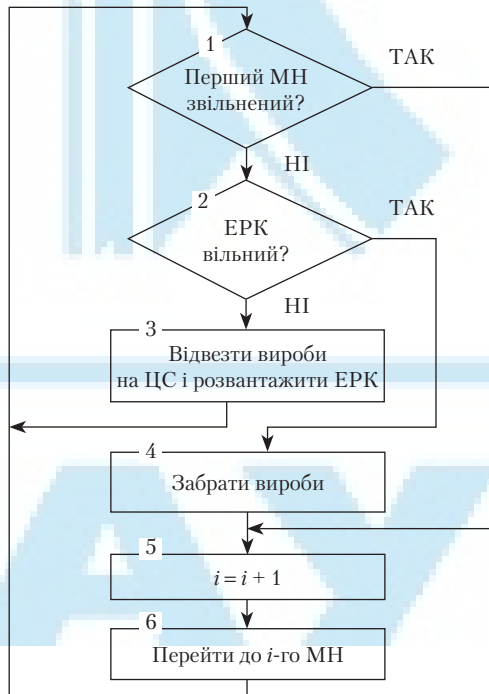


Рис. 2.4. Схема алгоритму керування ЕРК

Фрагмент продукційної системи, що описує алгоритм керування ЕРК, можна подати у вигляді такого набору правил:

- Правило 1: ЯКЩО  $i$ -й МН є порожнім, ТО перейти до  $(i + 1)$ -го МН.
- Правило 2: ЯКЩО в  $i$ -му МН є вироби І ЕРК не заповнений, ТО звільнити МН і перейти до  $i = (i + 1)$ -го МН.
- Правило 3: ЯКЩО ЕРК заповнений, ТО перевезти вироби на ЦС і повернутися до  $i$ -го МН.

### 2.3. Моделі подання знань семантичними сітками

**Базові положення теорії семантичних сіток.** Комп'ютерні семантичні сітки (СС) були детально розроблені у 1956 р. Річардом Річенсом у рамках проекту Кембриджського центру вивчення мови за допомогою машинного перекладу. Процес перекладу розподілявся на дві частини: переклад вихідного тексту у проміжну форму подання, а далі ця проміжна форма транслюється в необхідну мову. Саме такою проміжною формою і були СС.

Формально семантична сітка визначається як набір  $\langle I, C_1, C_2, \dots, C_n, D \rangle$ , де  $I$  – множина інформаційних одиниць;  $C_1, C_2, \dots, C_n$  – типи зв'язків між інформаційними одиницями;  $D$  – відображення, яке задає зв'язки між інформаційними одиницями.

Отже, в моделях знань на СС основний акцент робиться на описі різноманітних зв'язків між інформаційними одиницями.

Спробуємо зобразити наведений у підрозд. 1.2 приклад 1.1 і дещо несуттєво скориговане при цьому твердження «Робот ЕРК1 доставив до складальної позиції СкП2 сотовий (С) та шахтний (Ш) накопичувачі, наповнені комплектуючими складального вузла СВ2 – відповідно, деталями класів А та В» концептуальним графом.

Для наведеної на рис. 2.5 реалізації структурованої СС характерна певна внутрішня структура.

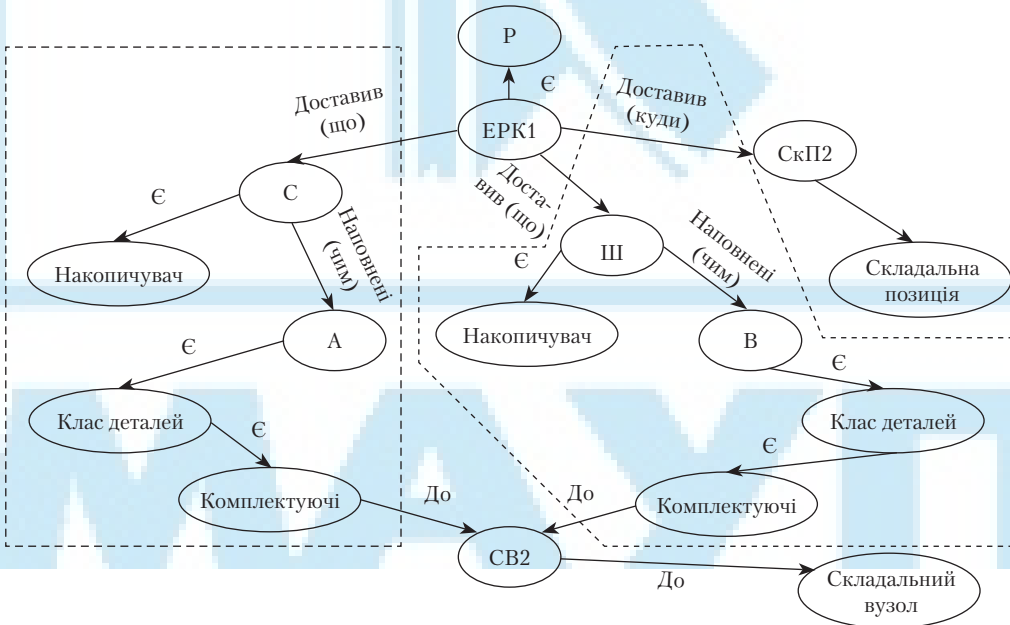


Рис. 2.5. Приклад структурованої СС

Отже, основою формалізації семантичних знань про предметну галузь є спрямований дводольний граф з позначеними вершинами та дугами, який дістав назву *СС* – що й визначає першу умову формальної системи, а саме, **алфавіт СПЗ на СС**. У назві поєднано терміни двох наук: *семантика* в мовознавстві вивчає смисл речень, а сітка в математиці являє собою різновид графа. Роль вершин в *СС* виконують поняття БЗ, яким ставляться у відповідність конкретні факти  $F_i \mid i = 1, n$  (позначаються колами, причому  $n$  – кількість фактів в описі предметної галузі) та конкретні значення атрибутів  $A_j$  (тобто доменів  $\text{DOM}(A_j)$ , які позначаються прямокутниками), а дугам – семантичні відношення між ними (які, власне, і є атрибутами). Позначки вершин мають посилальний характер (тобто дуги виходять з вершин-фактів і спрямовуються на вершини-значення атрибутів) і є деякими іменами. Як імена можуть, наприклад, виступати слова природної мови. Позначки дуг означають елементи множини відношень, завдяки яким реалізується відповідність *СС* другій умові формальної системи, а саме, наявності **механізму створення складних розгалужень**. Таким чином, *СС* відображає семантику предметної галузі у вигляді понять і відношень, де кожна вершина відповідає елементу предметної множини, а дуга – предикату, тобто функції з областю значень  $\{0, 1\}$  (або «Істина», «Хибність»).

Розглянемо більш чітко означення *СС* [102]. Нехай задано скінченну множину символів  $A = (A_1, \dots, A_n)$ , які називаються *атрибутами*, і скінченну множину відношень  $R = (R_1, \dots, R_n)$ . *Схемою*, або *інтенціоналом відношення*  $R_i$ , називається набір пар

$$\text{INT}(R_i) = \{ \dots [A_j, \text{DOM}(A_j)], \dots \}, \quad (2.16)$$

де  $R_i$  – ім'я відношень;  $\text{DOM}(A_j)$  – домен  $A_j$ , тобто множина значень атрибута  $A_j$  відношення  $R_i$ . Таким чином, атрибут  $A_j$  із конкретним значенням  $\text{DOM}(A_j)$  являє собою об'єкт, якому й присвоюється ім'я  $R_j$  цього відношення.

**Означення 2.14.** *Об'єктом СС* є атрибутивна пара типу «атрибут  $A_j$  – його значення  $\text{DOM}(A_j)$ », якій присвоюється ім'я цього відношення.

Об'єднання всіх доменів є *екстенціоналом відношення*  $R_p$ , тобто множиною

$$\text{EXT}(R_i) = \{F_1, \dots, F_p\},$$

де  $F_k (k = 1, p)$  – факт відношення  $R_i$ . Отже:

**Означення 2.15.** *Факт задається* сукупністю атрибутивних пар-об'єктів типу «атрибут  $A_j$  – значення  $\text{DOM}(A_j)$ », тобто під фактом розуміють конкретизацію певного відношення між зазначеними об'єктами, і саме факти становлять основу **аксіоматичних тверджень**, тобто третьої умови відповідності *СС* формальній системі.

Нарешті, четверта умова відповідності *СС* формальній системі полягає в наявності **механізмів виведення** в МПЗ на *СС* через організацію *запитів на існування і перелічення*, пошук відповідей на які полягає в розв'язанні задачі ізоморфного вкладення графа запиту (або його підграфа) в *СС* (див. далі підрозд. «*Формування виведення в СС*»).

**Означення 2.16.** У графічній інтерпретації факт – це підграф *СС*, який має зіркоподібну структуру. Корінь підграфа – вершина предикатного типу – має унікальну позначку, що містить ім'я відповідного відношення. З вершини факту виходять ребра, які позначаються іменами атрибутів факту й спрямовуються до вершин базової множини, що є значеннями цих атрибутів.

**Приклад 2.8.** Розглянемо приклад, що ілюструє наведені означення. Нехай задано базову множину моделі – сукупність цілих чисел  $\{0, 1, 2\}$ , а також множину відношень  $R$ , яка містить відношення «менше – більше» та «сума» (позначимо відношення відповідно « $<$ » « $>$ »). Іntenсiонал цих відношень відповідно до (2.16) можна записати у вигляді:

$INT(+)=\{[\text{перший доданок}, (0, 1, 2)], [\text{другий доданок}, (0, 1, 2)], [\text{сума}, (0, 1, 2)]\};$

$INT(<)=\{[\text{менше, більше}, (0, 1)], [\text{менше, більше}, (1, 2)], [\text{менше, більше}, (0, 2)]\}.$

Екстенсiонали відношень можна записати у вигляді фактів, наприклад так:

- $F_1 <$ : (менше 0, більше 1);
- $F_2 <$ : (менше 1, більше 2);
- ...
- $F_9 +$ : (перший доданок 2, другий доданок 0, сума 2).

Іноді зручно скористатися табличною формою подання семантичних знань (табл. 2.1, 2.2). Графічне зображення екстенсiональної СС подано на рис. 2.6. При її побудові використано такі правила та позначення: базовій множині моделі покладено у відповідність вершини 0, 1, 2, які позначено квадратами; вершини факту зображено колами з позначками, що утворюють пари ідентифікаторів – факт і відношення; дугам присвоєно позначки відповідних атрибутів.

Таблиця 2.1

Таблична форма подання семантичних знань стосовно відношення «менше – більше»

Позначка факту	Відношення «менше – більше»	
	Значення атрибута	Значення атрибута
$F_1 <$	0	1
$F_2 <$	1	2
$F_3 <$	0	2

Таблиця 2.2

Таблична форма подання семантичних знань стосовно відношення «сума»

Позначка факту	Відношення «сума»		
	Перший доданок	Другий доданок	Сума
$F_4 +$	0	0	0
$F_5 +$	0	1	1
$F_6 +$	0	2	2
$F_7 +$	1	0	1
$F_8 +$	1	1	2
$F_9 +$	2	0	2

Екстенсiональні СС описують екстенсiональні знання про об'єкти, що моделюються, і є, так би мовити, «фотографією» їх теперішнього стану. Іntenсiональні СС містять іntenсiональні знання й описують загальну структуру галузі, що моделюється, на основі абстрактних об'єктів і відношень, тобто узагальнених представників деяких класів об'єктів і відношень. Наприклад, такі об'єкти, як виробнича дільниця, вантаж, деталь, можуть бути узагальненими поняттями, множина значень яких утворює множину імен конкретних виробничих дільниць (токарна дільниця, ковальсько-пресова дільниця тощо), множину імен вантажів (касета, заготовка тощо), множину імен деталей (вал, шасі, болт тощо).

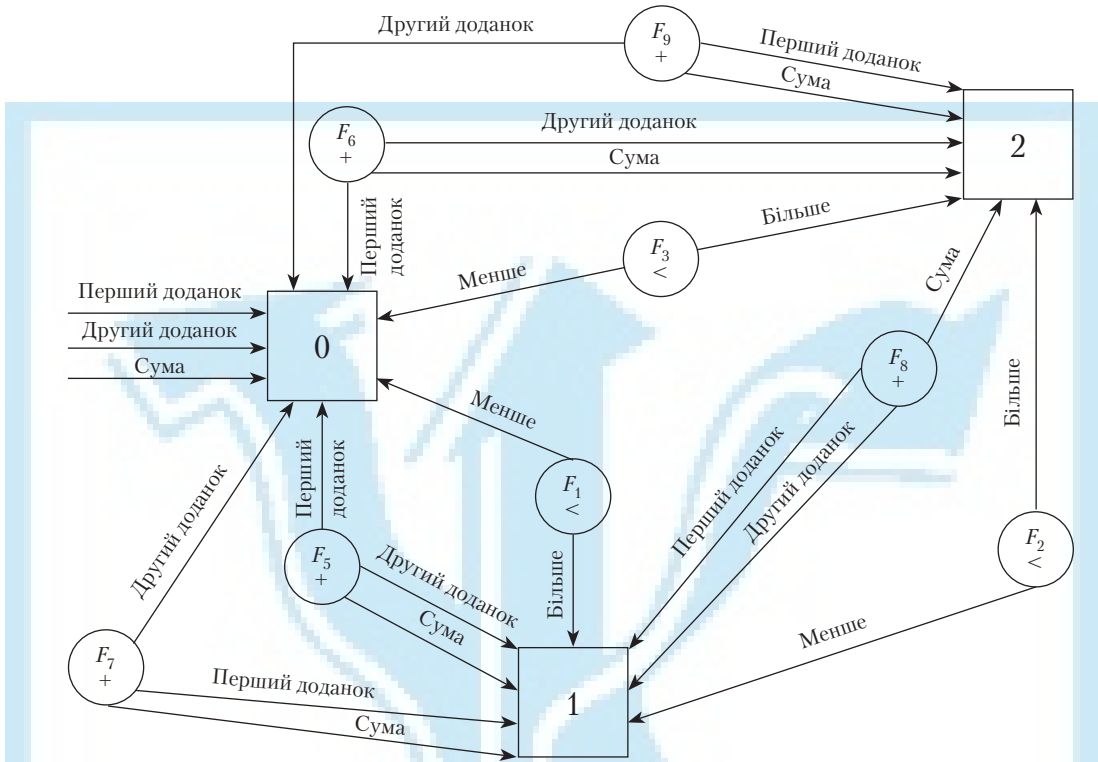


Рис. 2.6. Екстенціональна СС

**Основні типи об'єктів у СС.** У СС застосовуються три основних типи об'єктів: поняття, події та властивості.

**Поняття** — це відомості про абстрактні чи фізичні об'єкти предметної галузі. Загальні поняття інтерпретуються як множини доменів (параметрів або констант).

**Подія** — це дія, що може внести зміни в предметну галузь. Результатом події є деякий новий стан предметної галузі. Можна задати деякий бажаний (цільовий) стан предметної галузі, поставивши завдання відшукування в СС послідовності подій, що приводить до цільового стану.

**Властивості** використовуються для уточнення понять, подій або інших властивостей об'єктів. Стосовно понять властивості описують їх особливості або характеристики (колір, розмір, якість); стосовно подій — тривалість, місце, час тощо.

**Класифікація СС.** Для всіх СС справедливим є розподіл за арністю, кількістю, ієрархічністю та функціональністю [88]:

- за **арністю** вирізняють СС з **бінарними** (зв'язують рівно два поняття) і з **N-арними** (зв'язують більше двох об'єктів) відношеннями;
- за **кількістю типів відношень** СС поділяються на **однорідні** (володіють тільки одним типом відношень — стрілки) і на **неоднорідні** (кількість типів відношень більша двох). Кількість типів відношень визначається розробником СС виходячи з конкретної мети з урахуванням того, що швидкість роботи з БЗ у моделі на СС залежить від ефективності створених програм обробки необхідних відношень;

- за **ієрархічністю** (або **теоретико-множинні**) типи відношень поділяються на *класифікації IS-A* (тобто «Сутність») — означає, що об'єкт належить до множини об'єктів, причому множина (клас) класифікує свої екземпляри, і властивості об'єкта успадковуються від множини; *Example* («Наприклад») — це обернене до *IS-A* відношення, яке використовується для позначення прикладів; *гіпонімії A Kind Of* (тобто «Різновид») — визначає, що кожний елемент надмножини входить і у підмножину (тобто виконується *IS-A* для кожного елемента); *мерономії HasPart* (тобто «Ціле — частина») — описують складові об'єкта. Як приклад на рис. 2.7 зображено фрагмент таксономічної структури поняття «роботи»;
- за **функціональністю** типи відношень поділяються на: *функціональні зв'язки* (зазвичай визначаються дієсловами «виконує», «впливає» тощо); *кількісні* («більше», «менше», «дорівнює», ...); *просторові* («далеко від», «близько від», «поза», «під», «над» тощо); *часові* («раніше», «пізніше», «в проміжок часу», ...); *атрибутивні* («мати властивість», «мати значення»); *лінгвістичні відношення* (є відмінкові, до яких, зокрема, належать: «об'єкт — відношення» між подією й тим, над чим виконується дія; «інструмент — об'єкт», за допомогою якого здійснюється подія, та ін.); *логічні відношення* — це операції, що використовуються при обчисленні висловлювань (наприклад, диз'юнкція, кон'юнкція, імплікація, заперечення); *квантифіковані* — це логічні квантори загальності й існування (про що вже йшлося раніше).

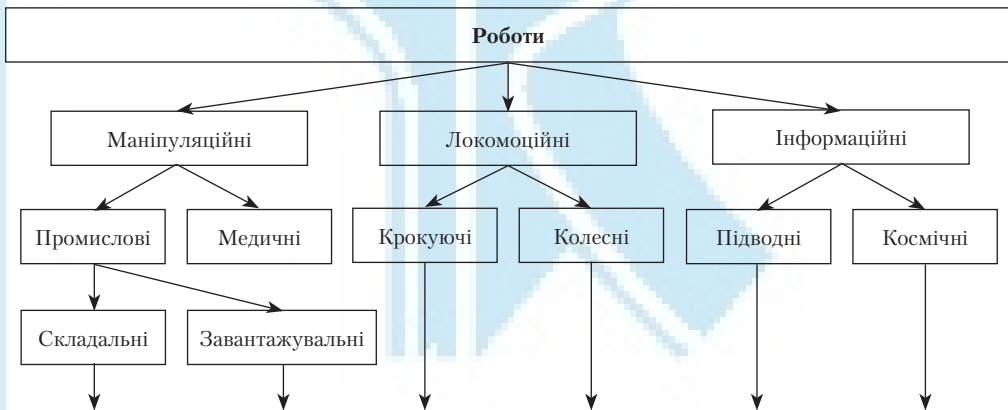


Рис. 2.7. Фрагмент таксономічної структури поняття «роботи»

Логічні квантори застосовуються для подання декларативних знань, наприклад, для запису таких тверджень, як «Кожний верстат потребує профілактичного ремонту» або «Існує робот А, що може обслужити всі верстати групи В». Цей перелік може бути як завгодно великим через те, що в реальному світі **кількість відношень є невичерпною**.

**Приклад 2.9.** Розглянемо приклад СС при поданні знань, які містяться в реченні «Якщо верстат закінчив обробку, то робот завантажує касету з деталями на ЕРК, який перевозить їх на ЦС, де штабелер розміщує касету в комірку складу». Виділимо п'ять фактів: верстат закінчив обробку ( $F_1$ ), робот завантажує ( $F_2$ ), ЕРК перевозить ( $F_3$ ), касета містить ( $F_4$ ), штабелер розміщує ( $F_5$ ). Зазначимо, що при описі фраз природної мови факти часто називаються *висловлюваннями*.

Як і в попередньому прикладі, факт позначатимемо колом, а пов'язані з ним поняття – прямокутниками. Дуги позначимо назвами відношень, які вони передають. Схему СС показано на рис. 2.8.

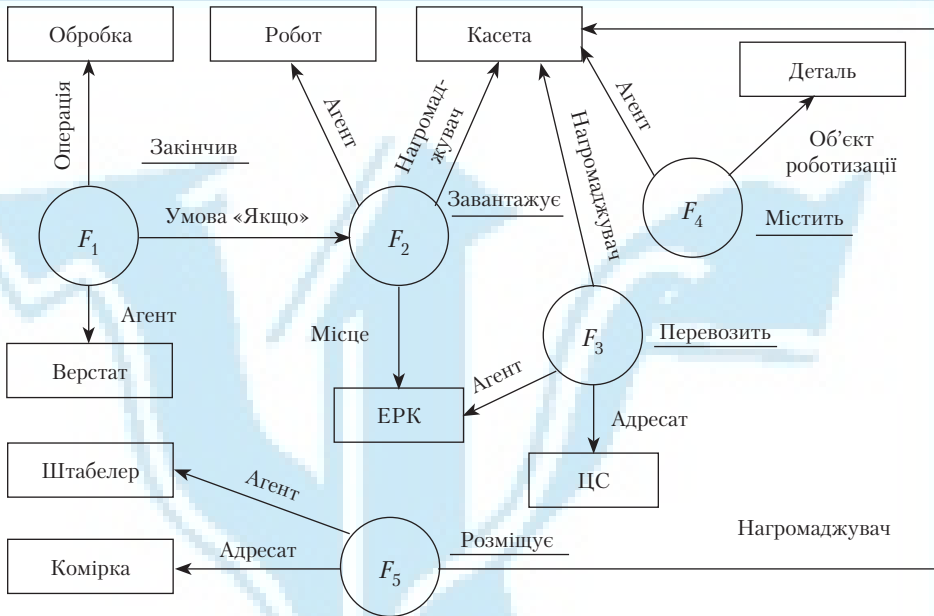


Рис. 2.8. Схема СС

Важливим поняттям у СС є *десигнат* – унікальне внутрішньосистемне ім'я, що ставиться у відповідність деякому об'єкту предметної галузі, коли про нього на даний момент немає повної інформації. Десигнат відображає найважливішу суть об'єкта, зокрема просто факт його існування. Наприклад, фраза типу «Верстат В1 має нагромаджувач» містить невизначеність щодо характеристик нагромаджувача.

Уявімо собі, що в міру надходження інформації (що може відбуватися в процесі проектування) будуть з'ясовані місткість нагромаджувача, його розміри тощо. Тоді наведену фразу можна подати так:

( $F_1$ : МАЄ агент ВЕРСТАТ об'єкт D1)

(D1: ім'я НАГРОМАДЖУВАЧ)

(D1: розмір)

(D1: місткість).

Тут  $F_1$  – позначка факту, D1 – позначка десигнату. Порожні місця заповнюються в процесі поповнення даних значеннями відповідного атрибута. Отже, при першому згадуванні про об'єкт у БД заноситься його десигнат, яким можна маніпулювати, не чекаючи повної інформації про об'єкт. Уведення десигнатів підвищує гнучкість СС, проте їх використання потребує додаткових обчислювальних ресурсів.

Розглянемо деякі питання побудови інтелектуальних БД на основі СС. Основний принцип їх організації полягає в поділі на екстенсiональні та інтенсiональні знання, коли екстенсiональна СС є основою БД, а інтенсiональна – основою БЗ. Таке «розшарування» СПЗ суттєво підвищує її виразні спроможності порівняно, наприклад, з логічними моделями.

Структуру СПЗ розглядуваного типу зображено на рис. 2.9. Умовність зображення системи пов'язана зокрема з відсутністю чіткого розподілу зазначених компонентів СПЗ при їх реалізації.

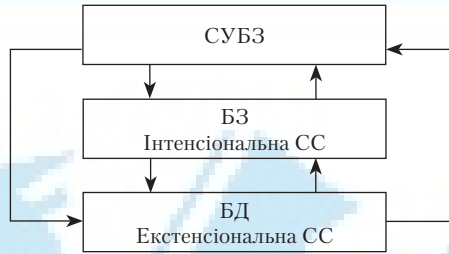


Рис. 2.9. Структура СПЗ

**Формування виведення в СС.** При функціонуванні СПЗ основним є інформаційно-пошуковий режим. *Запитом* називається набір фактів (ситуацій), при описі яких дозволяється використовувати змінні в позиціях значень атрибутів та імен відношень. Запит можна зобразити графом, вершини якого, що відповідають деяким змінним, не визначено. Пошук відповіді полягає в розв'язанні задачі ізоморфного вкладення графу запиту (або його підграфа) в СС.

Можна виокремити два основних типи запитів — на існування та на перелічення. *Запит на існування* не містить змінних і передбачає відповідь типу «ТАК», якщо ізоморфне вкладення графу запиту в СС здійснилось, і «НІ» — в протилежному разі. При обробці *запиту на перелічення* відбуваються пошук (перелічення) всіх можливих ізоморфних графу запиту підграфів в СС, а також конкретизація змінних.

**Приклад 2.10.** Як приклад розглянемо такий запит: «Оператор повідомив, що ЕРК щось перевозить. Треба визначити, що й куди перевозить ЕРК». Пошук відповіді здійснюється за СС, схему якої показано на рис. 2.8.

Запит можна зобразити графом (рис. 2.10), вершини  $X$  і  $Y$  якого потребують конкретизації, а  $F_6$  є фактом повідомлення. Очевидно, зіркоподібний підграф факту  $F$  у графі запиту (на рис. 2.10 показано штриховою лінією) може бути ізоморфно вкладеним у СС при суміщенні з  $F_3$  на рис. 2.8. При цьому конкретизуються вершини  $X$  і  $Y$ , а саме:  $X$  — касета,  $Y$  — ЦС. Відповідь виглядатиме так: «ЕРК перевозить касету до ЦС».

Цей приклад ілюструє пошук відповіді в СС у спрощеній формі, оскільки тут не розглядається процес інтерпретації запиту, який початково висловлюється спеціальною мовою спілкування.

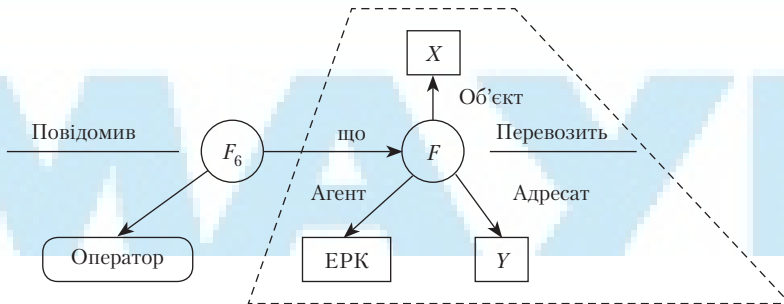


Рис. 2.10. Граф запиту



Семантична сітка може використовуватися для подання як декларативних, так і процедурних знань. При поданні декларативних знань застосовуються базові відношення, які характеризуються тим, що їх екстенціонал повністю зберігається в БД. Процедурні знання пов'язані віртуальними відношеннями, які не мають екстенціоналу, що зберігається в явній формі. В описах цих відношень є правила (програми), що дають змогу конструювати факти за необхідності.

Віртуальні відношення реалізуються або у вигляді обчислюваних відношень, які мають функціональний або предикатний характер, або у вигляді правил з використанням властивості симетричності, рефлексивності або транзитивності (в тих випадках, коли вони стосуються предметної галузі, що моделюється).

Нарешті, слід зазначити, що подана за допомогою СС на певному ієрархічному рівні об'єктна галузь може на вищому рівні абстрагування подаватися об'єктом — атрибутивною парою, і для подання подій на цьому ієрархічному рівні необхідно буде створити екстенціональну СС, в якій зазначений об'єкт — атрибутивна пара (з описом об'єктної галузі нижчого рівня) матиме відношення через відповідні факти з іншими об'єктами — атрибутивними парами на цьому вищому рівні подання узагальненого об'єкта. Так, розглянутий приклад взаємодії окремого виробничого верстатного модуля з транспортно-нагромаджувальною системою можна узагальнити на вищому рівні подання функціонуванням дільниці, яка складається з декількох верстатних модулів, які на рівні виробничої дільниці представлятимуть відповідні об'єкти — атрибутивні пари.

#### 2.4. Подання знань у вигляді фреймів

**Базові положення теорії і структура фреймів.** Фрейми були запропоновані у 1975 р. Марвіном Мінським [71] як *спосіб опису просторових сцен*. Фрейм є парадигмою для подання знань з метою використання цих знань комп'ютером. В основі теорії фреймів лежить *сприйняття стереотипних ситуацій*, що виникають, наприклад, у процесі функціонування складних об'єктів, зокрема й виробничих. Для подання та опису стереотипних об'єктів, подій або ситуацій введено поняття фреймів, що є *складними структурами даних*.

**Означення 2.17.** Фрейм — це мінімально можливий опис суті якогось явища, події, ситуації, процесу або об'єкта (мінімально можливий означає, що при подальшому спрощенні опису втрачається його повнота, він перестає визначати ту одиницю знань, для якої й призначався).

З іншого боку, фрейм — це спеціальна структура даних для подання деякого концептуального об'єкта. Всі фрейми пов'язані і утворюють єдину фреймову систему, в якій органічно *об'єднані декларативні та процедурні знання*.

*Розв'язання конкретних задач (дій)* здійснюється з використанням специфічного набору фреймів, що були створені раніше. Зміна задач (дій) спричинює зміну фреймів, що використовуються для її розв'язання, в яких необхідно буде відобразити нові знання.

У загальному вигляді фрейм можна розглядати як сітку (або мережу), що складається з кількох вершин і відношень — саме вони і є **алфавітом формальної фреймової системи**. На верхньому рівні фрейму міститься фіксована інформація: факт стосовно стану об'єкта, який зазвичай вважається істинним — це й приймається за **аксіоматичні твердження**. На наступних рівнях подається інформація

стосовно фрейму, яка міститься в множині *термінальних слотів* (терміналів), які обов'язково повинні бути заповнені конкретними значеннями та даними.

Щоб подати семантичну сітку у вигляді сукупності фреймів, треба вмти зображати відносини між вершинами сітки. Для цього також використовуються *слоти фреймів*, які можуть мати імена вигляду «Зв'язок  $Y$ », де  $Y$  – ім'я того відношення (його тип), яке встановлює певний фрейм-вершина з іншим фреймом-вершиною.

Фрейм утворюється шляхом об'єднання всіх бінарних фактів, пов'язаних з описуванним об'єктом, який формально у рамках фреймової моделі подається сукупним описом усіх основних характеристик об'єкта так:

**Ім'я фрейму, ((Атрибут\_1, значення\_1),**

**(Атрибут\_2, значення\_2),**

...

**(Атрибут\_m значення\_m))**

Окремі атрибути у фреймі описуються структурами даних, які називаються *слотами* цього фрейму.

Фреймова структура описує узагальнене, родове поняття – групу (клас) однотипних об'єктів з однаковими характеристиками, тоді як окремі конкретні об'єкти є *екземплярами* фреймів, опис яких формується шляхом конкретизації останніх, тобто заповненням слотів фрейму конкретними значеннями (що відповідає процедурі визначення значень атрибутів). Якщо ж при описі загального фрейму деякі слоти уже заповнені конкретними значеннями, ці значення передаються всім екземплярам цього фрейму. Таким чином, будь-який промисловий робот (ПР) може бути описаний на основі фрейму «Промисловий робот», якщо підставити у відповідні слоти конкретні дані цієї моделі ПР.

Для фреймових моделей характерна *ієрархія понять*. Наприклад, фрейм «Промисловий робот» є похідним від загальнішого фрейму «Автоматична машина», і тоді він *наслідує* (успадковує) слоти останнього, який в цьому разі заведено називати *батьківським фреймом*. Для нашого прикладу фрейм «Автоматична машина» може мати слоти «Модель», «Серія», «Номер», і тоді відповідні слоти фрейму «Промисловий робот» можна не задавати, а навести інші, які розкривають його властивості, наприклад, «Робоча система координат», «Вантажопідйомність». Заповнені конкретними значеннями слоти можуть успадковуватися фреймами-нащадками – *дочірніми фреймами*.

Як значення слота може бути новий фрейм, що дає змогу на множині фреймів здійснювати ієрархічну класифікацію.

Отже, важливою властивістю слота є *режим успадкування*: *обов'язкове успадкування; успадкування певної умови за певної умови; відмова від успадкування* тощо. У кожному слоті задається умова, яка повинна виконуватися при встановленні відповідності між значеннями (слот або сам встановлює відповідність, або це робить дрібніша складова фрейму). Проста умова позначається позначкою і може, наприклад, містити вимоги, щоб відповідність устанавлював користувач або щоб опис значень був досить повним тощо. Складні умови визначають відношення між фактами, що відповідають кільком терміналам. Поеднавши множину фреймів, які є відношеннями, можна побудувати фреймову систему, найважливішою перевагою якої є можливість перетворювати фрейми в одній системі – саме таким чином реалізується другий постулат формальної системи, тобто *механізм створення складних утворень* у вигляді ієрархічної фреймової системи.

Отже, можна сформулювати означення сценарію у фреймовій СПЗ.

**Означення 2.19.** *Сценарій* – це фреймоподібна структура знань, яка визначає послідовність подій, характерних для певного процесу чи для певної ситуації, або причинно-наслідкові зв'язки між подіями [21]. За таким означенням процеси в дискретно-подійних системах, якими і є ГКІС, можна подавати фреймовими моделями. До речі, поняття «сценарію» притаманне і моделям знань на СС, коли останні розглядаються як сітка, всі зв'язки якої мають тип «причина – наслідок» або часові «раніше – пізніше», що ще раз підкреслює органічний зв'язок між СС та ФС.

Що стосується четвертої необхідної умови відповідності фреймової моделі подання знань формальній системі – наявності **механізму логічного виведення**, то, оскільки в самій фреймовій системі його немає, такий механізм створюється штучно за допомогою відповідних процедур, наприклад, процедур мови LISP (про що детальніше говориться далі в підрозд. «Способи управління виведенням»).

Якщо аналізується видимий об'єкт, то різні фрейми однієї системи описують його з різних кутів зору, а перетворення одного фрейму на інший показує результат переходу з одного пункту спостереження до іншого. В одній системі різні фрейми можуть мати спільні термінали.

Зображається фрейм у вигляді ланцюжка:

Фрейм=<слот1><слот2>...<слотN>.

Як приклад розглянемо фрейм для поняття «взяття».

«Взяття»:

(Суб'єкт,  $X_1$ );

(Об'єкт,  $X_2$ );

(Місце,  $X_3$ );

(Час,  $X_4$ );

(Умова,  $X_5$ ).

У цьому фреймі вказано імена слотів (суб'єкт, об'єкт і т. д.), але замість їхніх значень стоять змінні ( $X_1$ ,  $X_2$  і т. д.). Такий фрейм називається *фреймом-прототипом*, або *протофреймом*.

Протофрейми зберігають знання про саме поняття. Наприклад, поняття «взяти» пов'язано з наявністю слотів з вказаними іменами. Взяття здійснює  $X_1$  в місці  $X_3$  під час  $X_4$ , якщо виконана умова  $X_5$ . Бере  $X_1$  щось, позначене як  $X_2$ . Підставляючи замість всіх змінних конкретні значення, отримуємо конкретний факт-опис:

«Взяття»:

(Суб'єкт, РОБОТ);

(Об'єкт, ДЕТАЛЬ);

(Місце, ПРИЙМАЮЧИЙ БУНКЕР);

(Час,  $X_4$ );

(Умова, В БУНКЕРІ Є ДЕТАЛЬ, А У РОБОТА ЇЇ НЕМА).

У штучному інтелекті фрейми, в яких позначені всі основні слоти (вони якимсь чином позначаються в описі фрейму), називаються *фреймами-екземплярами*, або *екзофреймами*. В нашому прикладі, очевидно, основними для фрейму «взяття» можна вважати слоти з іменами «суб'єкт» і «об'єкт». Оскільки до складу фрейму можуть входити слоти з іменами дій, фрейми підходять для подання як декларативних, так і процедурних знань.

**Властивості фреймів.** Розглянемо основні властивості фреймів.

**Базовий тип.** При ефективному використанні фрейму можна досягти швидкого розуміння суті цього предмета та його стану, проте для запам'ятовування різних

позицій у вигляді фреймів потрібна пам'ять. Тому тільки найважливіші об'єкти цієї позиції запам'ятовуються у вигляді базових фреймів, на основі яких будуються фрейми для нових станів. При цьому кожен фрейм містить слот, оснащений покажчиком підструктури, що дає змогу різним фреймам спільно використовувати однакові частини. Така структура не змінюється при зміні кута зору. Завдяки цій властивості фреймів можливі подання та використання інформації, одержаної в різний час і з різних ділянок як незалежної інформації, пов'язаної з цим кутом зору (тобто як знання).

*Процес зіставлення.* Процес, у ході якого перевіряється правильність вибору фрейму, називається процесом зіставлення. Фрейм містить умови, які обмежують значення слота. Мета полягає в тому, щоб визначити, які з цих умов, стосуючись цієї ситуації, є релевантними.

*Ієрархічна структура.* Фрейм зазвичай відповідає поданню загального поняття з класифікаційною ієрархічною структурою, а особливість останньої полягає в тому, що інформація про атрибути, які містить фрейм верхнього рівня, може спільно використовуватися всіма фреймами нижчих рівнів, що з ним пов'язані.

*Міжфреймові сітки.* Запам'ятовування концептуального об'єкта, який має класифікаційну ієрархічну структуру, легко пояснюється фреймовою моделлю. Проте якщо процес зіставлення закінчується неуспіхом, постає необхідність пошуку фрейму, подібного до попереднього. Такий пошук можливий завдяки поєднанню фреймів, які описують об'єкт з незначними відмінностями, з наявними покажчиками й утворенню сітки подібних фреймів.

*Значення за умовчанням.* Коли людина щось розглядає і думає про те, що б це означало, або щось уявляє собі і думає про те, що б це могло бути, то такі процеси можна подати як розподіл конкретних значень між термінальними слотами фрейму. При цьому в разі уявного подання межі стосовно розподілу цих значень є широкими. В таких випадках уявне значення називається *значенням за умовчанням*, яке слабо пов'язане зі слотами і далі поступово замінюється достовірною інформацією.

Виведення за умовчанням виконує дуже важливу функцію при розпізнаванні образів або мовленні. Наприклад, якщо видно тільки частину образу, то, замінивши решту його частин значеннями за умовчанням, можна окреслити повний образ. Так само, користуючись значенням за умовчанням, можна відновити зміст контексту, з якого взято окремі речення.

*Відношення «абстрактне – конкретне» та «ціле – частина».* Відношення «абстрактне – конкретне» характерні тим, що на верхніх рівнях розташовані абстрактні об'єкти (концепти), а на нижчих – конкретні об'єкти, причому останні успадковують атрибути об'єктів верхніх рівнів. Ці відношення називаються ще відношеннями типу *IS – A* (це є) або *KIND – OF* (це різновид).

Відношення «ціле – частина» стосується структурованих об'єктів і свідчить про те, що об'єкт нижчого рівня є частиною об'єкта верхнього рівня.

Практично у фреймових системах застосовуються лише відношення «абстрактне – конкретне». Відношення «ціле – частина» використовують при описі й управлінні структурованим об'єктом, наприклад, у системах автоматизованого проектування (САПР) та робототехніці.

**Особливості подання знань фреймами.** До основних особливостей подання знань фреймами насамперед слід зарахувати структуру їх даних, а також способи управління виведенням.

Фрейм характеризує певну структуру, а фреймова система — це ієрархічна структура, вузлами якої є подібні фрейми. Розглянемо значення кожного з елементів, з яких складається фрейм (рис. 2.11).

*Ім'я фрейму* — це ідентифікатор, який присвоюється фрейму. Він повинен мати лише одне в цій фреймовій системі ім'я (*унікальне ім'я*). Кожний фрейм, як випливає з рис. 2.11, складається з довільної кількості слотів, деякі з яких визначаються самою системою для виконання специфічних функцій, а решту встановлює користувач. До складу фрейму входять: слот *IS – A* (вказує на фрейм-батька цього фрейму); слот покажчиків дочірніх фреймів, які є списком покажчиків цих фреймів; слот для виведення імені користувача, дати визначення фрейму, дати зміни коментарів, тексту та інші слоти. Кожен слот, у свою чергу, має певну структуру даних.



Рис. 2.11. Структура даних фрейму

*Ім'я слота* — це ідентифікатор, який присвоюється слоту. Слот повинен мати унікальне ім'я у фреймі, до складу якого він входить. Зазвичай ім'я слота не несе жодного змістового навантаження, а є тільки ідентифікатором цього слота, хоча іноді й може мати специфічний зміст. До таких імен, крім *IS – A* (відношення *IS – A*), *DDESENDENTS* (показник прямого дочірнього фрейму), *FINEDON* (користувач, який визначає фрейм), *DIFINEDON* (дата визначення фрейму), *MODIFINEDON* (дата модифікації фрейму), *COMMENT* (коментар) тощо можна зарахувати імена, що використовуються для подання структурованих об'єктів, наприклад *HASPART*, *RELATION* та ін. Ці слоти називаються системними і застосовуються при реалізації БЗ та управлінні виведенням.

*Показники спадковості.* Ці показники стосуються тільки фреймових систем ієрархічного типу, що ґрунтуються на відношеннях «абстрактне — конкретне». Вони показують, яку інформацію про атрибути слотів у фреймі верхнього рівня успадковують слоти з такими ж іменами у фреймі нижчого рівня. Типовими показниками спадковості є *Unique* (*U*: унікальний), *Same* (*S*: такий самий), *Range* (*R*: встановлення меж), *Override* (*O*: ігнорувати) тощо. Показник *U* свідчить про те, що кожен фрейм може мати слоти різного призначення; *S* — що всі слоти повинні мати однакові значення; *R* — значення слотів фрейму нижчого рівня повинні перебувати у межах, зазначених значеннями слотів фрейму верхнього рівня; *O* — за відсутності вказівки значення слота фрейму верхнього рівня стає значенням слота

фрейму нижчого рівня, проте при визначенні нового значення слотів фреймів нижчих рівнів ці значення застосовуються для слотів фреймів нижчого рівня. Показчик *O* виконує одночасно функції показчиків *U* та *S*. Незважаючи на те, що більшість систем передбачає кілька варіантів позначення спадковості, існують і такі, які передбачають тільки один варіант. При цьому можна вважати, що використується показчик *O* значення за умовчанням.

**Показчик типу даних.** Зазначається, що слот або має числове значення, або служить показником іншого фрейму (його імені). До типів даних належать *FRAME* (показчик), *INTEGER* (цілий), *REAL* (дійсний), *BOOL* (булевий), *LISP* (приєднана процедура), *TEXT* (текст), *LIST* (список), *TABLE* (таблиця) та ін.

**Значення слота.** Воно має збігатися з вказаним типом даних цього слота і, крім того, повинна виконуватись умова спадковості.

**Демон** — це процедура, що здійснюється автоматично при виконанні деякої умови. Демони застосовуються при зверненні до відповідного слота. Наприклад, демон *IF – NEEDED* використовується тоді, коли на момент звернення до слота значення останнього не було встановлено; демон *IF – ADDED* — при підставлянні в слот його значення; демон *IF – REMOVED* — при стиранні значення слота. Зазначимо також, що демон є різновидом приєднаної процедури.

**Приєднана процедура.** Як значення слота застосовується програма процедурного типу. Приєднана процедура здійснюється за повідомленням, яке надходить з іншого фрейму. Коли кажуть, що в моделях подання знань фреймами об'єднуються процедурні та декларативні знання, то демони і приєднані процедури є процедурними знаннями. Крім того, в мові подання знань фреймами відсутній спеціальний механізм управління виведенням, тому користувач повинен реалізувати цей механізм за допомогою приєднаної процедури.

Отже, як **алфавіт** у фреймовій системі подання знань виступають фрейми зі слотами, як **механізм створення складних утворень** — відношення типу «абстрактне — конкретне» (*IS – A*) та «ціле — частина» або «різновид» (*KIND – OF*), за **аксіому** береться фіксована інформація фрейму верхнього рівня, яка вважається істинною, **механізмом управління виведенням** служить приєднана процедура мови *LISP*, а також демони *IF – NEEDED*, *IF – ADDED* та *IF – REMOVED*.

**Способи управління виведенням.** Розглянемо конкретні структури систем і способи управління виведенням, які описано в [132].

**Приклад 2.11.** На рис. 2.12 показано простий приклад ієрархічної структури фреймової системи планування. Ця структура ґрунтується на відношеннях *IS – A* між фреймами, які описують деяку конференцію. Вважається, що ці фрейми повинні містити такі пункти, як дата та місце проведення конференції, назва теми і прізвища доповідачів. Таким чином, на найвищому рівні ієрархії розміщується фрейм «конференція», що містить чотири згаданих слоти.

У розглядуваному випадку йдеться про конференцію з комерційних питань і конференцію з розвитку, які визначено як відповідні дочірні фрейми. Загальною темою конференції з комерційних питань є торгівля, а конференції з розвитку — освоєння нових видів продукції, що підставляються як значення у відповідні слоти. Крім того, у фреймі «конференція з комерційних питань» необхідним є пункт «мета» (обсяг торгівлі), а у фреймі «конференція з розвитку» задається бюджет (на освоєння), тому до цих фреймів додано відповідні слоти. Як фрейм, що є дочірнім щодо фрейму «конференція з комерційних питань», призначений фрейм «четверта конференція з комерційних питань», у чотири слоти якого (виняток становить слот «тема») введено конкретні значення.

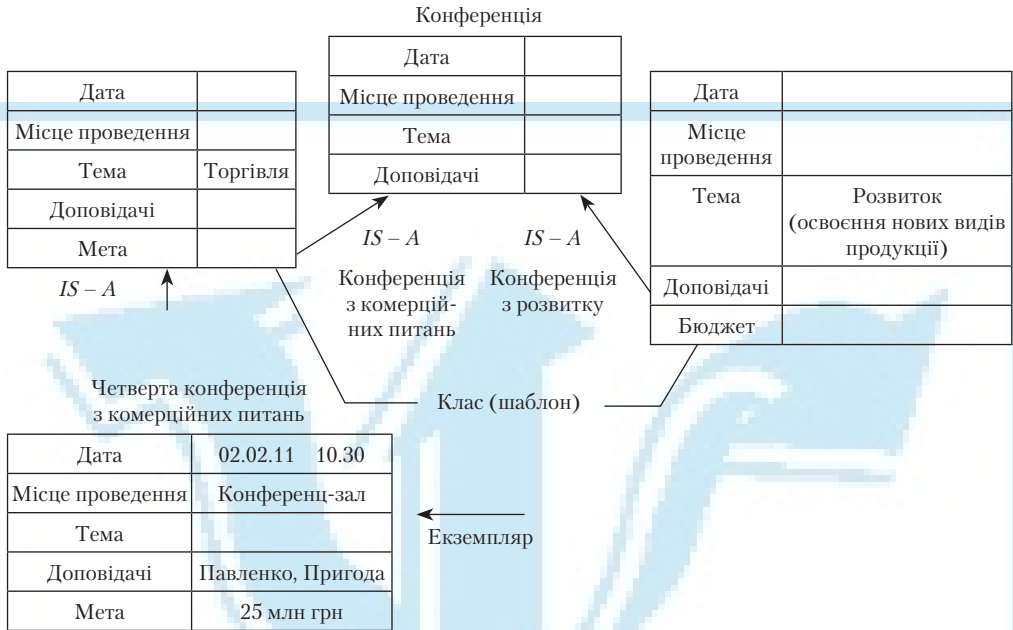


Рис. 2.12. Ієрархічна структура фреймової системи планування (на відношеннях IS-A між фреймами)

Фрейми, які описують різні об'єкти, називаються *шаблонами* (template), а фрейми верхнього рівня, що використовуються для подання цих шаблонів, — *фреймами класу*. Що ж стосується теми, то тут виконується така обробка. В момент звернення до слота «тема» фрейму «четверта конференція з комерційних питань» вводиться механізм управління спадковістю, за допомогою якого здійснюються пошук і застосування значення цього слота у відповідному фреймі верхнього рівня (якщо в цьому фреймі це значення відсутнє, то пошук відбувається в іншому фреймі верхнього рівня). В реальних системах у БЗ для економії пам'яті й через інші причини слоти з однаковими значеннями у фреймах верхнього рівня, як правило, не використовують.

**Приклад 2.12.** На рис. 2.13 наведено приклад фрейму «четверта конференція з комерційних питань» з приєднаними демонами. Як бачимо, в кожному слоті можна вказати три типи демонів. У загальних рисах демон функціонує так. У розглядуваному прикладі в слоті «місце проведення» визначено демон *IF - ADDED* з ім'ям «бронювання» (ім'я функції мови *LISP*), що автоматично вводиться при підставлянні в цей слот значення «конференц-зал». Якщо останній можна орендувати, то він бронюється. Якщо ж зал вже зайнятий, викликається повідомлення «бронювання неможливе».

Демон *IF - NEEDED* з ім'ям «хто?», приєднаний до слота «доповідачі», в разі якщо при зверненні до цього слота його значення було *NIL* (нуль), генерує запит: «хто виступатиме на четвертій конференції з комерційних питань?». Відповідь на цей запит передається при підставлянні вхідних даних користувача як значення слота.

Таким чином, з описаного випливає, що демони вводяться автоматично при зверненні до відповідного слота.

Четверта конференція з комерційних питань

Ім'я слота	Значення	if – needed	if – added	if – removed
Дата	02.02.11 10:30			
Місце проведення	Конференц-зал		Бронювання	
Тема				
Доповідачі	Павленко, Пригода, Гунько	Хто?		
Мета	25 млн грн			

*LISP*  
**proc** бронювання (назва конференції, місце проведення, дата)  
**if** можливо (місце проведення, дата)  
**then** забронювати (назва конференції, місце проведення, дата)  
**else** сповістити («бронювання можливе», місце конференції)  
**end**

Декларативні знання

Процедурні знання

Рис. 2.13. Приклад приєднаної процедури типу демона, що доповнює фрейм

**Приклад 2.13.** Рис. 2.14 ілюструє приклад приєднаної процедури, яка називається *службовою*, а також спосіб передавання повідомлень із застосуванням цієї процедури. Функція (команда) *MSG*, яка в приєднаній процедурі має назву «обчислити» і стоїть у слота «обчислення» фрейму *AA*, призначена для передавання повідомлень.

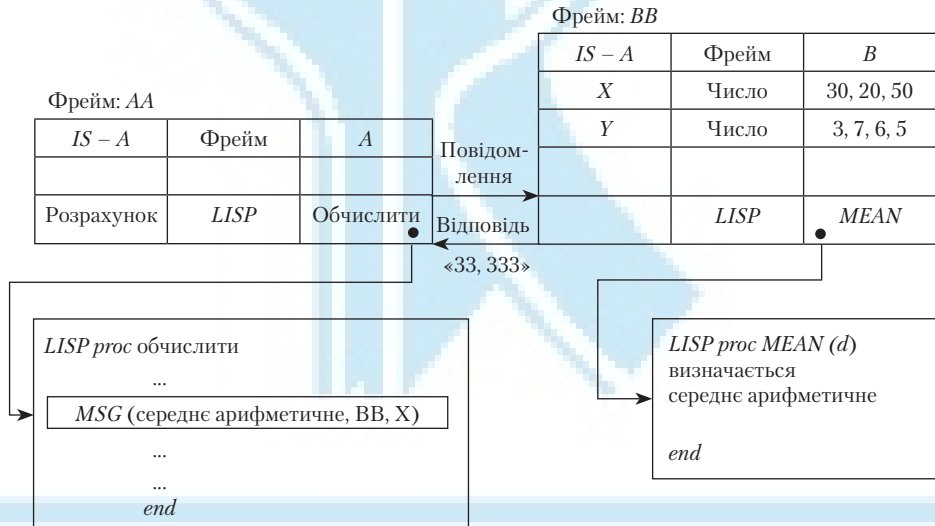


Рис. 2.14. Приклад службової процедури, що доповнює фрейм

Речення цієї функції має таку структуру:

*MSG* (ім'я фрейму, ім'я слота, параметр, ...).

Отже, три аргументи «середнє арифметичне», «*BB*» та «*X*» функції *MSG* є відповідно аргументами імені фрейму, імені слота та процедурної функції фрейму, що передає повідомлення. Механізм передавання (приймання) повідомлень за допомогою функції *MSG* працює за такою схемою. Якщо за повідомленням від іншого фрейму ініціюється приєднана процедура «обчислити» фрейму *AA*, то за допомогою функції *MSG* фрейму *BB* передається повідомлення, згідно з яким ініціюється приєднана процедура «середнє арифметичне» фрейму *BB*. За допомогою



цієї процедури обчислюється значення  $d$ , тобто середнє арифметичне трьох значень 30, 20, 50 слота «X». Результат обчислення (тобто число 33,333) передається у фрейм AA.

Таким чином, у рамках фреймового підходу припускають, що знання в системі подаються у вигляді окремих кластерів знань або підструктур, які містять відомості про стереотипи об'єктів (ситуацій). Тоді розуміння останніх для системи означає пошук у переліку накопичених структур такої, яка найкраще описує розглядуваний об'єкт. При цьому слоти заповнюються деякою інформацією, і заповнений фрейм (як вже наголошувалося, такий фрейм має назву *екземпляра фрейму* або *екзофрейму*) перевіряється на адекватність цьому об'єктові. В разі незбігу відшукується новий фрейм, і процес продовжується. Отже, технологія фреймового виведення знань складається з трьох складових: *створення екземпляра фрейму; активізації фреймів; власне організації виведень перевіркою адекватності екземпляра фрейму об'єкта (ситуації)*, що описується.

**Поповнення первинних описів у ФМ.** Первинний опис ситуації (об'єкта) людиною відбувається в тому вигляді, в якому він безпосередньо сприймається її органами чуттів, і надалі, з отриманням нових наявних знань про цю ситуацію, опис поповнюється та уточнюється. Отже, інтелектуальна система, сприймаючи деякий об'єкт, формує первинний його опис і зіставляє цей опис з деяким фреймом, що зберігається в її пам'яті (процеси формування первинного опису та зіставляння можуть відбуватися паралельно). При успішному зіставленні з деяким фреймом система відносить новий об'єкт до поняття, яке описується цим фреймом. Процедурно це відбувається так:

- формується загальна структура опису нового об'єкта (який розглядається тепер як *екземпляр первинного фрейму*);
- заповнюються всі слоти нового екземпляра, значення яких визначаються в описі родового (*батьківського*) фрейму або задаються в первинному описі;
- якщо опис недостатньо повний (не вистачає значень деяких важливих слотів) або суперечливий (наприклад, це може відбуватися при зіставленні з кількома фреймами або коли інформація, що вводиться, суперечить визначенням за успадкуванням дочірнім фреймам), активізуються процедури уточнення.

**Приклад 2.14.** Нехай у пам'яті інтелектуальної системи зберігається фрейм опису деякого технологічного процесу (ТП) в гнучкій комп'ютерно-інтегрованої системі (ГКІС):

**ТП\_В\_ГКІС ((Вид\_ТП, СКЛАДАННЯ),  
 (Початок,  $t_{\text{поч}}$ ),  
 (Кількість\_операцій, \_),  
 (Закінчення, Початок + Кількість\_операцій \*  $t_{\text{он}}$  +  
 $t_{\text{під_зак}}$  \* (Кількість\_операцій - 1),  
 (Зміна, \_),  
 (Шо, \_),  
 (Устаткування, \_),  
 (Підрозділ, <Механічний цех, Науково-виробниче об'єднання>)).**

Атрибут **Закінчення** є обчислюваним, бо його значення залежить від кількості операцій (кожна тривалістю  $t_{\text{он}}$  та підготовчо-заклучним часом  $t_{\text{під_зак}}$ ).

Нехай тепер у систему надходить інформація «*Технологічний процес виготовлення захватного пристрою (ЗП) відбувається в першу зміну на роботизованому складальному центрі (РСЦ)*». На основі вкладених у систему алгоритмів остання

виявляє, що ця інформація зіставляється з фреймом **ТП**, і формує опис на основі цього фрейму (фактично, новий екземпляр фрейму), присвоюючи цьому опису назву **ТП\_«Виготовлення\_ЗП»** з відображенням у заголовку, що це екземпляр фрейму **ТП\_В\_ГКІС**. Заповнюються відповідні слоти, і опис набуває вигляду:

**ТП\_Виготовлення\_ЗП (ТП\_В\_ГКІС) ((Вид\_ТП, СКЛАДАННЯ),  
(Початок,  $t_{\text{поч}}$ ),  
(Кількість\_операцій,  $\_$ ),  
(Закінчення, Початок + Кількість\_операцій \*  $t_{\text{оп}}$  +  
 $t_{\text{під\_зак}}$  \* (Кількість\_операцій - 1)),  
(Зміна, ПЕРША),  
(Що, Складання\_ЗП),  
(Устаткування, РСЦ),  
(Підрозділ, <Механічний цех, Науково-виробниче об'єднання>)).**

Слоти **Вид\_ТП**, **Підрозділ** та **Початок** були заповнені на основі базового (батьківського) фрейму, а слоти **Зміна**, **Що**, **Устаткування** – з нової інформації, що надійшла до системи. Водночас слоти **Кількість\_операцій** і **Закінчення** залишилися незаповненими, оскільки для цього системі не вистачає знань. Тому при ініціюванні користувачем запити «Коли закінчиться ТП складання ЗП в ГКІС (або РСЦ)?» система має активізувати **процедуру уточнення** через формування запити до користувача: «Із скількох операцій складається ТП виготовлення ЗП?» Після введення користувачем кількості операцій (тобто уточнення відповідного слота **Кількість\_операцій**) та здійснення після цього системою необхідних обчислень нею формується висновок (продукується виведення) щодо загальної тривалості та часу закінчення ТП.

Узагальнюючи цей приклад, можна вважати, що у типовому фреймі зберігаються такі знання:

- ім'я і загальний опис (*шаблон*) фрейму;
- інформація про батьківський фрейм;
- інформація про окремі слоти: їх імена та значення, властивості (зокрема, тип слота, умови коректності інформації, яка в ньому зберігається, тощо).

**Недоліки та переваги фреймової СПЗ.** Фреймова модель подання знань *характеризується значною гнучкістю*: вважається можливою комбінація декларативних і процедурних знань в одній одиниці їх подання – фреймі; можливими також є *ієрархічна побудова БЗ* згідно зі ступенем абстракції поняття та реалізація будь-якої системи *виведення з використанням об'єктно-орієнтованого методу управління* виведенням – обміну повідомленнями. Крім того, *наочність та зведення до мережних* та інших подань при використанні фреймових МПЗ дають переваги цій моделі над іншими МПЗ. Нарешті, фреймові подання пояснюють *швидкий доступ до пам'яті*, що виводить пам'ять на провідне місце при обговоренні проблем штучного інтелекту.

Однак окрім наведених вище переваг при застосуванні фреймових моделей виникає ряд проблем, які треба враховувати при виборі МПЗ.

*По-перше*, відповідно до прикладної задачі постає необхідність будувати модель ЕС з використанням багаторівневої ієрархічної структури.

*По-друге*, існує проблема, пов'язана з приєднувальною процедурою при виборі фреймової структури, оскільки мови цього рівня орієнтовані на фахівців у галузі штучного інтелекту та інженерії знань. Інакше кажучи, обробки знань здебільшого написані саме мовами штучного інтелекту типу Лісп та Пролог, проте нерідко для підвищення швидкості виконання програм і використання не найновіших комп'ютерів їх описують мовами високого рівня (Сі, ФОРТРАН). Отже,

залишається суперечність між необхідним рівнем інтелекту для опису фреймової структури та прагненням обмежитись при цьому рамками простих методів подання знань.

*По-третьє*, фреймові системи можуть використовуватися не тільки для опису знань, а й для написання алгоритмів їх виведення. З точки зору користувачів, які добре розуміються на обробці знань, це дає переваги, але для звичайного користувача є великим навантаженням.

*По-четверте*, та обставина, що у фреймових системах знання задаються, по суті, процедурами, порівняно з іншими методами процес здобуття знань ускладнюється, а можливості динамічної адаптації фреймової системи до змін зовнішнього середовища збіднюються. Відношення ж між фреймами в складних проблемах стають складнішими.

*По-п'яте*, зв'язок слотів з процедурами, а також те, що вони передають один одному управління завдяки обміну повідомленнями, означає, що структура фреймової системи описує структуру управління процедурою обробки знань. Внаслідок цього фреймова система зводиться до форми подання одиниць керуючих структур і одиниць структур даних у загальних фреймах, через що дуже часто вдаються до досить надуманих прийомів. При цьому втрачається легкість інтуїтивного розуміння процесів подання та виведення знань, що є особливістю цієї системи.

Нарешті, *по-шосте*, однією з труднощів подання знань і мови фреймів є відсутність формальної семантики, що ускладнює порівняння властивостей подання знань різних мов фреймів, а також повне логічне пояснення мови фреймів.

## 2.5. Модель дошки оголошень

**Основні положення моделі дошки оголошень (ДО).** У продукційній системі в БЗ накопичуються однакові за формою правила, і тому, якщо припустити, що в одній і тій самій системі вирішуються проблеми з кількох різних прикладних галузей, то в ній будуть змішані правила з різними властивостями, які стосуються цих галузей, що спричинює суттєве зниження ефективності обробки інформації. Отже, оперування різноманітними проблемами в одній продукційній системі небажане.

Однак бувають такі випадки, коли одна проблема конструюється з кількох окремих проблем із різними властивостями. До того ж ці проблеми можуть бути не обов'язково роз'єднаними і можуть зберігати незалежність. Наприклад, роз'єднати окремі проблеми неможливо за наявності між ними такого слабкого зв'язку, коли результатом вирішення однієї з них ініціюється початок вирішення наступної.

Однією з таких проблем є розпізнавання мови. Свого часу було створено систему HEARSAY-II [92], в якій використано модель ДО, або класної дошки (blackboard model). В цій моделі для кожної окремої проблеми, які в сукупності становлять одне ціле, є відповідна множина знань, яка через загальну робочу пам'ять (що й дістала назву ДО) управляється таким чином, що всі знання використовуються узгоджено як одне ціле. Такі відокремлення множини знань називаються джерелами знань, причому кожне з них самостійно будується як продукційна система.

Уся робоча пам'ять розподіляється на рівні (наприклад, рівень складів і рівень слів), а дані зберігаються у формі «об'єкт — атрибут — значення». Кожне джерело знань (ДЗ) визначається як сукупність правил, що виконують спеціальні функції між рівнями робочої пам'яті (наприклад, виведення між рівнем окремих складів і рівнем слів).

Таким чином, можна побудувати систему, що здійснює виведення шляхом ініціювання ДЗ відповідно до ситуації, причому робоча пам'ять у системі відіграє роль ДО, яка спільно використовується ДЗ (рис. 2.15) і водночас є засобом для одержання інформації.

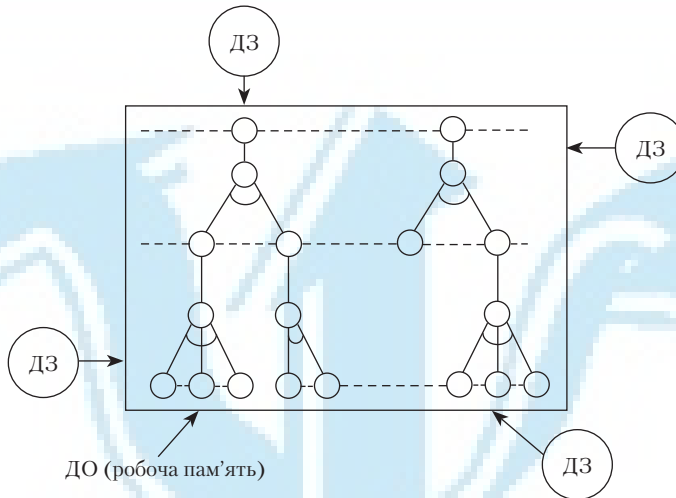


Рис. 2.15. Модель ДО

У моделі ДО кожне ДЗ служить вузлом зв'язку між даними нижчого рівня, на основі яких робиться висновок, і даними верхнього рівня, що характеризують результат виведення. Ці дані змінюються відповідно до своїх об'єктів. Якщо вважати, що кожне ДЗ належить об'єкту, який характеризує результат виведення на верхньому рівні (об'єкт верхнього рівня), то можна сказати, що цей об'єкт дуже близький до фрейму в тому значенні, що він доповнений правилами механізму виведення, які стосуються даних і функцій управління даними.

Справді, якщо в таку систему ввести концептуальну ієрархію об'єктів і відповідну спадковість атрибутів, то вона перетвориться на фреймову систему (дуже близьку до неї). І навпаки, фреймову систему з розподіленими на групи правилами можна використовувати для визначення стратегії виведення. До того ж така фреймова система доповнюється потужними засобами, пов'язаними з процедурою виведення. Ось чому методи побудови систем із ШІ, які об'єднують МЛВ та фреймові системи, є перспективними.

Відповідність моделі ДО умовам формальної системи (див. підрозд. 1.7) полягає в такому:

- як **базовий алфавіт** у моделі ДО виступають власне ДО і ДЗ;
- правила створення ієрархічного зв'язку гіпотез між рівнями у ДО є **механізмом створення складних утворень**;
- самі джерела знань і правила міжрівневого зіставлення останніх у робочій пам'яті (тобто ДО) моделі є **аксіоматичними твердженнями** — третьою умовою визначеності СП як формальної системи;
- нарешті, можливості управління даними (висхідне виведення) або самою моделлю (низхідне виведення) формують **механізми логічного виведення**.

**Концептуальна модель ДО.** Концептуальну схему моделі ДО показано на рис. 2.16.

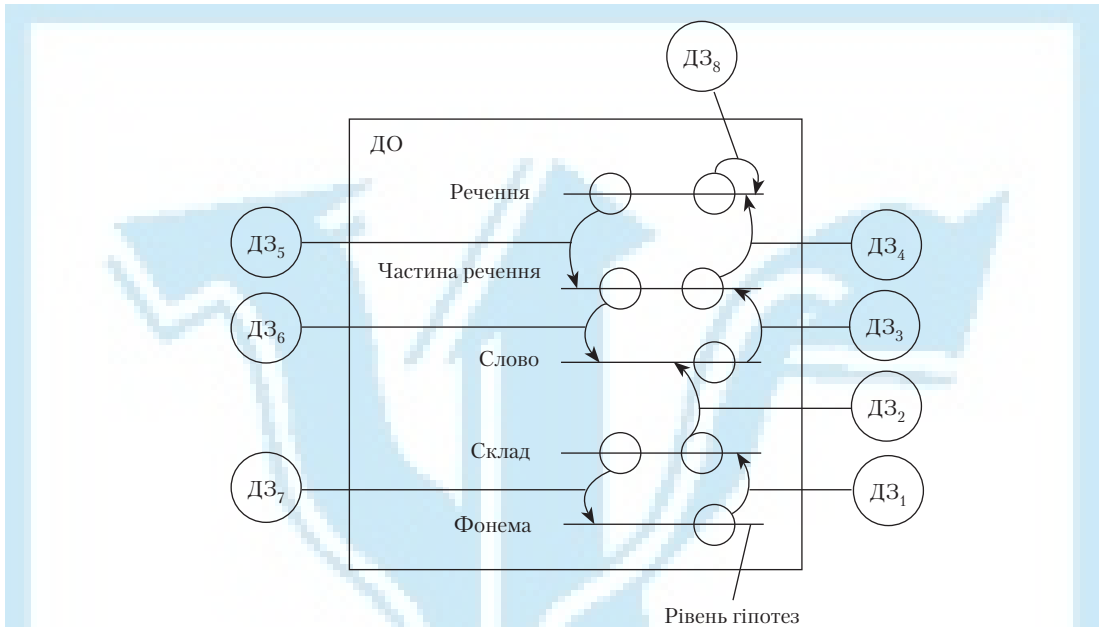


Рис. 2.16. Концептуальна схема моделі ДО

Центральна частина ДО ієрархічно визначає гіпотези: на найвищому рівні розташовані висновки, на найнижчому — факти, на проміжних — проміжні гіпотези. Наприклад, при введенні мови у вигляді сигналів часового ряду в системі, що синтезує з них повний текст, визначаються рівні гіпотез зверху до низу за типом «речення — частина речення — слово — склад — фонема».

Навколо ДО розміщуються ДЗ, які діють на власних рівнях гіпотез або між рівнями. Наприклад, ДЗ<sub>1</sub> призначене для формування гіпотез оцінювання фонем. Таким чином, послідовно з низу до верху формуються гіпотези верхніх рівнів, а на найвищому з них утворюється повне речення.

Проте утворення речення з послідовності сигналів не завжди є успішним: через шуми та нечіткість виникає велика кількість гіпотез-кандидатів з високою ймовірністю їх достовірності. В таких випадках у напрямку зверху до низу використовуються знання з граматики з урахуванням контексту, щоб запобігти великій кількості кандидатів. Наприклад, ДЗ<sub>8</sub> можна підібрати так, щоб воно оцінювало придатність висунутої гіпотези-кандидата як остаточного варіанта речення.

*Висхідним* називається виведення, що управляється даними, а *низхідним* — виведення, що управляється моделлю. Ці терміни відповідають зазначеним вище прямому та зворотному виведенням.

Таким чином, модель ДО відіграє роль каналу зв'язку між ДЗ. Крім того, оскільки в цій моделі чітко визначено функції кожного ДЗ, а також ситуації, коли їх слід використовувати, та ефективність застосування ДЗ, вона має істотну перевагу з точки зору систематизації знань і підвищення ефективності виведення.

Оцінювання продукції, що міститься в ДЗ, виконується тільки при його активізації, тому важливою проблемою при управлінні виведенням є вибір послідовності ініціювання ДЗ. Основні методи, що тут використовуються, це метод «планувальника», а також метод «висування кандидатури відправника». В першому випадку планувальник оцінює ситуацію на ДО і послідовно називає відправників – ДЗ. У другому випадку відправник сам називає ім'я наступного відправника.

## 2.6. Моделі штучних нейронних сіток

**Загальні положення.** Кора великих півкуль головного мозку людини містить близько 14 млрд нейронів, їх короткі та довгі відростки – *дендрити*, якими надходять вхідні дії, та *аксони*, що відводять вихідні реакції, утворюють надзвичайно складні переплетення зв'язків. Склад і закони функціонування нейрона також дуже складні. Тому при моделюванні нейронів користуються спрощеними описами моделі нейронних сіток, які дістали назву *нейроподібних сіток*.

Нейронні сітки, або штучні нейронні сітки, відображають розвиток моделей, які з'явилися завдяки спробам імітування механізму мислення людини. Для певних класів задач, що стосуються ШІ (насамперед розпізнавання, інтелектуального управління динамікою складних механічних систем тощо), нейросіткові моделі часто забезпечують більш ефективні рішення, ніж традиційні символічні підходи.

Нейронна сітка є сукупністю нейронів і зв'язків між ними. Кожний нейрон виконує порівняно просту функцію, але, будучи об'єднаними численними зв'язками в єдину структуру, всі вони разом утворюють систему, здатну розв'язувати складні задачі.

Оскільки модель штучної нейронної сітки (ШНС) служить моделлю подання знань у системах ШІ, визначимо відповідність апарату ШНС умовам формальної системи (див. підрозд. 1.7), а саме:

- **базовим алфавітом** у ШНС є *вершини* (входів, виходів, тіл нейронів, вагових настроювачів та активаційних функцій), спрямовані дуги (у прямо спрямованих та обернено спрямованих топологіях ШНС);
- правила з'єднання ребрами (спрямованими дугами) вершин є **механізмом створення складних утворень** з елементів базового алфавіту ШНС, тобто формальним апаратом подання топологій ШНС;
- навчання ШНС (з учителем, без вчителя або методами самоорганізації) шляхом визначення настроювальних ваг становить основу **аксіоматичного твердження** – третьої умови визначеності СП як формальної системи;
- нарешті, можливість ініціювання зміни вхідних векторів (образів) у ШНС в результаті їх ідентифікації, класифікації, кластеризації шляхом ототожнювання з аксіоматичними твердженнями – результатами навчання – формує **механізм логічного виведення**.

**Функціональні особливості нейрона.** Елементарна складова сітки – нейрон – має кілька входів і один вихід. Структуру нейрона зображено на рис. 2.17, а (це Adeline – adaptive linear element [43; 44; 199; 200]). Елементи вхідного вектора множаться на вагові коефіцієнти  $W_1, W_2, \dots, W_n$ ; зважені значення підсумовуються й надходять до входу порогового елемента, двійковий вихід якого є виходом нейрона. У загальному випадку для відтворення нелінійностей у нейроні можуть застосовуватися не тільки ступінчасті, а й похилі функції (рис. 2.17, в та г відповідно).

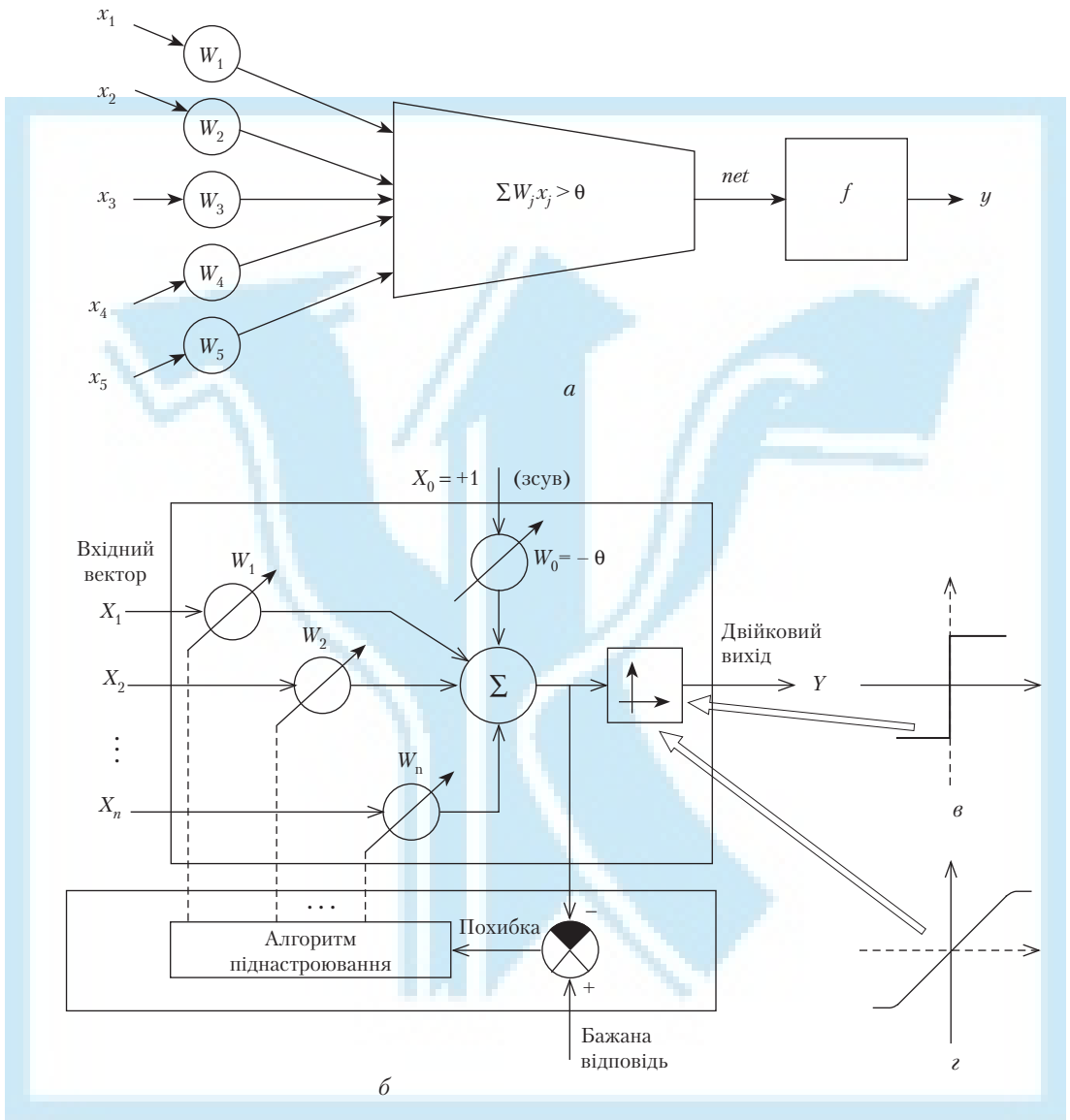


Рис. 2.17. Спрощена структура нейрона (а) (тут  $x_1 \dots x_5$  – вхідні сигнали;  $W_1 \dots W_5$  – синапси (вагові коефіцієнти);  $\theta$  – порогова величина;  $f$  – активаційна функція;  $y$  – вихідний сигнал), зображення нейрона у вигляді процесорного блоку (б) та варіанти нелінійностей у ньому (в, г)

Нейрон може набувати лише два стани: або він збуджений, або пасивний. Тому для цих двох станів досить використовувати двійкові числа, наприклад, числа 0 і 1. Входами для нейрона є збудження, які надходять до нього дендритами. Через те що ці сигнали є виходами попередніх нейронів, їхні величини також будуть дорівнювати 1 або 0. Сигнали дендритів перед надходженням до нейрона підсилюються або послаблюються. Це легко піддається моделюванню,

при якому їх перемножують з додатними або від'ємними дійсними числами: додатні числа призначаються для посилення, а від'ємні — для ослаблення сигналів. Коли оброблені в такий спосіб вхідні сигнали досягають нейрона, вони збуджують його лише тоді, якщо сума усіх величин потенціалів перевищить деяке граничне значення.

У моделі нейрона формується сума всіх входів, перемножених на значення синапсів, і порівнюється з *граничним значенням*  $\theta$ . Якщо сума перевищує  $\theta$ , вихідний сигнал нейрона (величина сигналу аксона) встановлюється на 1, в протилежному разі — на 0. Формальне подання функцій нейрона наведено на рис. 2.17, а, де  $x_1, x_2, x_3, x_4, x_5$  — вхідні величини;  $w_1, w_2, w_3, w_4, w_5$  — величини синапсів (ваг), якщо сума  $\sum w_i x_i > \theta$ , то нейрон збуджений, тобто  $y = 1$ .

При цьому для нейрона справедливі такі кількісні співвідношення:

- вхідні величини  $x_1, x_2, \dots, x_a$  дорівнюють 1 або 0;
- вагові коефіцієнти  $w_1, w_2, \dots, w_a$  — будь-які дійсні числа.

Ці вхідні сигнали, у сукупності позначені вектором  $\overline{X}$ , відповідають сигналам, що надходять у біологічні нейрони. Аналогічно множину ваг позначають вектором  $\overline{W}$ . Підсумовуючий блок, що відповідає тілам біологічних елементів (SOM'и), додає зважені входи алгебраїчно, утворюючи вихід, що дістав назву *net*. У векторних позначеннях це має вигляд:

$$net = \overline{XW}.$$

У розглянутому прикладі  $y = f(net - \theta)$  — вихідний сигнал нейрона, в якому активаційна функція  $f(x)$  може, наприклад, набувати значення:

$$f(x) = \begin{cases} 1, & \text{якщо } x > 0 \\ 0, & \text{якщо } x \leq 0. \end{cases}$$

Тут  $net = \sum w_i x_i$  — алгебраїчна сума зважених входів.

Поведінку нейрона з трьома входами можна проілюструвати так:  $y$  дорівнює точно одиниці, якщо входи 1,  $x_1, x_2, x_3$ , перемножені відповідно на ваги  $\theta, w_1, w_2, w_3$ , в сумі перевищують 0, тобто

$$\theta \times 1 + w_1 x_1 + w_2 x_2 + w_3 x_3 > 0.$$

Для введення граничного значення  $-\theta$  у загальну суму передбачається (див. рис. 2.17, б) додатковий вхід  $X_0$ , що завжди дорівнює 1. Цей допоміжний вхід спрощує формалізацію, при якій граничне значення  $-\theta$  зараховується до ваг, і опитування нормується на число 0. Цей додатковий вхід зі стандартною величиною 1 і вагою  $W_0 = -\theta$  дістав назву *Bias* (зсув).

*Активувальна функція*  $f(x)$  може в деталях варіюватися. Вище стани нейрона були позначені двійковими величинами 0 і 1. Можуть застосовуватися й інші пари. У багатьох випадках зручніше працювати з величинами  $\pm 1$ . При цьому двійкові активаційні функції замінюються в такий спосіб:

$$f(x) = \begin{cases} 1, & \text{якщо } x > 0 \\ -1, & \text{якщо } x \leq 0. \end{cases}$$

В інтелектуалізованих НС (ІНС) набули застосування різні активаційні функції  $f(x)$ , зображені на рис. 2.18 [76]:



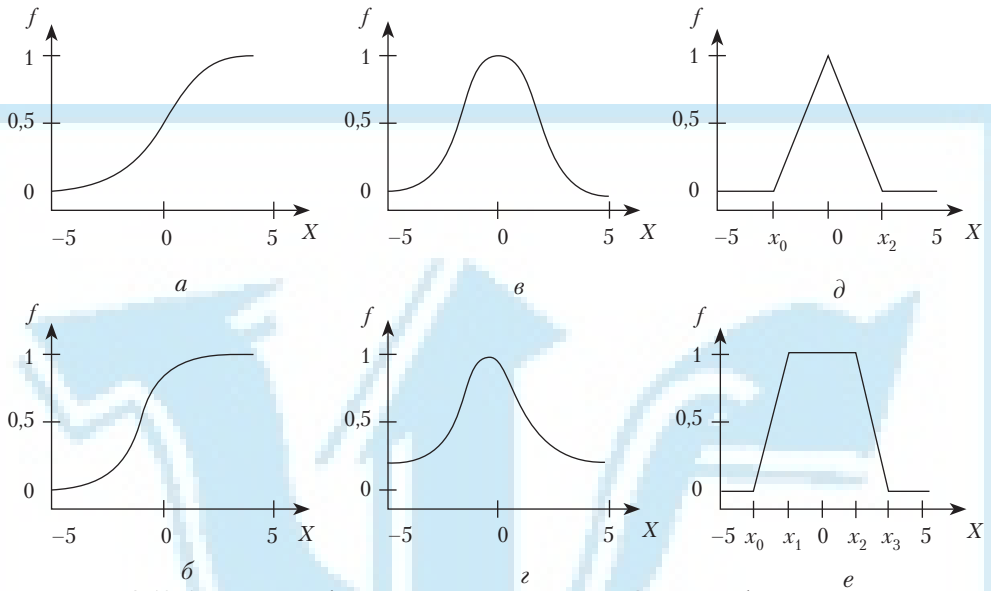


Рис. 2.18. Активаційні функції для різних типів ІНС і нейро-фаззі-систем

а) сигмоїдна функція:

$$f(x) = \frac{1}{1 + e^{-cx}}$$

Її застосування дає змогу уникнути проблем, пов'язаних з недиференційованістю активаційних функцій, і поліпшує властивості процедур навчання;

б) функція гіперболічного тангенса:

$$f(x) = \text{th}(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}};$$

в) гаусівська функція:

$$f(x) = e^{-\frac{(x-c)^2}{2\sigma^2}};$$

г) інверсна мультікватратична функція:

$$f(x) = \frac{1}{\sqrt{(x-c)^2 + d^2}};$$

д) трикутна функція (для нейро-фаззі-систем):

$$f(x) = \begin{cases} c_1(x - x_0) & \text{для } x \in (x_0, 0) \\ -c_2(x - x_2) & \text{для } x \in (0, x_2) \\ 0 & \text{в інших випадках;} \end{cases}$$

е) трапецієподібна функція (для нейро-фаззі-систем):

$$f(x) = \begin{cases} c_1(x - x_0) & \text{для } x \in (x_0, x_1) \\ 1 & \text{для } x \in (x_1, x_2) \\ -c_2(x - x_2) & \text{для } x \in (x_2, x_3) \\ 0 & \text{в інших випадках;} \end{cases}$$

**Узагальнені топології НС.** Як наголошувалося у підрозд. 1.8, вирізняють два типи нейроподібних сіток. У *першому* з них вузлами сітки є формальні елементи, що описують окремі нейрони. В сітках *другого* типу вузлами є формальні елементи, що відповідають не окремим нейронам, а особливим їх сукупностям – нейронним ансамблям.

**Означення 2.19.** Під *нейронним ансамблем* розуміють таку сукупність взаємопов'язаних нейронів, яка збуджується повністю при збудженні окремої її частини.

Нейронна сітка утворюється шляхом послідовного або паралельного сполучення ряду нейронів. Так, наприклад, на рис. 2.19, *a* зображено найпростішу нейронну сітку з двох нейронів. На вхід сітки подаються сигнали  $x_1, x_2$ , на виході формуються величини  $y_1, y_2$ . Ваги (значення синапсів) включені у зв'язуючі лінії. У разі відсутності Bias (зсуву) опитування при двійковій активаційній функції орієнтоване на  $\theta = 0$ . При розрахунку кожній парі входів  $x_1, x_2$  відповідає одна пара виходів  $y_1, y_2$ . Через те що всі величини двійкові (0 або 1), одержуємо логічну функцію, передану сіткою.

Вирази для розрахунку нейронної сітки (рис. 2.19, *a*) мають вигляд:

$$\left. \begin{aligned} y_1 &= f(\text{net}_1 - 0), \\ y_2 &= f(\text{net}_2 - 0), \\ \text{net}_1 &= -1x_1 + 2x_2, \\ \text{net}_2 &= 2x_1 - 1x_2 \end{aligned} \right\}$$

де  $f$  – двійкова функція.

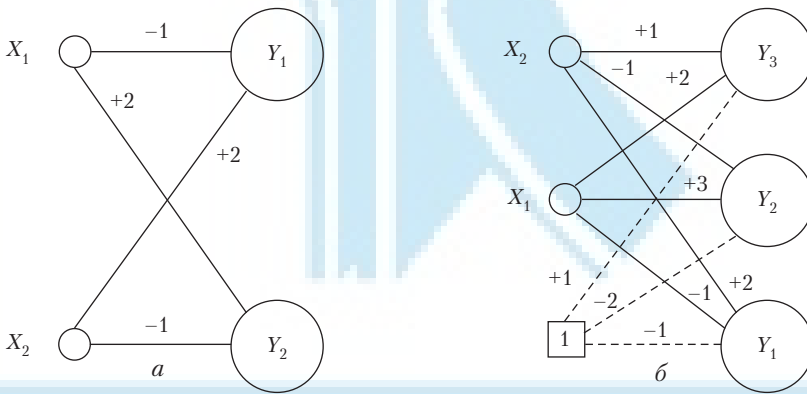


Рис. 2.19. Нейронні сітки: *a* – найпростіша сітка; *б* – сітка зі зсувом

Задаючи для входів  $x_1, x_2$  будь-які можливі величини, одержуємо відповідні до них вихідні величини  $y_1, y_2$  згідно з такою умовою:

$x_1$	$x_2$	$y_1$	$y_2$
0	0	0	0
1	0	0	1
0	1	1	0
1	1	1	1

Тоді розглянута сітка являє собою логічну двійкову вектор-функцію. Вводячи вектори

$$\bar{X} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \bar{Y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

і матрицю

$$\bar{M} = \begin{pmatrix} -1 & 2 \\ 2 & -1 \end{pmatrix},$$

одержуємо

$$\begin{aligned} \overline{net} &= \bar{M}\bar{X}, \\ \bar{y} &= \bar{f}(\overline{net} - \bar{0}), \end{aligned}$$

де  $\bar{0}$  – нуль-вектор,  $\overline{net}$  – вектор сумарного імпульсу та

$$\bar{f}(\dots) = \begin{pmatrix} f(net_1 - 0) \\ f(net_2 - 0) \end{pmatrix}.$$

У розглянутому прикладі всі граничні величини  $\theta = 0$ . У протилежному разі граничні величини відображаються через зсуви (*Bias*), як це показано на рис. 2.19, б. Цій нейронній сітці відповідає така процедура розрахунку:

$$\left. \begin{aligned} y_1 &= f(-1x_1 + 2x_2 - 1) \\ y_2 &= f(3x_1 - 1x_2 - 2) \\ y_3 &= f(2x_1 + 1x_2 + 1) \end{aligned} \right\}.$$

Відповідно у векторній формі

$$\left. \begin{aligned} \bar{y} &= f(\overline{net} - \theta) = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}; \\ \overline{net} &= \begin{pmatrix} -1 & 2 \\ 3 & -1 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; \\ \theta &= \begin{pmatrix} -1 \\ -2 \\ 1 \end{pmatrix} \end{aligned} \right\}.$$

Таким чином, вхідний вектор  $\bar{X}$  двовимірний, а вихідний вектор  $\bar{Y}$  так само, як і граничний вектор  $\theta$ , є тривимірним, і матриця має вигляд « $2 \times 3$ ». Легко побачити, що сіткою (рис. 2.19, б) задається така функція:

$x_1$	$x_2$	$y_1$	$y_2$	$y_3$
0	0	0	0	1
1	0	0	1	1
0	1	1	0	1
1	1	0	0	1

На основі цих прикладів можна пояснити поняття «нейронна сітка»: одношарова нейронна сітка задається за допомогою  $n \times m$  матриці  $\bar{M}$ , елементами якої є

дійсні числа, а також векторною активаційною функцією  $\bar{f}$  так, що кожному двійковому імпульс-вектору  $\bar{X}$  відповідає один вихідний вектор  $\bar{Y}$  і відповідна умова  $\bar{y} = f(\text{net} - \theta)$ , де  $\theta$  – вектор граничної величини,  $\text{net} = \bar{M}\bar{X}$ .

Отже, будь-яка нейронна сітка може бути подана графом, у якому кожна вхідна величина зв'язана з кожною вихідною. На лініях зв'язку вказуються відповідні ваги (елементи матриці). Граничні величини можуть при цьому вноситися в граф за допомогою зсувів. Необхідно зазначити, що нейрони лівого ряду (рис. 2.19) виконують функції розподілу вхідних сигналів. Вони не виконують якихось обчислень і тому не вважаються шаром. Структури (рис. 2.19) являють собою одношарові нейронні сітки.

**Прямоспрямовані нейронні сітки.** Вирізняють два основних класи сіток: сітки зі зворотним зв'язком, у яких вихідний сигнал подається назад на вхід сітки, і сітки, у яких для кожного вхідного вектора розраховується вихідний вектор, що зчитується з вихідних нейронів. Тут відсутні зворотні зв'язки й у цьому сенсі відбувається прямоспрямований потік інформації. Сітки з такою структурою називають також «*feed forward Networks*» або «*hetero assotiative Networks*». Нижче розглядатимуться насамперед сітки без зворотних зв'язків.

Розглянуті вище одношарові сітки можуть описуватися в матричній формі

$$\bar{y} = f(\bar{\text{net}} - \bar{\theta}),$$

де  $\bar{\text{net}} = \bar{M}\bar{X}$ ;  $\bar{X}$  і  $\bar{Y}$  – вхідний і вихідний вектори;  $f$  – активаційна функція.

У 1969 р. М. Мінський і С. Паперт [184; 185] показали, що є важливі логічні функції висловлювань, які не піддаються опису у формалізмах зазначених одношарових типів сіток. Однією з таких функцій є, наприклад, булева *XOR-функція* (*Exclusiven Order* – «виключає АБО»). Однак через 10 років було доведено, що положення Мінського і Паперта втрачає силу стосовно багатошарових сіток, коли один вихідний шар є вхідним для наступного нейронного шару. Якщо ці шари кілька разів включаються один за одним, виходить багатошаровий тип нейронної сітки (рис. 2.20, а, б) – *багатошаровий перцептрон MLP* (*Multilayer Perceptron Networks*). Ці різновиди сіток можуть являти собою усілякі функції, і для них з 1985 р. використовується прийнятний алгоритм навчання (*Backpropagation Algorithm*).

Розглянемо спочатку (рис. 2.21, а, б) топологію *тришарової прямоспрямованої нейронної сітки*. Тут (рис. 2.21, а) середній шар служить входом для наступного за ним шару чи нейронів, що у свою чергу являє собою вихід усієї нейронної сітки. Перший шар (вхідний), як уже було зазначено, використовується лише для розподілу вхідних сигналів. Він не виконує якихось обчислень, і його часто не вважають окремим шаром. У цій сітці всі граничні величини дорівнюють нулю (відсутній *Bias*). Вхідними величинами є  $x_1, x_2, x_3$ . На виході нижнього шару формуються величини  $k_1, k_2$  і ці величини потім як вхідні надходять до верхнього шару. Цей верхній шар формує остаточні вихідні величини сітки  $y_1, y_2, y_3$ . Проміжні нейрони запам'ятовують допоміжні величини  $k_1, k_2$ . Їх називають схованими нейронами (*hidden neurons*), а весь середній шар – схованим шаром (*hidden layer*).

З математичної точки зору маємо нелінійну комбінацію. Важливо, що для визначення завантаження схованих нейронів застосовується активаційна функція, тому що в протилежному разі весь опис звівся б до перемножування двох матриць, і тоді всю сітку можна було б звести до одношарової.

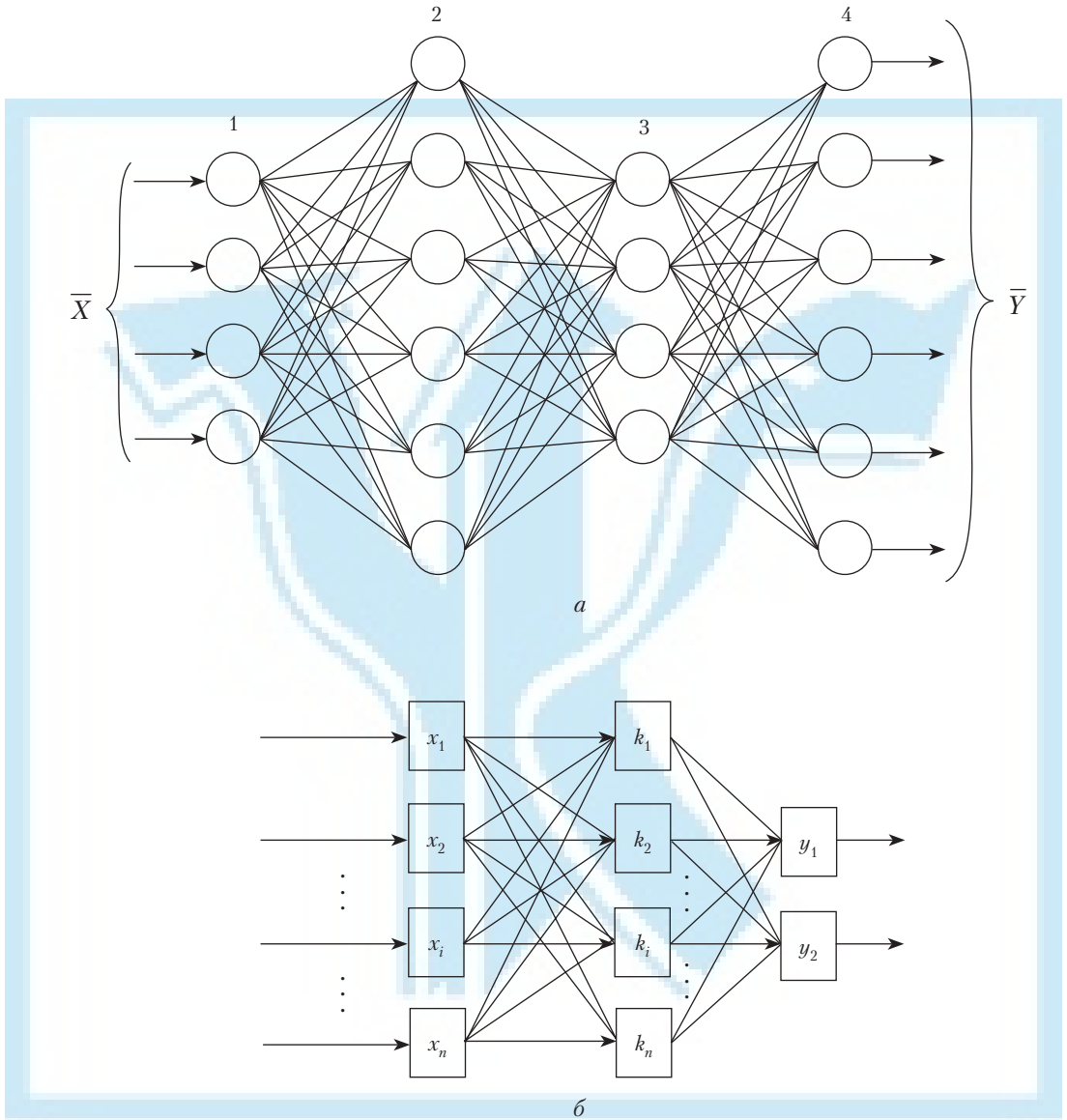


Рис. 2.20. Структура MLP (а) (де  $\bar{X}$  – вектор вхідних сигналів;  $\bar{Y}$  – вектор вихідних сигналів; 1 – вхідний шар нейронів – розподільників вхідних сигналів; 2, 3 – сховані (проміжні) шари ІНС; 4 – вихідний шар нейронів) і тришарового (б) перцептрона

Для багатошарової сітки застосовуємо такі ж формалізми, як і для одношарової. Так, для тришарової сітки, зображеної на рис. 2.21, а:

$$\bar{k} = \bar{f}(\overline{net}_1 - \bar{\theta}_1),$$

де  $\overline{net}_1 = \overline{W}_1 \bar{x}$ . Тут  $\bar{k}$ ,  $\bar{x}$  і  $\bar{\theta}_1$  – вектори відповідно схованого шару, вхідний, граничний;  $net_1$  розраховується за матрицею  $\overline{W}_1$ .

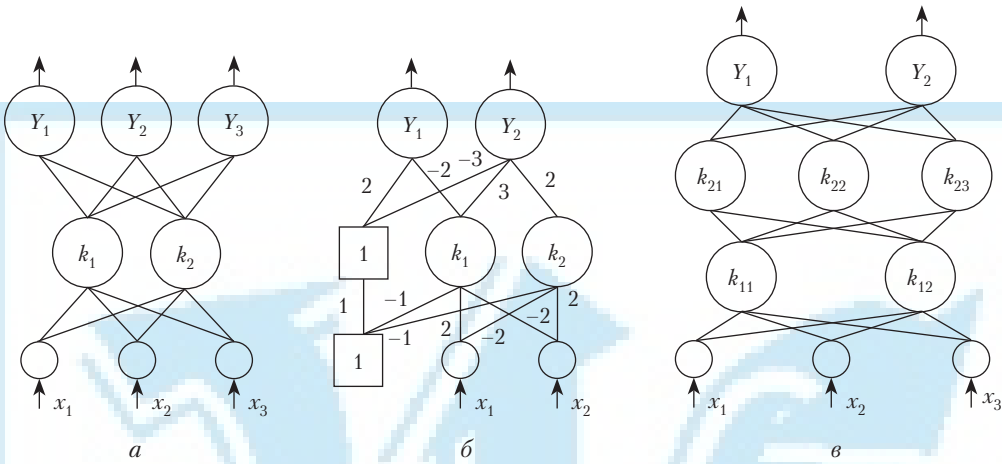


Рис. 2.21. Варіанти архітектури прямоспрямованих нейронних сіток: а – тришарова ІНС із двома схованими нейронами; б – тришарова ІНС зі зсувами у вхідному і схованому шарах; в – нейронна сітка з двома схованими шарами

Другий шар описується співвідношенням

$$\bar{Y} = f(\overline{net}_2 - \bar{\theta}_2),$$

де  $\overline{net}_2 = \overline{W}_2 \bar{k}$ ;  $\bar{Y}$  – вихідний вектор;  $\overline{W}_2$  – вагова матриця другого шару.

Для пояснення розглянемо приклад (рис. 2.21, б), де для кожного шару додатково до ваг введено по *Bias*'у (зсуву).

Розрахувавши величини схованого шару і вихідні значення, одержимо

$$\left. \begin{aligned} k_1 &= f(2x_1 - 2x_2 - 1) \\ k_2 &= f(-2x_1 + 2x_2 - 1) \\ y_1 &= f(-2k_1 + 0 \times k_2 + 2) \\ y_2 &= f(3k_1 + 2k_2 - 3) \end{aligned} \right\}$$

У векторній формі це можна подати так:

$$\left( \begin{array}{c} 1 \\ k_1 \\ k_2 \end{array} \right) = f \left[ \left( \begin{array}{ccc} 1 & 0 & 0 \\ -1 & 2 & -2 \\ -1 & -2 & 2 \end{array} \right) \cdot \left( \begin{array}{c} x_1 \\ x_2 \end{array} \right) \right];$$

$$\left( \begin{array}{c} y_1 \\ y_2 \end{array} \right) = f \left[ \left( \begin{array}{ccc} 2 & -2 & 0 \\ -3 & 3 & 2 \end{array} \right) \cdot \left( \begin{array}{c} 1 \\ k_1 \\ k_2 \end{array} \right) \right],$$

причому

$$\left. \begin{aligned} \bar{k} &= f(W_1 \bar{x}), \\ \bar{y} &= f(W_2 \bar{k}), \end{aligned} \right\} \quad (2.17)$$

де  $\bar{x}$  – розширений на число 1 вхідний вектор, а  $W_1$  і  $W_2$  – матриці.

Як бачимо, *Bias* може бути інтегрований простим розширенням вектора на координату 1, завдяки чому надалі розглядаються лише матриці, в яких не враховуються граничні вектори.

Простим перерахуванням можна визначити, що сітка (рис. 2.21, б) подає такі функції (сюди додано сховані нейрони  $k_1, k_2$ ):

$x_1$	$x_2$	$k_1$	$k_2$	$y_1$	$y_2$
0	0	0	0	1	0
0	1	0	1	1	0
1	0	1	0	0	0
1	1	0	0	1	0

У загальному вигляді дворівневу сітку можна подати у формі (2.17), причому тепер:  $x$  – вхідний вектор;  $k$  – вектор схованого шару;  $y$  – вектор вихідного сигналу. Можлива наявність *Bias*'у заздалегідь інтегрована у вхідний вектор, тобто передбачається, що нейрон входу  $x_1$  як *Bias* постійно встановлений на 1, і відповідні елементи матриць  $W_1, W_2$  мають ті самі пороги.

З наведених рівнянь одержуємо:

$$\bar{y} = f(W_2(f(W_1\bar{x}))).$$

За аналогією можна визначити тришарову сітку

$$\bar{y} = f(W_1f(W_2f(W_3\bar{x})))$$

або  $n$ -шарову сітку

$$\bar{y} = f(W_1f(W_2f(W_3\dots f(W_n\bar{x}))))).$$

На рис. 2.21, в показано топологію ІНС з двома схованими шарами, побудованої за описаним вище принципом.

*Нейросітки зі зворотними зв'язками.* На відміну від розглянутих прямоспрямованих сіток існують НС, в яких вихідний сигнал знову спрямовується на вхід сітки, через що виникає ітеративний процес – це так звані *сітки зі зворотним зв'язком (Feed-back Nets)*, або автоасоціативні НС. До таких структур належать *сітки Хопфілда* [169], які мають лише один шар нейронів, причому кожен нейрон пов'язаний з усіма іншими і сам із собою, в результаті чого спостерігаються численні зворотні зв'язки. До всіх нейронів підводяться вхідні сигнали, які можуть набувати значення +1 або -1. Оскільки, крім того, всі ваги є цілочисловими, подальші розрахунки легко виконуються. Основною сферою застосування сіток Хопфілда є розпізнавання образів. Процес розпізнавання полягає в тому, що коли задано сітку Хопфілда з вагами і пороговими значеннями, то вона перебуватиме в динамічній рівновазі за умови, якщо на її вхід подано образ, заданий певним стартовим вектором у процесі їх генерації. Ці стани рівноваги характеризуються тим, що загальна енергія (гамільтонова функція) системи є мінімальною. Перенесення цього положення на сітку Хопфілда приводить до «енергетичної функції», яка буде мінімальною при наданні сітці деякого образу. Оскільки при цьому запам'ятовуються всі двійкові величини, для яких енергетична функція мінімальна, то зафіксовані мінімуми є локальними.

Проте деякі застосування сіток потребують пошуку глобального мінімуму, тобто абсолютного мінімуму енергетичної функції. Такі застосування являють собою базу Больцман-машини, яку обробляють так звані сітки Больцмана [143].

До інших застосувань Хопфілд-ітерацій належать задачі оптимізації, в яких здійснюється ідентифікація енергетичної функції з деяким виразом. Цю процедуру можна розв'язати за допомогою сітки Хопфілда.

До прийнятної різновиду НС із зворотним зв'язком належить також так звана **карта Кохонена** [177], яка ще відома під назвою «самоорганізуючої карти» (*Self Organizing Ftafure MAP, або скорочено FM*). Найважливішою властивістю цього різновиду НС є те, що аналогічні вхідні сигнали збуджують сусідні нейрони в карті Кохонена, що може здійснюватися завдяки зворотним зв'язкам. Завдяки цьому групи подібних вхідних сигналів з багатовимірного вхідного простору можуть геометрично зображатися дискретною низьковимірною нейронною ґраткою, і тому цей різновид НС дістає різноманітні практичні застосування: для візуалізації даних, зниження різноманітності даних, контролю процесів тощо.

Існують й інші типи НС, серед яких для задач адаптивного розпізнавання образів певного поширення набули **когнітрон** та **неокогнітрон** — різновиди прямо-спрямованих багатопшарових сіток, причому кожний шар має два типи нейронів: збуджуючих (активізуючих) та гальмуючих [162]. Збуджуючі нейрони мають додатні, а гальмуючі — від'ємні ваги. Оскільки в неокогнітроні відсутні обмеження для використання тільки для обробки візуальних даних, то він може знайти широке застосування як узагальнена система розпізнавання образів за наявності значних обчислювальних ресурсів.

Для гібридних нейро-фаззі-систем застосовуються топології сіток **радіальних базисних функцій** (RBF-Nets) [154; 182].

**Способи навчання нейронних сіток.** Навчання здійснюється шляхом послідовного подання вхідних векторів з одночасним піднастроюванням ваг відповідно до певної процедури. В процесі навчання ваги сітки поступово змінюються так, щоб кожний вхідний вектор виробляв відповідний йому вихідний вектор.

Можливі, як мінімум, три підходи до навчання нейронних сіток [176]:

- **Навчання з вчителем.** На кожному кроці навчання вектор  $\bar{Y}$ , що видається сіткою, порівнюється з цільовим вектором  $\bar{Y}_c$ . Ваги змінюються за однією з формул, наприклад «Delta — Regal» — дельта-правилом, якщо  $\bar{Y} \neq \bar{Y}_c$ .
- **Навчання без вчителя.** Тут точний цільовий вектор не задається, а ваги на кожному кроці трохи змінюються у довільному напрямку. За допомогою оцінювальної функції визначається, поліпшується чи ні розрахований вихідний вектор при зміні ваг порівняно з попереднім значенням. Якщо поліпшення досягнуто, змінені ваги запам'ятовуються, а в протилежному разі вони «забуваються».
- **Навчання шляхом самоорганізації.** Ваги змінюються на кожному кроці навчання, і зміни залежать від: а) сусідніх вхідних образів; б) імовірного розподілу, за яким пропонуються до навчання припустимі вхідні образи.

**Правило Д. Хебба.** Як було показано, нейронна сітка являє собою двійкові функції, що відображають вхідні і вихідні вектори. Ці функції можуть бути математичними залежностями, способом дій, що керують інструкціями для роботів, тощо. Для побудови такої сітки необхідні вказівки з розрахунку вагових коефіцієнтів за правилами, яких для більшості застосувань немає. Тому необхідні спроби пошуку різних методик, які дали б змогу виконувати розрахунки ваг для сітки із заданими функціями.

Дональд Хебб [164] висловив гіпотезу про те, що мозок збільшує підсилювальні властивості синапсів тоді, коли нейрони перед і за синапсами одночасно активізо-



вані. У більш строгому формулюванні гіпотеза Хебба говорить: *властивість синапса (посилення чи уповільнення) змінюється пропорційно добутку перед- і післясинаптичної активності*. Якщо  $w_{ij}$  – вага в графічній сітці від входу  $x_i$  до виходу  $y_j$ , то алгоритм реалізації цього правила для НС у момент надходження вхідного сигналу можна подати математичною формою:

$$\Delta w_{ij} = \sigma x_i y_j \text{ (правило Хебба),} \quad (2.18)$$

де  $\sigma > 0$  – постійна;  $x_i$  – вхід;  $y_j$  – вихід.

В алгоритмах навчання правило Хебба використовується з незначними змінами. Замість виходу  $y_j$  приймають  $\Delta y_j$ , при цьому величина  $\Delta y_j$  дорівнює різниці між бажаним і фактичним виходами, що відповідає моменту навчання. Тоді одержують так зване *дельта-правило*:

$$\Delta w_{ij} = \sigma x_i \Delta y_j. \quad (2.19)$$

Як бачимо, подальша корекція ваги не буде потрібна, якщо отриманий вихід узгоджується з бажаним.

В разі необхідності практичного застосування НС у загальному випадку потрібно виконати такі заходи:

- вибрати для сітки *необхідну топологію*;
- задати представницькі *логічні функції*, а саме: вхідні вектори  $x_1, x_2, \dots, x_n$  і відповідні їм вихідні вектори  $y_1, y_2, \dots, y_n$
- визначити *значення ваг*, при яких сітка реалізовуватиме задані функції. Для визначення ваг застосувати процедуру навчання сітки;
- після навчання сітка підготовлена до роботи і багаторазового використання.

Таким чином, розробка і застосування НС містить дві важливі фази: *навчання (Learning Phase)* та *експлуатації (Recall-Phase)*.

Відповідно до досвіду фаза навчання в плані обчислень дуже трудомістка, але її здійснюють лише один раз, а фаза експлуатації за витратами на обчислення безпроблемна і може як завгодно багато повторюватися.

Визначення ваг здійснюється зазвичай у такому порядку:

*Крок 1.* Для усіх ваг задаються випадкові числа.

*Крок 2.* Вибирається випадковий вхідний вектор  $x_i$ .

*Крок 3.* Розраховують з вагами на поточний момент вихідний вектор  $y_j$ .

*Крок 4.* Порівнюють  $y_j$  з цільовим вектором  $y_{jц}$ . Якщо  $y_j = y_{jц}$ , то повторюють процедуру кроку 2. У протилежному разі коригують ваги за однією з прийнятних формул корекції і далі діють за кроком 2.

Як формули корекції на кроці 4 можна застосувати, наприклад, вираз (2.18) або дельта-правило (2.19).

Наведене правило навчання називають *«навчанням з вчителем»*. Вказівки вчителя полягають у повідомленні на кожному кроці системі точного вихідного вектора – навчання йде під наглядом учителя, якому знайомий зміст досліджуваного предмета, і він включає його в процедуру навчання.

Інший метод навчання полягає в змінюванні ваг без знання цільової величини, і якщо при навчанні ми наближаємося до мети, воно заохочується, у протилежному разі – штрафується. При комп'ютерній процедурі оцінюю функцією зважується кожний акт навчання. У разі успіху ваги змінюються, у протилежному разі залишаються попередніми. Цей метод дістав назву *«навчання без учителя» (Unsupervised Learning)*.

Нарешті, існують також процедури навчання, в яких ваги самоорганізуються в аспекті задачі навчання.

**Навчання перцептрона.** Алгоритм навчання перцептрона містить три кроки:

*Крок 1.* Для ваг  $w_{ij}$  вибираються випадкові числа.

*Крок 2.* Задається випадковий вхідний вектор  $\bar{X} = (x_1, x_2, \dots, x_n)$  і приймаються значення  $\bar{X}_i = 1$  або 0.

*Крок 3.* Змінюють ваги за співвідношенням

$$w_{ij}^n = w_{ij}^{n-1} + \Delta w_{ij},$$

де  $w_{ij}^{n-1}, w_{ij}^n$  — попередня і нова ваги;  $\Delta w_{ij} = \beta x_i \varepsilon_j$ ;  $\varepsilon_j = y_j - f(\sum_i w_{ij} x_i)$ , тобто  $\varepsilon_j = y_j - f(\text{net}_j)$  — різниця між цільовим і фактичним виходом у точці  $j$ ;  $x_i, y_j$  — вхідний і вихідний сигнали сітки;  $\beta > 0$  — мале число.

*Крок 4.* Повернення до кроку 2.

Робота алгоритму завершується в момент, коли сітка для всіх вхідних векторів почне видавати правильні цільові вектори. Тут, звичайно, йдеться про варіант «навчання з учителем».

З теореми збіжності для перцептрона випливає, що будь-яка класифікаційна задача, для якої при навчанні існує розв'язок, буде в остаточному підсумку розв'язана за декілька кроків.

*Збіжність процесу навчання перцептрона.* Якщо функцію можна подати перцептроном, то алгоритм навчання сходиться за скінченну кількість кроків. У теорії нейронних сіток доведення цього виконується за глобальними функціями, які поєднують у собі інформацію про всю сітку. Ці функції можна порівняти з енергетичними функціями фізики. Так само, як там фізична система зазнає змін доти, поки загальна енергія буде мінімізована, так і в нейронній сітці алгоритм навчання змінює ваги доти, поки буде виконана задача навчання і «енергетична функція» буде мінімізована.

Для перцептрона ця функція, що зазнає мінімізації, вкрай проста, а саме є сумою квадратів помилки:

$$\Phi = 0,5(\sum_j \varepsilon_j^2), \quad (2.20)$$

де  $\varepsilon_j$  — сумарна помилка вихідних величин. Можна показати, що функція  $\Phi$  зменшується з кожним кроком навчання і в остаточному підсумку сходиться до 0.

**Навчання штучної нейросітки методом зворотної передачі помилки (Back-propagation Method).** Опис методу градієнтного спуску [165] можна інтерпретувати таким прикладом для тришарової НС (рис. 2.20, б).

Нижче наведено опис методу для тришарової сітки (рис. 2.20), але формально наведені міркування можна легко поширити на  $n$ -шарові сітки. У розглянутій сітці маємо: вхідний вектор  $\bar{x}$ , вектор  $\bar{k}$  схованого шару і вихідний вектор  $\bar{y}$ . Для цих векторів вище були отримані співвідношення

$$\bar{y} = \bar{f}(W_1 \bar{k}); \quad (2.21)$$

$$\bar{k} = \bar{f}(W_2 \bar{x}), \quad (2.22)$$

де  $W_1, W_2$  — матриці, які ми будемо диференціювати.

Для диференціювання як активаційну вибираємо сигмоїдну функцію

$$f(x) = \frac{1}{(1 + \exp(-cx))}, \quad (2.23)$$

де  $c > 0$ .

Формально при виведенні процедури навчання поводяться як для перцептрона: ваги вибираються так, щоб функція помилки була мінімальною. Її подають у вигляді

$$E = (1/2) \sum_1 (z_1 - y_1), \quad (2.24)$$

де  $z_1$  — задані цільові величини досліджуваної функції;  $y_1$  — фактичні значення вихідного сигналу сітки, які у цілому забезпечують суму всіх квадратів помилки. Чим менша величина  $E$ , тим краща сітка, а при  $E = 0$  (абсолютний мінімум) сітка працює відмінно.

Зміни ваг у процесі навчання вибираються так, щоб  $E$  на кожному кроці зменшувалася, тобто процес повинен виходити на одержання мінімуму. Математично це формулюється в такий спосіб. Нехай  $w_{ij}^1$  — елементи матриці  $W_1$  у (2.21) і  $w_{ij}^2$  — елементи матриці  $W_2$  у (2.22). Тоді кожен крок навчання полягає в тому, щоб ці елементи матриць (ваги) змінювалися на прирощення

$$\Delta w_{ij}^1 = \beta \varepsilon_i k_j, \quad (2.25)$$

$$\Delta w_{ij}^2 = \beta \sum_m \overline{x_m} w_{mi}^1 \overline{x_j}. \quad (2.26)$$

Тут  $x_m$  — вхідні величини;  $k_j$  — сховані величини;  $\varepsilon_i = z_i - y_i$  — помилка (різниця між цільовою і фактичною величинами);  $\beta$  — коефіцієнт навчання ( $\beta > 0$  — вибирається довільно). У результаті функція помилки  $E$  поступово зменшується, кроки навчання змінюють сітку в напрямку мінімуму (локального). Розширена формула корекції відповідає формулі корекції для одношарового перцептрона.

Для доведення використовується метод градієнтного спуску, при якому функцію помилки  $E$  диференціюють і показують, що при прийнятих змінах ваг величина  $\Delta E < 0$  і, отже,  $E$  зменшується.

Реалізація алгоритму навчання Backpropagation (методу зворотної передачі помилки) виконується так:

*Крок 1.* Задання усіх ваг випадковими числами.

*Крок 2.* Задання випадкових образів входу/виходу для досліджуваної функції, розрахунок зайнятості  $k_j$  схованих шарів.

*Крок 3.* Для заданих вхідних  $x_i$  і цільових  $z_i$  величин виконано коригування ваг за формулами (2.25). Далі — повернення до кроку 2.

Цей метод має ту перевагу, що для нього існує математична формалізація, яку можна застосовувати для будь-якої сітки, і не потрібно враховувати особливостей досліджуваних функцій. На жаль, кількість кроків навчання найчастіше буває значною, через що фаза навчання в обчислювальному аспекті є трудомісткою.

Хоча для двошарових сіток алгоритм зворотної передачі дає змогу подавати будь-які можливі функції, він у ряді випадків дає неправильні розв'язки. Істотний недолік цього методу в тому, що функція помилки підлягає мінімізації. При цьому, як при будь-якій мінімакській задачі, можна попадати на локальний мінімум, тоді як для навчальної функції потрібен абсолютний мінімум. У випадках локального мінімуму одержувані величини часто близькі до шуканого значення, але все ж таки відрізняються від точного значення.

Іншою проблемою є вибір кількості схованих вузлів сітки. При виборі малої кількості можлива ситуація, коли навчальну функцію неможливо подати через недостатню потужність сітки. При збільшенні кількості нейронів у схованому шарі збільшується кількість незалежних змінних функції помилок, а це призводить до збільшення кількості побічних мінімумів, причому програма приводить до найближчого з них. Поряд з цим наростання кількості схованих точок збільшує час розрахунків у фазі навчання, який і так досить великий.

Для більшості зазначених ускладнень немає теоретичних вказівок, які могли б допомогти запобігти розвитку помилкового ходу процесу навчання. Залишається лише експериментальний контроль, тобто виконання ряду пробних тестів. Для прискорення збіжності для корекції ваг  $w_{ij}$  користуються формулою

$$w_{ij}^n = \mu w_{ij}^{n-1} + (1 - \mu)\Delta w_{ij}, \quad (2.27)$$

де  $w_{ij}^{n-1}$ ,  $w_{ij}^n$  — попередня і нова ваги;  $\mu$  — додатне число, яке на початку процесу навчання близьке до нуля, потім у ході навчання збільшується і наприкінці наближається до одиниці;  $\Delta w_{ij}$  — корекція ваги.

Очевидно, що спочатку коефіцієнти корекції ваги чинять сильний вплив, який поступово послаблюється з поліпшенням значень ваг. Існує строге доведення збіжності методу зворотної передачі помилки [176].

Ефективного зниження кількості кроків навчання можна досягти, якщо вхідні і вихідні вектори підібрати так, щоб подібні вхідні вектори відтворювали також подібні вихідні вектори. Відображення з такими особливостями у певному сенсі є неперервним.

Насамкінець зазначимо, що існує багато прикладів застосування сіток, у яких вхідними або вихідними величинами служать не (0,1) біт-последовності, а реальні числа  $x$  у діапазоні  $0 \leq x \leq 1$ .

**Сфера застосування НС.** Результати досліджень у галузі моделювання нейронних сіток значно розширили клас задач, що розв'язуються за допомогою нейроподібних сіток.

Тепер до цього класу належать комбінаторні, оптимізаційні та інші задачі. Завдяки успіхам мікроелектроніки підготовлено технологічну базу для створення обчислювальних пристроїв, які здатні здійснювати паралельну обробку інформації. Ці два фактори зумовили появу нейрокомп'ютерів — ЕОМ, архітектура яких найкраще пристосована до розв'язання задач моделювання нейронних сіток. За нашого часу нейрокомп'ютери створюються у вигляді компактних приставок до персональних ЕОМ, суттєво підвищуючи їх функціональні можливості.

## 2.7. Сітки Петрі як формальний апарат подання і відображення процесів складних систем

**Особливості застосування апарату сіток Петрі (СП).** Основні особливості СП полягають у можливості відображувати *паралелізм, асинхронність, ієрархічність* складних об'єктів простіше та наочніше, аніж у разі використання інших формальних засобів подання знань про об'єкт дослідження. Це й зумовило широке використання апарату СП, зокрема для розробки алгоритмів керування, через те, що сама процедура зміни станів СП однозначно відображає послідовність і логіку станів досліджуваного об'єкта [54; 55; 83].

В СП як базову інформацію використовують дані про логічні взаємозв'язки подій. Причинно-наслідковий зв'язок подій в СП задається множиною відношень виду «умови — події». Побудова моделей об'єкта управління (ОУ) у вигляді СП пов'язана з такими діями:

- процеси та явища, які подаються та які відбуваються в ОУ, описують множиною подій та умов, якими ці події визначають, а також причинно-наслідковими відношеннями;

- визначають події, послідовність появи яких керується станами об'єкта. Ці стани задаються множиною умов, які формулюються у вигляді предикатів. Кількісно умови характеризуються ємністю, вираженою числами натурального ряду;
- умови (предикати) можуть виконуватися і не виконуватися. Тільки виконання умов забезпечує можливість появи подій. Умови, при виконанні яких можлива реалізація подій, називаються *передумовами*;
- після того, як подія настає, забезпечується виконання інших умов, які перебувають у причинно-наслідковому зв'язку з передумовами. Ці умови називаються *постумовами*.

Як правило, в СП умови — це *позиції*, а події — *переходи* (можливий і зворотний принцип ідентифікації умов і подій, що обумовлюється при формуванні формальної системи). Послідовності подій відображаються спрацьовуваннями переходів. Виконання деякої умови пов'язано з появою маркера (мітки) у відповідній цій умові позиції. Угода про правила спрацьовування переходів є способом вираження концепції причинно-наслідкових зв'язків між умовами та подіями в ОУ і водночас — формуванням механізму *логічного виведення* у формальній моделі подання знань на СП.

Проте порівняно з іншими сітковими моделями подання знань (таких як семантичні сітки, нейросітки, автоматні системи взаємодіючих графів — АСВГ [113] тощо) апарат СП дає ряд важливих додаткових можливостей: синтезувати алгоритми керування, аналізувати останні на коректність і здійснювати їх перетворення й розв'язувати ряд інших задач на основі розширень і модифікацій класичного апарату СП, які посилюють його виразні спроможності. Крім того, апарат СП дає змогу враховувати специфічні властивості ГКІС як об'єкта подання знань та обмеження, які на нього накладаються (зокрема, ємності нагромаджувачів і складів, багатомономенклатурність одночасно оброблюваних заготівок та виробів тощо). При цьому необхідно встановити вимоги до моделюючих і розв'язувальних потужностей СП для формального подання процесів у складних системах. Адже системи складаються з окремих взаємодіючих компонент, кожна з яких має свій *стан* (абстракцію відповідної інформації, необхідної для опису її дії), який з часом може змінюватися. Діям компонент притаманні сумісність і паралелізм, тобто дії однієї компоненти системи можуть відбуватися одночасно з діями інших компонент. Проте оскільки компоненти взаємодіють, необхідно встановити *синхронізацію* їх дії, а це може призвести до того, що одна компонента чекатиме на іншу. Саме ця властивість є характерною для великих систем, що містять значну кількість компонент, кожна з яких, у свою чергу, також може складатися з ряду компонент (тобто мати ієрархічний і системний характер побудови). Власне, для подання знань про процеси, що відбуваються в таких (зокрема, обчислювальних) системах, і був запропонований апарат СП.

### Загальні поняття про СП

**Означення 2.20.** Формально *сітка Петрі* — це набір виду:

$$N = (P, T, F, H, \mu_0), \quad (2.28)$$

де  $P$  — скінченна непорожня множина позицій (станів або місць);  $T$  — скінченна непорожня множина переходів (подій);  $F: P \times T \rightarrow \{0, 1, 2, \dots\}$  — вхідна функція (вхідний потік — відображення переходів у позиції);  $H: T \times P \rightarrow \{0, 1, 2, \dots\}$  — функція на виході (потік на виході — відображення з позицій у перехід);  $\mu_0: P \times K$  — початкове маркування (розмітка) сітки, яке задає початковий розподіл міток за позиціями сітки ( $K$  — множина невід'ємних цілих чисел).

Графічно структура СП зображається орієнтованим біграфом, множину вершин  $V$  якого розбивають на дві неперетинні множини (долі)  $T$  і  $P$  такі, що  $V = T \cup P$ ,  $T \cap P = 0$  (рис. 2.22, а).

Вершини  $p_i \in P$  (позиції зображаються колами) і  $t_j \in T$  (переходи зображуються вертикальною рискою або вузьким вертикальним прямокутником) з'єднані між собою орієнтованими ребрами (дугами) так, що разом з вершинами  $T$  і  $P$  ребра на вході та на виході забезпечують усім простим циклам біграфа парну довжину. Отже, дуги (зв'язки) графа можуть бути спрямовані тільки від позицій до переходів або тільки від переходів до позицій. Інші комбінації у графі не допускаються. Таким чином, функції входу та виходу задають множину одинарних дуг (ребер)  $(t_j, p_i)$  і  $(p_i, t_j)$ , що зв'язують вершини біграфа.

Наприклад, дуги, які передують позиції  $p_i$ , містяться у множині  $H(p_i) = \{(t_j, p_i) \mid H(t_j, p_i) = 1\}$ , а дуги, які передують переходові  $t_j$ , – у множині  $H(t_j) = \{(p_i, t_j) \mid H(p_i, t_j) = 1\}$  (тут значення  $H(t_j, p_i) = 1$  вказує на наявність ребра, яке з'єднує перехід  $t_j$  і позицію  $t_j$ ), і аналогічно значення  $H(p_i, t_j) = 1$  вказує на дугу, спрямовану від позиції  $p_i$  до переходу  $t_j$ ). Дуги, які прямують з  $p_i$  і  $t_j$ , можуть бути подані відповідно множинами:

$$F(p_i) = \{(p_i, t_j) \mid F(p_i, t_j) = 1\}; \quad F(t_j) = \{(t_j, p_i) \mid F(t_j, p_i) = 1\}.$$

Вхідний потік  $PRE(t_j)$  переходу  $t_j$  об'єднує множину позицій, що передують йому; аналогічно для вхідного потоку  $PRE(p_i)$  позиції  $p_i$  об'єднується множина переходів, що передують їй:

$$PRE(t_j) = \{p_i \in P \mid F(p_i, t_j) = 1\}; \quad PRE(p_i) = \{t_j \in T \mid H(t_j, p_i) = 1\}. \quad (2.29)$$

За аналогією можна записати витічні потоки  $j$ -го переходу  $POST(t_j)$  та  $i$ -ї позиції  $POST(p_i)$ , відповідно:

$$POST(t_j) = \{p_i \in P \mid H(t_j, p_i) = 1\}; \quad POST(p_i) = \{t_j \in T \mid F(p_i, t_j) = 1\}. \quad (2.30)$$

Усі перелічені множини можна знайти з відображень  $H$  і  $F$ .

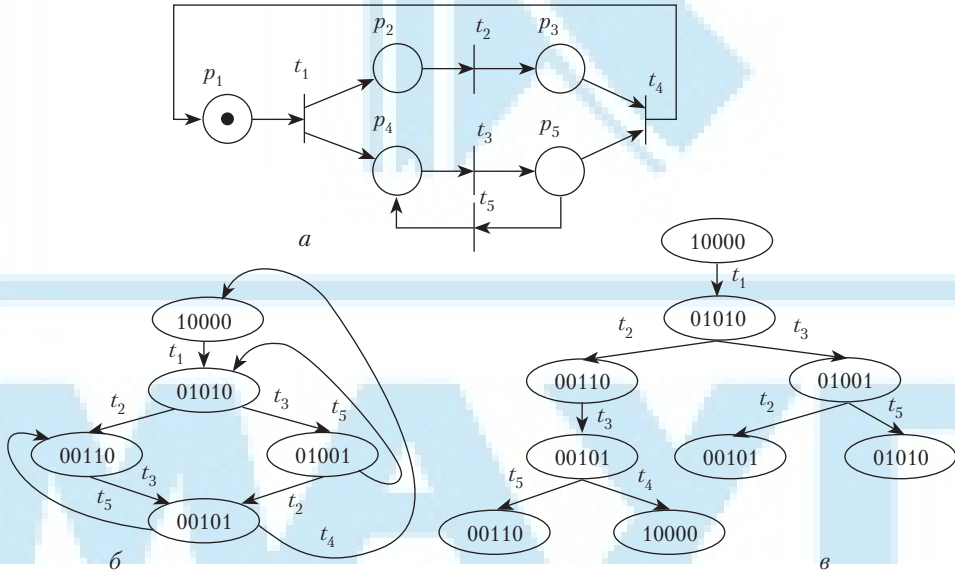


Рис. 2.22. Схематичне подання сіток Петрі: а – структура СП; б – граф досяжності; в – дерево досяжності

При маркуванні сітки всім позиціям СП приписують деякі натуральні числа, і маркування сітки подають вектором:

$$\mu = \begin{pmatrix} \mu(p_1) \\ \mu(p_2) \\ \dots \\ \mu(p_n) \end{pmatrix},$$

де  $\mu(p_i)$  — кількість міток у позиції  $p_i$ ,  $i = \overline{1, n}$ .

Перехід від одного маркування до іншого здійснюється через спрацьовування (або збудження) переходів. При цьому граф досяжності (рис. 2.22, б) відображує можливість переходу маркера з будь-якої позиції до іншої, а дерево досяжності (рис. 2.22, в) відображує шляхи таких переходів.

Формальна умова спрацьовування (збудження) переходу

$$\mu(p_i) - F(p_i, t_j) \geq 0, \forall p_i \in P \quad (2.31)$$

порівнює кількість міток у  $p_i$  з кількістю дуг між  $p_i$  та  $t_j$ . Детальніше про рух міток в СП та особливості її функціонування, а також про об'єктно-орієнтовані на процеси в ГКІС розширення (модифікації) сіток Петрі йдеться в розділах 7 та 8 книги. Тут тільки визначимо відповідність апарату СП умовам формальної системи (див. підрозд. 1.7), а саме:

- **як базовий алфавіт** в СП виступають *переходи, позиції, ребра* (спрямовані дуги), *маркери* (мітки — визначають стан, або завантаженість позицій; для кольорових розфарбованих СП — певні умовні пофарбування міток). Спрацьовування переходів (формальна умова (2.31)) свідчить про настання ситуацій в СП, за яких можлива (і відбувається) зміна її станів. Останнє характеризується графічно змінами в маркуванні позицій, які передують  $PRE(t_j)$  та йдуть після  $POST(t_j)$  збудженого переходу  $t_j$ , а математично — у відповідності умовам (2.29) та (2.30);
- правила з'єднання ребрами (спрямованими дугами) вершин — позицій та переходів (*по-перше*), правила коректного заміщення переходів і позицій при розгортанні і згортанні, тобто абстрагуванні та уточненні СП (*по-друге*), а також відображення навантажених станів СП (*по-третьє*) є **механізмом створення складних утворень** з елементів базового алфавіту СП, тобто формальним апаратом подання знань при відображенні станів процесів у складних системах;
- початкове маркування сітки відповідає наявності **аксіоматичного твердження** — третьої умови визначуваності СП як формальної системи;
- нарешті, можливість ініціювання зміни маркувань в СП в результаті спрацьовування переходів при чергуванні умов і подій (тобто станів об'єкта керування/моделювання), а також множина досяжності СП (тобто множина всіх доступних маркувань) формують **механізм логічного виведення** за деревом досяжності сітки (детальніше див. розділи 7 та 8).

## 2.8. Апарат процесних алгебр як основа формального визначення поведінки системи у семантичному, модульному та ієрархічному поданні

**Визначення понять процесної алгебри (ПА).** Для опису формального апарату ПА визначимо основні його поняття [113].

**Означення 2.21.** *Процесна алгебра* — формальна мова для специфікації та обґрунтування поведінки процесів при їх взаємодії.

Існує багато різноманітних видів процесних алгебр. Процесна алгебра оперує поняттями дій. За допомогою операторів ПА будуються вирази процесів, які й визначають модель системи.

Серед найбільш відомих прикладів відзначимо *CSP (Communicating Sequential Processes)* [128]. Пізніше *CSP* було використано як основу для мов паралельного програмування та специфікації. Серед найбільш поширених результатів є мова програмування *Oscam* [155].

Ще одним прикладом є числення Мілнера — *CCS (Calculus of Concurrent Systems)* [183], яке має близьку до *CSP* сферу застосування.

Як *CSP*, так і *CCS* в їхніх звичайних формах підтверджують, що процеси синхронізації є однаковими при взаємодії, але мають тонкі семантичні відмінності [153].

**Означення 2.22.** *Подія* є основною дискретою — станом системи у момент часу. Послідовність або набір таких подій описує поведінку ГКІС, що є основним завданням моделювання.

**Означення 2.23.** Множина імен подій, відібраних для конкретного опису об'єкта, називається його *алфавітом* або *алфавітом формальної СПЗ на основі процесних алгебр* (*перша умова* відповідності апарату ПА формальній СПЗ). При виборі алфавіту не обов'язково розрізняти події, що ініціюються самим об'єктом або деяким зовнішнім щодо нього фактором. Імена подій позначатимемо малими літерами  $a, b, c, d, e, \dots$ , а імена процесів — великими літерами  $A, B, C, D, E, \dots$ .

**Означення 2.24.** *Процесом* називається чергування (послідовність) станів (подій) системи, виражене за допомогою *операторів ПА*, які і є *механізмом створення складних утворень в СПЗ на основі апарату ПА* в алгебраїчній (наприклад, див. табл. 2.3) або графічній (наприклад, див. рис. 2.24–2.27, де спрямовані та поіменовані дуги позначаються малими латинськими літерами  $a, b, c, \dots$  і означають дії, а вершини — великими латинськими літерами  $A, B, C, \dots$  і називаються *факторами* або *агентами*) формах (*друга умова* відповідності апарату ПА формальній СПЗ). Отже, поняття «процес» означає поведінку об'єкта, яка описується в термінах обмеженого набору подій. Вважається, що конкретна подія в «житті» об'єкта відбувається миттєво і не має тривалості в часі. Тривалу дію розглядатимемо як пару подій, де перша означає початок дії, а друга — кінець. Дві тривалих дії можуть перекриватися в часі, якщо початок кожної з них передує завершенню другої.

**Означення 2.25.** Процес (фактор) з алфавітом  $A$ , такий, що в ньому не відбувається жодної події з  $A$ , називатиметься *stop A*. Цей процес описує поведінку пасивного об'єкта: маючи фізичну здатність брати участь у подіях з  $A$ , він ніколи її не використовує.

**Означення 2.26.** *Протоколом поведінки* процесу називається скінченна послідовність символів, що фіксує події, в яких процес брав участь до деякого моменту часу. Протокол позначається послідовністю символів, які розділені комами та охоплені кутувими лапками:



$\langle x, y \rangle$  (протокол для двох подій:  $x$  та наступної  $y$ );

$\langle x \rangle$  для протоколу з однією подією  $x$ ;

$\langle \rangle$  для протоколу з порожньою послідовністю, що не містить подій.

**Означення 2.27.** Опис ймовірної поведінки досліджуваного об'єкта є його *специфікацією*, яка являє собою предикат з вільними змінними, кожна з яких відповідає деякому спостережуваному аспекту поведінки об'єкта. Наприклад, специфікація електронного підсилювача з вхідним діапазоном в 1 В та з коефіцієнтом підсилення 30 задається предикатом:  $U30 = (0 \leq u \leq 1 \Rightarrow |u' - 30 \times u| \leq 1)$ . У наведеній специфікації  $u$  означає вхідну, а  $u'$  — вихідну напругу.

Якщо  $P$  — об'єкт, що відповідає специфікації  $S$ , то  $P$  *satisfy*  $S$  (тобто задовольняє  $S$ ). Це означає, що  $S$  описує всі можливі результати спостереження за поведінкою  $P$ :

$$\forall p.p \in \text{protocols}(P) \Rightarrow S.$$

**Твердження 2.3.** Якщо із специфікації  $S$  логічно випливає інша специфікація  $T$ , то будь-яке спостереження, що описується  $S$ , описується також і  $T$ . Отже, кожний об'єкт, що задовольняє  $S$ , у цьому разі повинен також задовольняти і більш слабку специфікацію  $T$ :

$$\text{Якщо } P \text{ satisfy } S \text{ та } S \Rightarrow T, \text{ то } P \text{ satisfy } T.$$

Протокол поведінки процесу (**Означення 2.26**) і специфікація об'єкта (**Означення 2.27**), а також **твердження 2.3** виконують роль **аксіоматичного твердження в СПЗ на основі апарату ПА (третя умова** відповідності апарату ПА формальній СПЗ).

**Означення 2.28.** Процесом взаємодії називається такий процес, який об'єднує два процеси для спільного виконання. Взаємодії, які полягають у передачі повідомлень і описуються парою  $s, v$ , де  $s$  — ім'я каналу, по якому відбувається взаємодія, а  $v$  — значення повідомлення, яке передається, є спеціальним класом подій.

**Означення 2.29.** Ситуація, за якої процеси не можуть дійти узгодженості щодо того, яка дія буде наступною, і при цьому жодні дії не можуть відбуватися у подальшому, називається *дедлоком* або *тутиковою ситуацією*. Іноді ж процес має деякий спектр можливої поведінки, причому його оточення не має можливості ані впливати на вибір між різними альтернативами, ані навіть спостерігати його. Іншими словами, вибір здійснюється довільним і недетермінованим способом.

**Означення 2.30.** У загальному випадку нехай  $X$  — множина подій, які пропонуються на першому кроці оточенням процесу  $P$ , їх алфавіти збігаються з алфавітом процесу  $P$ . Якщо ж при розміщенні  $P$  в це оточення існує ризик дедлоку на першому кроці, то  $X$  — це *множина відмов процесу  $P$* .

**Означення 2.31.** Процес називається *детермінованим*, якщо він ніколи «не відмовляється» від події, в якій може брати участь. Іншими словами, множина є відмовою детермінованого процесу, тільки якщо ця множина не містить подій, в яких процес може брати участь на першому кроці.

Отже, якщо можливе здійснення більш як однієї події, вибір між ними визначається зовнішнім щодо процесу середовищем, тобто оточення процесу може насправді здійснювати вибір або вибір стає відомим оточенню в той самий момент, коли він відбувається.

**Означення 2.32.** *Недетермінованим* є процес, який хоча й міг би взяти участь у певній події, але в результаті деякого внутрішнього недетермінованого вибору може відмовитися від виконання цієї події, навіть якщо оточення не перешкоджає

виконанню події. При цьому вибір здійснюється довільним і недетермінованим способом, і висновок про те, який саме вибір був зроблений, можна зробити тільки пізніше на основі аналізу подальшої поведінки процесу.

**Означення 2.33.** Канали використовуються для передачі повідомлень тільки в одному напрямку і тільки між двома процесами. Канал, що використовується процесом тільки для виведення повідомлень, будемо називати *вихідним каналом* цього процесу, а канал для введення повідомлень — *вхідним каналом*.

**Означення 2.34.** У ГКІС процес, що має в алфавіті лише вхідний та вихідний канали, називатимемо *транспорттером*.

**Означення 2.35.** Процес, який виводить з можливою деякою затримкою введени в нього послідовність інформаційних або матеріальних об'єктів, називається *буфером* і є завжди непорожнім і готовим до виведення об'єктів. Буфери зберігають об'єкти, які очікують на обробку, і є корисними для специфікації поведінки протоколів зв'язків, чие призначення полягає у передачі об'єктів у тій послідовності, в якій вони надходять до буфера. Такі протоколи складаються з двох процесів — передавача  $T$  та приймача  $R$ , які поєднані в ланцюжок ( $T \gg R$ ). Якщо протокол правильний, то ( $T \gg R$ ) є буфер.

**Твердження 2.4.** Якщо  $Q$  виступає як основний чи головний процес, то в комбінації ( $P \parallel Q$ ) процес  $P$  є підпорядкованим процесом.

**Означення 2.36.** Катастрофічним перериванням процесу  $P$  є переривання, яке викликано не самим  $P$ .

**Означення 2.37.** Контрольною точкою процесу є такий його стан, за яким катастрофа потребує *перезапуску* процесу шляхом повернення до деякого попереднього стану системи, про який відомо, що він є задовільним. Оскільки може статися, що однієї контрольної точки може виявитися недостатньо, важливим завданням є вибір і збереження в системі набору контрольних точок.

**Базові оператори ПА.** Оператори ПА змінюють стан процесу і поєднують процеси один з одним. Основні оператори ПА наведено в табл. 2.3.

Таблиця 2.3

Оператори процесної алгебри

Назва оператора	Умовне позначення оператора
пасивність	<i>stop</i>
дія-префікс	$a; B$ чи $a \rightarrow B$ чи $a.B$
вибір	$B + C$ чи $\Sigma B_i$ чи $B C$
композиція	$B \parallel_A C$ чи $B/[A] C$
сховування	$B/A$ чи <i>hide A in B</i>
визначення	$p := B$
застосування	$p$

**Оператор пасивності.** Оператор пасивності — *stop* процесу  $A$  визначає процес з алфавітом  $A$  такий, що в ньому не відбувається жодної події з  $A$ , і позначається *stop A*. Цей оператор описує поведінку «пошкодженого» об'єкта: маючи фізичну здатність брати участь у подіях  $A$ , він ніколи її не використовує.

**Оператор дія-префікс.** Оператор дія-префікс визначає об'єкт, який спочатку бере участь у деякій події  $x$ , а потім поводить себе так само, як процес  $P$ . При цьому

$(x \rightarrow P)$  читається як « $P$  за  $x$ ». Взагалі процес  $(x \rightarrow P)$  має за визначенням той самий алфавіт, що й  $P$ , тому це позначення можна використовувати тільки за умови, що  $x$  належить тому самому алфавіту. Більш формально:  $a(x \rightarrow P) = aP$ , якщо  $x \in aP$ .

**Оператори вибору.** *Оператори вибору послідовності* описують об'єкт, який спочатку бере участь в одній з подій. Наприклад, якщо  $b$  і  $c$  – різні події, то  $(b \rightarrow B \mid c \rightarrow C)$  або  $(b.B + c.C)$ . Подальша поведінка об'єкта описується процесом  $C$ , якщо першою відбулася подія  $c$ . Оскільки  $b$  і  $c$  – різні події, то вибір між  $B$  та  $C$  визначається тим, яка з подій насправді настала першою. Вертикальну риску  $\mid$  слід читати як «чи»: « $B$  за  $b$  чи  $C$  за  $c$ ».

Загалом оператор вибору послідовності  $(x : B \rightarrow P(x))$  використовується для визначення процесу, який володіє цілим спектром можливої поведінки. Зокрема, оператор  $\parallel$  дозволяє деякому іншому процесу робити вибір між альтернативами з множини  $B$  – це *оператор детермінованого вибору*.

*Оператор недетермінованого вибору* описує процес, який здійснюється довільно без відома чи контролю з боку зовнішнього оточення. Так, якщо  $P$  та  $Q$  – процеси, то оператор  $P \sqcap Q$ , або  $P$  чи  $Q$  визначає процес, який поводить себе або як  $P$ , або як  $Q$  (алфавіти операндів збігаються:  $a(P \sqcap Q) = aP = aQ$ ). Отже, описаний за допомогою оператора  $P \sqcap Q$  процес може бути реалізований або як  $P$ , або як  $Q$ .

Оскільки оточення процесу  $(P \sqcap Q)$  не може не тільки впливати на вибір процесу, але навіть не знає (у разі недетермінованого вибору) результату вибору, в ПА вводиться операція (з відповідним оператором) *генерального вибору*  $(P \square Q)$ , за яким оточення вже може керувати вибором між  $P$  та  $Q$  за умови, що цей вибір буде зроблено на першому кроці. Це означає, що якщо перша дія неможлива для  $P$ , то вибирається  $Q$ ; і навпаки, якщо  $Q$  не може виконати першу дію, то вибирається  $P$ . Але якщо перша дія можлива як для  $P$ , так і для  $Q$ , то вибір між ними залишається недетермінованим і маємо:  $a(P \square Q) = aP = aQ$ .

Випадок, коли операція генерального вибору аналогічна операції вибору:

$$(c \rightarrow P \square d \rightarrow Q) = (c \rightarrow P \mid d \rightarrow Q), \text{ якщо } c \neq d.$$

Випадок для однакових початкових умов:

$$(c \rightarrow P \square d \rightarrow Q) = (c \rightarrow P \sqcap d \rightarrow Q), \text{ якщо } c \neq d.$$

Відмінність між процесами  $(P \sqcap Q)$  та  $(P \square Q)$  є тонкою. Ці процеси можна помістити в таке середовище, в якому процес  $(P \sqcap Q)$  на першому кроці ввійде у тупиковий стан (дедлок), а процес  $(P \square Q)$  – ні. Нехай, наприклад,  $x \neq y$ , причому

$$P = (x \rightarrow P), Q = (y \rightarrow Q), aP = aQ = \{x, y\}.$$

Тоді процес  $(P \square Q) \parallel P + (x \rightarrow P) = P$ , а процес

$$(P \square Q) \parallel P = (P \parallel P) \sqcap (Q \parallel P) = P \sqcap \text{stop}.$$

Отже, в оточенні  $P$  процес  $(P \sqcap Q)$  може прийти в тупиковий стан, а процес  $(P \square Q)$  – ні.

Інколи умови функціонування ГКІС потребують об'єднання процесів з однаковими алфавітами для паралельного виконання, в ході якого не відбувається безпосередньої їх взаємодії чи синхронізації. Тоді кожна дія системи – це дія в точності одного з процесів.

Існує ще один оператор вибору – це *оператор чергування*. Справді, якщо один процес  $P$  не може виконати дію, це повинен зробити інший процес  $Q$  (генеральний вибір). Якщо ж одну дію готові виконати обидва процеси  $P$  та  $Q$ , то вибір між ними

є недетермінованим і для упорядкування недетермінованості вводиться оператор чергування  $\parallel$ , тобто якщо  $P$  чергується з  $Q$ , то  $P \parallel Q$ .

**Оператори взаємодії (композиції).** Ці оператори призначені для об'єднання і подання спільного виконання двох процесів. Взаємодії процесів можна розглядати як події, які потребують одночасної участі обох процесів. Нехай алфавіти цих процесів збігаються. Якщо  $P$  та  $Q$  — процеси з однаковими алфавітами, то позначимо через  $P \parallel Q$  процес, який веде себе як система, складена з процесів  $P$  і  $Q$ , взаємодія між якими покороково синхронізована. Тобто для таких процесів вводиться *оператор паралельної композиції*, який має позначення  $\parallel$ . Внутрішні події, що відповідають внутрішнім переходам процесів, які відбуваються одночасно, і взаємодії та зв'язкам між компонентами, що працюють паралельно, супроводжуються діями, які мають відбуватися автоматично в той момент, коли це можливо, і при цьому залишаються недосяжними для контролю та спостереження з боку оточення процесів. Слід також зазначити, що якщо в комбінації  $(P \parallel Q)$  процес  $Q$  виступає як основний, то, відповідно до **твердження 2.4**, процес  $P$  буде підпорядкованим. Якщо відношення основного та підпорядкованого процесів необхідно приховати від їх спільного оточення, застосовується позначення:  $P \parallel Q = (P \parallel Q) \setminus aP$ . Крім того, підпорядкованість може бути *вкладеною*, наприклад  $(n : (m : P \parallel Q) \parallel R)$ . Тоді всі входження подій з ім'ям  $m$  ховаються до того, як ім'я  $n$  приписується подіям, що залишилися, кожна з яких входить до алфавіту  $Q$  і не входить до алфавіту  $P$ ; процес  $R$  при цьому не має можливості ані безпосередньо взаємодіяти з  $P$ , ані знати про існування  $P$  та його імені  $m$ .

**Оператор послідовної композиції  $(P; Q)$ .** При проектуванні часто складна задача розподіляється на дві підзадачі, одна з яких успішно завершується до початку іншої. Така складна задача являє собою процес, який веде себе спочатку як  $P$ , а після завершення  $P$  продовжує вести себе як  $Q$ . Якщо ж успішного завершення  $P$  не відбувається, то не завершується й  $(P; Q)$ . В такій послідовній композиції виконання наступного процесу залежить від успішного виконання попереднього процесу.

**Оператор переривання** позначається як  $(P \wedge Q)$  і застосовується для подання композиції процесів з перериванням, якщо послідовна композиція  $(P; Q)$  не залежить від успішного завершення  $P$ , і при першій же з подій процесу  $Q$  виконання  $P$  переривається і вже не поновлюється.

**Оператор катастрофічного переривання** використовується тоді, коли переривання  $P$  викликане не самим процесом  $P$  (так званий вихід процесу у нештатну ситуацію), і тоді процес, який веде себе до катастрофи як  $P$ , а після неї як  $Q$ , подається так:  $P \wedge \perp Q = P \wedge (\perp \rightarrow Q)$ , причому тут  $Q$  може відігравати роль процесу, який ліквідує наслідки катастрофи.

**Оператор перезавпуску первинного процесу** використовується (відповідно до *означень 2.36–2.37*), якщо визначити  $P \wedge$  як процес, який веде себе як  $P$  до настання переривання  $\perp$ , а після кожної події  $\perp$  знов веде себе як  $P$ , але з самого початку (інакше кажучи, цей процес є *таким, що перезавпускається*), і визначається рекурсією:

$$aP \wedge = aP \cup \{\perp\}, \quad P \wedge = \mu X.(P \wedge \perp X) = P \wedge \perp (P \wedge \perp \dots),$$

тобто  $P \wedge$  — циклічний процес, навіть якщо  $P$  таким не є.

**Зв'язок подання об'єктів опису за допомогою ПА з алгоритмічним поданням.** Однією зі сфер застосування ПА стали CSP-подібні мови паралельного програмування [155]. Так, операторами-прототипами основних алгоритмічних структур стали: *присвоєння, умова, цикл*.

**Присвоєння.** Нехай  $x$  – програмна змінна,  $e$  – вираз, а  $P$  – процес. Тоді  $(x := e; P)$  – це процес, який веде себе як  $P$  з початковим значенням  $x$ , яке дорівнює початковому значенню виразу  $e$ .

**Умова.** Нехай вираз  $b$  обчислює істинність логічної функції (значення  $TRUE$  або  $FALSE$ ). Якщо  $P$  та  $Q$  – процеси, то  $P \Leftarrow b \Rightarrow Q$  – це процес, який веде себе як  $P$ , якщо початкове значення  $\in TRUE$ , або як  $Q$ , якщо початкове значення  $\in FALSE$ , тобто

*if b then P else Q.*

**Цикл.** Цикл позначається через  $b * Q$ , а традиційний цикл:

*while b do Q.*

**Операції над протоколами.** Четвертою складовою відмінності ПА, яка дає підстави говорити про належність апарату ПА до СПЗ, є **наявність правил виведення (перетворень) протоколів (специфікацій) (четверта умова** відповідності апарату ПА формальній СПЗ). Найважливішими операціями над протоколами, яким належить основна роль у фіксації, описі та розумінні поведінки процесів, є **конкатенація, звуження, а також вибірка.**

**Означення 2.38.** *Конкатенація* – це операція побудови нового протоколу з пари операндів – протоколів  $s$  та  $t$  шляхом з'єднання їх у зазначеному порядку як  $s \wedge t$ .

Властивості конкатенації:

- *асоціативність*, а також те, що порожній протокол  $\langle \rangle$  служить для неї одиницею, наприклад:  $s \wedge \langle \rangle = \langle \rangle \wedge s = s$ , а також  $s \wedge (t \wedge u) = (s \wedge t) \wedge u$ ;
- *строгість*, коли, наприклад, функція  $f$  відображає протокол у порожній протокол:  $f(\langle \rangle) = \langle \rangle$ ;
- *дистрибутивність*, якщо:  $f(s \wedge t) = f(s) \wedge f(t)$ .

**Твердження 2.5.** *Всі дистрибутивні функції є строгими;*

- *індукційність за  $n$* , де  $n$  – натуральне число. Наприклад,  $t^n$  є конкатенацією  $n$  копій протоколу  $t$ . Таким чином, спостерігається:  $t^0 = \langle \rangle$ , а також  $t^{n+1} = t \wedge t^n$ .

Іншою важливою операцією над протоколами є їх *звуження*.

**Означення 2.39.** *Звуження протоколу* – це операція  $(t \uparrow A)$  відкидання з протоколу  $t$  символів, які не належать до  $A$  (тобто звуження на множину символів  $A$ ).

**Твердження 2.6.** *Всі звуження є дистрибутивними:*

$$\langle \rangle \uparrow A = \langle \rangle, \text{ а також } (s \wedge t) \uparrow A = (s \uparrow A) \wedge (t \uparrow A).$$

Для одноелементних послідовностей ефект звуження:

$$\langle x \rangle \uparrow A = \langle x \rangle, x \in A; \langle y \rangle \uparrow A = \langle \rangle, y \notin A.$$

**Твердження 2.7.** *Від протоколу, звуженого на порожню множину, нічого не залишається:*

$$s \uparrow \{\} = \langle \rangle,$$

а послідовне звуження на дві множини рівнозначне одному звуженню на перетин цих множин:

$$(s \uparrow A) \uparrow B = s \uparrow (A \cap B).$$

**Означення 2.40.** *Головою протоколу* є перший елемент  $s_0$  непорожньої послідовності  $s$ .

**Означення 2.41.** *Залишок протоколу* – це результат, що отримується при видаленні  $s_0$  з  $s$  і позначається через  $s'$ :

$$\langle x, y, x \rangle_0 = x, \text{ причому } \langle x, y, x \rangle' = \langle y, x \rangle.$$

**Означення 2.42.** Множина скінченних протоколів  $A^*$  — це набір всіх скінченних протоколів (включно  $\langle \rangle$ ), складених з елементів множини  $A$ . Після звуження на  $A$  такі протоколи залишаються незмінними, звідки:  $A^* = \{s \mid s \uparrow A = s\}$ .

**Означення 2.43.** Довжина протоколу  $\#t$  — це кількість подій, що входять до протоколу. Наприклад,  $\# \langle x, y, x \rangle = 3$ .

До початку процесу  $P$  невідомо, який саме з можливих протоколів буде запитаний, бо його вибір залежить від зовнішніх щодо процесу факторів. Проте повний набір всіх можливих протоколів процесу  $P$  може бути відомий раніше, тому вводиться функція  $protocols(P)$  для позначення цієї множини. Протокол процесу ( $c \rightarrow P$ ) може бути порожнім через те, що  $\langle \rangle$  є протоколом поведінки будь-якого процесу до моменту початку його першої події. Тому кожний непорожній протокол цього процесу починається з  $c$ , а його хвіст повинен бути можливим протоколом  $P$ .

$$\begin{aligned} Protocols(c \rightarrow P) &= \{t \mid t = \langle \rangle \vee (t_0 = c \ \& \ t' \in protocols(P))\}; \\ &= \{\langle \rangle \cup \{\langle c \rangle \wedge t \mid t \in protocols(P)\}\}. \end{aligned}$$

Існує тісний зв'язок між протоколами процесу та зображенням поведінки останнього у вигляді дерева.

**Твердження 2.8.** Для кожної вершини дерева протоколом процесу до моменту досягнення цієї вершини буде послідовність міток на шляху з кореня дерева до цієї вершини.

**Твердження 2.9.** Протоколи процесу являють собою множину всіх шляхів, які ведуть з кореня до різних вершин дерева.

**Твердження 2.10.** Через те що всі дуги, які ведуть з кожної вершини, помічені різними подіями, кожний протокол процесу однозначно визначає шлях з кореня дерева до деякої з вершин.

**Твердження 2.11.** Якщо  $s \in protocols(P)$ , то  $P/s$  ( $P$  після  $s$ ) — це процес, який веде себе так, як веде себе  $P$  з моменту завершення усіх дій, що записані у протоколі  $s$ . Якщо ж  $s$  не є протоколом  $P$ , то  $(P/s)$  не визначене.

Нарешті, ще однією важливою операцією над протоколами є вибірка.

**Твердження 2.12.** Якщо  $s$  — послідовність пар, то  $s \downarrow x$  визначає результат вибірки з  $s$  усіх пар, першим елементом яких є  $x$ , і заміни такої пари на її другий елемент, причому перший і другий елементи пари розділяються крапкою: нехай  $s = \langle a.7, b.9, a.8, c.0 \rangle$ , тоді  $s \downarrow a = \langle 7, 8 \rangle$ , проте  $s \downarrow d = \langle \rangle$ . Якщо ж  $s$  не є послідовністю пар, то  $s \downarrow a$  позначає кількість входження  $a$  в  $s$ .

Якщо  $t$  — протокол, який відображає послідовність подій з того моменту, коли деяка послідовність  $s$  успішно завершилася, то їх композиція позначається як  $(s \checkmark; t)$ , де символ  $\checkmark$  означає успішне завершення послідовності  $s$ . В разі відсутності в  $s$  такого символу протокол  $t$  в композиції  $(s; t)$  не може розпочатися, тобто  $(s; t) = s$ , якщо  $\neg(\langle \checkmark \rangle \text{ в } s)$ .

У разі присутності символу  $\checkmark$  наприкінці  $s$  він видаляється, а  $t$  приєднується до результату, наприклад:  $(s \wedge \langle \checkmark \rangle); t = s \wedge t$ , якщо  $(\langle \checkmark \rangle$  після  $s)$ .

Протоколи процесів відіграють основну роль у специфікаціях систем.

**Взаємодія процесів.** Взаємодія процесів — це спеціальний клас подій. Взаємодія полягає в передачі повідомлень і власне є подією, яка описується парою  $c.v$ , де  $c$  — це ім'я каналу, по якому відбувається взаємодія, а  $v$  — значення повідомлення, яке передається.

**Процеси введення та виведення.** Процес  $P$  може здійснювати взаємодію по каналу  $c$  через множину повідомлень:  $ac(P) = \{v \mid c.v \in aP\}$ ; при цьому функції, що вибирають ім'я каналу  $c$  та значення повідомлення  $v$ , визначаються, відповідно, як:

$$\begin{aligned} canal(c.v) &= c. \\ inform(c.v) &= v. \end{aligned}$$

Нехай повідомлення  $v$  є елементом множини повідомлень  $caP$ . Тоді процес, який спочатку виводить  $v$  по каналу  $c$ , а потім веде себе як  $P$ , позначається так:

$$(c!v \rightarrow P) = (c.v \rightarrow P).$$

З іншого боку, процес, який у початковому стані готовий ввести будь-яке значення  $x$ , яке передається через канал  $c$ , а потім веде себе як  $P(x)$ , позначається так:

$$(c?v \rightarrow P(x)) = (y : \{y \mid canal(y) = c\} \rightarrow P(inform(y))).$$

**Означення 2.44.** Відтворюваною подією процесу введення-виведення є взаємодія  $c.v$ , де  $v$  — значення, яке визначається процесом виведення.

Отже, якщо дотримуватися правил сполучення процесів  $P$  (той, що виводить повідомлення) і  $Q$  (той, що вводить водночас це повідомлення), то взаємодія їх по каналу  $c$  визначається як  $ac(P) = ac(Q)$  із збігом алфавітів на обох кінцях каналу  $c$  і відбувається за такими законами:

1.  $(c!v \rightarrow P) \parallel (c?v \rightarrow Q(x)) = c!v \rightarrow (P \parallel Q(v))$ ;
2.  $(c!v \rightarrow P) \parallel (c?v \rightarrow Q(x)) \setminus \tilde{N} = (P \parallel Q(v)) \setminus \tilde{N}$ , де  $\tilde{N} = \{c.v \mid v \in ac\}$ .

**Процеси-транспортери.** Такі процеси відіграють важливу роль при композиційному поданні транспортних операцій.

**Означення 2.45.** Процес, що має в алфавіті лише два канали — вхідний та вихідний, називається *транспортером*. Результатом послідовного поєднання транспортерів  $P, Q, R$  тощо є  $P \gg Q \gg R$  і так далі.

**Означення 2.46.** Специфікація транспортера — відношення  $S(in, out)$  між послідовністю повідомлень (об'єктів, процесів) з вхідного  $in$  та вихідного  $out$  каналів.

Оскільки при поєднанні двох транспортерів у ланцюжок спільна послідовність ховається, то для запобігання ризику замкнення виводиться правило:

- |      |   |
|------|---|
| ЯКЩО | $P \text{ satisfy } S(in, out)$ , а $Q \text{ satisfy } T(in, out)$ , |
| ТА   | $P$ попередній зліва, а $Q$ попередній справа,                        |
| ТО   | $(P \gg Q) \text{ satisfy } \exists s. S(in, s) \ \& \ T(s, out)$ .   |

**Процес-буфер.** Такий вид процесів є дуже поширеним у ГКІС і, як правило, коригує дисбаланс продуктивностей послідовних технологічних процесів у системі.

**Означення 2.47.** Буфер — це процес, який виводить, після деякої затримки, таку послідовність, яка вводиться. Буфер є непорожнім і завжди готовий до виведення.

Наступні процеси є буферними:  $BUF, (COP \gg COP)$ .

Буфери використовуються або для зберігання матеріальних об'єктів (у тому числі інформаційних та енергетичних), або для специфікації поведінки протоколу зв'язків (забезпечуючи передавання об'єктів у тій послідовності, в якій вони надходять). В останньому випадку протокол складається з двох процесів — завантажувача  $T$  та приймача  $R$ , поєднаних у ланцюжок ( $T \gg R$ ). Оскільки на практиці приймач та завантажувач поєднуються доволі довгим з'єднувачем, що може призвести до пошкодження і втрати повідомлень, які надсилаються, то з'єднувач моделюється окремим процесом  $CONNECT$ , тобто:

$$(T \gg CONNECT \gg R).$$

**Приклад 2.15.** Простий однопозиційний буфер (рис. 2.23) подається як  $Buf := in; out; Buf$ .

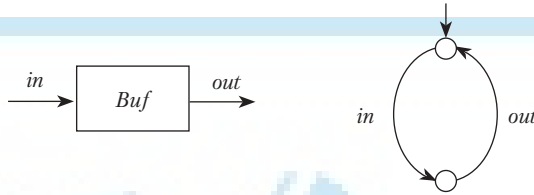


Рис. 2.23. Схема та граф однопозиційного буфера

**Приклад 2.16.** Складний однопозиційний буфер, утворений послідовним з'єднанням двох простих однопозиційних буферів (рис. 2.24).

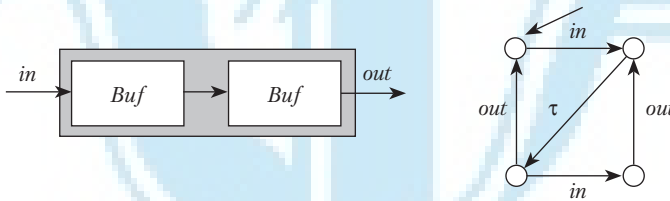


Рис. 2.24. Схема та граф складного однопозиційного буфера

**Приклад 2.17.** Двопозиційний буфер (рис. 2.25) подається як  
 $buf\ 2 := in; Half$   
 $Half := in; Half + out; Buf\ 2$   
 $Full := out; Half$ .

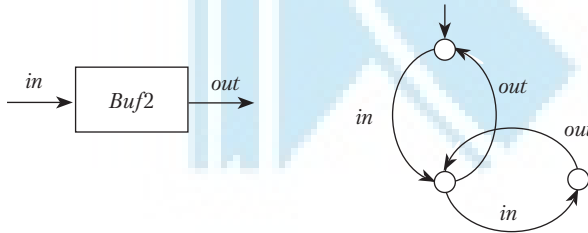


Рис. 2.25. Схема та граф двопозиційного буфера

**Алгебраїчні закони та їх узагальнення.** Тотожність процесів з однаковими алфавітами можна визначати за допомогою алгебраїчних законів, скориставшись поняттям *еквівалента*. Еквіваленти вивчають різноманітні подання понять і відрізняються властивостями. Прикладами еквівалентів можуть бути: еквіваленти шляху, тестування тощо. Застосування еквівалентностей підпорядковується закономірностям, що зумовило появу наведених нижче алгебраїчних законів:

- *Рівнозначність:*  $A + stop = A$ ;  $A + A = A$ ;
- *Комутативність:*  $B + C = C + B$  та  $B \parallel_A C = C \parallel_A B$ ;
- *Асоціативність:*  $(B + C) + D = B + (C + D)$  та  $B \parallel_A (C \parallel_A D) = (B \parallel_A C) \parallel_A D$ .

Для ПА існує своє тлумачення алгебраїчних законів.



### 1. Стосовно оператора вибору.

**Означення 2.48.** Два процеси, визначені за допомогою оператора вибору, є різними, якщо на першому кроці вони пропонують різні альтернативи або після однакового першого кроку поведуться по-різному, і є тотожними, якщо множини початкового вибору виявляються рівними і для кожної початкової альтернативи подальша поведінка процесів збігається, тобто:

$$(x : A \rightarrow P(x)) = (y : B \rightarrow Q(y)) \equiv (A = B \ \& \ \forall x \in A. P(x) = Q(x)).$$

### 2. Як керують поведінкою процесів з паралельною композицією ( $P \parallel Q$ ):

а) закон відображення логічної симетрії «процес – оточення»:

$$P \parallel Q = Q \parallel P;$$

б) закон відображення спільності роботи трьох і більше паралельних процесів:

$$P \parallel (Q \parallel R) = (P \parallel Q) \parallel R,$$

тобто при спільній роботі декількох процесів не має значення, в якому порядку вони об'єднані оператором паралельної композиції;

в) закон виникнення тупикової ситуації системи:

$$P \parallel \text{stop } aP = \text{stop } aP.$$

Якщо процес перебуває в тупиковій ситуації, то це призводить до дедлоку всієї системи;

г) закон одночасності виникнення однакових дій парою процесів:

$$(c \rightarrow P) \parallel (c \rightarrow Q) = (c \rightarrow (P \parallel Q)); \quad (c \rightarrow P) \parallel (d \rightarrow Q) = \text{stop}, \quad \text{якщо } c \neq d.$$

При комбінації процесів залишаються можливими ті події, які є у множинах вибору обох процесів:

$$(x : A \rightarrow P(x)) \parallel (y : B \rightarrow Q(y)) = (z : (A \cap B) \rightarrow (P(z) \parallel Q(z))).$$

Цей закон узагальнюється на випадки, коли в одного чи обох процесів є вибір початкової події. Саме цей закон дає змогу описати систему, визначену у термінах паралельного виконання без використання паралелізму.

### 3. Послідовної композиції:

$$(P; Q); R = R; (Q; R);$$

$$(x : B \rightarrow P(x)); Q = (x : B \rightarrow (P(x); Q));$$

$$(a \rightarrow P); Q = a \rightarrow (P; Q);$$

$$\text{Stop}; Q = \text{stop}.$$

**Реалізація процесів.** Для охарактеризування загальних умов реалізації процесів скористаємось таким твердженням:

**Твердження 2.13.** Будь-який процес  $P$ , який записаний за допомогою введених у цій роботі позначень, можна подати у вигляді:  $x : B \rightarrow F(x)$ , де  $F$  – функція, що ставить у відповідність множині символів множини процесів.

Власне, кожний процес можна розглянути як функцію  $F$  з областю визначення  $B$ , яка виділяє множини подій, що є початковими подіями процесу. Якщо першою відбулася подія  $x$  з  $B$ , то  $F(x)$  визначає подальшу поведінку процесу. При цьому множина  $B$  може бути порожньою (випадок *stop*), мати один (випадок *префікса*) або більше як один елемент (випадок *вибору*).

**Сфера застосування ПА.** Отже, в сучасному стані ПА – це формалізм для визначення поведінки системи у семантичному, модульному та ієрархічному поданні [93; 152; 180]. За допомогою операторів ПА будуються вирази процесів, дій, що за-

безпечують композиційність системи. Застосування операторів ПА спрощує внутрішню будову системи через внутрішні перетворення та безпосереднє подання системи за рахунок того, що оператори, в свою чергу, можуть складатися з дрібніших операторів.

Найбільш поширеною сферою застосування ПА є специфікація, а також розробка та реалізація систем, які неперервно діють та взаємодіють зі своїм оточенням. При цьому основна ідея полягає в тому, що ці системи можна розкласти на паралельно працюючі підсистеми, які взаємодіють як одна з одною, так і зі своїм спільним оточенням. Такий підхід є надійною основою для уникнення тупикових ситуацій та зациклювань [128].

Моделі у ПА можна подати у вигляді орієнтованого графа, дуги якого позначаються латинськими літерами і називаються *діями*, а вершини позначаються великими літерами і називаються *факторами* (агентами). Приклади моделей, зображені на рис. 2.26, подано множиною дій  $\{a, b, c, d, e\} \cup \{t\}$  і факторів  $E, E', C1, C2, F, F1, F2, G1, G2$ .

Апарат ПА може допомогти відділити поняття сутності інтерактивно-керованої системи від особливості будь-якої певної взаємодії. Застосовуючи ПА, інколи можна сконструювати формальні доведення властивостей поданих систем [158]. У випадках, де шаблон взаємодій між програмними компонентами стає складним (комплексним), ПА використовуються в аналізі або для однакового визначення специфікації на верхньому рівні абстракції.

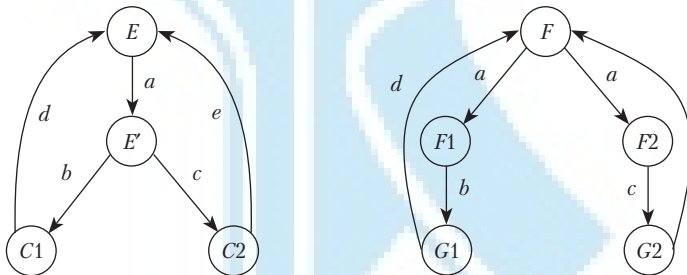


Рис. 2.26. Дві відмінні моделі з однаковим набором подання

### Подання взаємодії компонентів складних технічних систем на основі ПА.

Розглянемо ПА, яка пропонується для розв'язування задачі моделювання взаємодії компонентів складних систем. При цьому система розглядається як сукупність дій її компонентів, що взаємодіють один з одним. Відповідно, кожний компонент системи моделюється послідовним процесом. Взаємодія компонент у системі моделюється так: по фіксованих комунікаційних каналах приймаються і відправляються дії компонентів. Моделювання при поданні складних систем на основі цього підходу є більш легким, зрозумілим і зручним, ніж моделі, що використовують інші підходи.

Розглянемо деяку абстрактну систему, що складається з двох типових компонентів, які описуються моделями (рис. 2.27).

Кожну модель компонента також можна визначити у вигляді системи рівнянь. Так, для першої моделі:

$$P = a?.P + d!.P';$$

$$P' = d!.P' + a?P + d!.P',$$

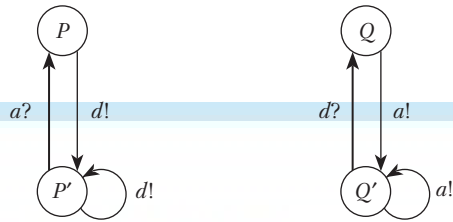


Рис 2.27. Моделі компонентів

і для другої моделі:

$$Q = a!.Q' + d?.Q;$$

$$Q' = a!.Q + d?Q + a!.Q'.$$

Тоді модель системи можна подати композицією цих двох компонентів. При цьому утворюються взаємні зв'язки та додаткові стани системи, що відображають взаємодію компонентів і в цілому моделюють поведінку системи. Модель системи, утвореної композицією моделей компонентів (рис. 2.27), наведено на рис. 2.28.

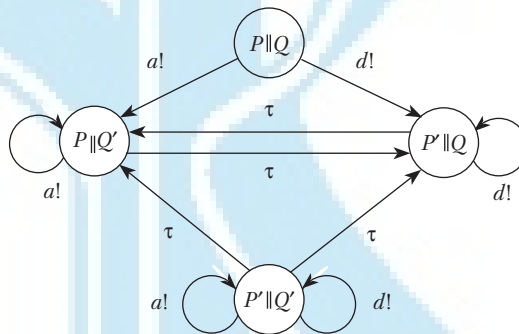


Рис 2.28. Модель взаємодії двох компонентів

Отримана модель відображає паралелізм і невизначеність при взаємодії двох компонентів у системі. У графічному вигляді подання на цій моделі як на прикладі можна розглянути всі зазначені вище моделі подання, наведені у прикладах 2.15–2.17.

Детальніше приклади використання апарату процесних алгебр в інтелектуалізованих моделях процесів функціонування складних дискретно-подійних систем розглядаються в розд. 11.

### Контрольні завдання

1. Дати визначення логічної системи подання знань.
2. Розкрити зміст чотирьох складових — умов відповідності логічної моделі формальній системі.
3. Пояснити, в чому полягає метод «зворотного міркування» — метод резолюцій при виведенні в логічних системах подання знань.

4. Прокоментувати методи прямого та зворотного виведення в логіці предикатів.
5. Навести означення вивідної та суперечливої формул і тавтологій у логіці висловлювань.
6. Пояснити, чим відрізняється логіка предикатів першого порядку від звичайної логіки висловлювань.
7. Дати тлумачення правильно побудованої формули і її попередньо нормальної форми в логіці предикатів.
8. Сформулювати правило резолюцій і розкрити його роль в утворенні істинної формули — резольвенти.
9. Пояснити, як будується дерево виведення при застосуванні методу резолюцій.
10. Розкрити склад загального вигляду формули продукції.
11. Охарактеризувати чотири складові — умови відповідності продукційної моделі формальній системі.
12. Пояснити, яку роль відіграє робоча пам'ять у продукційній СПЗ.
13. Навести методи обмеження конфліктних наборів і розв'язання конфліктів у продукційних СПЗ.
14. Сформулювати прямий і обернений способи виведення для продукційних СПЗ.
15. Дати визначення семантичної сітки.
16. Розкрити зміст чотирьох складових — умов відповідності моделі СС формальній системі.
17. Пояснити, що являє собою об'єкт в СС і які типи об'єктів в СС використовуються.
18. Розкрити можливість задання факту, пояснити його графічну інтерпретацію в СС.
19. Пояснити фізичний зміст екстенціоналу відношення в СС.
20. Розкрити зміст інтенціональної СС.
21. Навести й охарактеризувати основні класифікаційні ознаки СС.
22. Розкрити принципи, на яких будується виведення в СС.
23. Розкрити структуру і властивості фреймів і пояснити, що лежить в основі теорії фреймів.
24. Охарактеризувати чотири складові — умови відповідності ФМ формальній системі.
25. Охарактеризувати особливості подання знань фреймами.
26. Пояснити механізм управління виведеннями в ФМ.
27. Розкрити можливості поповнення первинних описів у ФМ.
28. Охарактеризувати концептуальну модель «дошки оголошень».
29. Навести чотири складові — умови відповідності ДО формальній системі.
30. Розкрити фізичний зміст джерел знань у моделі ДО і проілюструвати прикладом функціонування ДО як СПЗ.
31. Охарактеризувати структуру та функціональні особливості нейрона.
32. Розкрити чотири складові — умови відповідності моделі НС формальній системі.
33. Дати узагальнену характеристику узагальнених топологій НС.
34. Навести основні типи активаційних функцій (належності) і розкрити роль цих функцій у процедурах перетворення вхідних даних.

35. Пояснити, в яких випадках НС стає сіткою із семантикою.
36. Охарактеризувати НС із зворотними зв'язками і навести приклади таких топологій.
37. Розкрити основні способи і навести спрощені алгоритми навчання НС.
38. Охарактеризувати фізичний зміст моделі базового апарату СП, розкрити особливості її функціонування.
39. Навести чотири складові – умови відповідності моделі СП формальній системі.
40. Довести переваги використання апарату СП для відображення дискретно-подійних процесів.
41. Дати означення процесної алгебри.
42. Розкрити чотири складові – умови відповідності апарату ПА формальній системі.
43. Навести класифікацію процесів й охарактеризувати їх.
44. Розкрити зміст основних (дія-префікс, комунікації, переривання тощо) операторів ПА.
45. Пояснити особливості формування виведення в моделі ПА.
46. Охарактеризувати сферу застосування ПА.



## НЕДОСТОВІРНІ ТА НЕЧІТКІ ЗНАННЯ

---

---

*Теорії, в які ми віримо, ми називаємо фактами,  
а факти, в які не віримо — теоріями..*

Фелікс Коен

*Коли слухаючий не розуміє промовляючого,  
а промовляючий не знає, що він має на меті, — це філософія.*

Марі Француа Аруе

*Математик вже дещо може,  
проте, певна річ, не те, чого від нього жадають отримати на даний момент.*

Альберт Ейнштейн

### 3.1. Загальні положення

Найбільш вражаючою властивістю людського інтелекту є спроможність приймати правильні рішення в умовах неповної та нечіткої інформації. Побудова моделей наближених міркувань людини та використання їх в інтелектуальних комп'ютеризованих системах є нині одним з найперспективніших напрямків розвитку сучасної обчислювальної техніки.

Якщо для класичного поняття множини характеристична функція в інтервалі  $(0, 1)$  набуває значення 0 або 1, то, поширивши поняття на *нечіткі множини*, говорять, що *функція належності* (ФН) в цьому випадку може набувати будь-яких значень в межах цього інтервалу [202]. У цій праці американський вчений Лотфі Заде визначив також ряд операцій над нечіткими множинами і запропонував узагальнення відомих методів логічного виведення «modus ponens» і «modus tollens».

Запроваджуючи поняття *лінгвістичної змінної* і припустивши, що її значеннями (термами) виступають нечіткі множини, Л. Заде запропонував апарат для подання процесів інтелектуальної діяльності, включаючи нечіткість і невизначеність висловлювань. Це дало змогу створити фундамент теорії нечітких множин і нечіткої логіки, а також передумову для впровадження методів нечіткого управління в інженерну практику. У свою чергу, це зумовило постановку цілого ряду нових проблем, зокрема таких, як створення нових архітектур комп'ютерів для нечітких обчислень, елементної бази нечітких комп'ютерів і контролерів, методів та інструментальних засобів розробки нечітких систем управління тощо.

Математична теорія нечітких множин дає змогу подавати нечіткі поняття і знання, оперувати ними та робити нечіткі виведення. Нечітке управління є особливо корисним, коли досліджувані процеси є занадто складними для аналізу за допомогою загальноприйнятих методів або коли доступні джерела інформації інтерпретуються неякісно, неточно або невизначено. Нечітка ж логіка, на якій ґрунтується

нечітке управління, надаючи ефективні засоби відображення невизначеностей і неточностей реального світу, ближча до людського мислення та істотних мов, аніж традиційні логічні системи.

Характерним для фаззі-логіки і фаззі-управління є безпосереднє застосування експертних знань, які якісно формулюються для генерування керуючих впливів. Знання про взаємодію з процесом за цією методикою подається у формі правил продукції такого виду:

ЯКЩО (вихідна ситуація), ТО (відповідна реакція). Такі ЯКЩО-ТО-правила відповідають найпростішій формі людських взаємодій. На відміну від раніше розглянутого (див. підрозд. 2.2), у фаззі-логіці «вихідна ситуація» і «відповідна реакція» являють собою нечіткі висловлювання, наприклад:

ЯКЩО (швидкість пересування ланки маніпулятора *є надмірною*),

ТО (напругу на відповідний привідний двигун *зменшити*).

Або таке логічне висловлювання:

ЯКЩО (при захопленні деталі спостерігається її *незначне* проковзування в захватному пристрої ЗП промислового робота ПР) І (*є достатній* запас міцності деталі для збільшення зусилля захоплення), І (діапазон зусилля захоплення ЗП уможливує *деяке* його збільшення), ТО (необхідно *повністю* використати можливість ЗП).

Форму опису, що дає змогу операціоналізувати сформульовані вище знання, фаззі-теорія розглядає як розширення класичної теорії множин [202].

Отже, апарат нечітких множин і нечіткої логіки вже давно і з успіхом застосовується для розв'язування задач, в яких вихідні дані є ненадійними і слабоформалізованими.

Водночас для нечітких систем характерні й деякі недоліки, зокрема:

- вихідний набір постульованих нечітких правил формулюється експертом-людиною і може виявитися неповним або суперечливим;
- вид і параметри ФН, які описують параметри системи на вході та на виході, вибираються суб'єктивно і можуть виявитися не зовсім такими, що відображають реальну дійсність.

Для усунення, принаймні частково, наведених недоліків намагаються створювати нечіткі системи адаптивними, коригуючи, у міру їх функціонування, правила та параметри ФН. Найбільш вдалим прикладами таких систем є зокрема нечіткі нейронні сітки, які формально за структурою ідентичні з багатошаровими нейронними сітками з навчанням, наприклад, за алгоритмом зворотного поширення похибки (див. підрозд. 2.6), проте сховані шари в ній відповідають етапам функціонування нечіткої системи:

- перший шар нейронів виконує функцію введення нечіткості (fuzzification) на основі заданих ФН входів;
- другий шар відображує сукупність нечітких правил;
- третій шар виконує функцію зведення до чіткості (defuzzification).

Кожний з цих шарів характеризується набором параметрів (ФН, нечіткими розв'язувальними правилами, активаційними функціями, вагами зв'язків), настроювання яких виконується по суті так само, як і для звичайних нейросіток.

### 3.2. Поняття про недостовірні та нечіткі знання

**Неповні та недостовірні знання.** В розд. 1 і 2 розглядалися проблеми, пов'язані з використанням знань в ЕС та СПЗ. Розглянемо інші аспекти цієї проблеми.

**Неповні знання.** Ця категорія знань свідчить про те, що для доведення або спростування певного твердження може не вистачати необхідної інформації. У багатьох системах логічного виведення прийнято *постулат замкненості* світу [21], що означає, що на запит про істинність деякого твердження відповіддю буде «так» тоді і тільки тоді, коли його можна довести; якщо ж довести це твердження неможливо, то відповіддю стане «ні». Водночас «*неможливо довести через брак інформації*» (тобто неповноту знань) і «*доведено, що ні*» (тобто підтверджена неможливість використання знань через їх неповноту або хибність) — це зовсім не одне й те саме. Тому бажано, щоб в ЕС, якій доводиться мати справу з такими невизначеностями, формувалася запит до користувача про факти, яких не вистачає.

**Недостовірність знань.** На результат виконання деякого правила в продукційній СПЗ можуть впливати як випадкові чинники (*об'єктивна невизначеність*), так і такі, які залежать від невпевненості експерта в якомусь факті чи правилі виведення (*суб'єктивна невизначеність*).

Таким чином, ненадійність знань і недостовірність наявних фактів можуть значною мірою вплинути на результат логічного виведення, які формуються інтелектуалізованою системою, і цей фактор обов'язково повинен враховуватися в процесі логічних побудов. Якщо ж просто відкидати факти і правила виведення, які викликають сумнів, то доведеться відмовитися від цінної інформації, яка була неточно подана або неправильно інтерпретована системою. Часто буває так, що експерт не зовсім упевнений в тому чи іншому факті, але, незважаючи на це, інформація залишається цінною і повинна бути включена в базу знань. Тому необхідно розвивати процедури, які дають змогу здійснювати логічні побудови при недостовірних даних, використовуючи ці процедури в ЕС.

**Нечіткість знань.** На практиці часто доводиться мати справу з неточно визначеними або нечітко сформульованими поняттями — *лінгвістичними невизначеностями*, такими, як «багато», «мало», «сильно», «слабо» тощо.

Враховуючи викладене, наведемо деякі означення.

**Означення 3.1.** *Висловлювання називається неточним*, якщо його істинність або хибність не можуть бути встановлені однозначно, тобто твердження не є ані абсолютно достовірним, ані абсолютно хибним [42].

**Означення 3.2.** *Неточним виведенням називається логічне виведення в умовах неточності (недостовірності) знань.*

**Модальне твердження.** Неточне виведення традиційно розглядається як самостійний напрям, хоча неточне твердження можна було б інтерпретувати як частковий випадок *модального твердження*. Суть останнього полягає в тому, що воно оперує твердженнями, які дають оцінку іншим твердженням, або, інакше кажучи, є *опосередкованими* через інші твердження.

**Означення 3.3.** *Модальними логіками є логічні системи, які оперують модальними твердженнями.*

Як приклад наведемо такі модальні твердження, які можна занести в БЗ експертної системи: «ПР може захопити неправильно зорієнтовану деталь» (у таких випадках говорять про *логіки можливого*, або *алетичні логіки*) або «Система інформаційного забезпечення електроробочого визначає, що останній пройшов контрольну



(реперну) точку відліку його місцерозташування в просторі» (це приклад так званих *епістемічних логік*).

Неточне виведення слід відрізнити від роботи з нечіткими знаннями, хоча останні також формально підпадають під сформульоване вище визначення.

### 3.3. Чіткі та нечіткі множини

**Властивості чітких множин.** Нехай  $M$  — універсальна множина,  $x$  — елемент  $M$ , а  $G$  — деяка властивість. Звичайна (чітка) підмножина  $A$  універсальної множини  $M$ , елементи якої задовольняють властивості  $G$ , визначається як множина упорядкованих пар:

$$A = \{\mu_A(x)/x\},$$

де  $\mu_A(x)$  — *характеристична функція* (або ФН), для якої:

$$\mu_A(x) = \begin{cases} 1, & \text{якщо } x \text{ задовольняє властивості } G; \\ 0 & \text{— у протилежному разі.} \end{cases}$$

Види характеристичних (активаційних) функцій розглянуто в підрозд. 2.6.

**Властивості нечітких множин.** Великого значення для практичного застосування в задачах, які розв'язуються інтелектуальними роботами, набуває теорія нечітких множин, яка дає змогу будувати керування складними об'єктами в умовах лінгвістичної невизначеності. Наприклад, можна сказати: об'єкти роботизації, вага яких визначається грамами, *зовсім неважкі*, об'єкти вагою декілька кілограмів *більш-менш важкі*, а вагою десятки кілограмів — *досить важкі* і т. ін. При цьому так і залишається невизначеним поріг, з якого об'єкти роботизації стають важкими. Для формалізації таких умов використовують апарат *нечітких множин*.

Останнім часом у дослідженнях нечітких множин велика увага приділяється розробці *правил нечіткого умовного логічного виведення*, так званого Fuzzy Conditional Inference Rules, або скорочено FCIR. Ця обставина пов'язана з тим, що в семантиці звичайної мови існує певна кількість нечітких концепцій і робляться логічні висновки, в яких передумови і наслідки включають такі концепції.

У 1965 р. професором Л. Заде [30] було введено поняття *функції нечіткої належності* (ФНН)  $\mu_M(x)$  елемента до множини, дозволивши цій функції набувати будь-якого значення від 0 до 1. Чим більшим є значення ФНН, тим більшою мірою елемент належить до множини, причому якщо  $\mu_A(x) = 1$ , елемент  $x$  чітко належить множині  $A$ , якщо ж  $\mu_A(x) = 0$ , то елемент  $x$  чітко не належить цій множині. При  $0 < \mu_A(x) < 1$  елемент належить множині нечітко. Чим вагомішим є значення ФНН, тим більшою мірою елемент належить до множини. Іншими словами, нечітка множина відрізняється від звичайної тим, що для елементів  $x$  з  $M$  немає однозначної відповіді «так-ні» щодо властивості  $G$ . У зв'язку з цим нечітка підмножина  $A$  універсальної множини  $M$  визначається як множина упорядкованих пар

$$A = \{\mu_A(x)/x\},$$

де  $\mu_A(x)$  — характеристична ФН, яка набуває значення в деякій цілком упорядкованій множині  $E$  (наприклад,  $E = [0, 1]$ ).

**Означення 3.4.** *Нечіткою підмножиною  $A$  довільної непорожньої множини  $M$  називається множина пар  $A = \{(x, \mu_A(x))\}$ , де  $x \in M$ ,  $\mu_A(x) \in [0, 1]$ .*

**Означення 3.5.** *Функцією належності нечіткої підмножини  $A$  є функція  $\mu_A: E \rightarrow [0, 1]$ , базовою множиною (або базовою шкалою) якої є  $E$ .*

**Означення 3.6.** *Ступенем належності елемента  $x$  до нечіткої множини  $M$  називається значення  $\mu_A(x)$ , визначене для кожного  $x \in M$ .*

**Означення 3.7.** *Носієм нечіткої підмножини  $A$  називається чітка підмножина  $A^*$  множини  $M$ , що містить ті елементи з  $M$ , для яких  $\mu_A(x) = 0$ .*

Отже, нечітка підмножина характеризується своєю ФН, яка визначає, якою мірою елемент базової множини належить до нечіткої підмножини.

Отже, ФН вказує ступінь (або рівень) належності елемента  $x$  до підмножини  $A$ . Тоді  $E$  є множиною належностей, і якщо  $E = \{0, 1\}$ , то нечітку підмножину  $A$  можна розглядати як звичайну чітку множину.

**Твердження 3.1.** *Носієм нечіткої підмножини завжди є чітка звичайна підмножина базової множини.*

**Приклад 3.1.** Розглянемо як приклад формальне подання поняття «високий коефіцієнт завантаження обробного ресурсу». Для цього насамперед визначимо ФН до відповідної нечіткої множини.

Очевидно, базовою множиною  $M$  є множина невід'ємних дійсних чисел. Функцію ФН можна задати аналітичною формулою (рис. 3.1):

$$\mu_A(x) = \begin{cases} 0, & x < 0,5; \\ 1x - 0,5 & 0,5 \leq x \leq 0,9; \\ 1, & x > 0,9. \end{cases}$$

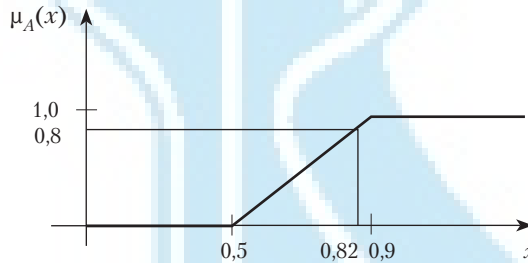


Рис. 3.1. Функція належності поняття «високий коефіцієнт завантаження обробного ресурсу»

Тоді відповіддю на запитання «Чи є високим коефіцієнт завантаження обробного ресурсу 0,82?» може бути відповідь «Ступінь належності 0,8», що можна проінтерпретувати як «досить високий».

Але цю саму нечітку множину можна задати і по-іншому, якщо явно перелічити деякі ступені належності, наприклад:

$$A = \{(0,5; 0), (0,6; 0,25), (0,7; 0,5), (0,8; 0,75), (0,9; 1)\}.$$

Необхідно звернути увагу на такі важливі моменти:

- *функція ФН не може бути визначена однозначно, оскільки вона є дуже суб'єктивною; кожна людина може запропонувати свої значення цієї функції;*
- *самі ступені належності можна розглядати як розмиті. Уявімо собі, що де-хто оцінив коефіцієнт завантаження обробного ресурсу (КЗОР) 0,85 см як високий зі ступенем належності 0,8. Якщо у нього спитати, чому 0,8, а не 0,85 чи 0,82, він не зможе чітко відповісти на це запитання. Тому замість конкретних числових значень ступенів належності можна говорити про інтервали оцінок ступенів належності. Якщо ступінь належності кожного елемента оці-*

нюється інтервалом, говорять про *нечіткі множини другого роду* [97] з ФН  $\mu_A: E \rightarrow [0, 1]^{[0,1]}$ . Такі міркування можна продовжити і далі. Втім, на практиці найчастіше користуються описаними раніше нечіткими множинами першого роду (з конкретними числовими значеннями ступенів належності);

- незважаючи на те, що ФН є суб'єктивною оцінкою, її не можна вибирати довільно (так, для прикладу з вагою об'єктів роботизації ФНН повинна бути *неспадною*). Наприклад, у нашому випадку ФН, яка описує поняття «високий зріст», повинна бути *неспадною*: неможливо, щоб хтось оцінював КЗОР 0,65 більшим ступенем належності, аніж для КЗОР 0,85.

**Основні характеристики нечітких множин.** Нехай  $E = [0, 1]$  та  $A$  — нечітка підмножина з елементами універсальної множини  $M$  та множиною належностей  $E$ .

**Означення 3.8.** Величина  $\sup_{x \in U} \mu_A(x)$  називається висотою нечіткої множини  $A$ . Зазначимо таке:

- якщо  $\sup_{x \in U} \mu_A(x) = \begin{cases} 1, & \text{то нечітка множина є нормальною;} \\ 0, & \text{то нечітка множина є субнормальною;} \end{cases}$
- нечітка множина є *порожньою*, якщо  $\forall x \in M$  і спостерігається  $\mu_A(x) = 0$ . *Непорожню субнормальну* множину можна нормувати за формулою:

$$\mu_A(x) := \mu_A(x) / \sup_{x \in U} \mu_A(x);$$

- нечітка множина *унімодална*, тобто  $\mu_A(x) = 1$  тільки на одному  $x$  з  $M$ ;
- *носієм* нечіткої множини  $A$  є звичайна підмножина із властивістю  $\mu_A(x) > 0$ , тобто *носії*  $A = \{x / \mu_A(x) > 0\}$ ,  $\forall x \in M$ ;
- елементи  $x \in M$ , для яких  $\mu_A(x) = 0,5$ , називаються *точками переходу* множини  $A$ .

**Методи побудови функції належності нечітких множин.** Virізняють *прямі* та *опосередковані* методи побудови ФН.

Як правило, **прямі методи** використовуються для вимірюваних понять (наприклад, швидкість, час, віддаль, тиск тощо) або якщо виокремлюються полярні значення. Тоді експерт задає для кожного  $x \in M$  значення  $\mu_A(x)$ . Так, виділивши при характеристиці об'єкта набір його ознак і для кожної з них визначивши полярні значення, які відповідають значенням функції належності 0 або 1, і виходячи з отриманої шкали задають  $\mu_A(x) \in [0, 1]$ , формуючи векторну ФН  $\{\mu_A(x_1), \mu_A(x_2), \dots, \mu_A(x_n)\}$ . Зокрема, різновидом прямих методів побудови функції ФН є *прямі групові методи*, коли, наприклад, групі експертів показують конкретний об'єкт, і кожний з експертів повинен дати одну з двох відповідей: належить чи ні цей об'єкт до заданої множини. Кількість ствердних відповідей, поділена на загальну кількість експертів, і визначає значення функції ФН об'єкта до заданої нечіткої множини.

**Опосередковані (непрямі) методи** визначення значень функцій ФН використовуються у тих випадках, коли немає вимірюваних елементарних властивостей, через які визначається нечітка множина. Якби значення функцій ФН було відомим (наприклад,  $\mu_A(x_i) \in w_i$ ,  $i = 1, 2, \dots, n$ ), то попарні порівняння можна подати матрицею відношень  $A = \{\alpha_{ij}\}$ , де  $\alpha_{ij} = w_i / w_j$  (операція ділення).

На практиці експерт сам формує матрицю  $A$ , при цьому припускається, що діагональні елементи дорівнюють 1, а для елементів, симетричних відносно головної діагоналі, співвідношення мають вигляд  $\alpha_{ij} = 1/\alpha_{ji}$ , тобто якщо один елемент оцінюється в  $\alpha$  разів більш значущим, ніж другий, то цей останній має бути в  $1/\alpha$  разів більш значущим, ніж перший. У загальному випадку задача зводиться до пошуку

вектора  $\mathbf{w}$ , що задовольняє рівняння виду  $\mathbf{A}\mathbf{w} = \xi_{\max} \mathbf{w}$ , де  $\xi_{\max}$  — найбільше власне значення матриці  $\mathbf{A}$ . Оскільки матриця  $\mathbf{A}$  додатна за побудовою, розв'язок цієї задачі існує і є додатним.

Нарешті, використовуються також *методи, які базуються на типових формах кривих* для ФН (зокрема, формі  $L$ - $R$ -типу [56]) з уточненням їх параметрів згідно з експериментальними даними.

**Операції над нечіткими множинами та нечіткі логічні виведення.** Наведемо означення деяких операцій над нечіткими множинами  $A$  і  $B$  на універсальній множині  $M$ .

**Означення 3.9.** *Рівними між собою* називаються дві нечіткі підмножини  $A$  та  $B$  (позначається як  $A = B$ ) базової множини  $M$ , якщо для будь-якого  $x \in M$  виконується  $\mu_A(x) = \mu_B(x)$ .

**Означення 3.10.** *Операція включення* відбувається тоді, коли підмножиною іншої множини  $B$  є деяка нечітка множина  $A$  (позначається  $A \subseteq B$ ), якщо для будь-якого  $x \in M$  виконується  $\mu_A(x) \leq \mu_B(x)$ .

**Означення 3.11.** *Перетином двох нечітких множин*  $A$  і  $B$  (позначається  $A \cap B$ ) базової множини  $M$  називається нечітка множина  $D = A \cap B$  з функцією належності  $\mu_M(x) = \min(\mu_A(x), \mu_B(x))$ .

**Означення 3.12.** *Об'єднанням двох нечітких множин*  $A$  і  $B$  (позначається  $A \cup B$ ) називається нечітка множина  $D = A \cup B$  з функцією належності  $\mu_M(x) = \max(\mu_A(x), \mu_B(x))$ .

**Означення 3.13.** *Доповненням нечіткої множини*  $A$  називатимемо нечітку множину  $B$ , якщо  $\mu_B(x) = 1 - \mu_A(x)$ . Очевидно, що  $(\overline{\overline{A}}) = A$  (тут доповнення визначено для  $E = [0, 1]$ , проте також очевидно, що його можна поширити на будь-яке упорядковане  $E$ ).

**Означення 3.14.** *Різницею двох нечітких множин*  $A$  і  $B$  базової множини  $M$  є нечітка множина  $C = A - B = A \cap \overline{B}$  з функцією належності  $\mu_C(x) = \mu_{A \cap \overline{B}}(x) = \min(\mu_A(x), 1 - \mu_B(x))$ .

**Означення 3.15.** *Диз'юнктивною сумою нечітких множин*  $A$  і  $B$  є нечітка множина  $H = A \oplus B = (A - B) \cup (B - A) = (A \cap \overline{B}) \cup (\overline{A} \cap B)$ , для якої існує ФН  $\mu_H(x) = \max\left\{\left[\min(\mu_A(x), 1 - \mu_B(x))\right]; \left[\min(1 - \mu_A(x), \mu_B(x))\right]\right\}$ .

Для нечітких множин виконуються всі основні властивості звичайних чітких множин, крім двох: об'єднання  $A \cup \overline{A}$  не обов'язково дорівнює базовій множині; перетин  $A \cap \overline{A}$  не обов'язково дорівнює порожній множині.

Як бачимо, для подання основних логічних операцій за допомогою характеристичних функцій застосовується відома з булевої алгебри двійкова логіка, що є основою всіх ЕОМ. Для графічної інтерпретації переведення основних логічних операцій у нечіткі множини наведемо спочатку значення двох множин  $A$  і  $B$  у формі характеристичних функцій 1 і 2 чітких множин і ФН 3 і 4 нечітких множин (рис. 3.2).

Пояснимо коротко окремі логічні операції:

- множина логічного об'єднання  $D = A \cup B$  (АБО) обох множин  $A$  і  $B$  визначається за всіма тими елементами  $x$ , що належать  $A$  або  $B$  або обом множинам. Характеристичну функцію множини  $D$ , як це впливає з означення 3.12, можна в такий спосіб отримати формуванням максимуму (АБО)  $\mu_M(x) = \max[(\mu_A(x), \mu_B(x))]$  (крива 5 на рис. 3.2);

- множина логічного перетину  $D = A \cap B$  (І) обох множин  $A$  і  $B$  визначається за всіма тими елементами  $x$ , що належать як  $A$ , так і  $B$ . Характеристичну функцію  $D$  можна за означенням 3.11 отримати формуванням мінімуму (І)  $\mu_M(x) = \min[\mu_A(x), \mu_B(x)]$  (крива 6 на рис. 3.2);
- множина логічного доповнення  $D = \bar{A}$  (НІ) до множини  $A$  визначається всіма тими елементами  $x$ , що не належать до множини  $A$ . Характеристичну функцію чіткої множини логічного доповнення  $D$ , згідно з означенням 3.13, можна подати у вигляді  $\mu_B(x) = 1 - \mu_A(x)$  (крива 7 на рис. 3.2).

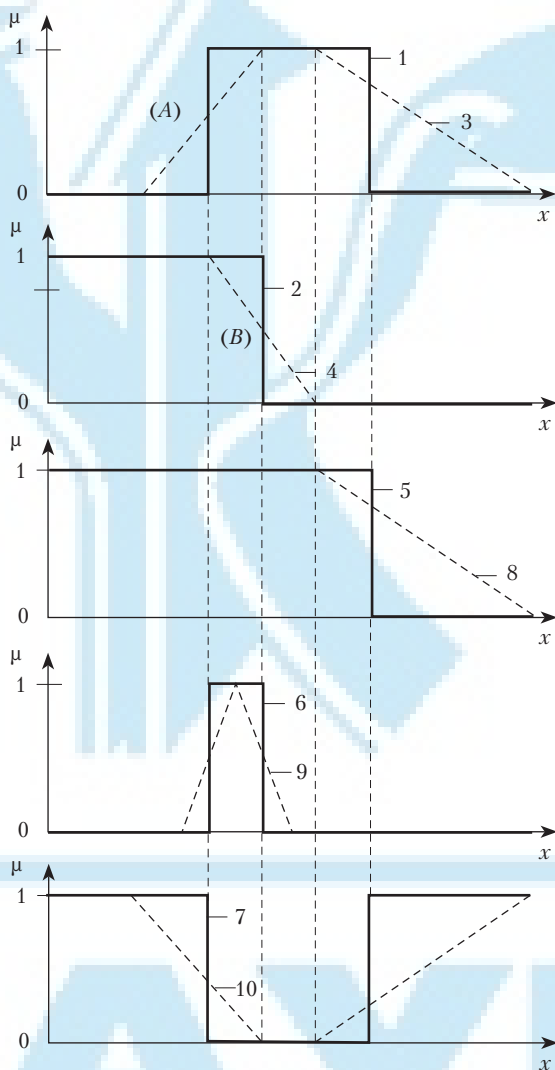


Рис. 3.2. Характеристичні функції множин  $A$  і  $B$  при двійковій логіці:

$\mu_A, \mu_B$  – графіки характеристичних функцій множин  $A$  і  $B$  при двійковій логіці (суцільні лінії 1 і 2) і відповідних їм ФН при фаззі-логіці (пунктирні лінії 3 і 4); графіки подання основних операцій при двійковій логіці (суцільні лінії 5, 6 і 7) і відповідні ФН при фаззі-логіці (пунктирні лінії 8, 9 і 10)

Оскільки характеристичну функцію чіткої множини можна розглядати як окремий випадок ФН нечіткої множини, наведені вище форми подання основних операцій з чіткими множинами можуть бути безпосередньо застосовані і для нечітких множин:

- для  $D = A \cup B$  (АБО) маємо  $\mu_{A \cup B}[\mu_A(x), \mu_B(x)]$  (крива 8 на рис. 3.2);
- для  $D = A \cap B$  (І) маємо  $\mu_{A \cap B}[\mu_A(x), \mu_B(x)]$  (крива 9 на рис. 3.2);
- для  $D = \bar{A}$  (НІ) маємо  $\mu_{\bar{A}}(x) = \mu_B(x) = 1 - \mu_A(x)$  (крива 10 на рис. 3.2).

Попарне порівняння кривих 5 і 8, 6 і 9, а також 7 і 10 наочно ілюструє процеси переходу від чіткої до фаззі-логіки.

Відзначимо також, що крім основних операцій для нечітких множин зберігають силу й інші відомі обчислювальні закони булевої алгебри, а саме:

- комутативність  
 $A \cup (B \cap C) = (A \cup B) \cap C$  або  $A \cap (B \cup C) = (A \cap B) \cup C$ ;
- асоціативність  
 $A \cup B = B \cup A$  або  $A \cap B = B \cap A$ ;
- дистрибутивність  
 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$  або  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ ;
- $A \cup \emptyset = A$   
 $A \cap \emptyset = \emptyset$  } де  $\emptyset$  – порожня множина, тобто  $\mu_{\emptyset}(x) = 0$  для  $\forall x \in M$ ;
- $A \cap M = A$   
 $A \cup M = M$  } де  $M$  – універсальна множина;
- $\overline{A \cap B} = \bar{A} \cup \bar{B}$   
 $\overline{A \cup B} = \bar{A} \cap \bar{B}$  } – формули де Моргана

та ін. (див. підрозд. 2.1), крім неприйнятності співвідношень  $A \cup \bar{A} = 1$ ,  $A \cap \bar{A} = 0$ .

**Зауваження 3.1.** Зазначимо, що введені вище операції над нечіткими множинами базуються на використанні операцій **max** і **min**. В теорії нечітких множин розглядаються питання побудови узагальнених, параметризованих операторів перетину, об'єднання та доповнення, які дають змогу враховувати різні змістові відтінки відповідних їм операторів «І», «АБО», «НІ».

Один з підходів до узагальнення операторів перетину та об'єднання полягає в їх визначенні у класі трикутних норм і конорм, які впливають з наведених на рис. 2.18 (підрозд. 2.6) нормованих активаційних функцій (або ще функцій належності).

**Означення 3.16.** Трикутною нормою (*T-нормою*) є двомісна дійсна функція  $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$ , що задовольняє такі умови:

$T(0, 0) = 0; T(\mu_A, 1) = \mu_A; T(1, \mu_A) = \mu_A$ – обмеженість;	(3.1)
$T(\mu_A, \mu_B) \leq T(\mu_C, \mu_D)$ , якщо $\mu_A \leq \mu_C, \mu_B \leq \mu_D$ – монотонність;	(3.2)
$T(\mu_A, \mu_B) = T(\mu_B, \mu_A)$ – комутативність;	(3.3)
$T(\mu_A, T(\mu_B, \mu_C)) = T(T(\mu_A, \mu_B), \mu_C)$ – асоціативність.	(3.4)

Прикладами *T*-норм можуть бути:  $\min(\mu_A, \mu_B)$ ; добуток  $\mu_A \cdot \mu_B$ ;  $\max(0, \mu_A + \mu_B - 1)$ .

**Означення 3.17.** Трикутною конормою ( $T$ -конормою) є двомісна функція  $S: [0, 1] \times [0, 1] \rightarrow [0, 1]$  з такими властивостями:

$S(1, 1) = 1; S(\mu_A, 0) = \mu_A; S(0, \mu_A) = \mu_A$ – обмеженість;	(3.5)
$S(\mu_A, \mu_B) \geq S(\mu_C, \mu_D)$ , якщо $\mu_A \geq \mu_C, \mu_B \geq \mu_D$ – монотонність;	(3.6)
$S(\mu_A, \mu_B) = S(\mu_B, \mu_A)$ – комутативність;	(3.7)
$S(\mu_A, S(\mu_B, \mu_C)) = S(S(\mu_A, \mu_B), \mu_C)$ – асоціативність.	(3.8)

Прикладами  $T$ -конорм є:  $\max(\mu_A, \mu_B)$ ;  $\min(1, \mu_A + \mu_B)$ .

**Нечіткі та лінгвістичні змінні.** Поняття нечіткої та лінгвістичної змінних застосовується при поданні об'єктів та явищ за допомогою нечітких множин.

**Означення 3.18.** Нечітка змінна характеризується трійкою параметрів  $\langle a, X, A \rangle$ , де  $a$  – ім'я змінної;  $X$  – універсальна множина (область визначення  $a$ );  $A$  – нечітка множина на  $X$ , що описує обмеження (тобто  $\mu_A(x)$ ) на значення нечіткої змінної  $a$ .

**Означення 3.19.** Лінгвістична змінна (ЛЗ) характеризується набором параметрів  $\langle b, T, X, S, G \rangle$ , де  $b$  – ім'я лінгвістичної змінної (символ  $b$  може використовуватися як для назви самої змінної, так і для позначення усіх її значень. Наприклад, терм «незначне», який є значенням лінгвістичної змінної  $b = \langle \text{зусилля захватного пристрою} \rangle$ , одночасно може бути і нечіткою множиною  $S$  («незначне», що може впливати з контексту подання лінгвістичної змінної);  $T$  – множина значень лінгвістичної змінної (терм-множина), що є назвою нечітких змінних, областю визначення кожної з яких є множина  $S$ . Множина  $T$  називається базовою терм-множиною ЛЗ;  $S$  – семантична процедура, яка дає змогу перетворити кожне нове значення лінгвістичної змінної, утворене процедурою  $G$ , в нечітку змінну, тобто сформувати відповідну нечітку множину;  $G$  – синтаксична процедура, яка дає змогу оперувати елементами терм-множини  $T$ , зокрема генерувати нові терми (значення).

**Означення 3.20.** Розширеною терм-множиною ЛЗ називається множина  $T \cup G(T)$ , де  $G(T)$  – множина згенерованих термів.

**Приклад 3.2.** Розглянемо приклад, за яким експерт визначає зусилля, яке необхідне захватному пристрою (ЗП), щоб утримувати крихкий об'єкт. При цьому він оперує поняттями «мале зусилля», коли об'єкт проковзує в ЗП, «нормальне зусилля», коли об'єкт утримується ЗП без пошкоджень поверхневого шару і руйнації об'єкта, і «велике зусилля», коли поверхневий шар об'єкта руйнується від тиску губок ЗП на поверхню цього об'єкта. Нехай також мінімальне зусилля ЗП визначається значенням  $10 \text{ г/мм}^2$ , а максимальне –  $100 \text{ г/мм}^2$ .

Формалізацію такого подання можна виконати за допомогою такої лінгвістичної змінної  $\langle b, T, X, S, G \rangle$ , де  $b$  – зусилля ЗП;  $T = \{ \langle \text{мале зусилля} \rangle, \langle \text{нормальне зусилля} \rangle, \langle \text{велике зусилля} \rangle \}$ ;  $X = [10, 100]$ ;  $G$  – процедура утворення нових термів за допомогою зв'язок «і», «або» та модифікаторів типу «дуже», «ні», «слабо». Наприклад: «мале або нормальне зусилля», «дуже мале зусилля» тощо;  $S$  – процедура задавання на  $X = [10, 100]$  нечітких підмножин  $A_1 = \langle \text{мале зусилля} \rangle$ ,  $A_2 = \langle \text{нормальне зусилля} \rangle$ ,  $A_3 = \langle \text{велике зусилля} \rangle$ , а також нечітких множин для термів з  $G(T)$  згідно з правилами трансляції нечітких зв'язок і модифікаторів [56] «і», «або», «дуже», «ні», «слабо» та інших над нечіткими множинами виду:  $A \cap B, A \cup B, \bar{A}$ ,  $\text{CON } A = A^2$ ,  $\text{DIL } A = A^{0.5}$  тощо.

Треба зазначити, що поряд з наведеними вище базовими значеннями ЛЗ «зусилля» ( $T = \{ \langle \text{мале зусилля} \rangle, \langle \text{нормальне зусилля} \rangle, \langle \text{велике зусилля} \rangle \}$ ) можливі значен-

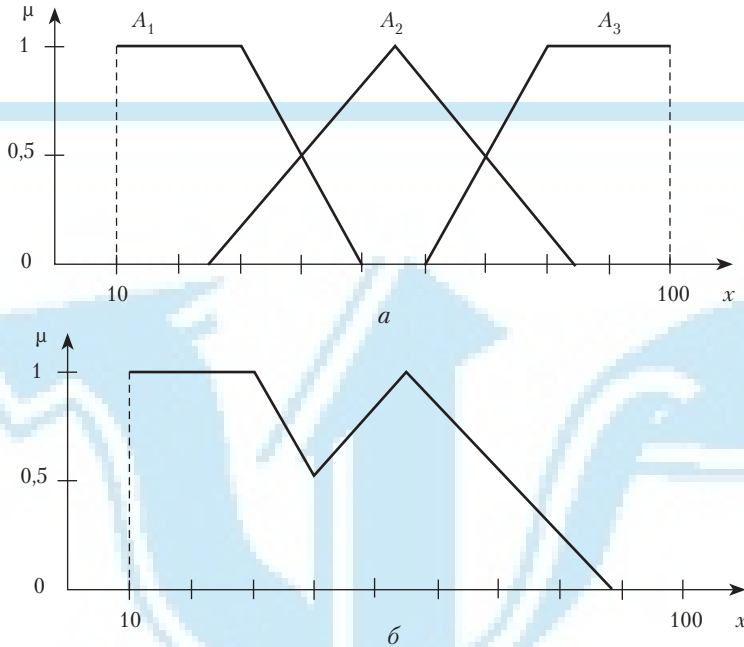


Рис. 3.3. Функції належності нечітких множин:  $A_1 = \langle \text{мале зусилля} \rangle$ ,  $A_2 = \langle \text{нормальне зусилля} \rangle$ ,  $A_3 = \langle \text{велике зусилля} \rangle$  (а) та  $A_1 \cup A_2 = \langle \text{мале або нормальне зусилля} \rangle$  (б)

ня, які залежать від області визначення  $X$ . Нехай у розглядуваному випадку значення ЛЗ «зусилля ЗП» можуть бути визначені як «близько 30 г/мм<sup>2</sup>», «близько 60 г/мм<sup>2</sup>», «близько 80 г/мм<sup>2</sup>», тобто у вигляді нечітких чисел.

Терм-множину та розширену терм-множину в умовах наведеного прикладу можна охарактеризувати функціями належності, поданими на рис. 3.3, а та 3.3, б відповідно.

### Нечіткі числа.

**Означення 3.21.** *Нечіткі числа* — це нечіткі змінні, визначені на числовій осі, тобто нечітке число визначається як нечітка множина  $A$  на множині дійсних чисел  $R$  з ФН  $\mu_A(x) \in [0, 1]$ , де  $x \in R$ .

**Означення 3.22.** *Нечітке число  $A$  є нормальним*, якщо  $\max \mu_A(x) = 1$ , та є *опуклим*, якщо для будь-яких  $x \leq y \leq z$  виконується  $\mu_A(x) \geq \mu_A(y) \wedge \mu_A(z)$ .

**Означення 3.23.** *Множина  $\alpha$ -рівня нечіткого числа  $A$  визначається як  $A_\alpha = \{x/\mu_A(x) \geq \alpha\}$ .*

**Означення 3.24.** *Носієм нечіткого числа  $A$  називається підмножина  $C_A \subset R$ , якщо  $C_A = \{x/\mu_A(x) > 0\}$ .*

**Означення 3.25.** *Нечітке число  $A$  є унімодальним*, якщо умова  $\max_x \mu_A(x) = 1$  справедлива тільки для однієї точки дійсної осі.

**Означення 3.26.** *Нечітким нулем є опукле нечітке число  $A$ , якщо справедливе  $\mu_A(0) = \sup_x (\mu_A(x))$ .*

**Означення 3.27.** *Нечітке число  $A$  є унімодальним*, якщо  $\forall x \in C_A, x > 0$ , та є *від'ємним*, якщо  $\forall x \in C_A, x < 0$ .

Операції над нечіткими числами розглянуто у [56].



### 3.4. Нечітке логічне виведення

**Спрощена постановка задачі.** Найтипівіша задача логічного виведення в умовах нечіткості формулюється так:

«дано продукційне правило: **«Якщо  $A$ , То  $B$ ;  
спостерігається  $A^*$  ( $A$  певною мірою)».**

Яким повинно бути  $B$ ?»

Формальніше, задано носії  $C$  і  $D$  та їх нечіткі підмножини  $A$  і  $B$ . Задано або елемент  $c \in C$ , або множину  $A' \subseteq C$ . Потрібно визначити елемент  $d \in D$ , що являє собою виведення системи, яке визначає результат застосування нечіткого продукційного правила.

**Приклад 3.3.** Дано нечітке продукційне правило:

«Якщо промисловий робот  $PR$  інтенсивно задіяний у виробничому процесі, то в ньому високий показник споживання електроенергії (ПСЕ)».

Як множину  $C$  ми розглядаємо множину чисел, що визначають кількість годин на тиждень, які  $PR$  може бути задіяним у виробничому процесі. Візьмемо  $C$  як діапазон чисел від 0 до 40 і для простоти подальших міркувань обмежимося невеликою кількістю можливих значень:

$$C = \{0, 6, 12, 18, 24, 30, 35, 40\}.$$

Аналогічно, якщо за максимальну оцінку  $D$  показника ПСЕ для  $PR$  прийняти 10, а діапазон його значень взяти від 0,3 до 1,0, то знову обмежимося невеликим набором можливих значень:

$$D = \{3,0; 5,0; 6,5; 8,0; 9,0; 10,0\}.$$

Задамо  $\Phi N$  для нечітких множин  $A$  («інтенсивно задіяний у виробничому процесі») та  $B$  («високий ПСЕ») таким чином:

$$A = \{(6; 0); (12; 0,2); (18; 0,4); (24; 0,6); (30; 0,8); (35; 0,9); (40; 0,1)\}.$$

$$B = \{(3,0; 0,1); (5,0; 0,2); (6,5; 0,4); (8,0; 0,7); (9,0; 0,85); (10,0; 1,0)\}.$$

Нехай явним чином задано кількість годин функціонування  $PR$  у виробничому процесі, або ступінь належності, що визначає, чи інтенсивно він задіяний у виробничому процесі (що еквівалентно, бо, знаючи конкретну кількість годин, ми можемо визначити відповідний ступінь належності). Нехай ступінь належності дорівнює  $\alpha$ . Тоді для виведення можна застосувати *метод простої підстановки нечіткого значення*, відповідно до якого  $d$  вибирається з умови:

$$\mu_B(d) = \alpha.$$

Нехай у нашому прикладі дано, що  $PR$  задіяний у виробничому процесі 29 годин. При цьому ступінь належності дорівнює 0,77, і система повинна дійти висновку, що за такої інтенсивності використання  $PR$  у виробничому процесі його ПСЕ дорівнюватиме 8,5 («досить великий»). Оскільки в нашій спрощеній множині  $D$  не виявилось значення точно з таким ступенем належності, величину ПСЕ визначено шляхом інтерполяції (або вибором наближеного значення).

**Узагальнена постановка задачі нечіткого виведення.** Тепер розглянемо більш детально процедуру нечіткого виведення. Механізм нечітких виведень, який використовується в різного роду експертних і керуючих системах, формується спеціалістами предметної галузі у вигляді сукупності нечітких предикатних правил виду:

$$\begin{aligned} \Pi_1: & \text{ЯКЩО } x \in A_1, \text{ ТО } y \in B_1; \\ \Pi_2: & \text{ЯКЩО } x \in A_2, \text{ ТО } y \in B_2; \\ & \dots \\ \Pi_n: & \text{ЯКЩО } x \in A_n, \text{ ТО } y \in B_n, \end{aligned} \quad (3.9)$$

де  $x$  — вхідна змінна (ім'я відомих значень даних);  $y$  — змінна виходу (ім'я значення даних, яке буде обчислене);  $A$  і  $B$  — функції належності, визначені відповідно на  $x$  і  $y$ .

Отже, знання експерта  $A \rightarrow B$  відображає нечітке причинне відношення передумови і висновку, тому його називають *нечітким відношенням* і визначають як  $R = A \rightarrow B$ , де « $\rightarrow$ » — нечітка імплікація.

Відношення  $R$  можна розглядати як нечітку підмножину прямого добутку  $x$  та  $y$  повної множини передумов  $X$  і виведень  $Y$ . Тоді процес отримання нечіткого результату виведення  $B'$  із застосуванням цього спостереження  $A'$  та знання  $A \rightarrow B$  можна подати у вигляді композиційного правила *нечіткої «modus ponens»*:

$$B' = A' \bullet R = A' \bullet (A \rightarrow B), \quad (3.10)$$

де « $\bullet$ » — операція згортання (композиції).

Операції композиції та імплікації в алгебрі нечітких множин можна реалізувати по-різному (з різними результатами), проте у будь-якому разі загальне логічне виведення здійснюється за такі чотири етапи [56; 76]:

1. **Введення нечіткості** (фазифікація) — для визначення ступеня істинності кожної передумови кожного правила застосування функцій належності, визначених на вхідних змінних, до їх фактичних значень.

2. **Логічне виведення** — застосування обчисленого значення істинності для передумов кожного правила до виведень кожного правила. Це приводить до однієї нечіткої множини, яка буде призначена кожною змінною виведення для кожного правила. Як правила логічного виведення зазвичай використовуються тільки операції **min** (МІНІМУМ) або **prod** (ДОБУТОК). У логічному виведенні **min** функція належності виведення «відтинається» за висотою, відповідною обчисленому ступеню істинності передумови правила (так звана нечітка логіка «I»). В логічному виведенні **prod** функція належності виведення масштабується за допомогою обчисленого ступеня істинності передумови правила.

3. **Композиція** — об'єднання разом усіх нечітких підмножин, призначених до кожної змінної виведення (в усіх правилах), для формування однієї нечіткої підмножини для всіх змінних виведення. При такому об'єднанні зазвичай використовуються операції **max** (МАКСИМУМ) або **sum** (СУМА). При композиції **max** комбіноване виведення нечіткої підмножини конструюється як покроковий максимум за всіма нечіткими підмножинами (так звана нечітка логіка «АБО»). При композиції **sum** комбіноване виведення нечіткої підмножини формується як покрокова сума за всіма нечіткими підмножинами, призначеними змінною виведення правилами логічного виведення.

**Властивості max-min-композиції.** Операція  $R_3 \bullet (R_2 \bullet R_1) = (R_3 \bullet R_2) \bullet R_1$  **max-min**-композиції асоціативна, тобто дистрибутивна відносно об'єднання, але недистрибутивна відносно перетину:

$$\left. \begin{aligned} R_3 \bullet (R_2 \cup R_1) &= (R_3 \bullet R_2) \cup (R_3 \bullet R_1); \\ R_3 \bullet (R_2 \cap R_1) &\neq (R_3 \bullet R_2) \cap (R_3 \bullet R_1) \end{aligned} \right\} \quad (3.11)$$

Крім того, для **max-min**-композиції виконується така важлива властивість:

$$\text{ЯКЩО } R_1 \subset R_2, \text{ ТО } R_3 \bullet R_1 \subset R_3 \bullet R_2. \quad (3.12)$$

*Властивості max-\*-композиції.* Для **max-min**-композиції відношень  $R_1$  і  $R_2$  у виразі  $\mu_{(R_1 \cdot R_2)}(x, z) = \bigvee_y [\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z)]$  операцію « $\wedge$ » можна замінити будь-якою іншою, для якої виконуються ті самі обмеження (наприклад, асоціативність, монотонність), що й для « $\wedge$ ». Тоді маємо:

$$\mu_{(R_1 * R_2)}(x, z) = \bigvee_y [\mu_{R_1}(x, y) * \mu_{R_2}(y, z)]. \quad (3.13)$$

Зокрема, операція « $\wedge$ » може бути замінена алгебраїчним добутком, і тоді кажуть про **max-prod**-композицію.

**4. Зведення до чіткості** (дефаззифікація) — перетворення (за необхідності) нечіткого набору виведень до чіткої величини (числа). Існує багато методів зведення до чіткості, деякі з алгоритмів яких розглянуто далі. Але спочатку розглянемо приклад деякої системи, яка подається нечіткими правилами.

**Приклад 3.4.** Нехай деяка система задається нечіткими правилами вигляду (3.9):

$$\begin{aligned} \Pi_1: & \text{ЯКЩО } x \in A, \text{ ТО } w \in D; \\ \Pi_2: & \text{ЯКЩО } y \in B, \text{ ТО } w \in E; \\ \Pi_3: & \text{ЯКЩО } z \in C, \text{ ТО } w \in F, \end{aligned} \quad (3.14)$$

де  $x, y, z$  — імена вхідних змінних;  $w$  — ім'я змінної виведення;  $A, B, C, D, E, F$  — задані функції належності (наприклад, трикутної форми, див. розд. 2.6, рис. 2.18,  $\delta$ ).

Процедура отримання логічного виведення ілюструється рис. 3.4. Припускається, що задано конкретні (чіткі) значення вхідних змінних:  $x_0, y_0, z_0$

На *першому етапі* (фаззифікація) на основі цих значень і, виходячи з функцій належності  $A, B, C$ , визначаються ступені істинності  $\alpha(x_0), \alpha(y_0), \alpha(z_0)$  для передумов кожного з трьох наведених правил (3.14).

На *другому етапі* (логічне виведення) відбувається «відгинання» функцій належності виведень  $D, E, F$  правил (3.14) на рівнях  $\alpha(x_0), \alpha(y_0), \alpha(z_0)$ .

На *третьому етапі* (композиція) розглядаються функції належності, зрізані на попередньому етапі, і виконується їх об'єднання з використанням операції **max**, в результаті чого одержуємо комбіновану нечітку підмножину, яка подається функцією належності  $\mu_\Sigma(w)$  і відповідає логічному виведенню для вихідної змінної  $w$ .

На *четвертому етапі* (дефаззифікація) визначається, за необхідності, чітке значення вихідної змінної, що може реалізуватися із застосуванням, наприклад, *центроїдного методу*: чітке значення вихідної змінної визначається як центр ваги для кривої  $\mu_\Sigma(w)$ :

$$w_0 = \frac{\int \omega \mu_\Sigma(w) d\omega}{\int \mu_\Sigma(w) d\omega}. \quad (3.15)$$

Якщо проінтерпретувати наведені в *прикладі 3.4* результати *прикладом 3.2* з визначенням з виразу (3.15) чіткого значення  $w_0$  для нормального утримання об'єкта в захватному пристрої, то це значення лежить у площині функції належності  $E$  (рис. 3.4).

**Модифіковані алгоритми нечіткого виведення.** Для розгляду найбільш уживаних модифікацій алгоритмів нечіткого виведення спростимо задачу, вважаючи, що базу знань утворюють тільки два нечітких правила вигляду:

$$\begin{aligned} \Pi_1: & \text{ЯКЩО } x \in A_1 \text{ і } y \in B_1, \text{ ТО } z \in C_1; \\ \Pi_2: & \text{ЯКЩО } x \in A_2 \text{ і } y \in B_2, \text{ ТО } z \in C_2; \end{aligned} \quad (3.16)$$

де  $x$  і  $y$  — імена вхідних змінних;  $z$  — ім'я змінної виведення;  $A_1, A_2, B_1, B_2, C_1, C_2$  — деякі задані функції належності.

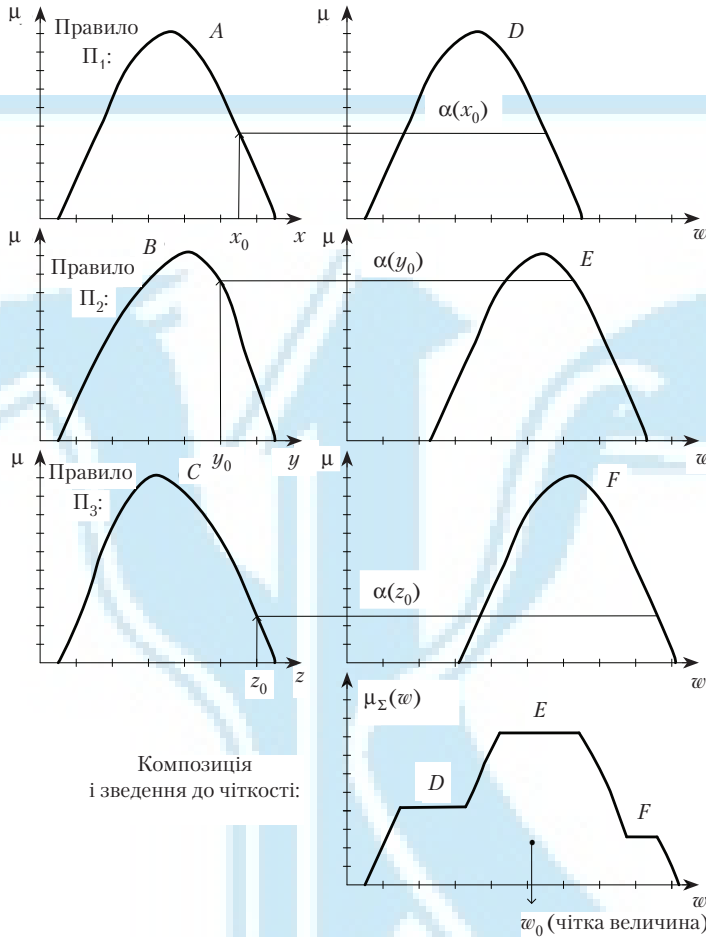


Рис. 3.4. Ілюстрація процедури логічного виведення

При цьому чітке значення  $z_0$  необхідно визначити на основі наведеної інформації і чітких знань  $x_0$  та  $y_0$ .

**Алгоритм Матдани.** Цей алгоритм відповідає вже розглянутому прикладу 3.4 і проілюстрований рис. 3.4. Математично для розглянутої ситуації його було подано чотирма етапами:

- *введення нечіткості* (фаззифікація) з визначенням ступеня істинності для передумов кожного правила:  $A_1(x_0)$ ,  $A_2(x_0)$ ,  $B_1(y_0)$ ,  $B_2(y_0)$ ;
- *логічне виведення* із знаходженням рівнів «відтинання» для передумов кожного з правил (із застосуванням операції **min**):

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0); \quad \alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

де через « $\wedge$ » позначено операцію логічного мінімуму **min**. Далі визначаються «відтяти» функції належності:

$$C'_1(z) = (\alpha_1 \wedge C_1(z)); \quad C'_2(z) = (\alpha_2 \wedge C_2(z));$$

- *композиція* з об'єднанням знайдених відтятих функцій із застосуванням операції **max** (яку надалі позначатимемо як  $\langle \vee \rangle$ , що дає змогу отримати підсумкову нечітку підмножину для змінної виходу з функцією належності:

$$\mu_{\Sigma}(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z));$$

- зведення до чіткості з визначенням чіткого значення  $z_0$ , наприклад, центроїдним методом.

**Алгоритм Tsukamoto.** Для цього алгоритму вихідні посилання такі самі, що й для алгоритму Mamdani, але тут припускається, що функції  $C_1(z)$  і  $C_2(z)$  є монотонними. Тоді алгоритм Tsukamoto визначається такою послідовністю:

- введення нечіткості (як і в алгоритмі Mamdani);
- нечітке виведення відбувається в два етапи. Спочатку визначаються рівні «відтинання»  $\alpha_1$  і  $\alpha_2$  (як в алгоритмі Mamdani), а потім, розв'язуючи рівняння

$$\alpha_1 = C_1(z) \text{ і } \alpha_2 = C_2(z),$$

знаходяться чіткі значення  $z_1$  і  $z_2$  для кожного вихідного правила;

- зведення до чіткості значень змінної виведення (як зважене середнє  $z_1$  і  $z_2$ ):

$$z_0 = (\alpha_1 z_1 + \alpha_2 z_2) / (\alpha_1 + \alpha_2), \quad (3.17)$$

або для загального випадку (дискретний варіант центроїдного методу):

$$z_0 = \left( \sum_{i=1}^n \alpha_i z_i \right) / \left( \sum_{i=1}^n \alpha_i \right). \quad (3.18)$$

**Приклад 3.5.** Нехай задано (рис. 3.5):

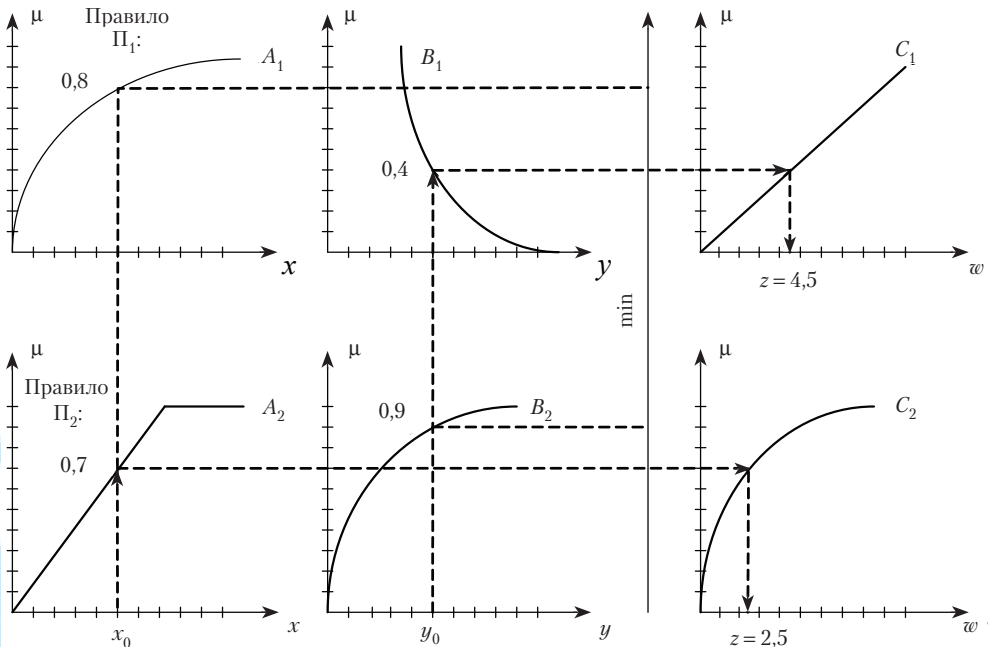


Рис. 3.5. Ілюстрація до алгоритму Tsukamoto

ступені істинності для передумов кожного правила (3.16)

$$A_1(x_0) = 0,8; A_2(x_0) = 0,7; B_1(y_0) = 0,4; B_2(y_0) = 0,9,$$

відповідні рівні «відтинання»:

$$\alpha_1 = \min\{A_1(x_0); B_1(y_0)\} = \min\{0,8; 0,4\} = 0,4,$$

$$\alpha_2 = \min\{A_2(x_0); B_2(y_0)\} = \min\{0,7; 0,9\} = 0,7,$$

а також значення  $z_1 = 4,5$  і  $z_2 = 2,5$ , які знайдено в результаті розв'язання рівнянь  $C_1(z_1) = 0,4$  і  $C_2(z_2) = 0,7$  (див. рис. 3.5).

Тоді в результаті дефаззифікації чітке значення змінної виведення, обчислене за виразом (3.17), має вигляд:

$$z_0 = (\alpha_1 z_1 + \alpha_2 z_2) / (\alpha_1 + \alpha_2) = (0,4 \cdot 4,5 + 0,7 \cdot 2,5) / (0,4 + 0,7) \cong 3,23.$$

**Алгоритм Sugeno** містить введення нечіткості (як в алгоритмі Mamdani) та нечітке виведення із знаходженням:

- рівнів «відтинання»:

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0) \text{ і } \alpha_2 = A_2(x_0) \wedge B_2(y_0);$$

- індивідуальних виходів правил:

$$z_1^* = a_1 x_0 + b_1 y_0;$$

$$z_2^* = a_2 x_0 + b_2 y_0;$$

- чіткого значення змінної виходу:

$$z_0 = (a_1 z_1^* + a_2 z_2^*) / (a_1 + a_2),$$

що ілюструється рис. 3.6.

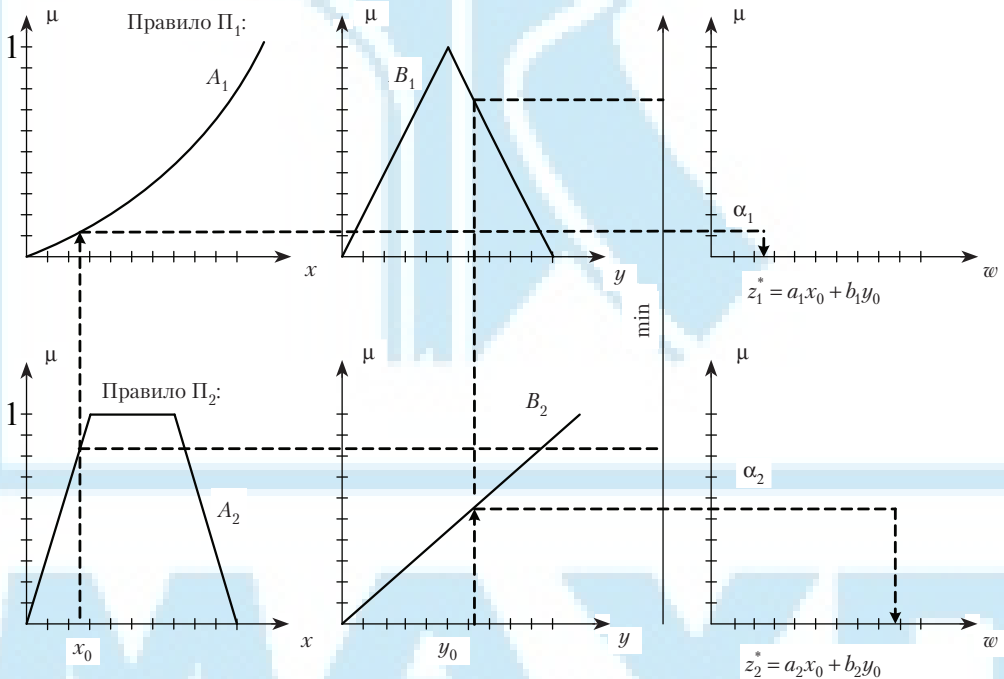


Рис. 3.6. Ілюстрація до спрощеного алгоритму Sugeno нечіткого логічного виведення

**Методи зведення до чіткості.** Вище розглядався *центроїдний метод*, який визначається залежностями (3.17) для загального випадку і (3.18) — для дискретного варіанта. Тепер наведемо ще деякі часто вживані методи.

**Метод першого максимуму.** Чітка величина виведення визначається як найменше значення, за яким досягається максимум підсумкової нечіткої множини

$$z_0 = \min \left\{ z \mid C(z) = \max_u C(u) \right\}. \quad (3.19)$$

Ілюстрацію методу першого максимуму наведено на рис. 3.7, а.

**Метод середнього максимуму.** Чітке значення за цим методом, зокрема для дискретного варіанта ( $C \in$  дискретним), знаходять за формулою (рис. 3.7, б):

$$z_0 = \frac{1}{n} \sum_{j=1}^n z_j.$$

**Метод критеріального максимуму.** За цим методом, на відміну від (3.19), чітке значення вибирається довільно серед множини елементів, для яких  $C$  сягає максимуму:

$$z_0 \in \left\{ z \mid C(z) = \max_u C(u) \right\}. \quad (3.20)$$

**Метод висотної дефаззифікації.** Для цього методу характерним є те, що елементи області визначення  $\Psi$ , для яких значення функції належності менше, ніж деякий рівень «відтинання»  $\alpha$ , не враховуються, і чітке значення розраховується за виразом:

$$z_0 = \left( \int_{C_\alpha} z \cdot C(z) dz \right) / \left( \int_{C_\alpha} C(z) dz \right), \quad (3.21)$$

де  $C_\alpha$  — нечітка множина  $\alpha$ -рівня.

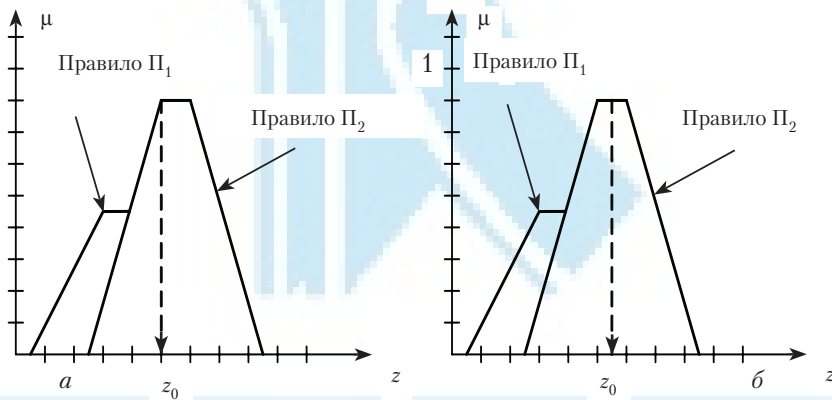


Рис. 3.7. Ілюстрація до методів зведення до чіткості:  
а — перший максимум; б — середній максимум

### 3.5. Ненадійні знання та виведення

На практиці в задачах, які розв'язуються інтелектуальними роботами, здебільшого рішення приймаються в умовах *невизначеності*. Остання може мати різну природу, тому застосовуються й різні способи роботи з невизначеними даними та значеннями: імовірнісна байєсівська логіка, нечітка логіка, багатозначна логіка, логіка умовчання, робастні методи оцінювання. Розглянемо детальніше реалізацію міркувань в умовах двох основних типів невизначеності: *випадковості* та *не-*

чіткості [75; 132], а також ознайомимося з посиланнями *логіки умовчання* та *теорії можливостей* (ця теорія має на меті розглянути з єдиних позицій модальні, недостовірні та нечіткі знання).

В усіх випадках недостовірність логічного виведення тісно пов'язана з апаратом теорії ймовірностей. Тому для розуміння цього розділу потрібно навести базові поняття з теорії ймовірностей.

**Базові посилання імовірнісної логіки.**

**Означення 3.28.** *Ймовірність події  $A$*  (позначається  $P(A)$ ) є мірою достовірності деякої події, за якої завжди виконується властивість  $0 \leq P(A) \leq 1$ . При цьому *неможлива подія* (подія, яка ніколи не може відбутися) має ймовірність 0, а ймовірність *достовірної події* дорівнюватиме 1.

Дамо більш формалізовані визначення.

Розглянемо *множину елементарних наслідків  $\Omega$* .

**Означення 3.29.** *Елементарні наслідки* — це можливі наслідки деякого експерименту, при цьому ці наслідки вважаються *рівноможливими* та *взаємовиключними*.

Тоді будь-яка подія  $A$  розглядається як підмножина множини елементарних наслідків.

**Твердження 3.2.** *Подія  $A$  відбувається, якщо існує будь-який елементарний наслідок, що входить до множини  $A$ .*

Якщо елементарний наслідок входить до множини  $A$ , кажуть, що він є *сприятливим* для події  $A$ .

**Твердження 3.3.** *Ймовірність події  $A$  визначається як відношення кількості сприятливих наслідків до загальної кількості елементарних наслідків.*

На основі цього твердження в ряді випадків можна безпосередньо розрахувати ймовірності.

Подією, *протилежною* події  $A$  (позначається  $\bar{A}$ ), називається та, яка полягає в тому, що подія  $A$  не відбувається. На основі класичного визначення ймовірностей легко переконатися в тому, що

$$P(\bar{A}) = 1 - P(A).$$

**Частотне визначення ймовірності.** Нехай проводиться серія однотипних експериментів і нехай  $K_A(n)$  — кількість випадків, в яких відбулася подія  $A$ , якщо було проведено  $n$  експериментів. Тоді

$$P(A) = \lim_{n \rightarrow \infty} \frac{K_A(n)}{n}.$$

Наприклад, якщо проводиться 1000 експериментів, а ймовірність події  $A$  дорівнює 0,3, то подія  $A$  настане приблизно в 300 випадках. При цьому зі збільшенням кількості експериментів частина випадків, в яких відбувається ця подія, щораз більше наближається до ймовірності.

**Означення 3.30.** *Ймовірність події* — це частота появи цієї події.

**Імовірнісна байєсівська логіка.** Випадкова невизначеність підпорядковується законам Байєса. Так, при функціонуванні виробничих дільниць не обійтись без відмов (виходу з ладу окремих елементів), можливих перебоїв у постачанні деталей або комплектуючих виробів. Природно, на стадії проектування та розроблення системи управління такими дільницями апіорі невідомо, який елемент і коли вийде з ладу, яких деталей і коли не вистачатиме.

Ймовірність прийняття рішення визначається за заданою сукупністю поточних даних, виходячи з апіорної ймовірності рішення (чи факту) й умовних ймовірностей, що пов'язують евристичні знання та спостереження з цими фактами.



Теорема Байєса записується у вигляді

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}, \quad (3.22)$$

де  $P(A/B)$ ,  $P(B/A)$  – умовна (або апостеріорна) ймовірність події  $A/B$  за наявності події  $B/A$ ;  $P(A)$ ,  $P(B)$  – безумовна (апріорна) ймовірність події  $A/B$  за відсутності супровідних умов.

Методи, які розроблено на основі байєсівського підходу, ґрунтуються на припущенні, що для будь-якої події існує (нехай дуже незначна) апріорна ймовірність того, що вона є істинною. Це дає змогу виконувати обчислення.

Якщо апріорно приймається ймовірність справедливості деякої гіпотези, то повинна існувати деяка сукупність даних, що підтверджують це припущення. Якби таких даних не було, то ця ймовірність завжди залишалася б незмінною. Проте за наявності пов'язаних із подією даних і відомостей апріорну ймовірність можна модифікувати для одержання вже апостеріорної ймовірності тієї самої гіпотези з урахуванням нових поточних даних.

Формула (3.22) називається **формулою Байєса** і має дуже велике значення, оскільки дає змогу обчислювати **апостеріорні** ймовірності  $P(B/A)$  «гіпотез»  $B$  за умови, що подія  $A$  відбулася через апріорні ймовірності  $P(A/B)$ . Більшість методик неточного логічного виведення так чи інакше пов'язані з формулою повної ймовірності та формулою Байєса.

**Нечітка логіка в задачах робототехніки.** Невизначеність другого типу пов'язана з поняттям нечіткості [77]. Розроблення й широке використання апарату теорії нечітких множин пов'язані з прагненням формально описувати лінгвістичні поняття, якими оперує людина в процесі прийняття рішень, а також імітувати міркування на основі тих категорій і правил, на які вона спиралася. Багато понять і правил нечіткої логіки є узагальненням або розвитком логіки предикатів.

Як вже наголошувалося, створення апарату теорії нечітких множин стало потужним інструментом при вирішенні широкого кола проблем, в яких важливе місце займають суб'єктивні знання, що важко формалізуються.

Нехай, наприклад, треба синтезувати систему керування транспортним роботом (ЕРК), який повинен зупинитися при зустрічі з перепоною недалеко від неї. Або нехай потрібно розробити алгоритм керування роботом, що відбраковує заготовки циліндричної форми, довжина яких є значно більшою (чи меншою), ніж задана. В наведених прикладах невизначеність зумовлена нечіткими поняттями «недалеко» і «значно більшою», «значно меншою». Такого роду *невизначеність називається лінгвістичною*.

Лінгвістична змінна може бути подана набором  $(X, T, U, G, M)$ , де  $X$  – назва змінної;  $T$  – терм-множина змінної  $X$ , тобто множина всіх назв лінгвістичних значень змінної  $X$  (до того ж кожне з таких значень є нечіткою змінною  $X$  із значеннями з універсальної множини  $U$  при базовій змінній  $u$ );  $G$  – синтаксичне правило, що зумовлює назву значення змінної  $X$ ;  $M$  – семантичне правило, яке ставить у відповідність кожній нечіткій змінній  $X$  зміст  $M(X)$ . Конкретна назва  $X$ , яка зумовлена синтаксичним правилом  $G$ , називається *термом*.

У свою чергу, нечітка множина  $M$  визначається як деяка множина значень  $X$  з функцією належності  $\mu_M(X)$ , що набуває значень в інтервалі  $[0, 1]$ :

$$M = \int_{x \in X} \mu_M(X) / X.$$

Для кожного терму з терм-множин, які визначено для лінгвістичних змінних, будується відповідна нечітка множина  $M$  із своїм носієм, що визначається після вивчення експертних оцінок (ЕО). При цьому під носієм нечіткої множини розуміють множину  $X'$  таку, що

$$X' = \{x / \mu_M(x) > 0, x \in X\}.$$

На наступному етапі будується набір правил, які описують, наприклад, технологічний процес. Ці правила мають вигляд:

$$\text{ЯКЩО } A = N, \text{ ТО } B = M, \quad (3.23)$$

де  $A, B$  — лінгвістичні змінні;  $N, M$  — їх терми.

На основі набору правил будується матриця нечітких відношень. Нечітке бінарне відношення  $R$  є підмножиною декартового добутку  $X \times Y$ , тобто

$$R = \int \mu_R(x, y) / (x, y), \quad (x, y) \in X \times Y. \quad (3.24)$$

Далі побудовані матриці (3.24) об'єднуються в матрицю

$$\mu_R = \max(\mu_{R_1}, \dots, \mu_{R_n}).$$

Для організації нечіткого умовного логічного виведення застосовуються композиційні правила, які охоплюють в основному три види умовних пропозицій:

$P1$ : Якщо  $x \in A$ , то  $y \in B$ ;

$P2$ : Якщо  $x \in A$ , то  $y \in B$ , інакше  $C$ ;

$P3$ : Якщо  $x_1 \in A_1$  і  $x_2 \in A_2$  і... і  $x_n \in A_n$ , то  $y \in B$ .

Концептуальною основою формалізації правил нечіткого умовного виведення є так зване правило «відділення» («Modus Ponens», див. підрозд. 2.1, вираз (2.11)), що свідчить: «якщо ( $a \rightarrow b$ ) істинне і  $a$  істинне, то  $b$  істинне». Інакше це правило називається *першою формою гіпотетичного силізму* (або *правилом виведення*).

**Означення 3.31.** Під *силізмом* розуміють дедуктивне виведення, в якому одна думка є необхідним наслідком двох інших.

У свою чергу, методологічною основою такої формалізації є запропоноване Л. Заде [30] так зване композиційне правило виведення, в якому логічними передумовами і наслідками є умовні вирази, що включають нечіткі концепції. Дещо пізніше Мамдані також сформулював своє правило виведення, яке, як і правило Л. Заде, було розроблене для умовного виразу вигляду  $P1$ . Іншими словами, розглядається нечітке умовне виведення такої форми:

Передумова 1: Якщо  $x \in A$ , то  $y \in B$ ;

Передумова 2:  $x \in A'$ ;

Наслідок:  $y \in B'$ .

(3.25)

Тут  $A$  і  $A'$  — нечіткі концепції, подані як нечіткі множини в універсумі  $U$ ;  $B$  — нечітка концепція або нечітка множина в універсумі  $V$ . Звідси  $B'$  є наслідком, поданим як нечітка множина у  $V$ . Використовуючи правило  $P1$  для отримання логічного наслідку  $B'$ , передумови 1 і 2 приводять до бінарного нечіткого відношення вигляду  $R(A_1(x), A_2(y))$  і унарного нечіткого відношення вигляду  $R(A_1(x))$ . Множини  $A_1(x)$  і  $A_2(y)$  визначаються атрибутами  $x$  і  $y$ , які набувають значення з універсумів  $U$  і  $V$  відповідно. При цьому

$$R(A_1(x)) = A, \quad (3.26)$$

а  $R(A_1(x), A_2(y))$ , згідно з правилами виведення Заде — Мамдані, визначаються так:

P1. Максимінне правило умовного виразу

$$R_m(A_1(x), A_2(y)) = (A \times B) \cup (\bar{A} \times V), \quad (3.27)$$

де  $\times$ ,  $\cup$  і  $\bar{A}$  – декартовий добуток, об'єднання та інверсія відповідно.

P2. Арифметичне правило умовної пропозиції

$$R_\alpha(A_1(x), A_2(y)) = (\bar{A} \times V) \oplus (U \times B), \quad (3.28)$$

де  $\oplus$  – операція «обмежена сума».

P3. Міні-функціональне правило умовної пропозиції

$$R_c(A_1(x), A_2(y)) = (A \times B). \quad (3.29)$$

Логічний наслідок  $R(A_2(y))$ , який подається як  $B'$  у (3.25), можна отримати з урахуванням (3.26) як

$$R(A_2(y)) = A' \circ (A \times B) \cup (\bar{A} \times V);$$

$$R(A_2(y)) = A' \circ (\bar{A} \times V) \oplus (\bar{U} \times B),$$

або

$$R(A_2(y)) = A' \circ (A \times B),$$

де  $\circ$  – операція максимінної композиції нечітких множин. Зазначимо, що з використанням цих же правил можна запропонувати правила для умовної пропозиції вигляду P2 [5], тобто

$$P4. R_4(A_1(x), A_2(x)) = [(\bar{A} \times V) \oplus (U \times B)] \cap [(A \times V) \oplus (U \times C)]; \quad (3.30)$$

$$P5. R_5(A_1(x), A_2(x)) = [(\bar{A} \times V) \cup (U \times B)] \cap [(A \times V) \cup (U \times C)]; \quad (3.31)$$

$$P6. R_6(A_1(x), A_2(x)) = (A \times B) \cup (\bar{A} \times C), \quad (3.32)$$

або правила для умовної пропозиції вигляду P3:

$$P7. R_7(A_1(x), A_2(x)) = \left[ \bigcap_{i=1, n} (\bar{A}_i \times V) \right] \oplus (U \times B); \quad (3.33)$$

$$P8. R_8(A_1(x), A_2(x)) = \left[ \bigcap_{i=1, n} (\bar{A}_i \times V) \right] \cup (U \times B); \quad (3.34)$$

$$P9. R_9(A_1(x), A_2(x)) = \left[ \bigcap_{i=1, n} \bar{A}_i \right] \times B. \quad (3.35)$$

Для з'ясування питання про те, наскільки запропоновані правила можуть служити для моделювання поведінки робота, можна використовувати деякі критерії для нечіткого умовного виведення, викладені в [30]. Ці критерії мають вигляд:

$$\text{Критерій Кр 1.} \left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \\ \text{Передумова 2: } x \in A'; \\ \text{Наслідок: } y \in B. \end{array} \right.$$

$$\text{Критерій Кр 2.} \left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \text{дуже } A; \\ \text{Наслідок: } y \in \text{дуже } B. \end{array} \right.$$

Критерій Кр 3.  $\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \text{ не дуже } A; \\ \text{Наслідок: } y \in \text{ не дуже } B. \end{array} \right.$

Критерій Кр 4.  $\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \text{ більш-менш } A; \\ \text{Наслідок: } y \in \text{ більш-менш } B. \end{array} \right.$

Критерій Кр 4-1.  $\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in \underline{A}, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \text{ дуже } \underline{A}; \\ \text{Наслідок: } y \in \text{ невідомо.} \end{array} \right.$

Критерій Кр 4-2.  $\left\{ \begin{array}{l} \text{Передумова 1: Якщо } x \in A, \text{ то } y \in B, \\ \text{Передумова 2: } x \in \overline{A}; \\ \text{Наслідок: } y \in \overline{B}. \end{array} \right.$

У [30] було також показано, що правила (3.27)–(3.29) не завжди задовольняють ці критерії.

У праці [5] було описано умови незадоволення критеріїв максимінного правила  $P1$ .

Умовою ж для вибору логіки при розробленні правил нечіткого умовного виведення є такі очевидні міркування.

Нехай нечіткі множини  $A$  з  $U$  і  $B$  з  $V$  задано у вигляді

$$A = \int_U \mu_A(u) / u, \quad B = \int_V \mu_B(v) / v.$$

Тоді, як було сказано вище, умовна пропозиція  $P1$  може бути переведена у бінарне нечітке відношення  $R(A_1(x), A_2(y))$  адаптацією операції імплікації в багатозначній логіці, тобто

$$R(A_1(x), A_2(y)) = A \times B \rightarrow U \times V = \int_{U \times V} (\mu_A(u) \rightarrow \mu_B(v)) / (u, v), \quad (3.36)$$

де значення  $\mu_A(u) \rightarrow \mu_B(v)$  визначається залежно від вибраної логіки. Далі, поклавши  $R(A_1(x)) = A$ , можна отримати логічний висновок  $R(A_2(y))$ , використовуючи критерій Кр1 для  $R(A_1(x))$  і  $R(A_1(x), A_2(y))$ . Звідси очевидно випливає (з урахуванням (3.33)):

$$\begin{aligned} R(A_2(y)) &= A \circ R(A_1(x), A_2(y)) = \int_U \mu_A(u) / u \circ \int_{U \times V} (\mu_A(u) \rightarrow \mu_B(v)) / (u, v) = \\ &= \int_{u \in U} \bigvee_{v \in V} (\mu_A(u) \wedge (\mu_A(u) \rightarrow \mu_B(v))) / (u, v). \end{aligned} \quad (3.37)$$

Для того щоб у виразі (3.37) критерій Кр 1 задовольнявся, необхідно, щоб виконувалася рівність  $R(A_2(y)) = B$  або

$$\bigvee_{u \in U} (\mu_A(u) \wedge (\mu_A(u) \rightarrow \mu_B(v))) = \mu_B(v),$$

або задовольнялася нерівність

$$\mu_A(u) \wedge (\mu_A(u) \rightarrow \mu_B(v)) \leq \mu_B(v), \quad (3.38)$$

яка спостерігається для довільних  $u \in U$  і  $v \in V$  або в термінах величин істинності  $v(P \wedge (P \rightarrow Q)) \leq v(Q)$ .

Резюмуючи сказане, сформулюємо основні дві умови, виконання яких необхідне для формалізації правил нечіткого умовного виведення, які повинні:

а) відповідати критеріям Кр 1–Кр 4; б) задовольняти нерівності (3.38).

У зв'язку з цим вважається, що формалізація удосконалених правил нечіткого умовного виведення повинна зводитися до пошуку такої багатозначної (нечіткої) логіки, яка відповідає пп. 1) і 2). Це спостерігається тільки за умови застосування  $S$ - і  $G$ -логік при використанні яких з'являється принципова можливість формалізувати модифіковані правила нечіткого умовного виведення:

$S$ -логіка (*Standart sequence*)

$$a \xrightarrow{S} b = \begin{cases} 1, & \text{якщо } a \leq b; \\ 0 & \text{у протилежному разі,} \end{cases}$$

$G$ -логіка (*Gudehian sequence*)

$$a \xrightarrow{G} b = \begin{cases} 1, & \text{якщо } a \leq b; \\ b & \text{у протилежному разі.} \end{cases}$$

Такі модифіковані правила в  $S$ - і  $G$ -логіках було запропоновано і досліджено в [30]. Зазначимо, що в цій праці було сформульовано правила тільки для умовної пропозиції вигляду  $P1$ . Ці правила мають такий вигляд:

$$1) R(A_2(y)) = R(A_1(x)) \circ R_S(A_1(x), A_2(y)).$$

Тут

$$\begin{aligned} R_S(A_1(x), A_2(y)) &= A \times B \xrightarrow{S} U \times V = \int_{U \times V} \mu_A(u) / (u, v) \xrightarrow{S} \int_{U \times V} \mu_B(v) / (u, v) = \\ &= \int_{U \times V} (\mu_A(u) \xrightarrow{S} \mu_B(v)) / (u, v), \end{aligned}$$

де

$$\mu_A(u) \xrightarrow{S} \mu_B(v) = \begin{cases} 1, & \text{якщо } \mu_A(u) \leq \mu_B(v); \\ 0 & \text{у протилежному разі.} \end{cases}$$

Це визначення ґрунтується на операції імплікації в  $S$ -логіці, тобто

$$v(P \rightarrow Q) = \begin{cases} 1, & \text{якщо } v(P) \leq v(Q); \\ 0, & \text{якщо } v(P) > v(Q). \end{cases}$$

$$2) R(A_2(y)) = R(A_1(x)) \circ R_G(A_1(x), A_2(y)),$$

$$\text{де } R_G(A_1(x), A_2(y)) = A \times V \xrightarrow{G} U \times B = \int_{U \times V} (\mu_A(u) \xrightarrow{G} (v)) / (u, v),$$

що відповідає

$$\mu_A(u) \xrightarrow{G} \mu_B(v) = \begin{cases} 1, & \text{якщо } v(P) \leq v(Q); \\ v(Q), & \text{якщо } v(P) > v(Q). \end{cases}$$

Наведені вище результати робіт у сфері формалізації правил нечіткого умовного виведення вказують на ряд істотних недоліків:

1) правила нечіткого умовного виведення, які ґрунтуються на ідеях Заде – Мамдані, не задовольняють нерівність (3.38) і критеріїв Кр 1–Кр 4;

2) модифіковані правила нечіткого умовного виведення в  $S$ - і  $G$ -логіках розроблено тільки для умовної логічної пропозиції вигляду  $P1$ ;

3) у багатозначній (нечіткій)  $S$ -логіці значення імплікації визначається за принципом класичної булевої логіки 1 або 0, що істотно обмежує можливості її застосу-

вання при побудові моделі світу роботів елементами штучного інтелекту, а також при плануванні їх поведінки, оскільки не враховує нечіткого характеру вхідних і вихідних посилань. При цьому для  $G$ -логіки при визначенні імплікації не враховується значення вхідної логічної посилки.

**Логіка умовчання.** Класична логіка виходить з того, що набір визначених в ній аксіоматичних знань є повним, і правильний висновок не зміниться навіть після додання нової аксіоми. Таке знання називається *одноманітністю*.

Проте при доданні нової аксіоми іноді можливе заперечення виведення, яке вважалося правильним у деякій системі аксіом (БЗ). Як засіб формальної обробки неповних знань, при якій необхідно здійснювати немонотонні виведення, запропоновано методи *немонотонної логіки*, що активно досліджуються. Немонотонна логіка ще повністю не вивчена, але функції значень за умовчанням уже знаходять практичне застосування, наприклад, при поданні фреймів у робототехніці.

Якщо в монотонній логіці атомарні формули зі знаком  $M$  можна записувати в довільному місці ППФ, то в логіці умовчання їх використовують тільки в посилках правил виведення.

Розглянемо формулу  $\frac{\alpha(x):M\beta(x)}{\beta(x)}$ , що означає  $\alpha(x) \wedge M\beta(x) \supset \beta(x)$ , тобто  $x \in \alpha, x \in \beta$ , якщо це не суперечить іншим значенням. Передумову правила записано над рискою й розділено знаком «:» на дві частини, де є формула без знака  $M$  і формула зі знаком  $M$ . Таким чином, вводиться обмеження на розташування формул із знаком  $M$ . При цьому можливі кілька розв'язків і виникають ситуації, коли розв'язки не можна визначити однозначно. Відповідний розв'язок розглядається як одна з можливих областей і називається розширенням (extension) із системи аксіом. Перехід від одного вибраного розв'язку до іншого вважається переходом між можливими областями. Правила виведення зі знаком  $M$  коротко називаються *правилами умовчання*.

Запис, при якому формула після знака  $M$  у передумові збігається з формулою у виведенні, називається *нормальним умовчанням*. Здебільшого значення за умовчанням можна подати в такій формі. Нормальне умовчання, в якому  $\alpha(x), \beta(y)$  не містять вільних змінних, називається *закритим нормальним умовчанням*. Якщо всі умовчання саме такі, то можна показати, що ця система знань має принаймні одне розширення (можливу область).

**Теорія можливостей.** У рамках теорії можливостей з кожним твердженням  $S$  пов'язуються дві міри: *міра можливості*  $Z(S)$  та *міра необхідності*  $U(S)$ . Ці міри, з одного боку, мають модальний характер, а з іншого — є узагальненням мір упевненості, які беруться до розгляду при неточному логічному виведенні.

Основною аксіомою теорії можливостей є *аксіома монотонності*, яка формулюється таким чином:

**Аксіома 3.1.** Якщо  $S \Rightarrow M$ , то  $Z(M) \geq Z(S)$ ;  $U(M) \geq U(S)$ . Інакше кажучи, міра можливості або необхідності висновку не може бути меншою, ніж відповідна міра передумови. У [28] наводяться такі основні властивості мір можливості та необхідності:

$$\begin{aligned} Z(S \vee M) &= \max((Z(S), Z(M))); \\ U(S \wedge M) &= \min((U(S), U(M))); \\ Z(S) &= 1 - U(\neg S). \end{aligned}$$

$Z(S) > U(S)$  (якщо твердження є *необхідно істинним*, воно повинно бути *можливо істинним*);

$$\begin{aligned} \text{Якщо } U(S) > 0, \text{ то } Z(S) &= 1. \\ \text{Якщо } U(S) < 1, \text{ то } Z(S) &= 0. \end{aligned}$$

### 3.6. Формалізація знань інтелектуалізованих робіт

**Логічні зв'язки в нечіткій логіці  $Z_1$ .** У багатьох технічних системах останнім часом використовуються моделі управління, в яких закладено механізми прийняття рішень в умовах нечіткості та невизначеності. До таких технічних об'єктів безперечно належать *інтелектуалізовані роботи* (ІР), функціонування яких у складних виробничих умовах пов'язане з моделюванням їх рефлекторної діяльності.

У підрозд. 3.3 було проаналізовано існуючі підходи до формалізації таких наближених міркувань і показано необхідність використання для цієї мети правил нечіткого умовного виведення, які ґрунтуються на реалізації операції імплікації в різних нечітких логіках. При цьому існуючі нечіткі логіки, за винятком  $S$ - і  $G$ -логік, не задовольняють класичного правила «модус-поненс», а запропоновані Л. Заде і Мамдані правила нечіткого умовного виведення, побудовані на їх основі, не задовольняють очевидних критеріїв [30]. Водночас значення істинності операції імплікації в нечіткій  $G$ -логіці рівні або 1, або 0, а при визначенні операції імплікації в  $G$ -логіці використовуються тільки значення істинності логічного висновку. Таким чином, знижується ступінь «розмитості» імплікації, що є істотним недоліком.

Тому далі розглядаються нечіткі логіки, в яких ступінь істинності операцій імплікації визначається з урахуванням ступеня істинності передумови і висновку і які дають змогу формалізувати правила нечіткого умовного виведення, що задовольняють зазначені вище критерії, при одночасному виконанні правила «модус-поненс» (2.11).

Позначимо ступінь істинності логічних посилок  $A$  і  $B$  через  $T(A) = a$  і  $T(B) = b$ . Логічні зв'язки в запропонованій нечіткій логіці наведено в табл. 3.1.

Отже, такий підхід до визначення нечітких множин відрізняється від існуючих новими операціями об'єднання  $\left(\overset{1}{\bigcup}\right)$  і перетину  $\left(\overset{1}{\bigcap}\right)$ . Зазначимо, що в цьому варіанті нечіткої множини нечіткої (багатозначної) логіки значення заперечення обчислюється за формулою:

$$T(\bar{A}) = 1 - T(A).$$

У свою чергу, зв'язки  $\overset{1}{|}$  і  $\overset{1}{\downarrow}$  виражаються як заперечення  $\overset{1}{\wedge}$ ,  $\overset{1}{\vee}$  відповідно, а суперечність і тавтологію визначено як

$$T\left(\overset{*}{A}\right) = T\left(\overset{1}{A} \overset{1}{\wedge} \bar{\overset{1}{A}}\right); \quad T\left(\overset{\cdot}{A}\right) = T\left(\overset{1}{A} \overset{1}{\vee} \bar{\overset{1}{A}}\right).$$

При цьому для цієї нечіткої логіки справедливі такі властивості:

**1. Некомутативність:**  $\overset{1}{a} \overset{1}{\vee} \overset{1}{b} \neq \overset{1}{b} \overset{1}{\vee} \overset{1}{a}$ ,  $\overset{1}{a} \overset{1}{\wedge} \overset{1}{b} \neq \overset{1}{b} \overset{1}{\wedge} \overset{1}{a}$ . Ця властивість безпосередньо впливає з рівнянь

$$\overset{1}{a} \overset{1}{\vee} \overset{1}{b} = \begin{cases} a, & a + b > 1; \\ 1, & a + b = 1; \\ b, & a + b < 1; \end{cases} \quad \overset{1}{b} \overset{1}{\vee} \overset{1}{a} = \begin{cases} b, & a + b > 1; \\ 1, & a + b = 1; \\ a, & a + b < 1. \end{cases}$$

Ця властивість дає змогу, не знаючи «фізики» процесу, на формально-логічному рівні виявити причинно-наслідкові пари того чи іншого явища, що становлять БЗ ІР. Зазначимо тільки, що вказана нечітка логіка може стати комутативною, якщо в ній використовувати інверсію вигляду

$$\bar{a} = \begin{cases} 1, & a = 0; \\ 0, & a \neq 0. \end{cases}$$

Таблиця 3.1

Логічні зв'язки в нечіткій логіці  $Z_1$

Назва зв'язки	Позначення зв'язки	Значення зв'язки
Тавтологія	$\overset{\cdot}{A}$	1
Суперечність	$\overset{\cdot}{\bar{A}}$	0
Диз'юнкція	$\overset{1}{A \vee B}$	$a, a + b > 1$ $1, a + b = 1$ $b, a + b < 1$
Заперечення	$\bar{A}$	$(1 - a)$
Кон'юнкція	$\overset{1}{A \wedge B}$	$a, a + b < 1$ $0, a + b = 1$ $b, a + b > 1$
Еквівалентність	$\overset{1}{A \leftrightarrow B}$	$1 - ka, a < b$ $1, a = b$ $1 - kb, a > b$
Штрих Шеффера	$\overset{1}{A   B}$	$1 - a, a + b < 1$ $1, a + b = 1$ $1 - b, a + b > 1$
Стрілка Пірса	$\overset{1}{A \downarrow B}$	$1 - a, a + b > 1$ $0, a + b = 1$ $1 - b, a + b < 1$

2. Відсутність асоціативності:

$$a \overset{1}{\wedge} (b \overset{1}{\wedge} c) \neq (a \overset{1}{\wedge} b) \overset{1}{\wedge} c; \quad a \overset{1}{\vee} (b \overset{1}{\vee} c) \neq (a \overset{1}{\vee} b) \overset{1}{\vee} c.$$

Покажемо це, для чого зауважимо, що

$$I. \quad b \overset{1}{\wedge} c = \begin{cases} b, & b + c < 1; \\ 0, & b + c = 1; \\ c, & b + c > 1. \end{cases}$$

Тоді

$$a \overset{1}{\wedge} \begin{cases} [b, & b + c < 1; \\ 0, & b + c = 1; \\ c, & b + c > 1 \end{cases} = \begin{cases} a \overset{1}{\wedge} b, & b + c < 1; \\ a \overset{1}{\wedge} 0, & b + c = 1; \\ a \overset{1}{\wedge} c, & b + c > 1, \end{cases}$$

$$a) \quad a \overset{1}{\wedge} b = \begin{cases} a, & a + b < 1; \\ 0, & a + b = 1; \\ b, & a + b > 1; \end{cases} \quad б) \quad a \overset{1}{\wedge} 0 = \begin{cases} a, & a \neq 1; \\ 0, & a = 1; \end{cases} \quad в) \quad a \overset{1}{\wedge} c = \begin{cases} a, & a + c < 1; \\ 0, & a + c = 1; \\ c, & a + c > 1. \end{cases}$$

$$II. \quad (a \overset{1}{\wedge} b) \overset{1}{\wedge} c = \begin{cases} [a, & a + b < 1; \\ 0, & a + b = 1; \\ b, & a + b > 1; \end{cases} \overset{1}{\wedge} c,$$



$$\text{а) } a \underset{1}{\wedge} c = \begin{cases} a, & a+c < 1; \\ 0, & a+c = 1; \\ c, & a+c > 1. \end{cases} \quad \text{б) } 0 \underset{1}{\wedge} c = \begin{cases} 0, & c < 1; \\ \equiv 0; & \\ 0, & c = 1; \end{cases} \quad \text{в) } b \underset{1}{\wedge} c = \begin{cases} b, & b+c < 1; \\ 0, & b+c = 1; \\ c, & b+c > 1. \end{cases}$$

Очевидно, що результати пп. I і II не збігаються, що й свідчить про відсутність асоціативності диз'юнкції і кон'юнкції в цій нечіткій логіці.

### 3. Часткова ідемпотентність:

$$a \underset{1}{\wedge} a = \begin{cases} a, & a \neq 0,5; \\ 0, & a = 0,5; \end{cases} \quad a \underset{1}{\vee} a = \begin{cases} 1, & a = 0,5; \\ a, & a \neq 0,5. \end{cases}$$

### 4. Відсутність дистрибутивності:

$$\left( a \underset{1}{\vee} b \right) \underset{1}{\wedge} c \neq \left( a \underset{1}{\wedge} c \right) \underset{1}{\vee} \left( b \underset{1}{\wedge} c \right), \quad \left( a \underset{1}{\wedge} b \right) \underset{1}{\vee} c \neq \left( a \underset{1}{\vee} c \right) \underset{1}{\wedge} \left( b \underset{1}{\vee} c \right).$$

### 5. Задоволення теорем де Моргана, узагальнених на випадок $M = [0, 1]$ :

$$\overline{\underset{1}{a \wedge b}} = \overline{\underset{1}{a}} \underset{1}{\vee} \overline{\underset{1}{b}}; \quad \overline{\underset{1}{a \vee b}} = \overline{\underset{1}{a}} \underset{1}{\wedge} \overline{\underset{1}{b}}.$$

Ця властивість доводиться так:

$$\overline{\underset{1}{a \wedge b}} = \text{not} \begin{cases} a, & a+b < 1; \\ 0, & a+b = 1; \\ b, & a+b > 1; \end{cases} = \begin{cases} 1-a, & a+b < 1; \\ 1, & a+b = 1; \\ 1-b, & a+b > 1. \end{cases}$$

$$\overline{\underset{1}{a \vee b}} = \begin{cases} 1-a, & \overline{a} + \overline{b} > 1; \\ 1, & \overline{a} + \overline{b} = 1; \\ 1-b, & \overline{a} + \overline{b} < 1; \end{cases} = \begin{cases} 1-a, & a+b < 1; \\ 1, & a+b = 1; \\ 1-b, & a+b > 1. \end{cases}$$

Зазначимо також, що спостерігаються три такі властивості:

$$1) a \underset{1}{\wedge} 0 = \begin{cases} a, & a < 1; \\ 0, & a = 1; \end{cases} \quad 2) a \underset{1}{\vee} 1 = \begin{cases} 1, & a \leq 0; \\ a, & a > 1; \end{cases} \quad 3) (a) = a.$$

**Логічні зв'язки та властивості нечіткої логіки  $Z_2$ .** Нижче розглядається комутативна нечітка логіка, в якій ступінь істинності операції імплікації визначається з урахуванням ступеня істинності передумови і висновку і яка дає змогу реалізувати правила нечіткого умовного виведення, що задовольняють вже згадані критерії М. Міцумото при одночасному виконанні правила «модус-поненс». Позначаючи, як і в попередньому випадку нечіткої логіки  $Z_1$ , ступінь істинності логічних посилок  $A$  і  $B$  через  $T(A) = a$  і  $T(B) = b$ , наведемо логічні зв'язки, характерні для логіки  $Z_2$  (табл. 3.2) [202].

Очевидно, що ця теорія нечітких множин відрізняється від існуючих новими операціями об'єднання  $\left( \overset{2}{\cup} \right)$  і перетину  $\left( \overset{2}{\cap} \right)$ . Значення заперечення визначається за формулою  $T(\overline{A}) = 1 - T(A)$ , а зв'язки  $\overset{2}{|}$  і  $\overset{2}{\downarrow}$  виражаються як заперечення  $\overset{2}{\wedge}$  і  $\overset{2}{\vee}$  відповідно. При цьому суперечність і тавтологія, як і раніше, мають вигляд

$$T\left(\overset{*}{A}\right) = T\left(\underset{1}{A \wedge \overline{A}}\right); \quad T\left(\overset{\cdot}{A}\right) = T\left(\overset{1}{A \vee \overline{A}}\right).$$

Логічні зв'язки в нечіткій логіці  $Z_2$

Назва зв'язки	Позначення зв'язки	Значення зв'язки
Тавтологія	$\overset{\cdot}{A}$	1
Суперечність	$\overset{*}{A}$	0
Заперечення	$\bar{A}$	$1 - a$
Диз'юнкція	$\overset{2}{A \vee B}$	1, $a + b \geq 1$ $\min(a, b), a + b < 1$
Кон'юнкція	$\overset{2}{A \wedge B}$	0, $a + b \leq 1$ $\max(a, b), a + b > 1$
Імплікація	$\overset{2}{A \rightarrow B}$	1, $a \leq b$ $\min[(1 - a), b], a > b$
Еквівалентність	$\overset{2}{A \leftrightarrow B}$	$\min[a, (1 - b)], a < b$ 1, $a = b$ $\min[(1 - a), b], a > b$
Штрих Шеффера	$\overset{2}{A   B}$	1, $a + b \leq 1$ $\min[(1 - a), (1 - b)], a + b > 1$
Стрілка Пірса	$\overset{2}{A \downarrow B}$	0, $a + b \geq 1$ $\max[(1 - a), (1 - b)], a + b < 1$

Значимо, що для цієї нечіткої логіки справедливі такі властивості:

1. **Комутативність:**

$$\overset{2}{a \vee b} = \overset{2}{b \vee a}, \quad \overset{2}{a \wedge b} = \overset{2}{b \wedge a}.$$

2. **Асоціативність:**

$$a \overset{2}{\wedge} (b \overset{2}{\wedge} c) = (a \overset{2}{\wedge} b) \overset{2}{\wedge} c; \quad a \overset{2}{\vee} (b \overset{2}{\vee} c) = (a \overset{2}{\vee} b) \overset{2}{\vee} c.$$

3. **Часткова ідемпотентність:**

$$a \overset{2}{\wedge} a = \begin{cases} 0, & a \leq 0,5; \\ a, & a > 0,5; \end{cases} \quad a \overset{2}{\vee} a = \begin{cases} 1, & a \geq 0,5; \\ a, & a < 0,5. \end{cases}$$

4. **Дистрибутивність:**

$$\left( \overset{2}{a \vee b} \right) \overset{2}{\wedge} c = \left( \overset{2}{a \wedge c} \right) \overset{2}{\vee} \left( \overset{2}{b \wedge c} \right), \quad \left( \overset{2}{a \wedge b} \right) \overset{2}{\vee} c = \left( \overset{2}{a \vee c} \right) \overset{2}{\wedge} \left( \overset{2}{b \vee c} \right).$$

Доведемо для прикладу наявність цієї властивості, тобто справедливості виразу

$$\left( \overset{2}{a \vee b} \right) \overset{2}{\wedge} c = \left( \overset{2}{a \wedge c} \right) \overset{2}{\vee} \left( \overset{2}{b \wedge c} \right):$$

$$I. \left( a \underset{2}{\vee} b \right) \overset{2}{\wedge} c = \begin{cases} 1, & a+b > 1 \\ \min(a+b), & a+b < 1 \end{cases} \overset{2}{\wedge} c = \begin{cases} 0, & 1+c \leq 1, \quad c=0; \\ \max\{\min(a,b), c\}, & \min(a,b)+c > 1. \end{cases}$$

Нехай для наочності  $\max(a,b) = a \underset{2}{\vee} b$ ;  $\min(a,b) = a \underset{2}{\wedge} b$ . Тоді отримаємо

$$\left( a \underset{2}{\vee} b \right) \overset{2}{\wedge} c = \begin{cases} 0, & c=0; \\ (a \underset{2}{\wedge} b) \underset{2}{\vee} c, & \min(a,b)+c > 1 \end{cases} = (a \underset{2}{\wedge} b) \underset{2}{\vee} c.$$

$$II. a \overset{2}{\wedge} c = \begin{cases} 0, & a+c \leq 1; \\ \max(a,c), & a+c > 1; \end{cases} b \overset{2}{\wedge} c = \begin{cases} 0, & b+c \leq 1; \\ \max(b,c), & b+c > 1, \end{cases}$$

звідки

$$\left( a \overset{2}{\wedge} c \right) \underset{2}{\vee} \left( b \overset{2}{\wedge} c \right) = \begin{cases} 1, & 0+0 \geq 1 \text{ суперечність} \\ \min(\max(a,c), \max(b,c)), & \max(a,c) + \max(b,c) < 1, \end{cases}$$

тобто, використовуючи раніше введені позначення, одержуємо

$$\begin{aligned} \left( a \overset{2}{\wedge} c \right) \underset{2}{\vee} \left( b \overset{2}{\wedge} c \right) &= (a \underset{2}{\vee} b) \underset{2}{\wedge} (c \underset{2}{\wedge} c) = (a \underset{2}{\vee} b) \underset{2}{\wedge} c = (a \underset{2}{\wedge} b) \underset{2}{\vee} \{c \underset{2}{\wedge} (a \underset{2}{\vee} b)\} \underset{2}{\vee} c = (a \underset{2}{\wedge} b) (b \underset{2}{\wedge} c) (a \underset{2}{\wedge} b) (c \underset{2}{\wedge} c) = \\ &= (a \underset{2}{\wedge} b) \underset{2}{\vee} c \underset{2}{\vee} \{(b \underset{2}{\wedge} c) \underset{2}{\vee} (a \underset{2}{\wedge} b)\} = (a \underset{2}{\wedge} b) \underset{2}{\vee} c \underset{2}{\vee} \{(a \underset{2}{\vee} b) \underset{2}{\wedge} c\} = (a \underset{2}{\wedge} b) \underset{2}{\vee} \{c \underset{2}{\vee} [(a \underset{2}{\vee} b) \underset{2}{\wedge} c]\} = \\ &= (a \underset{2}{\wedge} b) \underset{2}{\vee} \{c \underset{2}{\vee} [(a \underset{2}{\wedge} c) \underset{2}{\vee} (b \underset{2}{\wedge} c)]\} = (a \underset{2}{\wedge} b) \underset{2}{\vee} \{[c \underset{2}{\vee} (a \underset{2}{\wedge} c)] \underset{2}{\vee} [c \underset{2}{\vee} (b \underset{2}{\wedge} c)]\} = \\ &= (a \underset{2}{\wedge} b) \underset{2}{\vee} c \underset{2}{\vee} c = (a \underset{2}{\wedge} b) \underset{2}{\vee} c. \end{aligned}$$

5. **Задоволення теорем де Моргана**, узагальнених на випадок  $M = [0, 1]$ :

$$\begin{cases} \overline{\left( a \underset{2}{\wedge} b \right)} = \overline{a} \underset{2}{\vee} \overline{b}; & \left( a \underset{2}{\vee} (a \underset{2}{\wedge} b) \right) = a; \\ \overline{\left( a \underset{2}{\vee} b \right)} = \overline{a} \underset{2}{\wedge} \overline{b}; & \left( a \overset{2}{\wedge} (a \underset{2}{\vee} b) \right) = a. \end{cases}$$

Зазначимо також, що спостерігаються такі три властивості:

$$1) a \overset{2}{\wedge} 0 = 0; \quad 2) a \underset{2}{\vee} 1 = 1; \quad 3) (a) = a.$$

**Логічні зв'язки та властивості нечіткої логіки  $Z_3$ .** Розглянемо *некомутативну нечітку логіку*, в якій ступінь істинності операції імплікації визначається з урахуванням ступенів істинності передумови і висновку і яка дає змогу реалізувати правила нечіткого умовного виведення, що задовольняють критерії М. Міцумото при одночасному виконанні правила «модус-поненс».

Означивши ступені істинності логічних посилок  $A$  і  $B$  через  $T(A) = a$  і  $T(B) = b$ , в табл. 3.3 наведено основні логічні зв'язки в нечіткій логіці  $Z_3$ .

Очевидно, що ця теорія нечітких множин також відрізняється від існуючих і наведених вище новими операціями об'єднання і перетину. Зазначимо, що для відповідної теорії нечіткої множини нечіткої (багатозначної) логіки значення заперечення, штриха Шеффера, стрілки Пірса визначаються так само, як у логіках  $Z_1$  і  $Z_2$ .

Вказана нечітка логіка за своїми властивостями ідентична нечіткій логіці  $Z_1$ , тому зупинимося лише на ілюстрації основної з них — *властивості некомутативності*:

$${}_3 a \vee b = \begin{cases} 1, & a+b \geq 1; \\ \frac{b}{2-(a+b)}, & a+b < 1; \end{cases} \quad {}_3 b \vee a = \begin{cases} 1, & a+b \geq 1; \\ \frac{a}{2-(a+b)}, & a+b < 1; \end{cases}$$

звідки з очевидністю випливає, що

$${}_3 a \vee b \neq {}_3 b \vee a, \quad a+b < 1;$$

$${}_3 a \wedge b \neq {}_3 b \wedge a, \quad a+b > 1.$$

Таблиця 3.3

Логічні зв'язки в нечіткій логіці  $Z_3$

Назва зв'язки	Позначення зв'язки	Значення зв'язки
Тавтологія	$\overset{\cdot}{A}$	1
Суперечність	$\overset{\circ}{A}$	0
Заперечення	$\bar{A}$	$1-a$
Диз'юнкція	${}_3 A \vee B$	$1, \quad a+b \geq 1$ $b/[2-(a+b)], \quad a+b < 1$
Кон'юнкція	${}_3 A \wedge B$	$0, \quad a+b \leq 1$ $1-[(1-a)/(a+b)], \quad a+b > 1$
Імплікація	${}_3 A \rightarrow B$	$1, \quad a \leq b$ $b/[a+(1-b)], \quad a > b$
Еквівалентність	${}_3 A \leftrightarrow B$	$a/[b+(1-a)], \quad a < b$ $1, \quad a = b$ $b/[a+(1-b)], \quad a > b$
Штрих Шеффера	${}_3 A   B$	$1, \quad a+b \leq 1$ $(1-b)/(a+b), \quad a+b > 1$
Стрілка Пірса	${}_3 A \downarrow B$	$0, \quad a+b \geq 1$ $(1-b)/[2-(a+b)], \quad a+b < 1$

**Основні логічні зв'язки і властивості нечіткої логіки  $Z_4$ .** Тут теж розглядається комутативна нечітка логіка, в якій, як і раніше, ступінь істинності операції імплікації визначається з урахуванням ступенів істинності передумови і наслідку і яка дає змогу конструювати на її основі поліпшені правила нечіткого умовного виведення, що задовольняють наведені вище критерії М. Міцумото при одночасному виконанні правила «модус-поненс».

Так само, як і для викладення властивостей логік  $Z_1, Z_2, Z_3$ , позначимо ступені істинності логічних посилок  $A$  і  $B$  через  $T(A) = a$  і  $T(B) = b$ . Тоді основні логічні зв'язки для логіки  $Z_4$  можна подати табл. 3.4.

Зазначимо, що цей підхід нечітких множин відрізняється від існуючих і вже запропонованих новими операціями об'єднання  $\left(\overset{4}{\cup}\right)$  і перетину  $\left(\overset{4}{\cap}\right)$ . Нагадаємо, що

у відповідній теорії нечіткої множини нечіткої (багатозначної) логіки значення заперечення, штриха Шеффера і стрілки Пірса визначаються так само, як у логіках  $Z_1, Z_2$  і  $Z_3$ .

Для цієї нечіткої логіки справедливі такі властивості:

**1. Комутативність:**

$$a \underset{4}{\vee} b = b \underset{4}{\vee} a, \quad a \underset{4}{\wedge} b = b \underset{4}{\wedge} a.$$

Таблиця 3.4

Логічні зв'язки в нечіткій логіці  $Z_4$

Назва зв'язки	Позначення зв'язки	Значення зв'язки
Тавтологія	$\underset{4}{\cdot} A$	1
Суперечність	$\underset{4}{\circ} A$	0
Заперечення	$\bar{\underset{4}{A}}$	$1 - a$
Диз'юнкція	$\underset{4}{A \vee B}$	1, $a + b \geq 1$ $(1 - ka) \wedge (1 - kb), a + b < 1$
Кон'юнкція	$\underset{4}{A \wedge B}$	0, $a + b \leq 1$ $ka \vee kb, a + b > 1$
Імплікація	$\underset{4}{A \rightarrow B}$	1, $a \leq b$ $(1 - ka) \wedge (1 - kb), a > b$
Еквівалентність	$\underset{4}{A \leftrightarrow B}$	1, $a = b$ $(1 - ka) \wedge (1 - kb), a \neq b$
Штрих Шеффера	$\underset{4}{A   B}$	1, $a + b \leq 1$ $(1 - ka) \wedge (1 - kb), a + b > 1$
Стрілка Пірса	$\underset{4}{A \downarrow B}$	0, $a + b \geq 1$ $ka \vee kb, a + b < 1$

**2. Асоціативність:**

$$a \underset{4}{\wedge} (b \underset{4}{\wedge} c) = (a \underset{4}{\wedge} b) \underset{4}{\wedge} c; \quad a \underset{4}{\vee} (b \underset{4}{\vee} c) = (a \underset{4}{\vee} b) \underset{4}{\vee} c.$$

Доведемо як приклад наявність цієї властивості в нечіткій логіці  $Z_4$ :

$$(a \underset{4}{\wedge} b) \underset{4}{\wedge} c = a \underset{4}{\wedge} (b \underset{4}{\wedge} c),$$

$$I. \quad a \underset{4}{\wedge} b = \begin{cases} 0, & a + b \leq 1 \\ ka \vee kb, & a + b > 1 \end{cases} \underset{4}{\wedge} c = \begin{cases} 0 \underset{4}{\wedge} c; \\ [ka \vee kb] \underset{4}{\wedge} c; \end{cases}$$

$$a) \quad 0 \underset{4}{\wedge} c \equiv 0; \quad б) \quad [ka \vee kb] \underset{4}{\wedge} c = \begin{cases} 0, & [ka \vee kb] + c \leq 1; \\ k[ka \vee kb] \vee kc, & [ka \vee kb] + c > 1; \end{cases}$$

$$k[ka \vee kb] \vee kc = 1 - [ka \vee kb] \vee [ka \vee ab] \vee kc = ka \vee ab \vee kc.$$

$$\text{II. } b \underset{4}{\wedge} c = \begin{cases} 0, & b + c \leq 1 \\ kb \vee kc, & b + c > 1 \end{cases}, \quad a \underset{4}{\wedge} \{ \} = \begin{cases} a \underset{4}{\wedge} 0, & b + c \leq 1; \\ a \underset{4}{\wedge} [kb \vee kc], & b + c > 1; \end{cases}$$

$$\text{а) } a \underset{4}{\wedge} 0 \equiv 0; \quad \text{б) } a \underset{4}{\wedge} [kb \vee kc] = \begin{cases} 0, & a + [kb \vee kc] \leq 1; \\ ka \vee k[kb \vee kc], & a + [kb \vee kc] > 1; \end{cases}$$

$$ka \vee \{1 - [kb \vee kc]\} \vee kb \vee kc = ka \vee ab \vee kc.$$

### 3. Дистрибутивність:

$$\left( a \underset{4}{\vee} b \right) \underset{4}{\wedge} c = \left( a \underset{4}{\wedge} c \right) \underset{4}{\vee} \left( b \underset{4}{\wedge} c \right), \quad \left( a \underset{4}{\wedge} b \right) \underset{4}{\vee} c = \left( a \underset{4}{\vee} c \right) \underset{4}{\wedge} \left( b \underset{4}{\vee} c \right).$$

### 4. Задоволення теорем де Моргана, узагальнених на випадок $M = [0, 1]$ :

$$a \underset{4}{\vee} b = \overline{\overline{a \underset{4}{\wedge} b}}; \quad a \underset{4}{\wedge} b = \overline{\overline{a \underset{4}{\vee} b}}.$$

Зазначимо також, що спостерігаються такі три властивості:

$$1) \ a \underset{4}{\wedge} 0 = 0; \quad 2) \ a \underset{4}{\vee} 1 = 1; \quad 3) \ \overline{\overline{a}} = a.$$

Аналіз показує велику схожість властивостей нечіткої логіки  $Z_1$  і багатозначної КД-логіки, операція імплікації в якій має вигляд  $a \xrightarrow{\text{КД}} b = \max(1 - a, b)$ . У свою чергу, очевидна значна схожість властивостей нечітких логік  $Z_1$  і  $Z_4$ .

Повертаючись до проаналізованої нечіткої логіки  $Z_1$ , потрібно наголосити, що, на відміну від КД-логіки, вона задовольняє умову  $\mu_A x \wedge (\mu_A x \rightarrow \mu_B x) \leq \mu_B x$ , яка є необхідною при розробленні правил нечіткого умовного виведення. Ця обставина і те, що логіки  $Z_2$ – $Z_4$  також задовольняють цю нерівність, уможливило створення поліпшених правил нечіткого умовного виведення, яке детально розглядається в [5].

## Контрольні завдання

1. Розкрити зміст понять неповних, недостовірних і нечітких знань.
2. Охарактеризувати властивості і навести основні ознаки нечітких множин.
3. Розкрити зміст операцій над нечіткими множинами і пояснити особливості нечіткого логічного виведення.
4. Прокоментувати особливості використання трикутної норми і конорми.
5. Дати означення нечіткого числа та розкрити зміст понять носія та унімодальності нечіткого числа.
6. Навести спрощену та узагальнену постановку задачі нечіткого виведення.
7. Розкрити чотири етапи нечіткого логічного виведення: фаззифікацію, власне логічне виведення, композицію та дефаззифікацію (зведення до чіткості).
8. Охарактеризувати модифіковані алгоритми нечіткого виведення: Mamdani, Tsukamoto.
9. Розкрити основні методи зведення до чіткості: центроїдного, першого середнього та критеріального максимумів, висотної дефаззифікації.
10. Навести означення базових посилок імовірнісної логіки.

11. Пояснити зміст байєсівської імовірнісної логіки та розтлумачити теорему (формулу) Байєса.
12. Дати пояснення терміна лінгвістичної невизначеності.
13. Проілюструвати значення положень нечіткої логіки для задач робототехніки.
14. Охарактеризувати загальні положення теорії можливостей та розкрити зміст аксіоми монотонності.
15. Навести збіжності та відмінності в нечітких логіках  $Z_1, Z_2, Z_3, Z_4$  і прокоментувати їх переваги над правилами нечіткої логіки Л. Заде та Мамдані.



## НЕЙРО-ФАЗЗИ-ТЕХНОЛОГІЇ В СИСТЕМАХ ІЗ ШТУЧНИМ ІНТЕЛЕКТОМ

---

---

*Треба прагнути не до того,  
щоб нас будь-хто зрозумів, а до того,  
щоб нас неможливо було не зрозуміти.*

Марон Публій Вергілій

*Насамперед необхідні факти,  
а вже потім їх можна перекручувати.*

Марк Твен

### 4.1. Вихідні міркування

Різні типи інтелектуальних систем мають свої особливості, наприклад, за спроможністю до навчання, узагальнення та продукування результатів, що робить їх більш придатними до розв'язання одних класів задач і менш придатними — до інших. Зокрема, *системи з ШІ, в основі яких як СПЗ використовуються нейронні сітки*, застосовуються для розпізнавання образів, проте вони дуже незручні для пояснення того, *як вони таке розпізнавання здійснюють*. Нейросітки можуть автоматично здобувати знання, але процес їх навчання часто відбувається досить повільно, а аналіз навченої сітки дуже складний. При цьому якусь апріорну інформацію (знання експерта) для прискорення процесу її навчання в нейросітку ввести неможливо.

У *системах з нечіткою логікою*, навпаки, добре інтерпретуються отримані з їх допомогою виведення, проте *вони не можуть автоматично здобувати знання* для використання їх у механізмах виведень. Необхідність розбиття універсальних множин на окремі області, як правило, обмежує кількість вхідних змінних у таких системах несуттєвим значенням.

Теоретично системи з нечіткою логікою та штучні нейросітки подібні одна з одною, однак, згідно з викладеним вище, на практиці мають свої переваги і недоліки. Саме це й було покладено в основу створення апарату *нечітких нейронних сіток*, в яких виведення робляться на основі інструментаріїв нечіткої логіки, але відповідні функції належності піднастроюються з використанням алгоритмів навчання НС, наприклад, алгоритму зворотного поширення (підрозд. 2.6). У таких системах не тільки використовується апріорна інформація, а й існують механізми для здобуття нових знань, залишаючись при цьому логічно прозорими.

### 4.2. Принципи побудови і структура нейро-фаззі-систем

Як правило, складні об'єкти являють собою багатопараметричні системи, які характеризуються суттєво нелінійними і в ряді випадків змінними у часі процесами. Просте ПІД-регулювання на базі лінійної теорії у таких системах не завжди дає за-



довільні результати. Для застосування більш складних методів управління часто не вистачає інформації про процес і надійних математичних моделей, які адекватно його описують. Цим зокрема пояснюється той факт, що як і раніше деякі складні процеси керуються вручну досвідченими операторами. Знання про хід процесу, на які спирається досвідчений оператор, підсвідомо реалізуються ним у формі правил «ЯКЩО — ТО», що мають нечіткий інформаційний зміст. Цей же принцип використовується при автоматизації управління процесами на основі так званих *фаззи-контролерів* (ФК), які, по суті, реалізують наведені в підрозд. 3.4 чотири етапи загального логічного виведення.

Принципово узагальнену структуру фаззи-управління можна подати рис. 4.1, де зображено функціональні компоненти ФК, які й виконують процедури чотириетапного логічного виведення: *введення нечіткості — логічне виведення — композиція — зведення до чіткості*.

1. *Процедура фаззифікації параметрів процесу* полягає в переведенні поточних значень вхідних змінних ФК в лінгвістичні величини істинності. Для цього можливий числовий діапазон розглянутих параметрів процесу якісно оцінюють лінгвістичними величинами, зокрема «мало», «середньо», «багато». Кожна лінгвістична величина інтерпретується як фаззи-множина й описується функціями належності. Цим якісне висловлювання переводиться в кількісну величину в тому розумінні, що вона для кожного поточного числового значення змінної величини процесу відображає ступінь належності до тієї нечіткої підмножини, яка символізує конкретну лінгвістичну величину. Оскільки функції ФН, як правило, перекривають одна одну (див. рис. 3.3–3.6), то для тієї самої змінної процесу декілька ФН можуть відображати різні величини істинності, які відрізняються від нуля.

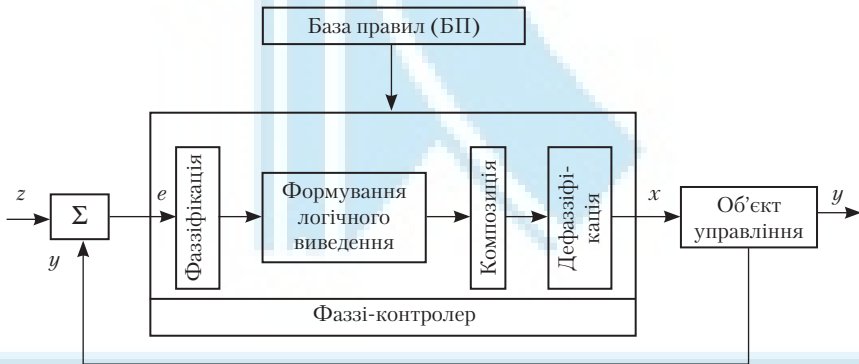


Рис. 4.1. Спрощена узагальнена структура контуру управління з фаззи-контролером:  $z$  — дія, яка задається;  $x$  — керуюча дія;  $\Sigma$  — обробка і перетворення вхідних і вимірюваних величин;  $y$  — величина, яка регулюється; БП — сукупність емпіричних знань про процес у формі правил «ЯКЩО—ТО»

2. *Формування логічного розв'язку* реалізується за допомогою лінгвістичних правил вигляду «ЯКЩО  $A$ , ТО  $B$ », які дістали назву *робочих правил*, що описують поведінку ФК відповідно до зазначених підстратегій: частина «ЯКЩО» (передумова) може при цьому означати поєднання логічних операцій будь-якої складності (див. підрозд. 2.2). Частина «ТО» (розв'язок, висновок) являє собою просте посилення на лінгвістичну величину для вихідної дії ФК. Відповідним формулюванням правил

досягається результат, при якому для будь-якої лінгвістичної величини  $x$  керуючої дії як мінімум одне з правил виявляється прийнятним. Математично як об'єднання фаззі-множини можуть бути прийняті ті правила, які визначені в процесі фаззіфікації. А це означає, що як результат обробки деякого правила з поточною зміною величини процесу ФН частини «ТО» (на виході) буде обмеженою величиною істинності, яка відповідає результуючій величині вхідної частини «ЯКЩО» цього правила.

3. *Процедура композиції* полягає в тому, що визначення величини самої частини «ЯКЩО» виконується за допомогою формування «максимуму» величини істинності сполучення «АБО» або побудовою «мінімуму» величини істинності сполучення «І». Залежно від вибору методу побудови логічного розв'язку кожне робоче правило у будь-який момент процедури оцінки результату видає обмежену або зв'язану функцію ФН частини «ТО». Сукупність усіх сформульованих правил (база правил) є ядром ФК. Перетворення правил в єдину стратегію здійснюється об'єднанням (накладанням) окремих правил з метою одержання обмеженої (мінімізованої) кількості ФН за рівняннями (3.11)–(3.13) шляхом формування «максимуму» (див. приклад 3.4 з рис. 3.3). В результаті отримують лише стільки ФН, скільки лінгвістичних значень має вихідна змінна ФК. Придатна результуюча ФН визначає собою поточний вплив бази правил. Процедура обробки бази правил з формуванням результуючої ФН являє собою логічний розв'язок для розрахунку вихідної величини  $x$  ФК. У найпоширенішому методі логічного розв'язку – «мінімаксному» – за рівнянням (3.13) на першому кроці (**min**-крок) ФН частини «ТО» кожного з правил об'єднується з величиною істинності «ЯКЩО» (**min**-операція), а на другому кроці (**max**-крок) сполучені ФН окремих правил взаємно накладаються відповідно до вимог **min-max**-операції.

4. *Дефаззіфікація* відтворює процедуру генерування керуючої дії  $x$  на виході ФК як нечіткої множини у формі ФН.

Деякі спеціальні методи дефаззіфікації описано в підрозд. 3.4.

### 4.3. Особливості формування нейро-фаззі-систем

**Основні поняття і визначення нечітких нейросіток.** Деякі фаззі-правила легко піддаються інтерпретації як локальні задачі нейронної сітки. Ряд фаззі-правил системи взаємно незалежні і можуть оброблятися паралельно. Стосовно лінгвістичних змінних, які застосовуються у правилах, а також їх структурних взаємозв'язків у нейро-фаззі-системі задається структура задачі і тим самим виконується вимога до чітко структурованої локальної нейронної сітки.

Для пояснення сутності нечітких нейросіток нагадаємо принцип формування і функціонування простої НС, що складається з одного нейрона з двома входами (див. підрозд. 2.6). Вхідні сигнали  $x$  «взаємодіють» із синаптичними вагами  $w_i$ :

$$p_i = w_i x_i, \quad i = 1, 2.$$

Ці частинні добутки підсумовуються, утворюючи значення *net* нейрона:

$$net = p_1 + p_2 = w_1 x_1 + w_2 x_2.$$

Вихід нейрона утворюється в результаті перетворення значень *net* деякою активаційною ФН  $f$ :

$$y = f(net) = f(w_1 x_1 + w_2 x_2).$$

Наведена однеїронна сітка, в якій використовуються операції добутку, підсумовування та сигмоїдна функція активації, являє собою стандартну НС.

**Означення 4.1.** *Нечіткою нейроною сіткою* називають НС з чіткими сигналами, вагами та активаційною функцією, але з об'єднанням  $x_i$  і  $w_i$ ,  $p_1$  і  $p_2$  із застосуванням операцій *T-норми* або *T-конорми*, які задовольняють властивості (3.1)–(3.8), або деяких інших неперервних операцій. Виходи, входи та ваги нечіткої НС — дійсні числа, що належать відрізьку  $[0, 1]$ .

**Нейронні сітки з фаззі-структурою.** Основною метою при формуванні нейро-фаззі-систем є перетворення фаззі-системи в еквівалентну, так звану *інтелектуалізовану НС* (ІНС) таким чином, щоб ця сітка могла ініціюватися значеннями фаззі-системи. Після навчання результат з такої НС переводиться назад у фаззі-правила, які потім аналізуються експертом щодо їхньої прийнятності. Завдяки цій процедурі ІНС може доповнювати базу фаззі-правил.

Характерною для більшості прямих перетворень нейро-фаззі-структур є заміна всіх підфункцій фаззі-системи елементами, подібними до звичайних і спеціальних (нечітких) нейронів. Загалом все це дає змогу оптимізувати параметри, використовуючи нейронні алгоритми навчання.

**Фаззіфікуючий нейрон.** Вже на етапі фаззіфікації, завданням якої є переведення вхідних змінних за допомогою ФН у лінгвістичні значення істинності, повинна враховуватися роздільність ділянок нелінійності характеристик (див., наприклад, рис. 2.18, в, д, підрозд. 2.6).

**Приклад 4.1.** Найпростіша можливість перетворення (фаззіфікація) відтворюється наведеним на рис. 4.2 прикладом, коли фаззіфікуючий нейрон (рис. 4.2, в) формується шляхом апроксимації ФН трикутної форми (рис. 4.2, а):

$$y(x) = \max(0; 1 - |x - c| / d), \quad (4.1)$$

з площею апроксимованої поверхні  $F_{\Delta} = d$  у рівну за площею функцію Гауса (рис. 4.2, б), що може бути подано так:

$$f(x) = e^{-(ax-b)^2},$$

де  $a = \pi / d$ ;  $b = c\sqrt{\pi} / d$ , а площа функції Гауса  $F_{\Gamma} = \sqrt{\pi} / a$ .

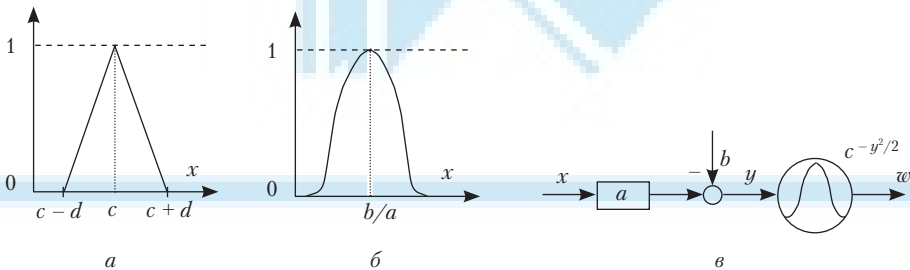


Рис. 4.2. Процес апроксимації ФН трикутної форми (а) у відповідну функцію Гауса (б) фаззіфікуючим нейроном (в)

Отже, з рис. 4.2 легко пересвідчитись, що фаззіфікуючий нейрон (рис. 4.2, в) на основі ФН (рис. 4.2, а) може бути безпосередньо параметризований. Розглянемо ще декілька прикладів елементарних нечітких НС.

**Приклад 4.2.** *Нечіткий нейрон «І»*. Сигнали  $x_i$  та ваги  $w_i$  у цьому випадку об'єднуються за допомогою *T-конорми* (див. означення 3.16–3.17):

$$p_i = S(w_i, x_i), \quad i = 1, 2,$$

а вихід утворюється із застосуванням  $T$ -норми (рис. 4.3, а):

$$y = \text{AND}(p_1, p_2) = T(p_1, p_2) = T(S(w_1, x_1), S(w_2, x_2)).$$

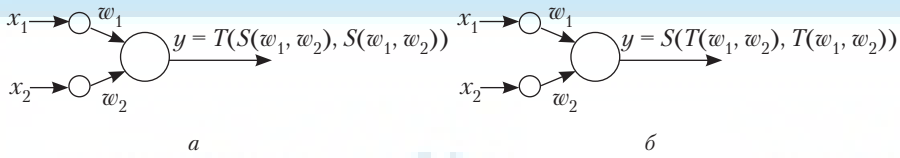


Рис. 4.3. Структури нечітких нейронів: а – «І»; б – «АБО»

Якщо прийняти  $T = \max$ ,  $S = \min$ , то нечіткий нейрон «І» реалізує композицію **min-max**:

$$y = \min \{ (w_1 \vee x_1), (w_2 \vee x_2) \}. \quad (4.2)$$

**Приклад 4.3. Нечіткий нейрон «АБО».** Сигнали  $x$  та ваги  $w$  в цьому нейроні об'єднуються за допомогою  $T$ -норми:

$$p_i = T(w_i, x_i), \quad i = 1, 2,$$

а вихід утворюється за допомогою  $T$ -конорми (рис. 4.3, б):

$$y = \text{OR}(p_1, p_2) = S(p_1, p_2) = S(T(w_1, x_1), T(w_2, x_2)).$$

Якщо прийняти  $T = \min$ ,  $S = \max$ , то нечіткий нейрон «АБО» реалізує композицію **max-min**:

$$y = \max \{ (w_1 \wedge x_1), (w_2 \wedge x_2) \}. \quad (4.3)$$

#### 4.4. Відображення системи фаззі-управління в нейросітках

Для ФН вхідних сигналів, як ми переконалися, прийнятні тільки симетричні функції трикутної форми, тоді як для вихідних сигналів – монотонні. Для логічних виведень застосовується максимінний метод, а для фаззіфікації – спрощений метод центра ваги за залежностями (3.17)–(3.18).

**Подання фаззіфікації.** Слід зазначити, що трикутне подання ФН наближено відображає поведінку рецепторів у природі, що відповідає рівнянню

$$z_i(x) = \max(0; 1 - y_i \mid x - x_i), \quad (4.4)$$

де  $x_i$  відповідає номінальній точці, у якій ФН набуває значення 1;  $y_i$  – нахил обох гілок цієї функції. При цьому  $x_i$  набуває весь заданий діапазон, а  $y_i$  – лише значення  $>0$ . Для подання усього заданого діапазону вхідної величини необхідно застосувати  $n$  таких функцій. Як свідчить досвід роботи з фаззі-системами, як правило, достатньо семи ФН на одну вхідну величину.

В роботі [203] розглянуто спеціальну структуру фаззі-нейрона, де ФН включено у ваги вхідних сигналів і вхідні величини є чіткими. При цьому рівняння (4.4) в цьому випадку можна розглядати безпосередньо як вагову функцію для чіткої вхідної величини  $x$ . Недоліком тут є те, що сама вага також являє собою функцію. Тому на рис. 4.4, а наведено більш прийнятний для оптимізації підхід, який макс-функцію за рівнянням (4.4) переводить у відображаючу функцію  $f(x) = \max(0; 1 - |\Sigma|)$ ,  $\Sigma = w_2x - w_1$ , а також реалізує розташування і форму ФН у вагах  $w_1 = yx$  і  $w_2 = y$ .

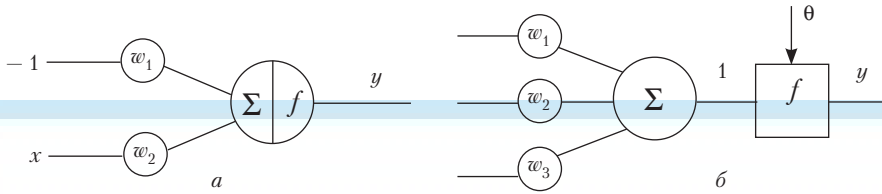


Рис. 4.4. Фаззі-нейрон для відображення фаззіфікації (а) і нейрон для формування max-min-операцій (б)

**Логічне розв’язання в нейро-фаззі-системі.** Фаззі-правило і його відповідне розв’язання, тобто зв’язок між нечіткими вхідною і вихідною множинами, є також нечіткими множинами, що являють собою сформований з цих двох множин висновок. Нехай маємо стандартне правило (див. підрозд. 3.4) «**ЯКЩО А, ТО В**» з елементами передумови А і постумови (висновку) В і відповідними ФН  $\mu_A(x)$  і  $\mu_B(y)$ . Тоді висновком буде основна множина  $A \times B$  і його можна подати у формі

$$R = \{(x, y), \mu_R(x, y), x \in A, y \in B\}. \quad (4.5)$$

Нагадаємо (див. підрозд. 3.4), що для визначення ФН  $\mu_R(x, y)$  існує декілька методів, з яких частіше застосовується **min-операція**, що відповідає логічному сполученню «І» нечітких множин:

$$\mu_R(x, y) = \min(\mu_A(x), \mu_B(y)). \quad (4.6)$$

Функція  $\mu_A(x)$  у рівнянні (4.6) задає для кожної величини  $x_i \in A$  максимум-обмеження величини ФН  $\mu_B(y)$ . Декілька правил зазвичай сполучаються за допомогою оператора «АБО» для того, щоб отримати діючу композицію усіх правил. Тому ФН композиції правил являє собою **max-утворення** ФН окремих правил (див. також приклади 3.4 і 3.5):

$$\mu_E(y) = \max(\mu_{R1}(x, y), \mu_{R2}(x, y), \dots). \quad (4.7)$$

Тоді для результуючого логічного розв’язку (виведення) за сумою правил (4.6) і (4.7) маємо:

$$\mu_E(y) = \max \min(\mu_{A_i}(x), \mu_{B_i}(y)), i = \overline{1, n}. \quad (4.8)$$

**Логічне розв’язання.** Логічне розв’язання повністю складається саме з таких аналогових **max-**, **min-**утворень. Для їх відображення на нейросітці спочатку визначаються її спеціальні тип і топологія, які здатні подати простою побудовою цю нелінійну функцію. Така схема відповідає лінійному типу НС (рис. 4.4, б), яка зазнає спеціального граничного випробування з вхідними величинами  $x_i$ , вагами  $w_i$  до цих вхідних величин, граничною величиною  $\theta$  і нелінійною функцією відображення

$$f(J) = 0,5 (\text{sign}(J - \theta) + 1) J = \begin{cases} J, & \text{якщо } J > \theta \\ 0,5J, & \text{якщо } J = \theta. \\ 0, & \text{якщо } J < \theta \end{cases} \quad (4.9)$$

За допомогою цього типу сітки з вагами  $w_1 = 1, w_2 = 1$  і  $f_2$  як такої, що лінійно відображає функції формування максимуму

$$y = y_1 + y_2 = 0,5(\text{sign}(x_1 + x_2) + 1)x_1 + 0,5(\text{sign}(x_2 - x_1) + 1)x_2 = \begin{pmatrix} \begin{cases} x_1 & x_1 > x_2 \\ 0,5x_1 & x_1 = x_2 \\ 0 & x_1 < x_2 \end{cases} + \begin{cases} 0 & x_1 > x_2 \\ 0,5x_2 & x_1 = x_2 \\ x_2 & x_1 < x_2 \end{cases} \end{pmatrix} = \max(x_1, x_2), \quad (4.10)$$

можна подати схемою (рис. 4.5, а).

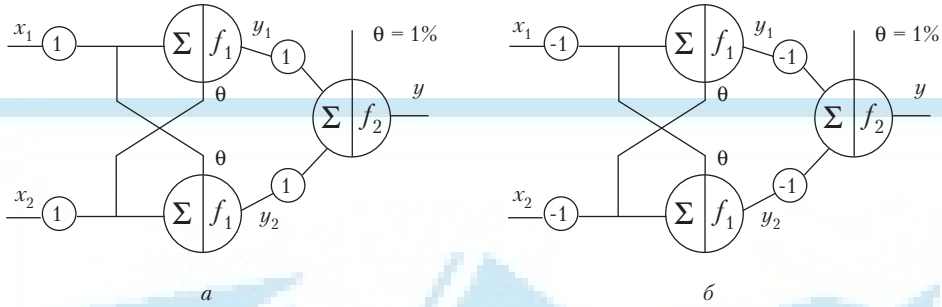


Рис. 4.5. Формування максимуму (а) і мінімуму (б) для двох функцій  $y_1$  та  $y_2$

Формування мінімуму виконується за тією ж схемою сітки, але зі зміненими вагами  $w_1 = -1, w_2 = -1$  (рис. 4.5, б):

$$y = -y_1 - y_2 = -0,5(\text{sign}(x_2 - x_1) + 1)(-x_1) - 0,5(\text{sign}(x_1 - x_2) + 1)(-x_2) =$$

$$= \begin{cases} 0 & x_1 > x_2 \\ 0,5x_1 & x_1 = x_2 \\ x_1 & x_1 < x_2 \end{cases} + \begin{cases} x_2 & x_1 > x_2 \\ 0,5x_2 & x_1 = x_2 \\ 0 & x_1 < x_2 \end{cases} = \min(x_1, x_2). \quad (4.10, a)$$

Для приглушення малих, близьких до нуля величин на вході доцільно для підсумовуючих нейронів на виході встановлювати певне граничне значення, наприклад, як це показано на рис. 4.5, б, на рівні  $\theta = 1\%$ . Це дасть змогу значення ФН, менші від 0,001, відображати точно в нуль. Це вкрай необхідно в разі зашумленості вхідних сигналів, що утруднює їх розшифрування і застосування градієнтного методу для оптимізації.

Відповідні структури для більш як двох входів можна побудувати послідовним з'єднанням (рис. 4.6, а). Зокрема, логічне заперечення деякої величини з урахуванням визначень фаззи-множин означає віднімання одиниці:

$$\bar{b} = 1 - b,$$

і формування такого доповнення може виконуватися одним нейроном з одним входом і однією опорною величиною (рис. 4.6, б).

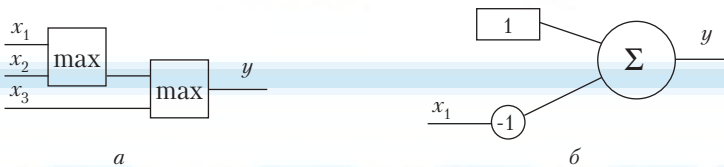


Рис. 4.6. Формування: а – максимуму при трьох вхідних сигналах; б – доповнення фаззи-параметра

Логічне розв'язання, тобто імплікація частини «ЯКЩО» до частини «ТО», легко відображається за умови, якщо для величини на виході застосувати так званий *фаззи-монотонний сигнал*. У цьому випадку кожна величина на виході фаззи-системи складатиметься з ряду дискретних значень, які при розв'язанні визначаються їх вагами. Нехай  $g$  є таким монотонним сигналом на виході з ФН  $\mu_g(y)$ , а  $\mu_w(x_1)$  – ре-

зультуючий ступінь належності частини «ЯКЩО» правила; тоді для розв'язання буде справедливим таке правило:

$$\mu_g(x, y) = (\mu_w(x_1), \mu_g(y)). \quad (4.11)$$

Отже, завдяки застосуванню фаззи-монотонного сигналу в нейронному поданні формування мінімуму вироджується в лінійну передатну функцію з підсиленням, яке дорівнює одиниці. Монотонні величини в межах подальшої дефаззифікації однозначно розподіляються, що забезпечує можливість визначення ФН  $\mu_g(y)$ .

**Відображення дефаззифікації.** Чітка величина сигналу  $z_i$  на виході при застосуванні фаззи-монотонного сигналу визначається залежностями (3.15), (3.19)–(3.21), і ця процедура детально розкривається у підрозд. 3.4. Інтерпретацію цієї процедури нейро-фаззи-компонентами, наприклад, для найпоширенішого методу розрахунку за законом центра ваги, можна подати такими міркуваннями. Скористаємося для цього видозміною залежності (3.15):

$$z_i = \sum_{j=1}^n y_{ij} x_{ij} / \sum_{j=1}^n y_{ij}, \quad (4.12)$$

де  $x_{ij}$  – керуюча дія, яка відповідає кожному фаззи-монотонному сигналу;  $y_{ij}$  – ваги, які визначаються логічним розв'язком фаззи-системи. Отже, якщо розглядати  $y_{ij}$  як ваги, то лічильник може легко відтворювати рівняння (4.12) як лінійний нейрон з входами  $x_{ij}$ . Аналогічно знаменник у виразі (4.12) можна відтворювати лінійним нейроном з входними вагами 1.

Для генерування сигналу  $Y$  на виході застосовується тип нейрона (рис. 4.7, а), який передбачає в активаційній функції керовану вагою активність. В такому нейроні поряд зі змінюваною кількістю входів тільки величини ваг  $y_{ij}$  можуть змінюватися. У результаті залежність (4.12) однозначно трансформується в його нейронне подання.

На рис. 4.7, б наведено схему НС з комплексним нейро-фаззи-перетворенням, що включає фаззифікацію, логічне розв'язання, **max-min**-композицію та дефаззифікацію. Вхід сітки утворюється трьома ФН, на виході використовуються дві ФН. Поряд з вхідним і останнім шарами присутні також три сховані шари  $h_1, h_2, h_3$ .

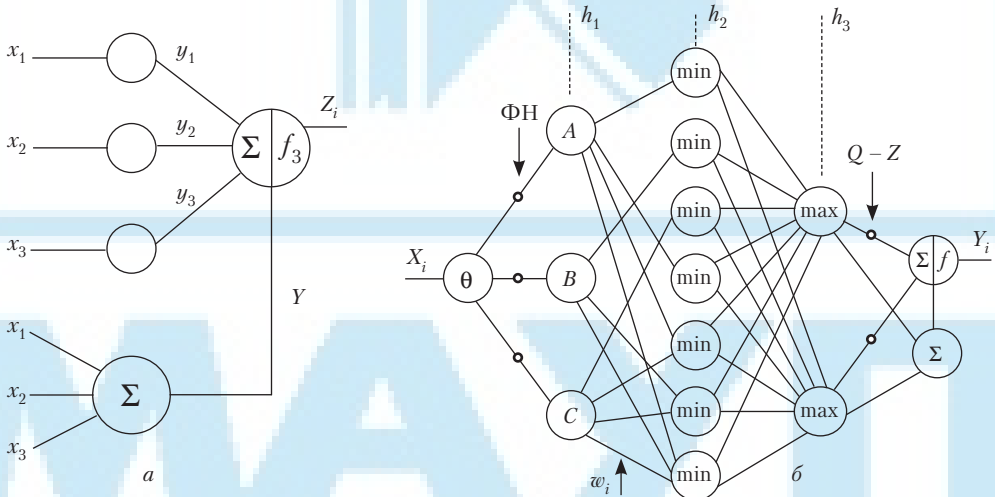


Рис. 4.7. Тип дефаззифікуючого нейрона (а) і нейросітка з комплексним нейро-фаззи-перетворенням (б)

Перший схований шар  $h_1$  з його вагами виконує *фаззифікацію*, другий шар  $h_2$  формує частину «ЯКЩО» правил, використовуючи при цьому виключно **min**-співвідношення, тоді як **max**-співвідношення між двома фаззі-вхідними величинами перекриваються застосуванням двох правил. Третій схований шар  $h_3$  формує *композиційний логічний розв'язок* за застосованими правилами. Між схованими шарами  $h_1$  і  $h_2$  вбудований настроювальний механізм для навчання сітки у вигляді регульованих ваг  $w_i$ , гранична величина  $\theta$  відіграє роль фільтруючого елемента, а збурювання у вигляді  $Q - Z$  діє на вході *дефаззифікуючого* нейрона.

Кожна з ваг схованого шару  $h_1$  подає пари взаємозалежних вільних настроювань  $(w_1, w_2)$ , які визначають параметри кожної ФН. Ваги шару  $h_2$  можуть бути двійковими (0/1) і в цьому разі вони щоразу вводять або виключають імплікацію (операцію «І»). Можуть бути також застосовані аналогові ваги між 0 і 1, які являють собою довірчі коефіцієнти імплікації.

Ваги  $Q - Z$  вихідного шару характеризують положення фаззі-монотонності у певному діапазоні сигналу на виході.

Якщо прийняти ваги шару  $h_1$  (ФН-входу) аналоговими, ваги шарів  $h_2$  і  $h_3$  двійковими, а ваги  $Q - Z$  останнього шару (фаззі-монотонності) теж аналоговими, то перетворення стає однозначним, тобто відображення буде на 100 відсотків достовірним. Це дає змогу скільки завгодно переходити від фаззі-зображення до нейро-зображення і навпаки.

Для оптимізації відомими методами навчання (див. підрозд. 2.6), наприклад, за допомогою алгоритму зворотного поширення, рекомендується використовувати аналогові ваги також і для схованих шарів  $h_2$  і  $h_3$ . Це супроводжується, однак, одним недоліком, який полягає в тому, що при зворотному перетворенні ці аналогові ваги повинні знову переводитися у двійкові.

#### 4.5. Алгоритми тренування нейро-фаззі-систем

**Особливості узгодження областей керування в інтегрованій нейро-фаззі-системі.** Фаззі-логіка і техніка НС ілюструє їх позитивні особливості в різних сферах застосувань. Перша дає змогу безпосередньо формулювати стратегії розв'язання задач при побудові систем управління процесами, тоді як НС мають бути відтреновані за введеними даними. Фаззі-системи самостійно не навчаються, і всі знання повинні вводитися в них ззовні, а спроможності НС, навпаки, за введеними даними мають бути відтреновані. Нарешті, фаззі-логіка успішно функціонує, якщо є знання у формі стратегій, необхідних для розв'язання поставленої мети, наприклад, у задачах моделювання, проектування та управління процесами в об'єктах автоматизації. Водночас за допомогою НС можуть успішно оброблятися передусім складні задачі розпізнавання та аналізу даних. Як бачимо, слабкі та сильні сторони обох цих підходів діаметрально протилежні. Тому зрозуміле мотивування до розробки *інтегрованих нейро-фаззі-систем* (ІНФС). Спряження НС з фаззі-системами можна виконувати в декількох варіантах, які визначають характер їх взаємодії. На рис. 4.8 наведено структуру ІНФС з паралельною обробкою сигналів у процесі її тренування. Ця схема відповідає випадкам, коли для фази навчання НС є дані тільки частини простору вхід/вихід (випадок часткового навчання), тому за межами гарантованої робочої області НС фаззі-система (або фаззі-контролер) має забезпечити заданий режим роботи. Як це досягається, наведено на рис. 4.8, б для векторів вхідної  $\bar{X}_{(i)}$  та вихідної  $\bar{Y}_{(i)}$  змінних.



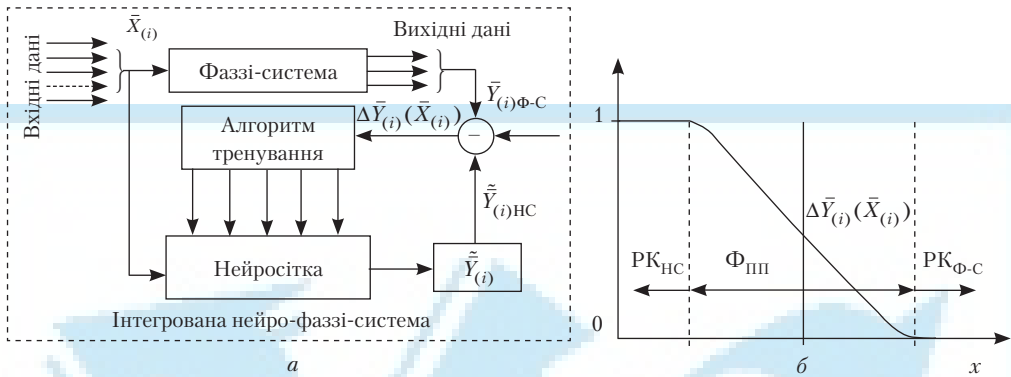


Рис. 4.8. Структура інтегрованої нейро-фаззі-системи з паралельною обробкою сигналів (а) і функція узгодження між фаззі-системою та НС областей керування (б)

Так, для плавного переходу на межу ділянку між вихідними змінними фаззі-системи і НС, як це впливає з рис. 4.8, б, з НС необхідно вивести сигнал  $\Delta \bar{Y}_{(i)}(\bar{X}_{(i)})$  достовірності, який вказує, де саме НС повинна діяти самостійно, а де достовірність її дії є ненадійною, а також де дії НС повинні бути зовсім виключені. Справді, як можна спостерігати з рис. 4.8, б, функція  $\Phi_{III}$  плавного переходу від області режиму  $PK_{НС}$  самостійної дії НС до області режиму  $PK_{\Phi-C}$  керування від фаззі-системи базується на значеннях сигналу  $\Delta \bar{Y}_{(i)}(\bar{X}_{(i)})$  достовірності керуючої дії. Нарешті, вихідний сигнал  $\tilde{\bar{Y}}_{(i)НС}$  і вихідна величина  $\bar{Y}_{(i)\Phi-C}$  фаззі-системи можуть далі підсумовуватися за принципом вагового формування середнього (див. підрозд. 4.4) для отримання ефективного значення результуючого сигналу:

$$\bar{Y}_{(i)} = \Delta \bar{Y}_{(i)}(\bar{X}_{(i)}) \Big|_{\bar{Y}_{(i)НС}} + |1 - \Delta \bar{Y}_{(i)}(\bar{X}_{(i)})| \Big|_{\bar{Y}_{(i)\Phi-C}} \quad (4.13)$$

**Означення 4.2.** Інтегрована нейро-фаззі-система – це програмно-алгоритмічна і технічна реалізація, яка характеризується за своєю суттю і функціональною спроможністю до навчання (що є характерною ознакою НС) і наочністю подання знань з елементами лінгвістичної невизначеності (характерна особливість фаззі-логіки).

Концепція, яка базується на використанні фаззі-систем із структурованими знаннями у вигляді СПЗ з можливістю подальшого навчання останніх, впливає з того, що технологічні процеси в об'єктах керування при постійному підвищенні їх складності неможливо точно математично описати, а вважається, що людина за інструкцією або з досвіду цілком спроможна складні взаємодії у ході протікання процесу як мінімум якісно оцінити та цілеспрямовано на них вплинути.

**Алгоритм тренування нечіткої НС.** Розглянемо типовий підхід до побудови алгоритмів навчання НС. Уявімо, що нечіткою НС повинно бути реалізоване невідоме відображення:

$$y^{(g)} = f(x^{(g)}) = f(x_1^{(g)}, x_2^{(g)}, \dots, x_m^{(g)}), \quad g = 1, \dots, M$$

за наявності навчаючої множини  $\left\{ \left( x^{(1)}, y^{(1)} \right), \dots, \left( x^{(M)}, y^{(M)} \right) \right\}$ .

Для моделювання невідомого відображення  $f$  використаємо спрощений алгоритм нечіткого виведення (див. підрозд. 3.4), застосовуючи таку форму запису предикатних правил:

$\Pi_i$ : ЯКЩО  $x_1 \in A_{j_1} \text{ I } x_2 \in A_{j_2} \text{ I } \dots \text{ I } x_m \in A_{j_m}$ , ТО  $y = z_j$ ,  $j = 1, \dots, k$ ,  
де  $A_{ij}$  – нечіткі числа трикутної форми;  $z$  – дійсні числа.

Ступінь істинності  $i$ -го правила визначається за допомогою операції добутку:

$$\alpha_j = \prod_{j=1}^m A_{ij} \left( x_j^{(g)} \right),$$

причому можна використовувати й будь-які інші подання для моделювання логічного оператора «I».

Вихід нечіткої системи визначається згідно з (3.18) для дискретного варіанта центроїдного методу:

$$z_0^{(g)} = \left( \sum_{j=1}^n \alpha_j z_j \right) / \sum_{j=1}^n \alpha_j.$$

Метою тренування є мінімізація похибки, що стосується кожного комплекту зразків (образів) так, щоб прийнятий комплект зразків був достатнім для виведення закону зміни ваги. Введення функції похибки для  $g$ -го поданого зразка вигляду

$$E_{(g)} = 0,5 \left( z_0^{(g)} - y^{(g)} \right)^2$$

дає змогу в подальшому, як і у звичайних НС, використовувати градієнтний метод для піднастроювання параметрів заданих предикатних правил. Зокрема, величини  $z_j$  можна коригувати за співвідношенням

$$z_j := z_j - \xi \frac{\partial E_{(g)}}{\partial z_j} = z_j - \xi \left( z_0^{(g)} - y^{(g)} \right) \frac{\alpha_j}{\alpha_1 + \alpha_2 + \dots + \alpha_n}, \quad j = 1, \dots, n,$$

де  $\xi$  – константа, що характеризує швидкість навчання.

**Процес тренування.** На *першому кроці* реалізується грубий підхід на базі фаззі-системи. Цю первинну структуру загалом можна досить швидко знайти і часто можна безпосередньо застосовувати для поєднання з процесом, що моделюється, сприяючи одержанню попередніх результатів, які, однак, ще далекі від оптимальних. Сформована таким шляхом базисна структура вже визначає всі сигнали на вході і виході, які мають застосовуватися у фаззі-структурі, і являє собою *стартовий комплект параметрів для подальшої оптимізації із застосуванням НС*.

На *другому кроці* у фазі навчання НС отримує від фаззі-системи всі сформовані нею прийнятні сигнали  $\bar{Y}_{(i)}$  процесу і керуючі сигнали  $\bar{X}_{(i)}$ , які надходять, наприклад, від людини-оператора (див. рис. 4.8, а). Вони рекурсивно спрягаються. На основі отриманої таким шляхом моделі *виконується розрахунок похибки за критерієм якості* (наприклад, формуванням середнього значення), з якого визначають, чи достатньо добре збігаються модель і дійсність. Якщо отримано задовільний збіг, подальше тренування припиняється і НС або трансформована фаззі-структура готові до роботи в активному режимі з керованим процесом.

У тих ситуаціях, які не відтреновані НС, відбувається керування процесом оператором у ручному режимі, ІНФС перестає брати участь у роботі і переходить в режим нової фази тренувань.

**Приклад 4.4.** Як приклад розглянемо більш складну нечітку систему, яка має наведену базу знань і до якої застосовується алгоритм Tsukamoto:

П1: ЯКЩО  $x_1 \in A_1 \text{ I } x_2 \in A_2 \text{ I } x_3 \in A_3$ , ТО  $y \in D$ ;

П2: ЯКЩО  $x_1 \in B_1 \text{ I } x_2 \in B_2 \text{ I } x_3 \in A_3$ , ТО  $y \in E$ ;

П3: ЯКЩО  $x_1 \in B_1 \text{ I } x_2 \in B_2 \text{ I } x_3 \in B_3$ , ТО  $y \in F$ ,

де  $x_1, x_2, x_3$  – вхідні змінні;  $y$  – вихід системи;  $A_1, A_2, A_3, B_1, B_2, B_3, D, E, F$  – деякі нечіткі множини з функціями належності сигмоїдного типу:

$$B_j(x_i) = 1 / [1 + \exp(-b_j(x_i - c_j))], \quad A_j(x_i) = 1 / [1 + \exp(b_j(x_i - c_j))], \quad j = 1, 2, 3;$$

$$D(z_1) = 1 / [1 + \exp(-b_4(z_1 - c_4 + c_5))], \quad E(z_2) = 1 / [1 + \exp(-b_4(z_2 - c_4))],$$

$$F(z_3) = 1 / [1 + \exp(b_4(z_3 - c_4))].$$

Для визначення вихідної змінної скористаємося алгоритмом Tsukamoto (ілюстративно наведено на рис. 4.9), за яким:

- підраховуються значення істинності передумов для кожного правила:

$$\alpha_1 = A_1(x_1) \wedge A_2(x_2) \wedge A_3(x_3); \quad \alpha_2 = H_1(x_1) \wedge H_2(x_2) \wedge H_3(x_3);$$

$$\alpha_3 = B_1(x_1) \wedge B_2(x_2) \wedge B_3(x_3),$$

де  $x_1, x_2, x_3$  – поточні значення входів системи;

- для кожного правила визначаються частинні виходи:

$$z_1 = D^{-1}(\alpha_1) = c_4 + c_5 + (1/b_4) \ln[(1 - \alpha_1) / \alpha_1];$$

$$z_2 = E^{-1}(\alpha_2) = c_4 + (1/b_4) \ln[(1 - \alpha_2) / \alpha_2];$$

$$z_3 = F^{-1}(\alpha_3) = c_4 + (1/b_4) \ln[(1 - \alpha_3) / \alpha_3];$$

- визначається загальний вихід системи:

$$z_0 = (\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3) / (\alpha_1 + \alpha_2 + \alpha_3).$$

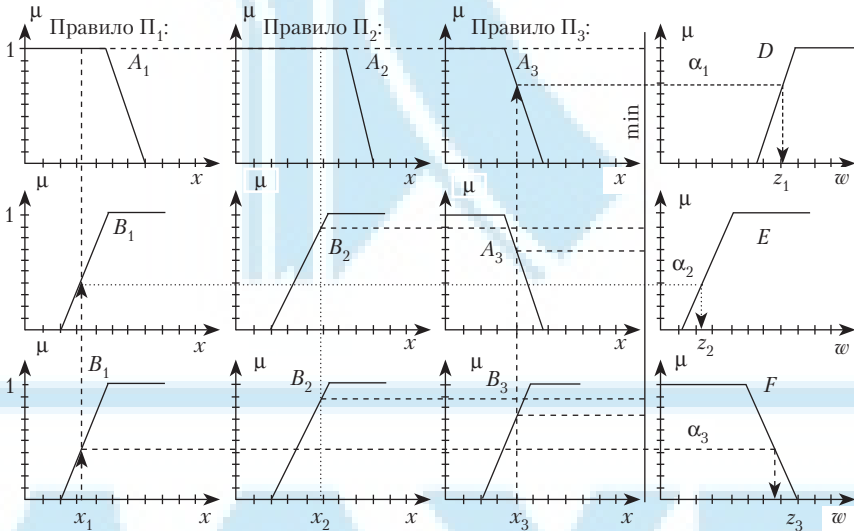


Рис. 4.9. Ілюстративне подання прикладу 4.4 нечіткої системи

Нечітку НС, яка реалізує наведений у цьому прикладі механізм тренування і виведення, подано на рис. 4.10. Зауважимо, що сітка з такою архітектурою дістала також назву *адаптивної системи з нейро-фаззі-виведеннями* (Adaptive Neuro-Fuzzy Inference System).

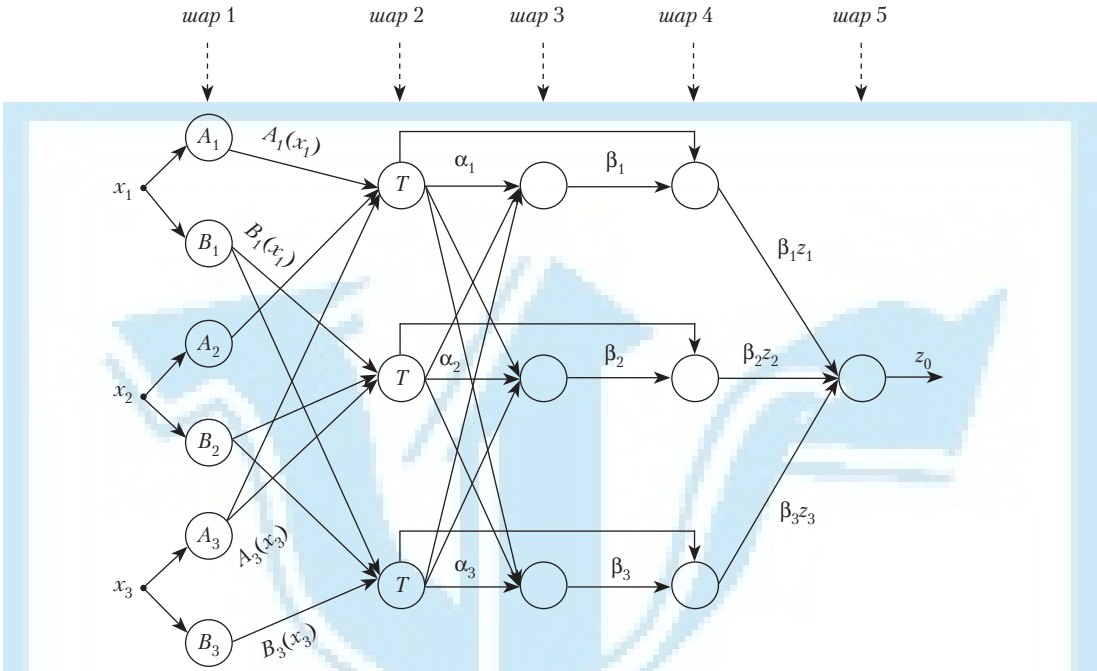


Рис. 4.10. Структура адаптивної системи з нейро-фаззі-виведеннями

Такі сітки можна описати так:

**Шап 1.** Виходи нейронів цього шару являють собою значення функцій належності при конкретних (заданих) значеннях входів.

**Шап 2.** Виходами нейронів цього шару є ступені істинності передумов кожного правила бази знань системи, які обчислюються за формулами:

$$\begin{aligned} \alpha_1 &= A_1(x_1) \wedge A_2(x_2) \wedge A_3(x_3), \\ \alpha_2 &= B_1(x_1) \wedge B_2(x_2) \wedge A_3(x_3), \\ \alpha_3 &= B_1(x_1) \wedge B_2(x_2) \wedge B_3(x_3). \end{aligned}$$

Усі нейрони цього шару позначені символом  $T$ , і це означає, що вони можуть реалізовувати довільну  $T$ -норму (див. підрозд. 3.3) для моделювання операції «І».

**Шап 3.** Нейрони цього шару обчислюють величини (операція нормування):

$$\beta_1 = \alpha_1 / (\alpha_1 + \alpha_2 + \alpha_3); \quad \beta_2 = \alpha_2 / (\alpha_1 + \alpha_2 + \alpha_3); \quad \beta_3 = \alpha_3 / (\alpha_1 + \alpha_2 + \alpha_3).$$

**Шап 4.** Нейрони цього шару виконують операції:

$$\beta_1 z_1 = \beta_1 D^{-1}(\alpha_1); \quad \beta_2 z_2 = \beta_2 E^{-1}(\alpha_2); \quad \beta_3 z_3 = \beta_3 F^{-1}(\alpha_3).$$

**Шап 5.** Єдиний нейрон цього шару обчислює вихід сітки:

$$z_0 = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3.$$

Коригування параметрів системи для функцій належності  $D, E, F$  (див. рис. 4.9, приклад 4.4) виконується відповідно до раніше розглянутих підходів. Так, наприклад, настроювання коефіцієнтів  $b_4, c_4, c_5$  для функцій належності здійснюється за формулами:

$$b_4 := b_4 - \xi \frac{\partial E(g)}{\partial b_4} = b_4 - \frac{\xi}{b_4^2} \delta(g) \frac{\alpha_1 + \alpha_2 - \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3};$$

$$c_4 := c_4 - \xi \delta \frac{\partial E_{(g)}}{\partial c_4} = c_4 + \xi \delta_{(g)} \frac{\alpha_1 + \alpha_2 + \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} = c_4 + \xi \delta_{(g)};$$

$$c_5 := c_5 - \xi \frac{\partial E_{(g)}}{\partial c_5} = c_5 + \xi \delta_{(g)} \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3},$$

де  $\delta_{(g)} = y^{(g)} - z_0^{(g)}$ .

Відповідні вирази можна отримати і для решти коефіцієнтів  $b_j$  і  $c_j$ ,  $j = 1, 2, 3$ .

Визначають два підходи до модифікації топології нечіткої НС на етапах навчання та використання. Перший, традиційний, базується на введенні додаткових продукційних правил у базу знань системи з урахуванням виконання вимоги несуперечливості її поповнення.

Другий підхід, запропонований у роботі [56], передбачає генерацію нових продукційних правил (що не суперечать правилам з бази знань системи), виходячи з аналізу експериментальних даних про об'єкт.

### Контрольні завдання

1. Розкрити функції складових в узагальненій структурі контуру управління фаззи-контролером при виконанні процедури фаззифікації параметрів процесу.
2. Пояснити принципи утворення min-max-операцій у фаззи-контролері.
3. Навести і прокоментувати означення нечіткої НС.
4. Охарактеризувати НС з фаззи-структурою і пояснити принципи функціонування фаззифікуючого нейрона при реалізації операцій «І» та «АБО».
5. Розкрити зміст логічного розв'язання в нейро-фаззи-системі.
6. Дати інтерпретацію відображення дефаззифікації у НС з комплексним нейро-фаззи-перетворенням.
7. Пояснити, що являє собою інтегрована нейро-фаззи-система.
8. Навести і прокоментувати алгоритм тренування нечіткої НС.

## РОЗПІЗНАВАННЯ ОБРАЗІВ

---

---

*Нема чого на дзеркало нарікати,  
коли «образ» кривий.*

(майже дослівний епіграф до гоголівського «Ревізора»)

*Ви можете засмучуватися від того, що у троянд є шпичаки,  
або радіти з того, що у шпичаків є троянди.*

Том Вілсон

### 5.1. Основні принципи розпізнавання

**Постановка задачі розпізнавання.** Розпізнавання образів є однією з найфундаментальніших проблем теорії інтелектуальних систем (ІС). Водночас розпізнавання образів має величезне практичне значення.

Замість терміна «розпізнавання» часто вживається інший — «класифікація». Ці два терміни у багатьох випадках розглядаються як синоніми, але не є повністю взаємозамінюваними. Кожний з них має свої сфери застосування, й інтерпретація обох термінів часто залежить від специфіки конкретної задачі. Тому обидва терміни можна вважати синонімами, якщо не буде явно обумовлено щось інше.

Наведемо деякі типові постановки задач класифікації.

1. Задача **ідентифікації**, яка полягає в тому, щоб вирізнити певний конкретний об'єкт серед схожих до нього (наприклад, розпізнати серед інших об'єктів роботи-зації або їх помітних положень у просторі конкретну топологію).

2. Задача **зарахування об'єкта до того чи іншого класу**. Це може бути, наприклад, задача розпізнавання літер або прийняття рішення про наявність дефекту у деякій технічній деталі. Зарахування об'єкта до певного класу відображає найтипівішу проблему класифікації, і коли говорять про розпізнавання образів, найчастіше мають на увазі саме цю проблему. Тому розглядатимемо її у першу чергу.

3. Задача **кластерного аналізу** полягає в поділі заданого набору об'єктів на класи — групи об'єктів, схожих між собою за тим чи іншим критерієм. Цю задачу часто називають *класифікацією без учителя*, оскільки, на відміну від задачі 2, класи апріорно не задані.

Проблеми розпізнавання легко вирішуються людьми, як правило, підсвідомо. Основна проблема побудови штучних систем розпізнавання полягає у визначенні ознак, за якими здійснюється розпізнавання. Для задач, для яких такі ознаки вдається виокремити, штучні системи розпізнавання набули значного поширення. До них зокрема належать:

- **Технічна діагностика.** На виробництві часто виникає потреба автоматизувати контроль якості деталей. Задача полягає в тому, щоб виявити, чи є деталь дефектною. Якщо ж з'ясується, що деталь має дефект, часто потрібно визначити тип цього дефекту.

- *Медицина діагностика.* Найтипівіша ситуація полягає в тому, що ті чи інші захворювання діагностуються на основі аналізу кардіограм, рентгенівських знімків і т. ін.
- *Розпізнавання літер.* Окрім усього іншого, ця проблема має велике значення для власне комп'ютерних технологій. Системи розпізнавання літер працюють разом зі *сканерами* — пристроями, які використовуються для введення у комп'ютер друкованих зображень і текстів. При введенні друкованого тексту сканер формує лише графічне зображення; для того щоб створити текстовий документ, з яким може працювати текстовий редактор, необхідно впізнати на цьому зображенні окремі літери. Аналогічно розпізнавання літер є необхідним для підтримки *пристроїв рукописного введення*. Цими пристроями часто комплектуються надпортативні комп'ютери. Основна мета цих пристроїв — замінити введення з клавіатури, що незручно для багатьох користувачів.
- *Розпізнавання мови.* Сьогодні інтенсивно розвиваються технології, пов'язані, по-перше, з голосовим керуванням комп'ютером, а по-друге — з введенням текстів з голосу.
- *Робототехніка.* Застосування методів розпізнавання в робототехніці є абсолютно природним і необхідним, оскільки роботи повинні безпосередньо сприймати зовнішній світ і, відповідно, мати пристрої машинного зору.
- *Охоронні системи.* Застосування методів розпізнавання в охоронних системах пов'язано насамперед з проблемою ідентифікації. Наприклад, потрібно ідентифікувати певну особу, щоб визначити, чи має вона право входити на територію, що охороняється. Розвиваються також системи, які вирішують проблему ідентифікації відбитків пальців і т. ін.

**Класи та їх властивості.** Одне з найвдаліших означень ключової парадигми теорії розпізнавання формулюється так: *будь-який об'єкт у природі є унікальним, всі об'єкти є типізованими.*

Зміст цієї парадигми полягає в тому, що кожний об'єкт характеризується тими чи іншими властивостями, або *ознаками* об'єкта. *Унікальність* будь-якого об'єкта означає те, що в природі не існує двох різних об'єктів, для яких збігаються абсолютно всі ознаки, і це дає змогу принаймні теоретично відрізнити один об'єкт від іншого. Але деякі ознаки різних об'єктів можуть збігатися, і це дає підстави говорити про те, що ці об'єкти належать до одного типу або класу, тобто що такі об'єкти є *типовими*.

Фундаментальні поняття «клас» та «об'єкт» неможливо повністю формалізувати. Спробуємо навести їх неформальні визначення.

**Означення 5.1.** *Об'єктом* у теорії розпізнавання прийнято називати будь-яку сутність, що існує або могла б існувати в реальному світі, а також будь-яке явище або процес.

Це дуже широке визначення, подальші уточнення можуть бути пов'язані з тим чи іншим звуженням нашого розуміння про те, що саме слід вважати об'єктом.

**Означення 5.2.** *Класом* у теорії розпізнавання образів прийнято називати сукупність об'єктів, які мають ті чи інші спільні ознаки.

**Означення 5.3.** Ознаки, що дають можливість відрізнити представників одного класу від іншого, прийнято називати *інформативними ознаками*.

**Означення 5.4.** Ознаки, спільні для всіх представників класу, називатимемо *інваріантами класу*.

Інколи, щоб уникнути непорозумінь, ми називатимемо об'єкти і класи реального світу відповідно *P-об'єктами* і *P-класами*. Зрозуміло, що *P-об'єкт* може належати будь-якій кількості *P-класів*.

Часто *P-об'єкти* називають *реалізаціями*, або *зразками P-класів*.

Можна виокремити основні властивості класів, які є твердженнями.

**Твердження 5.1.** *Усі представники класу мають певний набір спільних ознак (впливає з визначення 5.2).*

**Твердження 5.2.** *Різні об'єкти, що належать до одного класу, можуть бути не схожими між собою. Вони повинні мати спільні інваріантні ознаки, але всі інші ознаки можуть як завгодно варіюватися.*

**Твердження 5.3.** *Один і той самий об'єкт може змінюватися з часом і навіть поступово переходити від одного класу до іншого (наприклад, перетворення заготовки в процесі обробки на деталь). Це свідчить про те, що розпізнати чіткі межі класу часто неможливо.*

**Твердження 5.4.** *Ознайомлення з деякою скінченною кількістю представників певного класу дає можливість впізнавати інших представників цього класу.*

Загалом властивість, яка впливає з твердження 5.4, може не виконуватися. Але якщо ця властивість виконується (принаймні теоретично), це є дуже сильним і важливим твердженням. Воно по суті означає можливість навчатися на прикладах, тобто на основі спостереження певної кількості прикладів (можливо, разом з контр-прикладом) сформулювати *правило розпізнавання* (або *правило виведення*), яке дає змогу відрізнити представників цього класу від представників іншого (можливо, з певною достовірністю, тобто з певним процентом помилок). У деяких випадках правилом розпізнавання може бути предикат, який залежить від інформативних або інваріантних ознак, інколи правило розпізнавання реалізується у вигляді деякої складної процедури. Якщо ж навчання на прикладах неможливе або неефективне, правило розпізнавання інколи можна задати явно. Якщо цього не вдається, єдиною можливістю для надійного розпізнавання залишається запам'ятовування всіх можливих представників цього класу.

**Модельні описи класів. Розпізнавання як зіставлення.** Інтелектуальна система не в змозі сприймати реальні об'єкти (*P-об'єкти*) безпосередньо. Вона сприймає їх за допомогою органів чуття, фізичних датчиків, різноманітних технічних пристроїв і т. ін. При цьому формуються первинні описи об'єктів — *C-об'єкти*.

**Означення 5.5.** *C-об'єкт* — первинний опис *P-об'єкта* у тому вигляді, в якому він сприймається ІС.

Далі вважатимемо, що ІС може сприймати лише окремих представників класу, а не весь клас загалом. Відповідно до цього поняття «*C-класу*» далі не розглядається.

*C-об'єкти* можна було б називати образами, але в літературі термін «образ» часто вживається як синонім терміна «клас». *C-об'єкти* часто (хоча й не завжди) є функціями зміни тих чи інших фізичних величин. Наприклад, при розпізнаванні зображень первинним описом може бути функція яскравості зображення в різних точках.

Процедури сприйняття можуть послаблювати або посилювати природну змінюваність *P-класів* і *P-об'єктів*. Отже, *C-об'єкти* як результати спостереження *P-об'єктів* істотно залежать від умов спостереження. Одному і тому самому *P-об'єкту* можуть відповідати різні *C-об'єкти*. Так, деякий предмет може сприйматися по-різному, якщо його спостерігати під різними кутами і з різної відстані (фреймами). На формування *C-об'єктів* значно впливають зовнішні шуми.



З іншого боку, і різним  $P$ -об'єктам може відповідати один і той самий  $C$ -об'єкт. Таке можливо, наприклад, якщо два  $P$ -об'єкти належать до одного класу і з високим ступенем точності вимірюються лише інваріантні ознаки цього класу. Тоді  $C$ -об'єктами є набори таких ознак, і природно, що вони збігаються.

Як відомо, ІС завжди зіставляє первинні описи зі своїми знаннями. Відповідно до цього *в основі розпізнавання лежить метапроцедура зіставлення зі зразком* (згадаймо хоча б, яким чином виконувалося виведення в системах знань на семантичних сітках у підрозд. 2.3 або як виконувалася процедура перевірки на адекватність екзофрейму описуваному образу-об'єктові у підрозд. 2.4). В результаті такого зіставлення формується вторинний опис об'єкта — *М-об'єкт* (модельний опис, модель об'єкта).

**Означення 5.6.** *М-об'єкт* — це вторинний опис  $P$ -об'єкта у тому вигляді, в якому він сформувався в результаті зіставлення за допомогою ІС з наявними описами в її БЗ.

Модельним описом може бути просто набір інформативних ознак або складніша структура, наприклад, семантична сітка.

У пам'яті системи зберігаються також модельні описи різних класів — *М-класи*.

**Означення 5.7.** *М-класи* — сукупність вторинних описів  $P$ -об'єктів, які зберігаються в структурованій за інформативними ознаками та інваріантами класу БЗ ІС.

$M$ -класи багато в чому аналогічні фреймам класів, а  $M$ -об'єкти — екзофреймам (див. підрозд. 2.4. Тут доречно нагадати, що в термінології об'єктно-орієнтованого програмування фрейми називаються саме класами).

Тепер можна сформулювати загальний принцип розпізнавання.

**Лема 5.1.** *С-об'єкти, що спостерігаються ІС, або відповідні їм М-об'єкти (модельні описи об'єктів), що формуються в процесі роботи, зіставляються з М-класами (модельними описами класів), і ІС приймає рішення про віднесення об'єкта до того класу, зіставлення з модельним описом якого було найбільш успішним.*

**Постановка задачі й основні режими розпізнавання.** Тут і далі розглядається розпізнавання як задача віднесення об'єкта до того чи іншого класу. Нехай є  $m$  класів. Задача розпізнавання полягає у віднесенні об'єкта, який спостерігається системою, до одного з них.

Виокремлюють три режими функціонування системи розпізнавання: *робочий режим, навчання та екзамен*.

У **робочому режимі** система розпізнавання повинна прийняти рішення про віднесення об'єкта, який спостерігається системою, до того чи іншого класу. Критерії, які використовуються для прийняття такого рішення, і власне процедура прийняття рішення утворюють *правило розпізнавання*.

Система розпізнавання повинна насамперед *сприйняти реальний об'єкт* ( $P$ -об'єкт). У найпростіших випадках датчики просто здійснюють *дискретизацію*, але за допомогою спеціально сконструйованих датчиків можуть вимірюватися й інші характерні ознаки.

Загальну схему розпізнавання в робочому режимі можна зобразити структурою, поданою на рис. 5.1.

*Блок попередньої обробки* здійснює необхідну попередню обробку первинного опису ( $C$ -об'єкта). Основне його завдання — зменшення рівня шумів, що виникають внаслідок похибок вимірювання, впливу сторонніх факторів, недостатньої якості вимірювальної апаратури і т. ін.

Блок формування модельного опису формує модельний опис об'єкта ( $M$ -об'єкта). Як уже зазначалося, модельним описом може бути просто набір інформативних ознак або ж структура об'єкта.

Нарешті, блок класифікації (класифікатор) застосовує до модельного опису правило розпізнавання і на основі цього виробляє рішення про належність об'єкта до того чи іншого класу.

Перед функціонуванням системи розпізнавання в робочому режимі часто буває доцільно здійснити **навчання** цієї системи. До основних цілей навчання можна зарахувати формування правила розпізнавання, а також вибір множини інформативних ознак.

Навчання здійснюється на основі навчальної вибірки.

**Означення 5.8.** Навчальною вибіркою називається деяка множина об'єктів, про кожний з яких відомо, до якого класу він належить. Як правило, до навчальної вибірки включають достатню кількість представників кожного класу.

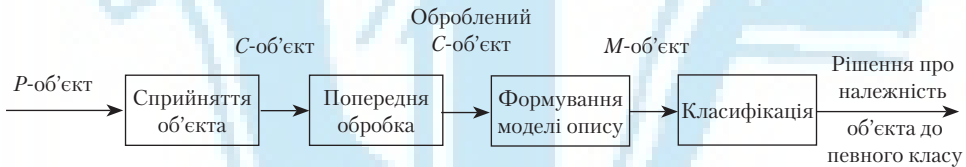


Рис. 5.1. Типова схема розпізнавання

Після завершення навчання доцільно провести **екзамен**, метою якого є оцінка якості навчання, передусім оцінка надійності сформованого правила розпізнавання. Екзамен здійснюється на основі екзаменаційної вибірки (певний набір об'єктів, про які відомо, до якого класу вони належать). Здійснюється розпізнавання об'єктів екзаменаційної вибірки, і на основі порівняння відповідей з правильними робиться висновок про якість навчання і можливість переходу до робочого режиму.

Можна виокремити такі типові схеми роботи системи розпізнавання:

- *схема без навчання*, за якої навчання не здійснюється, а правило розпізнавання жорстко задається під час проектування і розробки системи;
- *схема з одноразовим навчанням*, за якої правило розпізнавання виробляється один раз і надалі не змінюється;
- *схема з періодичним перенавчанням*, за якої навчання періодично повторюється.

При цьому навчальна вибірка може поповнюватися за рахунок об'єктів, які розпізнавалися в робочому режимі.

**Розпізнавання як прийняття рішень.** Задачу розпізнавання образів можна подавати як задачу прийняття рішень. Справді, розпізнавання і є прийняття рішення про те, до якого класу належить той чи інший об'єкт.

Насамперед розглянемо розпізнавання як деяку «гру з природою». Нехай є  $m$  класів і потрібно здійснити розпізнавання деякого об'єкта. Система розпізнавання має вибір між  $m$  стратегіями:  $i$ -та стратегія означає прийняття рішення про те, що об'єкт належить до  $j$ -го класу. Природа також має одну з  $m$  стратегій:  $j$ -та стратегія означає те, що насправді об'єкт належить до  $j$ -го класу.

Маємо матрицю ризиків  $R = (r_{ij}, i, j = 1, \dots, m)$ ;  $r_{ij}$  — втрати від віднесення об'єкта до  $i$ -го класу, тоді як насправді він належить до  $j$ -го.

Очевидно, метою системи розпізнавання є мінімізація середнього ризику.

Інтуїтивно зрозумілими є такі властивості матриці ризиків:

$$r_{ij} \leq 0 \text{ при } i = j; r_{ij} \geq 0 \text{ при } i \neq j.$$

Матриця ризиків, як правило, асиметрична. Справді, розглянемо такий приклад. Нехай на виробництві діє система автоматизованого контролю якості продукції роботизованої системи. Є два класи: деталі без дефектів і дефектні. Якщо деталь приймається за справну, її використовують, якщо ж ні — її викидають. Очевидно, що використання далі в машині некондиційної деталі може бути значно гіршим, аніж викинути справну.

**Класифікація основних методів розпізнавання.** Насамперед потрібно згадати прості спеціалізовані методи розпізнавання на основі фіксованих наборів еталонних ознак. Самі об'єкти при цьому вважаються порівняно незмінними. Кожен з цих методів розрахований на окрему специфічну задачу.

Існує декілька груп таких методів. Для прикладу згадаємо один з них — *метод зіставлення з еталоном*. Основний принцип полягає в порівнянні зображення з набором попередньо сформованих еталонів. Порівняння здійснюється з використанням спеціальних технічних пристроїв. Можуть застосовуватися, зокрема, фотомаски, матриці електричних елементів і т. ін. Наприклад, при використанні фотомасок зображення безпосередньо накладається на еталонні маски, набір яких охоплює всю множину еталонів. Міра збігу визначається за допомогою фотоелемента, розташованого за маскою.

Змінюваність реалізацій класів потребує розвитку більш універсальних методів. Основна відмінність між різними групами методів розпізнавання полягає в тому, *яким чином формуються модельні описи об'єктів*.

Найбільш вживаною є класифікація, згідно з якою методи розпізнавання образів поділяються на *дискримінантні* (часто вони називаються розпізнаванням у просторі ознак) і *структурні*. При використанні дискримінантних методів розпізнавання кожен об'єкт описується точкою у просторі ознак, тобто координатами цієї точки є значення ознак об'єкта. Відповідно до цього основною метою навчання є поділ усього простору на зони, кожна з яких відповідає певному класу.

Дискримінантні методи розпізнавання поділяються на *імовірнісні* та *детерміністські*.

При використанні *імовірнісних методів* кожен об'єкт описується вектором, який розглядається як реалізація деякого випадкового вектора. Якщо відомі імовірнісні характеристики розподілу випадкових векторів, то можна визначити ймовірності, з якими об'єкт може належати до того чи іншого класу, і на основі цього прийняти рішення про класифікацію. В основі імовірнісних методів розпізнавання лежить прийняття рішень на основі *байєсівського підходу*.

При використанні *детерміністських методів* жодних імовірнісних припущень не робиться, а рішення приймається на основі аналізу геометричних мір близькості між різними точками.

Структурні методи беруть до розгляду структуру об'єктів, які розпізнаються. Вони поділяються на *синтаксичні* (інша назва — *лінгвістичні*) і *логічні*.

При використанні *синтаксичних методів* кожен клас описується формальною мовою, породженою деякою формальною граматикую. Об'єкт, що розпізнається, описується певною фразою, і головний принцип розпізнавання полягає в тому, щоб визначити, до якої мови належить отримана фраза.

Нарешті, *логічні методи* розпізнавання ґрунтуються на аналізі комбінацій характерних ознак. Кожен клас описується деяким предикатом, і об'єкт належить до класу, якщо відповідний предикат набуває значення «істина». Використання логічних методів розпізнавання є, по суті, реалізацією деякої експертної системи.

Можна розглянути й іншу класифікацію методів розпізнавання, яка враховує, чи є помилки на навчальній вибірці. Використання деяких методів гарантує, що в робочому режимі всі об'єкти навчальної вибірки розпізнаватимуться безпомилково. Для інших методів це не відбувається.

## 5.2. Розпізнавання в просторі ознак

**Загальні особливості дискримінантного розпізнавання.** Як уже зазначалося, для дискримінантного розпізнавання, або для розпізнавання в просторі ознак, характерно те, що кожен об'єкт зображається окремою точкою в деякому просторі. Координатними осями цього простору виступають ознаки, за якими здійснюється розпізнавання. Таким чином, координатами об'єктів виступають значення відповідних ознак. Далі розпізнавання здійснюється на основі аналізу мір близькості між об'єктами.

**Типи ознак, міри відстаней.** Виокремлюють різні типи ознак: *дихотомічні* (ознака може бути присутня або відсутня (наприклад, наскрізний отвір у деталі є або його немає)); *номінальні* (наприклад, матеріал деталі: бронза, латунь, сталь і т. ін.); *порядкові* (наприклад, розмір деталі: габаритна — середніх розмірів — мініатюрна); *кількісні*. Для кожного типу ознак можна вводити свої міри відстані між об'єктами.

Для спрощення подання матеріалу обмежимося лише кількісними ознаками, які можуть мати числові значення. Для таких ознак, як міри близькості між об'єктами, найчастіше беруть *звичайні евклідові відстані* або їх квадрати. Евклідова відстань в  $n$ -вимірному метричному просторі між векторами  $a = (a_1, \dots, a_n)$  та  $b = (b_1, \dots, b_n)$  обчислюється за формулою

$$D(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (5.1)$$

Якщо різні ознаки повинні враховуватися різною мірою, кожній ознаці приписується вага  $w_i$  і використовується *зважена евклідова відстань*:

$$D(a, b) = \sqrt{\sum_{i=1}^n w_i (a_i - b_i)^2}. \quad (5.2)$$

Існують спеціальні методики, які дають змогу здійснювати перехід від не кількісних типів ознак до кількісних.

**Вектори та матриці ознак.** Будь-який об'єкт при застосуванні дискримінантних методів розпізнавання, як ми пересвідчилися, зображається вектором ознак у деякому  $n$ -вимірному просторі, де  $n$  — кількість ознак. Таким чином, будь-якому об'єкту відповідає  $n$ -вектор:

$$x = (x_j; j = 1, \dots, n), \quad (5.3)$$

де  $x_j$  — значення  $j$ -ї ознаки.

*Навчальну вибірку* прийнято позначати у вигляді спеціальної матриці, яка часто називається *матрицею даних*:

$$Y = (y_{ij}; i = 1, \dots, q; j = 1, \dots, n + 1), \quad (5.4)$$

де  $q$  — загальна кількість представників усіх класів;  $n$  — кількість ознак;  $y_{ij}$ ,  $i = 1, \dots, q$  — значення  $j$ -ї ознаки для  $i$ -го об'єкта;  $y_{i, n+1}$  — клас, до якого належить  $i$ -й об'єкт.

**Гіпотеза компактності.** Усі дискримінантні методи так чи інакше спираються на *гіпотезу компактності* [13]. Відповідно до цієї гіпотези класу відповідає компактна множина точок у деякому просторі ознак.

Слово «компактний» тут вживається у дещо незвичному значенні. Компактною у цьому розумінні називається множина точок, для якої:

- кількість граничних точок мала порівняно із загальною кількістю точок;
- будь-які дві внутрішні точки можуть бути з'єднані плавною лінією так, щоб ця лінія проходила лише через точки цієї ж множини;
- майже будь-яка внутрішня точка має в досить великому околі лише точки цієї ж множини.

Важливість належного вибору простору ознак, в якому класам відповідають компактні множини точок, можна проілюструвати такими прикладами. На всіх наведених рисунках представники одного класу позначені кружечками, а іншого — зірочками.

**Приклад 5.1** — простір з *лінійною роздільністю* ознак класів.

У цьому випадку ознаки обрано дуже вдало. Класи добре відокремлені один від одного. До того ж можна провести пряму лінію так, що всі представники першого класу розміщуватимуться по один бік цієї лінії, а всі представники другого класу — по інший бік (рис. 5.2, а). У такому разі кажуть, що класи є лінійно роздільними.

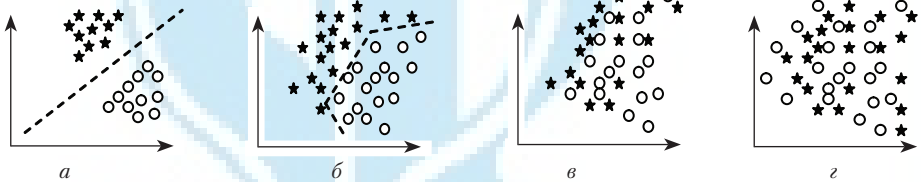


Рис. 5.2. Ілюстрація роздільності ознак класів: а — лінійна роздільність; б — складна роздільність; в — з частковим перекриттям; г — з повним перекриттям

Лінійна роздільність дуже спрощує задачу розпізнавання. Правило розпізнавання на основі лінійних роздільних функцій дуже просте і не потребує великих затрат часу і пам'яті. Роздільну пряму лінію дуже легко побудувати (деякі алгоритми обговорюватимуться в підрозд. 26.6).

**Приклад 5.2** — простір із *складною роздільністю* ознак класів.

У прикладі, наведеному на рис. 5.2, б, лінійної роздільності немає. Але все-таки можна побудувати лінію, хоча й непряму, яка відокремлює один клас від іншого. Процедури навчання і розпізнавання в цьому разі суттєво ускладнюються.

**Приклад 5.3** — простір з *частковим перекриттям* ознак класів.

У разі часткового перекриття ознак класів (рис. 5.2, в) надійно відокремити один клас від іншого не вдається, і це значно збільшує кількість помилок при розпізнаванні.

**Приклад 5.4** — простір з *повним перекриттям* ознак класів.

Про будь-яке розпізнавання в просторі з повним перекриттям ознак (рис. 5.2, г) говорити не доводиться.

**Типова схема розпізнавання в просторі ознак.** Схема розпізнавання в просторі ознак повністю відповідає загальній схемі, наведеній у підрозд. 5.1. В цьому випадку «модельним описом» об'єкта є його вектор ознак, і тому «формування модельного опису об'єкта» є не що інше, як виокремлення ознак, отримання їх числових значень. Типову схему розпізнавання в просторі ознак наведено на рис. 5.3.

Виокремлення ознак не слід змішувати із задачею *вибору інформативних ознак*, за якими потрібно здійснювати класифікацію. Такий вибір повинен відбуватися на етапі навчання. На етапі розпізнавання просто визначаються конкретні значення ознак для всіх об'єктів, що розпізнаються.

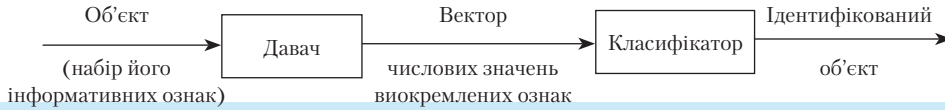


Рис. 5.3. Розпізнавання в просторі ознак

Як бачимо, всі ці етапи тісно пов'язані між собою. Уже з цього прикладу видно, що ідеальний класифікатор не потребує виокремлення ознак, і навпаки, при ідеальному виокремленні ознак робота класифікатора максимально спрощується. Основну проблему можна сформулювати так: *реальні методи отримання первинної інформації та доступні алгоритми виокремлення ознак часто не дають змоги отримати саме ті ознаки, які насправді відрізняють представників одного класу від представників іншого.*

**Роздільні функції як основа розпізнавання.** Для простоти розглядатимемо випадок двох класів; методологія, що буде описана, легко поширюється на більш загальні випадки.

Отже, нехай маємо два класи:  $K_1$  і  $K_2$ . Нехай довільний вектор  $x$  описується своїми координатами:  $x = (x_1, \dots, x_n)$ .

**Означення 5.9.** Функція  $g(x)$  розділяє класи  $K_1$  і  $K_2$ , або є *роздільною функцією*, якщо для всіх векторів класу  $K_1$  функція  $g(x) > 0$ , а для всіх векторів класу  $K_2$  функція  $g(x) < 0$ .

Побудова і використання роздільних функцій є основою багатьох методів розпізнавання. Насправді в робочому режимі ми, як правило, не можемо знати заздалегідь, до якого класу належить певний об'єкт (саме це й потрібно визначити). Натомість, якщо у нас є деяка роздільна функція, що є наближенням до справжньої роздільної функції, на основі її аналізу ми можемо прийняти певне рішення, а саме: рішення про належність вектора  $x$  до класу  $K_1$ , якщо  $g(x) > 0$ , і класу  $K_2$ , якщо  $g(x) < 0$ . Це рішення може виявитися помилковим. Отже, метою навчання має бути отримання справжньої роздільної функції, якщо вона існує, або побудова деякої наближеної роздільної функції в іншому випадку. Саме такий зміст надалі вкладатиметься в поняття «роздільна функція».

**Лінійні роздільні функції.** Як уже зазначалося, особливо важливим є випадок, коли роздільна функція є *лінійною*, тобто є рівнянням гіперплощини, і функція  $g(x)$  має вигляд:

$$g(x) = w_1 x_1 + \dots + w_n x_n + w_{n+1},$$

причому коефіцієнти  $w_i, i = \overline{1, n+1}$  повинні бути визначені в процесі навчання.

Одним з найпростіших методів підбору параметрів є *алгоритм перцептрона*, названий так через свій тісний зв'язок з навчанням перцептронів (див. підрозд. 2.6). Він полягає в тому, що задається початковий вектор параметрів  $w^{(0)}$ , який уточнюється у міру розпізнавання елементів навчальної вибірки. Правила уточнення такі: якщо черговий елемент навчальної вибірки  $x^{(i)}$  належить  $K_1$ , а  $g(x) \leq 0$ , то

$$w^{(i+1)} := w^{(i)} + cx^{(i)}, \text{ причому } c > 0; \quad (5.5)$$

черговий елемент навчальної вибірки  $x^{(i)}$  належить  $K_2$ , а  $g(x) \geq 0$ , то

$$w^{(i+1)} := w^{(i)} - cx^{(i)}, \text{ причому } c > 0. \quad (5.6)$$

У разі лінійної роздільності алгоритм збігається.

**Кусково-лінійні роздільні функції.** Цей клас функцій складніший за лінійні, і побудова їх також непроста. З іншого боку, кусково-лінійна роздільна функція

може з достатньою мірою точності апроксимувати практично будь-яку роздільну функцію. Одна з процедур побудови кусково-лінійних роздільних функцій була запропонована А. Аркадьєвим і Е. Браверманом [127].

**Методи дискримінантного розпізнавання.** Розглянемо детальніше деякі з класичних і найуживаніших методів дискримінантного розпізнавання.

**Метод найближчого сусіда.** Цей метод можна сформулювати так: *об'єкт відноситься системою розпізнавання до того класу, до якого належить його найближчий сусід з навчальної вибірки.*

Метод не дає помилок на об'єктах навчальної вибірки і не потребує попереднього формування розв'язувального правила. Але його недоліком є те, що в пам'яті доводиться зберігати всю навчальну вибірку і знаходити відстані до кожного її елемента. Це призводить до великих витрат як пам'яті, так і часу.

У разі застосування правила найближчого сусіда існує можливість виконати попереднє навчання, у процесі якого скоротити повну навчальну вибірку. Для цього можна використовувати так зване *стиснене правило найближчого сусіда*, запропоноване Е. Хартом, алгоритм якого наведено в [129], а тут подається у скороченому вигляді.

Для роботи алгоритму потрібні дві робочі множини — «Пам'ять» і «Відсів». Алгоритм виконує такі дії:

У «Пам'ять» заноситься по одному представнику кожного класу.

Кожен наступний об'єкт навчальної вибірки класифікується за допомогою правила найближчого сусіда; при цьому використовуються лише ті об'єкти, які містяться в множині «Пам'ять». Якщо цей об'єкт правильно класифікується, він викидається у «Відсів», у протилежному разі він заноситься в «Пам'ять».

Після перегляду всіх об'єктів описана процедура повторюється знову, але тільки для елементів множини «Відсів». Процедура завершується або при повному вичерпанні «Відсів», або якщо при черговому проході жоден об'єкт з «Відсів» не перейшов до «Пам'ять».

Після завершення роботи алгоритму як скорочена навчальна вибірка використовується множина «Пам'ять».

Модифікацією методу найближчого сусіда є *правило  $k$  найближчих сусідів*. Об'єкт відноситься системою розпізнавання до того класу, до якого належить більшість з  $k$  його найближчих сусідів з навчальної вибірки.

**Байєсівські методи розпізнавання.** Байєсівські методи належать до імовірнісних методів. Про деякі аспекти прийняття рішень в умовах невизначеності йшлося у підрозд. 2.8, а основні засади розв'язання задач в умовах лінгвістичної невизначеності розглядаються в [75; 77; 132].

Імовірнісні методи розпізнавання використовують для розгляду імовірнісні характеристики класів, і врахування такої інформації дає змогу суттєво підвищити якість розпізнавання

Основну ідею байєсівських методів можна неформально охарактеризувати так [21]. Нехай відомі імовірнісні характеристики того, що об'єкт, який належить до певного класу, описується деяким вектором ознак. Якщо ми отримали вектор ознак, який потрібно розпізнати, то на основі правил Байєса можемо розрахувати *апостеріорні ймовірності* належності цього вектора до кожного класу. Далі об'єкт відноситься системою розпізнавання до того класу, для якого ця апостеріорна імовірність найбільша (інакше кажучи, до найвірогіднішого класу).

Наведемо більш формалізований опис байєсівських методів.

Нехай маємо  $m$  класів  $K_1, \dots, K_m$ . З кожним класом пов'язана апіорна ймовірність  $P(K_i)$  – ймовірність появи об'єкта  $i$ -го класу. Нагадаємо, що ймовірності тісно пов'язані з частотою і тому вони, по суті, характеризують, наскільки часто зустрічаються об'єкти відповідних класів.

Основною імовірнісною характеристикою, яка встановлює розподіл векторів певного класу в просторі ознак, є *умовна щільність розподілу*. Наведемо необхідні визначення.

**Означення 5.10.** *Функцією розподілу* одновимірної неперервної випадкової величини  $\eta$  є функція

$$F_{\eta}(x) = P(\eta \leq x). \quad (5.7)$$

Аналогічно вводиться поняття функції розподілу від векторної випадкової величини.

**Означення 5.11.** *Щільністю розподілу* називається похідна від функції розподілу.

**Означення 5.12.** *Умовною щільністю розподілу*  $p(X_i/K_i)$  називається щільність розподілу вектора  $X_i$  за умови, що він належить до класу  $K_i$ .

Байєсівське розв'язувальне правило (2.21) можна інтерпретувати так.

Нехай ми маємо вектор  $Y$ . Для класифікації цього вектора потрібно для кожного класу розрахувати величини

$$p(K_i/Y) = \frac{P(K_i)p(Y/K_i)}{p(Y)},$$

які являють собою апостеріорні ймовірності належності цього вектора ознак для кожного класу. Слід обрати той клас, для якого ця ймовірність максимальна.

У знаменнику фігурує величина  $p(Y)$ , яка є спільною для всіх класів і яку можна скоротити. Остаточно маємо таке розв'язувальне правило: вектор  $Y$  потрібно віднести до того класу, для якого ця величина є максимальною.

Байєсівські методи можна застосовувати і тоді, коли необхідно враховувати ризики, пов'язані з неправильною класифікацією [27; 64].

### 5.3. Теоретико-множинна модель системи розпізнавання об'єктів роботизації

**Системний підхід до розпізнавання об'єктів.** Систему розпізнавання об'єктів, як і більшість складних технічних комплексів, можна розглядати з позицій системного підходу [139]. Моделі складних систем будуються з використанням апарату теорії множин, який використовується і при формалізованому описі розпізнавальної системи.

**Абстрактна модель розпізнавання.** Вихідним пунктом аналізу є абстрактна модель процесу розпізнавання об'єктів, яка описується орієнтованим графом

$$R'_{ij} \rightarrow \circ \xrightarrow{t \in T} \circ \xrightarrow{R_{ij} \quad q \in Q} \circ \xrightarrow{SP_{ij}^{(s)} \quad r \in RP} \circ \rightarrow K_{ij}, \quad (5.8)$$

де  $R'_{ij}$  – множина об'єктів;  $R_{ij}$  – множина їх зображень;  $SP_{ij}^{(s)}$  – система ознак;  $K_{ij}$  – класифікатор об'єктів;  $T$  – множина проєкційних перетворень;  $Q$  – сукупність аналітичних операцій та дій;  $RP$  – сукупність розв'язувальних правил.

Групування множин визначає склад основних параметрів, якими характеризується розпізнавальна система:

$$VS = \langle R'_{ij}, R_{ij}, SP_{ij}^{(s)}, K_{ij} \rangle. \quad (5.9)$$



Множина об'єктів задається в абсолютному тривимірному просторі поля спостережень. Кожен конкретний об'єкт описується кількістю точок простору, який він займає:

$$R'_{ij}(x_a, y_a, z_a) =$$

$$\left. \begin{array}{l} R'_{11}(x_a, y_a, z_a); R'_{12}(x_a, y_a, z_a); \dots; R'_{1j}(x_a, y_a, z_a) \\ R'_{21}(x_a, y_a, z_a); R'_{22}(x_a, y_a, z_a); \dots; R'_{2j}(x_a, y_a, z_a) \\ \dots \\ R'_{i1}(x_a, y_a, z_a); R'_{i2}(x_a, y_a, z_a); \dots; R'_{ij}(x_a, y_a, z_a) \end{array} \right\}, \quad (5.10)$$

де  $i$  – порядковий номер класу об'єктів у класифікаторі;  $j$  – порядковий номер підкласу, тобто конкретного об'єкта в класі;  $a = 1, 2, \dots$  – індекси точок об'єкта в системі координат  $xOyz$  поля спостереження.

**Особливості проектування об'єкта на чутливу поверхню.** На основі даних про об'єкт здійснюється процедура їх введення, яка полягає в проектуванні його на площину зображення. При цьому *проектування виконується паралельно або централью залежно від пристрою введення зображення*. Використання телевізійної або твердотільної камери для приймання відеоінформації ґрунтується на апараті центрального проектування об'єкта на площину зображення. Тоді кожна точка останнього описується двійкою чисел  $(x_k, y_k)$ , які є поточними координатами в декартовій системі  $Ox_ky_k$  кадрового поля.

Формалізовано процедуру введення даних зводиться до перетворення:

$$t: \{R'_{ij}(x_a, y_a, z_a)\} \rightarrow R_{ij}(x_k, y_k), \quad (5.11)$$

де  $t$  – лінійне невиворжене перетворення, що описується матрицею  $\mathbf{T}$  і відображає операцію проектування об'єкта;  $R'_{ij}, R_{ij}$  – об'єкт та його зображення;  $(x_a, y_a, z_a)$  – сукупність точок видимої частини об'єкта;  $(x_k, y_k)$  – координати відповідних точок його зображення.

Зупинимось докладніше на характері перетворення  $t$ . *Центральне проектування об'єкта* полягає в тому, що з деякого центра виходять промені, які проходять через площину зображення об'єкта, зв'язуючи центр з його точками. Якщо площина, на яку проектується об'єкт, розташована від центра проектування на відстані, що дорівнює фокусній, то одержане зображення є фотографічним відбитком об'єкта без збільшення. Якщо ж центр  $F$  задано в просторі поля спостереження і відомо координати  $x_a, y_a, z_a$  точки об'єкта, то останні та координати  $x_k, y_k$  проєкції цієї точки на площину зображення пов'язані між собою аналітичною залежністю

$$(x_k - f_1) / (x_a - f_1) = (y_k - f_2) / (y_a - f_2) = -f_3 / (z_a - f_3), \quad (5.12)$$

яка випливає з умов подібності. Тоді координати точки зображення об'єкта можна знайти за формулами

$$x_k = (f_3 x_a - f_1 z_a) / (f_3 - z_a); y_k = (f_3 y_a - f_2 z_a) / (f_3 - z_a). \quad (5.13)$$

Перетворенню центрального проектування об'єкта з точки  $F(f_1, f_2, f_3)$  на площину  $z_a = 0$  відповідає проєктивне перетворення, яке в однорідних координатах задається матрицею [23]

$$\mathbf{T}_f = \begin{vmatrix} f_3 & 0 & f_1 & 0 \\ 0 & f_3 & f_2 & 0 \\ 0 & 0 & f_3 & 0 \\ 0 & 0 & -1 & f_3 \end{vmatrix}. \quad (5.14)$$

Якщо вектор  $V_a = [x_a, y_a, z_a, 1]$  зображує точку в реальному просторі, то вектор  $V_k = \mathbf{T}_f V_a$  буде такою точкою у перспективному просторі, однорідні координати якої  $(x_k/1), (y_k/1)$  дорівнюють координатам проекції вихідної точки на площину зображень.

При *паралельному проектуванні об'єкта* центр  $F(f_3 \rightarrow \infty)$  розміщується в нескінченності й об'єкт зображується в реальному масштабі.

Повторивши багаторазово процедуру проектування, дістанемо зображення об'єкта, заданого сукупністю точок  $(x_k, y_k)$ , які відповідають збудженим рецепторам сітківки дискретизованого кадру:

$$R_{ij}(x_k, y_k) = \begin{vmatrix} R_{11}(x_k, y_k); R_{12}(x_k, y_k); \dots; R_{1j}(x_k, y_k) \\ R_{21}(x_k, y_k); R_{22}(x_k, y_k); \dots; R_{2j}(x_k, y_k) \\ \dots \\ R_{i1}(x_k, y_k); R_{i2}(x_k, y_k); \dots; R_{ij}(x_k, y_k) \end{vmatrix}. \quad (5.15)$$

Границя фон-об'єкта на зображенні подається порогом, згідно з яким функція яскравості  $B_{ij}$  для бінарного зображення має значення:

$$B_{ij} = \begin{cases} 1 & \text{— об'єкт накриває точку сітківки;} \\ 0 & \text{— не накриває цю точку.} \end{cases} \quad (5.16)$$

**Роздільна здатність чутливої поверхні.** Точність відтворення зображення об'єкта пропорційна геометричній роздільній здатності приймальної матричної сітківки і залежить від щільності чутливих елементів на одиниці площі.

*Дискретне зображення об'єкта* — це масив точок, кожна з яких характеризується значенням функції яскравості. Для його запису та зберігання необхідні значні обсяги машинної пам'яті, причому така інформація обробляється трудомісткими методами, що збільшує час на розпізнавання об'єкта. Тому автоматична класифікація об'єктів безпосередньо за масивом (5.15) даних про зображення неефективна.

Для усунення цього недоліку передбачається виконання групи аналітичних та інших операцій  $q \in Q$ , що визначають суть розрахунку ознак, яка економно і водночас повно й адекватно описує об'єкт та його зображення:

$$q : \{R_{ij}(x_k, y_k)\} \rightarrow SP_{ij}^{(s)}, \quad (5.17)$$

де  $i, j, s = 1, 2, \dots$ ;  $SP_{ij}^{(s)}$  — система ознак для опису  $j$ -го об'єкта  $i$ -го класу.

**Перетворення та дії над зображеннями об'єкта.** Ці операції виконуються з використанням математичного апарату аналітичної геометрії на площині, алгебри логіки та ін. Докладніше суть операцій, які виконуються при розрахунку системи ознак, розглядається у [18]. Зазначимо, що найчастіше в задачах розпізнавання об'єктів *застосовуються два методи*, причому для першого характерними є окремі точки зображення об'єкта, які задаються логічними аргументами, а для другого — коефіцієнти ряду Фур'є, який відображає аналітичну функцію, що описує контур зображення об'єкта.

З урахуванням викладеного системи ознак для опису об'єкта цими методами можна записати як

$$SP_{ij}^{(1)} = f[B_{ij}(x_k, y_k)]; \quad (5.18)$$

$$SP_{ij}^{(2)} = f(C_n, \alpha), \quad (5.19)$$

де  $C_n$  — коефіцієнти ряду Фур'є;  $\alpha$  — кут, який є параметром розвитку і характеризує орієнтацію об'єкта. У першому випадку система  $SP_{ij}^{(1)}$  задається в бінарному просторі ознак, бо функція яскравості  $B_{ij}$  є логічною змінною. В другому випадку система  $SP_{ij}^{(2)}$  задається в  $n$ -вимірному просторі коефіцієнтів  $C_n$ , які застосовуються для розпізнавання об'єктів, а орієнтація їх визначається за параметром  $\alpha$ .

Класифікатор  $K_{ij}$  містить множини значень еталонних ознак, які описують адекватно кожен об'єкт, що підлягає розпізнаванню. Він має тривірневу ієрархічну структуру, на верхньому рівні якої задаються  $i$ -ті номери класів об'єктів, на середньому —  $j$ -ті номери конкретних об'єктів, а на нижньому — еталонні значення ознак розпізнавання.

Процедура зведення до класифікатора формалізується виразом

$$r : \{SP_{ij}^{(s)}\} \rightarrow K_{ij}, \quad (5.20)$$

де  $r$  — операція, що відображає суть розв'язувального правила. Вибір останнього певною мірою залежить від специфіки розв'язуваної задачі розпізнавання. Для технічних об'єктів найбільш поширеним є правило, за яким виконується порівняння їх з еталоном, тобто значення ознак, розраховані для поточного поданого зображення об'єкта, порівнюються з еталонним.

При такому розв'язувальному правилі класифікатор зручно подати у вигляді орієнтованого графа  $K = (\overline{U}, \overline{V})$ , де  $U$  — кількість вершин графа, що відображають елементи класифікатора;  $V$  — кількість ребер, які задають внутрішні зв'язки та відношення підпорядкування в класифікаторі.

З урахуванням викладеного можна стверджувати, що розпізнавальна система здійснює *три види процедур*, які визначають її функції, а структура та склад процедур задаються графом процесу розпізнавання (5.8). Згідно з цим умовним розподілом теоретико-множинна модель розпізнавальної системи описується двома групами виразів, причому для першої групи

$$VS = \begin{cases} \langle R', R, SP, K \rangle; \\ \langle R, SP, K \rangle; \\ \langle SP, K \rangle, \end{cases} \quad (5.21)$$

а для другої

$$R' = \begin{cases} R'_{ij}(x_a, y_a, z_a); \\ t : \{R'_{ij}(x_a, y_a, z_a)\} \rightarrow R_{ij}; \end{cases} \quad (5.22)$$

$$R = \begin{cases} R_{ij}(x_k, y_k); \\ q : \{R_{ij}(x_k, y_k)\} \rightarrow SP_{ij}^{(s)}; \end{cases} \quad (5.23)$$

$$SP = \begin{cases} B_{ij}(x_k, y_k); \\ f(C_n, \alpha); \\ r : \{SP_{ij}^{(s)}\} \rightarrow K_{ij}; \end{cases} \quad (5.24)$$

$$K = (\overline{U}, \overline{V}). \quad (5.25)$$

#### 5.4. Основні методи попередньої обробки сигналів і зображень

**Суть попередньої обробки сигналів і зображень.** Як уже зазначалося, обробка експериментальних сигналів і зображень має велике значення для теорії і практики розпізнавання. Нагадаємо, що експериментальні сигнали описуються функцією однієї змінної (найчастіше — функцією, яка задає зміну деякої фізичної величини

з часом). Зображення ж описуються функцією двох змінних, яка задає значення яскравості в різних точках зображення.

На етапі попередньої обробки розв'язуються такі важливі задачі:

- отримання первинних ознак;
- зменшення шумів та підвищення якості зображення (наприклад, підвищення чіткості контурів фігур);
- сегментація (поділ зображення на зони, які оброблятимуться).

Існує величезна кількість задач, пов'язаних з обробкою сигналів і зображень [11; 14; 83]. Розглянемо особливість процедур, пов'язаних з деякими підходами до попередньої обробки зображень.

**Отримання первинних ознак на основі дискретизації.** Найтипівішим і найчастіше вживаним способом отримання первинних описів об'єктів, які задаються неперервними величинами, є *дискретизація* цих величин (інші назви — *оцифрування*, *цифрування*).

**Означення 5.15.** *Дискретизацією* є процедура аналогово-цифрового перетворення з реєструванням аналогової величини у вигляді набору своїх значень у дискретних точках.

Нехай неперервна функція  $f(t)$  спостерігається на проміжку  $[a, b]$ . Зафіксуємо  $n$  дискретних точок, в яких вимірюватимуться значення функції:  $t_1 = a < t_2 < \dots < t_{n-1} < t_n = b$ . Ці точки часто називаються *вузлами*. Тоді дискретизація функції  $f(t)$  полягає у зображенні її вектором

$$y = (y_1, \dots, y_n),$$

де  $y_j = f(t_j)$ , тобто кожна координата вектора являє собою значення функції в черговому вузлі.

Дискретні значення  $y_j = f(t_j)$  часто називають *первинними ознаками сигналу*. Найчастіше застосовується *рівномірна дискретизація*, за якої відстані між будь-якими сусідніми вузлами є однаковими. У такому разі величина  $h = t_{j+1} - t_j$ , однакова для всіх  $j$ , називається *кроком дискретизації*.

Крок рівномірної дискретизації можна обчислити за формулою

$$h = (b - a) / (n - 1). \quad (5.26)$$

Величина  $q = 1/n$  називається *частотою дискретизації*, яка характеризує кількість дискретних вузлів на одиничному інтервалі.

**Приклад 5.5.** Маємо набір дискретних значень функції  $f(t) = 2t + 1$  на проміжку  $[0, 1]$  з кроком 0,2.

Для  $n = 6$  маємо:  $t_1 = a = 0$ ;  $t_2 = 0,2$ ;  $t_3 = 0,4$ ;  $t_4 = 0,6$ ;  $t_5 = 0,8$ ;  $t_6 = b = 1$ .

Тоді вектор дискретних значень  $f(t) = (1; 1,4; 1,8; 2,2; 2,6; 3)$ .

Звичайно, дискретизовану функцію можна задавати і в табличному вигляді.

$t$	0	0,2	0,4	0,6	0,8	1
$f(t)$	1	1,4	1,8	2,2	2,6	3

Потрібно ретельно вибирати частоту дискретизації. Надто висока частота дискретизації (і, відповідно, надто малий крок дискретизації) приводить до завеликого обсягу інформації. Це, у свою чергу, спричиняє надмірні витрати на зберігання і обробку такої інформації. У зв'язку з цим постає проблема *стиснення інформації*, яка в цьому контексті означає зберігання оцифрованої інформації в економічнішому вигляді.

З іншого боку, зменшення частоти дискретизації призводить, як правило, до зменшення точності. Фундаментальна *теорема Котельникова* (аналогічний резуль-

тат пізніше був отриманий Шенноном) встановлює, якою повинна бути мінімальна частота дискретизації, за якої інформація не втрачається, тобто за якої неперервна функція ще може бути точно відновлена за дискретними даними [85].

Все наведене справедливе і для дискретизації зображень. На зображення накладається двовимірна сітка, і тоді зображення можна закодувати матрицею, елементи якої дорівнюють сумарній яскравості зображення в окремому квадратику сітки. Часто застосовується і простіший спосіб кодування, за якого елемент матриці дорівнює 1, якщо у відповідному квадратику є фрагмент зображення, і 0 — якщо немає.

**Приклад 5.6.** Нехай сканується зображення, яке являє собою деякий об'єкт (рис. 5.4).

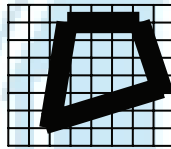


Рис. 5.4. Приклад дискретизації зображення об'єкта

Закодуємо це зображення за допомогою сітки розміром  $8 \times 8$  і подамо у вигляді матриці.

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Розглянемо обставини, які ускладнюють обробку таких зображень.

Форма відсканованого образу дещо спотворюється, що стає особливо помітно при великому розмірі клітинок сітки (при малому кроці дискретизації).

Сканер працює не бездоганно, і в результаті на відсканованому зображенні можуть з'являтися зайві лінії, яких не було на оригіналі, і навпаки — зникати ті, які були. В результаті сканований образ може бути сприйнятий як інший.

Результат кодування є дуже чутливим до позиціонування об'єкта, який розпізнається, відносно сканера. Якщо, скажімо, проекцію деталі (чи іншого об'єкта сканування) трошки зсунути вбік, його зображення потрапить в інші клітинки і матриця істотно зміниться.

Крім того, об'єкт сканування може займати на сканувальній (розпізнавальній) поверхні безліч відмінних (в тому числі і нестійких) положень. Щоб уникнути такої проблеми, удаються або до попереднього суворого орієнтування (просторового чи відносного) об'єкта сканування, або в базі даних розпізнавальної системи треба мати необхідну кількість описів відмінних положень одного й того самого об'єкта.

Усі ці обставини зумовлюють добре відомий факт: *методи розпізнавання в просторі ознак є дуже чутливими до перелічених вище факторів, і тому для розпізнавання технічних об'єктів насамперед використовуються структурні методи.*

**Ланцюговий код Фрімена.** Для опису ліній і контурів зображень широко застосовуються *ланцюгові коди*, основна ідея яких полягає в тому, що відрізки ліній апроксимуються векторами з певного фіксованого набору, кожний з яких, у свою чергу, кодується символом деякого алфавіту.

Найпростішим і найуживанішим є *код Фрімена*, за якого лінії кодуються за допомогою восьми можливих векторів, пронумерованих числами від 0 до 7 (рис. 5.5).

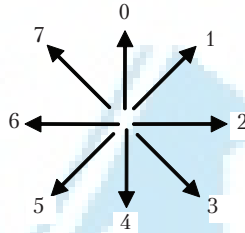


Рис. 5.5. Ланцюговий код Фрімена

Типова схема кодування полягає в тому, що вузли сітки, розташовані найближче до контуру, з'єднуються прямими лініями (векторами), які, у свою чергу, кодуються відповідними числовими значеннями.

**Параметризація неперервних функцій на основі лінійних ортогональних перетворень. Загальні принципи ортогональних перетворень.** Потужним засобом отримання вторинних ознак сигналів і зображень є використання *ортогональних перетворень*. Надалі розглядатимемо лише одновимірні сигнали.

З курсу математичного аналізу відомо, що неперервну функцію  $\eta(t)$ , яка спостерігається на відрізку  $[a, b]$ , можна розкласти в ряд:

$$\eta(t) = \sum_{k=1}^{\infty} c_k j_k(t). \quad (5.27)$$

При цьому координатні функції  $\{\varphi_k(t)\}$  можуть бути вибрані *ортонормованими*. Це означає, що скалярний добуток

$$\int_b^a \varphi_k(t) \varphi_l(t) dt$$

дорівнює 1, якщо  $k = l$ , і 0, якщо  $k \neq l$ .

Якщо координатні функції ортонормовані, коефіцієнт розкладу (5.27) обчислюється за формулою

$$c_k = \int_b^a \varphi_k(t) \eta(t) dt, \quad k = \overline{1, \infty}. \quad (5.28)$$

Формула (5.28) і являє собою *формулу ортогонального перетворення*.

Для функцій, які можуть бути фізично реалізовані, коефіцієнти  $c_k$  прямують до нуля, і, більше того, ряд (5.27) збігається. На практиці його можна апроксимувати скінченним рядом

$$\eta(t) = \sum_{k=1}^r c_k \varphi_k(t) \quad (5.29)$$

з певною похибкою.

Тоді спектральні коефіцієнти  $c_k$  можна розглядати як ознаки функції  $\eta(t)$ , а вектор  $(c_1, \dots, c_r)$  — як зображення цієї функції у відповідному просторі ознак.

**Перетворення Фур'є.** Найбільш відомим ортогональним перетворенням є *перетворення Фур'є*, формулу якого можна записати у вигляді

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \eta(t) e^{-ikt} dt, \quad k = \overline{1, \infty}. \quad (5.30)$$

Існує значна кількість перетворень, тісно пов'язаних з перетворенням Фур'є, насамперед синусне та косинусне. Так, якщо врахувати, що  $e^{-ix} = \cos x - i \sin x$ , то ряд Фур'є можна переписати за такою послідовністю.

Нехай неперервна функція  $f(x)$  задана на відрізку  $[-\pi, \pi]$ . До більш загального випадку можна перейти за допомогою заміни змінних. Ряд Фур'є такої функції записується у вигляді

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx), \quad (5.31)$$

де коефіцієнти розкладу обчислюються за формулами:  $a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) dx$ ;  
 $a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx dx$ ;  $b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx dx$ .

Безпосередня реалізація формули (5.28) та її дискретних апроксимацій пов'язана зі значними часовими затратами. Але через періодичність синусів і косинусів доданки, які входять до розрахункових формул, часто повторюються; крім того, розрахунковий алгоритм можна записати в рекурсивній формі. Ці міркування стали основою швидких алгоритмів реалізації перетворення Фур'є, які широко застосовуються на практиці [32; 81].

**Інтегральне перетворення Карунена – Лоева.** Інтегральне перетворення Карунена – Лоева є в певному розумінні оптимальним ортогональним перетворенням. Його оптимальність полягає в тому, що воно мінімізує похибку апроксимації ряду (5.27) при фіксованій кількості членів зрізаного ряду  $r$  або мінімізує  $r$  у межах заданої похибки. В цьому розумінні можна сказати, що перетворення Карунена – Лоева забезпечує оптимальне стиснення даних.

Координатні функції перетворення Карунена – Лоева обчислюються як власні функції рівняння

$$\int_a^b R(t, s) \phi_k(s) ds = \sigma_k \phi_k(t), \quad (5.32)$$

що відповідають власним функціям  $\sigma_k$ , впорядкованим за спаданням.

Тут  $R(t, s)$  – коваріаційна функція породжуючого випадкового процесу. Загалом вона є невідомою, але її можна оцінити за певним набором реалізацій цього процесу  $V$ . Для такого оцінювання використовується формула

$$\widehat{R}^{(v)}(t, s) = \sum \alpha_i \xi_i(t) \xi_i(s), \quad (5.33)$$

де сума береться за всіма функціями  $\xi_i(t)$ , що входять до набору  $V$ ;  $\alpha_i$  – задані коефіцієнти, сума яких дорівнює 1. Нехай множина  $V$  є базовою вибіркою, позначимо через  $q$  кількість її елементів.

Основним недоліком перетворення Карунена – Лоева є відсутність для нього швидких перетворень. Незважаючи на це, роль перетворення Карунена – Лоева збільшується в таких випадках:

- у задачах, в яких швидкість обробки не має суттєвого значення, а на перший план виходить саме міра стиснення інформації;
- при аналізі нестационарних процесів, оскільки їх спектр Фур'є залежить від двох змінних і практичне використання такого спектра ускладнюється;
- у ряді задач, в яких першочергове значення має некорельованість спектральних коефіцієнтів, яка в загальному випадку забезпечується лише перетворен-

ням Карунена — Лоева. Це, наприклад, задача відновлення сигналів і зображень, спотворених неідеальною вимірювальною апаратурою.

Алгоритми комп'ютерної реалізації власне інтегрального перетворення Карунена — Лоева сьогодні розвинені недостатньо, і їх ефективність суттєво залежить від способу апроксимації інтегралів. Через це зокрема на практиці замість інтегрального перетворення Карунена — Лоева використовується його *дискретний варіант*, який, по суті, є відомим у математичній статистиці методом головних компонент [4].

**Оперативний аналіз сенсорної інформації.** Аналіз зорової інформації про об'єкт здійснюється в такій послідовності: фільтрація зображень; виявлення об'єкта або інформативної області на його зображенні; формування опису об'єкта в термінах ознакового простору.

*Фільтрація зображень об'єктів* передбачається тоді, коли є зовнішні перешкоди і високий рівень шумів сенсорної системи, які унеможливають процес розпізнавання об'єктів або ведуть до необхідності організації числових алгоритмів перевірки вірогідності одержаних результатів. Великого поширення в практиці машинної обробки зображень набув метод *анізотропної фільтрації*, який дає змогу послабити вплив окремих плям, що не належать проекції об'єкта, та прогалин всередині проекції. Для бінарних зображень об'єктів при невеликому рівні перешкод ефективним є *метод логічного згладжування*, коли рівень довільного елемента зображення «вирівнюється» за потенціалами сусідніх його точок.

Для *виявлення об'єкта* аналіз фільтрованого зображення починається з виділення контурної межі, що відокремлює об'єкт від фону. Двовимірна функція апроксимуючого ступінчастого перепаду  $B(x, y)$ , знайдена на дискретному сенсорному полі, має вигляд:

$$B(x, y) = \begin{cases} b & \text{при } x \cos \theta + y \sin \theta < \rho \\ b + h & \text{при } x \cos \theta + y \sin \theta \geq \rho \end{cases} = B(x, y, \theta, b, h, \rho),$$

де  $\theta, \rho$  — полярні координати точки перепаду, найближчі до центра досліджуваної області.

Здебільшого процедури виділення контурних меж поєднуються з розв'язуванням задачі відстеження контуру, в процесі якого виявляються вершини контуру, їх координати та інші необхідні дані про проекцію об'єкта. При цьому виконується така послідовність дій: пошук точки в центрі растра, що переглядається; перевірка зв'язків усередині растра та пошук нової точки контуру; переміщення центра растра в нову точку контуру; виявлення вершини контуру та запам'ятовування її координат; перевірка замкненості контуру. Складність використання таких методів полягає у виключенні хибних вершин, у зв'язку з чим доводиться зберігати масиви довідкових даних з інформацією про попередні кроки руху вздовж контуру (див. блок-схему алгоритму ідентифікації на рис. 5.19).

Етапу побудови *ознакового простору* передують операції нормалізації зображень об'єкта, які складаються з визначення оператора нормалізації  $H(\Gamma)x_i = x_0$ , що дає змогу звести проекцію  $x_i$ , яка не виходить за межі множини  $\Gamma$  груп допустимих перетворень, до стандартної (еталонної) проекції  $x_0$  записаної моделі середовища.

Після завершення процедур нормалізації здійснюється формалізація об'єкта в термінах і числових характеристиках ознакового простору.



## 5.5. Формалізоване подання об'єктів ГКІС

**Основні методи формалізованого опису технологічних об'єктів.** Застосування обчислювальної техніки в технологічному проектуванні призвело до необхідності перетворення вихідної інформації у вигляді креслень об'єктів проектування, паспортів устаткування та інших документів у форму, зручну для сприйняття ЕОМ. Нині розроблено ряд методів формалізованого опису об'єктів проектування, орієнтованих як на використання *формалізованої технологічної мови (ФТМ)*, в основу якої покладено метод системно-структурного аналізу складних об'єктів, так і на аналітичний опис геометрії об'єкта, в тому числі й із застосуванням наближеного подання об'єктів.

**Формалізовані технологічні мови.** При використанні формалізованої технологічної мови просторовий образ деталі формується у вигляді сукупності об'ємних елементів і поверхонь, що кодується спеціально розробленими кодами. Один з основних принципів побудови ФТМ полягає в тому, що деталь подається як система, яка складається з множини елементів, пов'язаних між собою для виконання певної функції. Вирізняють такі основні підходи до опису об'єктів.

**Проекційний підхід** базується на описі проєкцій об'єкта на площині координат. Як змістові одиниці формальної мови виступають такі елементи опису, як точка, дуга, коло, відрізок прямої, відрізок довільної кривої. Зв'язки між цими елементами встановлюються шляхом визначення способів або координат точок перетину.

Цей спосіб застосовується при розробці програм для верстатів із ЧПУ, а також опису інформації, що вводиться у креслярсько-графічні автомати.

Оскільки цей підхід потребує логіко-математичного апарату розпізнавання об'єктів у просторі, він не застосовується при проектуванні технологічних процесів.

**Конструктивний підхід** полягає в тому, що в конфігурації деталі виділяють просторові образи, інформація про які в закодованому вигляді заноситься в кодуювальну відомість і утворює разом зі зв'язками цих образів формальний опис деталі. Прикладом такого підходу може бути система кодування КОД-78, призначена для кодування деталей типу тіл обертання. Змістовими одиницями ФТМ цієї системи є елементарні поверхні (наприклад, плоскі, циліндричні, сферичні), стандартизовані елементи, утворені типовими сполученнями елементарних поверхонь (пази, канавки, отвори). Зв'язок між елементами здійснюється за допомогою координуючих розмірів, вказівок на взаємне розташування елементів, конструктивних операцій типу «З'єднання», «Відсікання». Форма запису інформації про об'єкт різна — від табличної до складних мовних конструкцій. Система кодування деталей КОД-78 складається з предметної мови, кодувальних таблиць та інструкції щодо їх заповнення. Опис деталі поділяється на чотири основних модулі:

(опис деталі): := ПОЧ (модуль загальних характеристик)  
 [(модуль елементів)]  
 [(модуль зв'язків)]  
 [(логічний модуль)]  
 КІН.

Модуль загальних характеристик призначений для опису загальних характеристик деталей і містить оператори з іменами: ЗАГД — загальні дані; КОНФ — конфі-

гурація деталі; ШОРСТ – шорсткість поверхні та інших операторів, що надають інформацію про покриття, термічну обробку деталей. Нехай, наприклад, деяка деталь має такі параметри: позначення деталі (*ND*) – Ю-613541; найменування деталі (*IM*) – втулка; марка матеріалу (*MAR*) – сталь 50; ДСТ на матеріал (*GM*) – ДСТ 1050–74; маса деталі (*M*) = 0,15 кг. Тоді оператор загальних відомостей (ОВІД) має вигляд:

ОВІДND-Ю-613541, *IM* = ВТУЛКА, *MAR* = СТАЛЬ50, *GM* = 1050-74, *M* = 0,15.

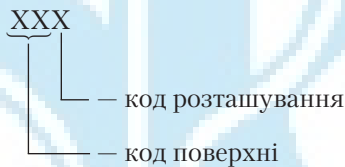
Оператор КОНФ містить такі параметри: ЄСКД – форма деталі за класифікатором ЄСКД; *L*, *D*, *DM* – відповідно довжина, найбільший зовнішній діаметр, найменший внутрішній діаметр. Наприклад:

КОНФ ЄСКД = 241500, *L* = 60, *D* = 28, *DM* = 16.

До оператора ШОРСТ входять такі параметри: *RZ* – висота нерівностей за десятима точками; *RA* – середнє арифметичне відхилення профілю; *MAX* – найбільша висота нерівностей профілю:

ШОРСТ *RZ* = 20.

Модуль елементів призначений для опису виділених елементів деталі, до яких належать елементарні поверхні, конструктивні елементи і комплекси. Елементарна поверхня має дворівневий вигляд:



Класифікатор елементарних поверхонь можна побудувати, наприклад, за таким принципом: 01 – зовнішня циліндрична поверхня; 02 – внутрішня конічна поверхня; 03 – зовнішня конічна поверхня і т. д. Розташування поверхонь кодується в такому порядку: 0 – довільне розташування поверхні; 1, 2, 3 – поверхня розташована відповідно у площині *xy*, *yz*, *zx*; 4, 5, 6 – поверхня паралельна відповідно площині *xy*, *yz*, *zx* і т. д. Конструктивні елементи являють собою поєднання елементарних поверхонь і поділяються на дві категорії. До першої зараховують елементи «з'єднання» (опуклі тіла), наприклад, циліндр, конус, сфера, опуклий багатогранник, до другої – «відсікання», наприклад, канавки, пази, отвори, шліци. Об'єднання елементів першої та другої категорій утворюють комплекси. Модуль зв'язків визначає зв'язок між елементами для сприйняття цілісного образу деталі, а також вимоги щодо точності взаємного розташування елементарних поверхонь. Взаємне розташування елементів описується оператором РОЗТ:

РОЗТ ВИД = код розташування,

СПОЛ = код сполучення,

ЕЛ = список елементів,

БЕЛ = базовий елемент,

у якому можуть бути використані такі коди: *код розташування*: 1 – співвісне розташування елементів; 2 – під прямим кутом; 3 – елементи розташовані по колу; 4 – мають паралельні осі та ін.; *код поєднання*: 1 – з'єднання елементів; 2 – відсікання від елемента; 3 – перетин елементів; 4 – конгруентність елементів та ін. Координуючі розміри, що зв'язують між собою елементи, задаються за допомогою оператора КООР, який містить код виду розміру (наприклад, 1 – охоплюваних, 2 – охоплю-

ючих, 3 – інших), номінальну величину, її точність, спосіб задання межі розміру. Відхилення від взаємного розташування фіксують за допомогою оператора ВІДХ, який містить код допуску розташування (наприклад, 1 – паралельність, 2 – перпендикулярність, 3 – нахил, 4 – співвісність, 5 – осьове биття), список елементів, величину допуску, довжину і ширину, на які задано допуск. Логічний модуль служить для відображення логічних зв'язків між елементами, що використовуються при описі складної, комплексної деталі.

**Аналітичні методи опису об'єктів.** Прикладом аналітичного опису складних геометричних об'єктів є *метод R-функцій*. Апарат R-функцій дає змогу складати рівняння складних креслень і просторових об'єктів, а також будувати рівняння сімей поверхонь, що включають заданий геометричний об'єкт. Суть методу R-функцій полягає в тому, що вказується спосіб побудови алгебраїчної функції  $F(x, y, z)$ , яка в усіх точках контуру додатна, поза контуром – від'ємна, на межі перетворюється в нуль.

**Приклад 5.7.** Нехай, наприклад, необхідно описати поверхню деталі, зображеної на рис. 5.6, а. Поверхня деталі може бути складена з границь таких областей:

- а) нескінченного циліндра  $D_1$ , обумовленого нерівністю  $16 - x^2 - y^2 \geq 0$ ;
- б) нескінченного циліндра  $D_2$ , обумовленого нерівністю  $64 - x^2 - y^2 \geq 0$ ;
- в) шару  $D_3$  між площинами  $z = 0$  та  $z = 12$ ;
- г) шару  $D_4$  між площинами  $z = 0$  та  $z = 20$ ;
- д) нескінченної призми  $D_5$  з ребрами, паралельними осі  $Ox$ , яка має в перпендикулярному перерізі прямокутник (рис. 5.6, б).

Область, що визначає розглянуту деталь, подається логічною формулою

$$D = [(D_1 \wedge D_4) \wedge (D_2 \wedge D_4 \wedge \overline{D_3})] \wedge \overline{D_5}.$$

Рівняння поверхні деталі записується у вигляді

$$[(16 - x^2 - y^2) \wedge (20z - z^2)] \wedge [(64 - x^2 - y^2) \wedge (20z - z^2) \wedge (z^2 - 12z)] \vee [(1 - y) \wedge (20z - z^2) \wedge (z^2 - 16z)] = 0.$$

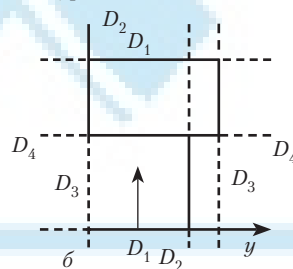
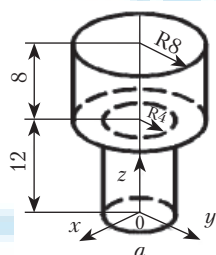


Рис. 5.6. Поверхня описуваної деталі (а), її інтерпретація за допомогою елементарних областей (б)

Цей метод опису застосовується для розв'язання багатьох задач математичної фізики, механіки, машинного розпізнавання геометричних образів тощо. З цією ж метою можна застосовувати *метод математичного опису моделей та їх взаємної взаємодії*, який належить до методів обчислювальної геометрії. В основу цього методу покладено способи і прийоми складання математичних залежностей, що характеризують геометрію об'єктів проектування засобами елементарної нарисної та аналітичної геометрії.

**Наближені методи подання об'єктів.** Застосування апарату аналітичної геометрії або апарату дискретних функцій дає змогу описувати геометричні об'єкти досить точно, але при цьому збільшується кількість параметрів опису, ускладнюються функціональні зв'язки між ними. Для практичних цілей технологічного проектування із застосуванням ЕОМ зручно користуватися *наближеними методами*, що описують з достатньою точністю геометричні об'єкти, наприклад, методом формального опису форми і просторового розташування об'єктів, у якому геометричний образ об'єкта подається у вигляді *математичної координатної моделі* (МКМ).

При цьому МКМ містить інформацію про геометрію деталі у вигляді координат поверхневих точок, що визначаються з рівняння поверхні, заданого в параметричній формі:

$$x = x(u, v); y = y(u, v); z = z(u, v).$$

При описі деталі не завжди можливо визначити її у вигляді однієї обмежуючої поверхні чи поверхні одного типу (сферичної, циліндричної тощо). Тому доцільно розбивати поверхню на елементарні поверхні. Таким чином, якщо визначити ряд елементарних типових поверхонь, то з їх сукупності можна синтезувати просторовий образ об'єкта з достатнім ступенем точності. Це означає, що МКМ деталі в загальному випадку являє собою набір *елементарних координат моделей* (ЕКМ) різних типів, жорстко зв'язаних між собою. Для опису форми і розташування в просторі кожної з ЕКМ, що описують деталь, необхідно: вибрати базову систему координат; задати початок відліку власних систем координат сукупності ЕКМ, що відображають геометрію об'єкта; задати параметри опису форми ЕКМ.

Розглянемо основні типи ЕКМ і способи їх опису.

1. ЕКМ «Паралелепіпед» (ЕКМП) (рис. 5.7, а). Параметри початку відліку власної системи координат —  $X_{ij} = X$ ,  $Y_{ij} = Y$ ,  $Z_{ij} = Z$ ; параметри форми ЕКМ —  $x, y, z$ . Параметричні рівняння поверхні мають вигляд:

$$x_{ij} = \alpha x; y_{ij} = \beta y; z_{ij} = \gamma z,$$

де коефіцієнти  $\alpha, \beta, \gamma$  відображають відношення координат поточної точки, яка належить до відрізка числової осі, до загальної довжини відрізка:

$$0 \leq \alpha \leq 1; 0 \leq \beta \leq 1; 0 \leq \gamma \leq 1,$$

причому індекс  $i$  — номер деталі,  $j$  — номер ЕКМ деталі, що описується.

2. ЕКМ «Циліндр» (ЕКМЦ) (рис. 5.7, б). Початок відліку власної системи координат визначає параметри положення ЕКМ:  $X_{ij} = X$ ,  $Y_{ij} = Y$ ,  $Z_{ij} = Z$ . Рівняння поверхні циліндра в параметричній формі має вигляд:

$$x_{ij} = R \cos \varphi; y_{ij} = R \sin \varphi; z_{ij} = H(1 + \cos \psi) / 2; 0 \leq \varphi \leq 360^\circ; 0 \leq \psi \leq 180^\circ.$$

Для опису форми ЕКМЦ необхідно задати  $R, H, \varphi, \psi$ .

3. ЕКМ «Циліндрична труба» (ЕКМЦТ) (рис. 5.7, в).

Цей вид ЕКМ є ніби похідною від ЕКМЦ, але через велику кількість деталей подібного класу у вузлах машинобудівних виробів доцільно виділити його в окремий вид. Параметри положення ЕКМЦТ —  $X_{ij} = X$ ,  $Y_{ij} = Y$ ,  $Z_{ij} = Z$ . Поверхня ЕКМЦТ описується такими параметричними рівняннями:

$$\begin{aligned} x_{ij} &= [R_1 + (R_2 - R_1)(1 + \cos \alpha) / 2] \cos \beta; \\ y_{ij} &= [R_1 + (R_2 - R_1)(1 + \cos \alpha) / 2] \sin \beta; \quad z_{ij} = H(1 + \cos \gamma) / 2; \\ &0 \leq \alpha \leq 360^\circ; 0 \leq \beta \leq 180^\circ; 0 \leq \gamma \leq 180^\circ \end{aligned}$$

і задаються параметри форми  $R_1, R_2, H, \alpha, \beta, \gamma$ .

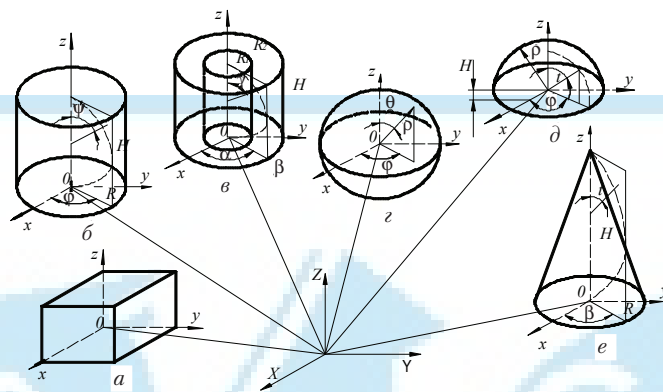


Рис. 5.7. Типи ЕКМ: паралелепіпед (а), циліндр (б), порожній циліндр (в), сфера (z), сегмент сфери (д), конус (е)

4. ЕКМ «Сфера» (ЕКМС) (рис. 5.7, z). Параметри початку відліку власної системи координат  $X_{ij} = X, Y_{ij} = Y, Z_{ij} = Z$ .

Параметричні рівняння поверхні сфери мають вигляд:

$$x_{ij} = \rho \cos \varphi \sin \theta; \quad y_{ij} = \rho \cos \varphi \sin \theta; \quad z_{ij} = \rho \cos \varphi;$$

$$0 \leq \varphi \leq 360^\circ; \quad 0 \leq \gamma \leq 180^\circ.$$

Для опису ЕКМС необхідно задати  $\rho, \varphi, \theta$ .

5. Похідна від ЕКМС «Сегмент сфери» (ЕКМСС) (рис. 5.7, д). Параметри положення –  $X_{ij} = X, Y_{ij} = Y, Z_{ij} = Z$ . Рівняння поверхні сегмента сфери в параметричній формі мають вигляд:

$$x_{ij} = (\rho^2 - H^2)^{\frac{1}{2}} \cos t \cos \varphi; \quad y_{ij} = (\rho^2 - H^2)^{\frac{1}{2}} \cos t \sin \varphi;$$

$$z_{ij} = [\rho^2 - (\rho^2 - H^2) \cos^2 t]^{\frac{1}{2}} - H; \quad 0 \leq t \leq 90^\circ; \quad 0 \leq \varphi \leq 360^\circ.$$

6. ЕКМ «Конус» (ЕКМК) (рис. 5.7, е). Параметри положення –  $X_{ij} = X, Y_{ij} = Y, Z_{ij} = Z$ . Рівняння бічної поверхні в параметричній формі мають вигляд:

$$x_{ij} = [(1 - \cos t) \cos \beta] R / 2; \quad z_{ij} = (1 + \cos t) H / 2; \quad 0 \leq t \leq 180^\circ; \quad 0 \leq \beta \leq 360^\circ.$$

Наведений ряд типів ЕКМ можна продовжити, наприклад, координатними моделями еліптичної, параболічної поверхонь другого порядку. Варіюючи параметрами форми, можна одержати деякі похідні цих типів ЕКМ.

**Приклад 5.8.** Розглянемо методику опису геометричних форм об'єктів методом ЕКМ на такому прикладі. Нехай необхідно побудувати координатну модель деталі, зображеної на рис. 5.8, а, б. Вибираємо положення базової системи координат так, щоб деталь розміщувалася в першому октанті за допомогою трьох ЕКМ:

1. ЕКМП з початком власної системи координат у точці  $O_{11}$ ; параметри положення –  $X_{11} = 22, Y_{11} = 30, Z_{11} = 5$ ; параметри форми –  $x_{11} = 18, y_{11} = 18, z_{11} = 18, \alpha = \beta = \gamma = 1$ .

2. ЕКМП з початком власної системи координат у точці  $O_{12}$ ; параметри положення –  $X_{12} = 31, Y_{12} = 39, Z_{12} = 23$ ; параметри форми –  $R = 15, H = 25, 0 \leq \varphi \leq 360^\circ$ .

3. ЕКМП з початком власної системи координат у точці  $O_{13}$ ; параметри положення –  $X_{13} = 31, Y_{13} = 39, Z_{13} = 48$ ; параметри форми –  $\rho = 22, 0 \leq \varphi \leq 360^\circ, 0 \leq \theta \leq 180^\circ$ .

Наведена сукупність ЕКМ повністю визначає геометрію об'єкта.

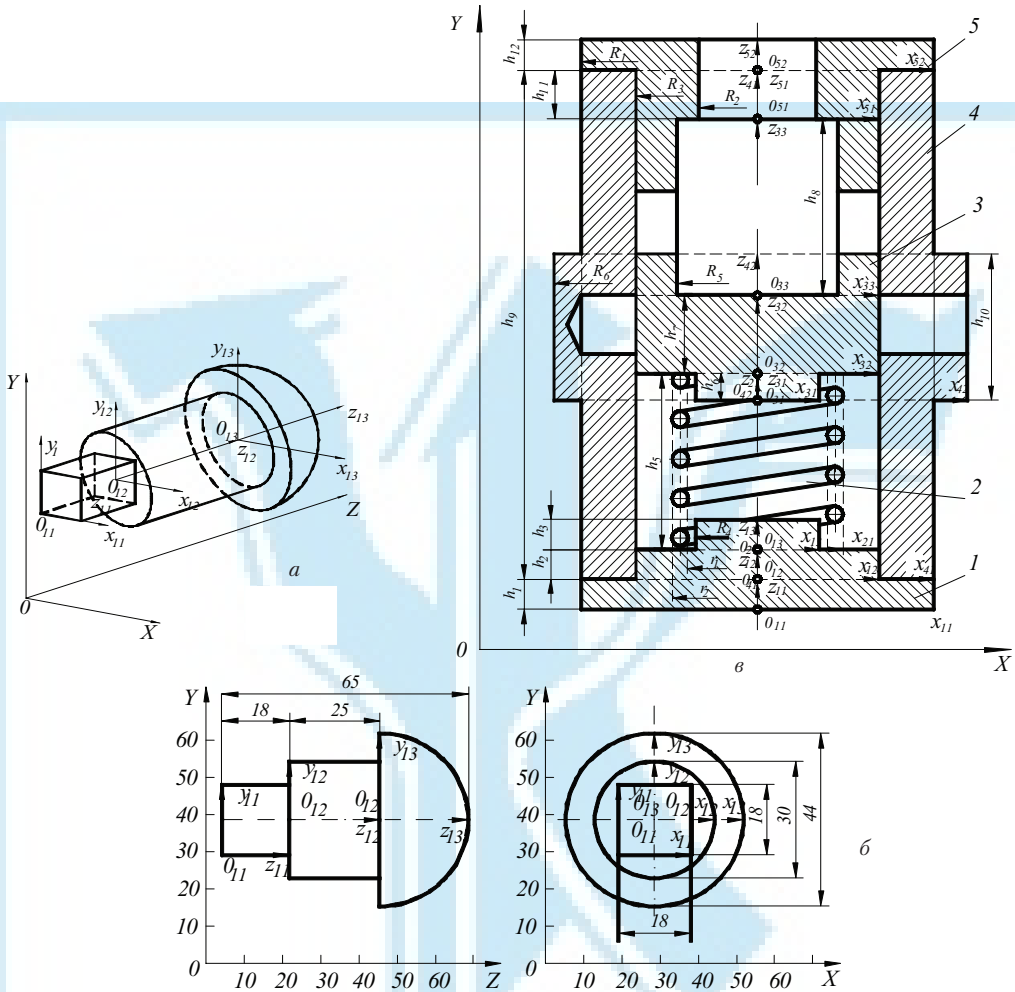


Рис. 5.8. Подання (а) та опис (б) деталі методом координатної моделі і складальна одиниця «Клапан редукційний» (в)

**Приклад 5.9.** Розглянемо опис складальної одиниці «Клапан редукційний», що складається з п'яти деталей (рис. 5.8, в). Кожну деталь розіб'ємо на ЕКМ. Результати формального опису складальної одиниці зведено в табл. 5.1.

Таблиця 5.1

Результати формального опису складальної одиниці

№ деталі	Номер і тип ЕКМ	Параметри положення	Параметри форми
1	2	3	4
1	1. ЕКМЦ	$O_{11}(X_{11}, Y_{11}, Z_{11})$	$R_1, h_1, \varphi$
	2. ЕКМЦ	$O_{12}(X_{12}, Y_{12}, Z_{12})$	$R_3, h_2, \varphi$
	3. ЕКМЦ	$O_{13}(X_{13}, Y_{13}, Z_{13})$	$R_1, h_3, \varphi$
2	1. ЕКМЦТ	$O_{21}(X_{21}, Y_{21}, Z_{21})$	$r_1, r_2, h_5, \varphi$

1	2	3	4
3	1. ЕКМЦ	$O_{31}(X_{31}, Y_{31}, Z_{31})$	$R_4, h_6, \varphi$
	2. ЕКМЦ	$O_{32}(X_{32}, Y_{32}, Z_{32})$	$R_3, h_7, \varphi$
	3. ЕКМЦТ	$O_{33}(X_{33}, Y_{33}, Z_{33})$	$R_5, R_3, h_8, \varphi$
4	1. ЕКМЦТ	$O_{41}(X_{41}, Y_{41}, Z_{41})$	$R_3, R_1, h_9, \varphi$
	2. ЕКМЦТ	$O_{42}(X_{42}, Y_{42}, Z_{42})$	$R_1, R_6, h_{10}, \varphi$
5	1. ЕКМЦТ	$O_{51}(X_{51}, Y_{51}, Z_{51})$	$R_2, R_3, h_{11}, \varphi$
	2. ЕКМЦТ	$O_{52}(X_{52}, Y_{52}, Z_{52})$	$R_2, R_1, h_{12}, \varphi$

Для розв'язання задач аналізу виробу, виходячи з умови його обслуговування промисловим роботом, використовується *метод формалізованого опису структури виробу скінченими неорієнтованими графами*, у яких вершини визначають помітні поверхні, а дуги — наявність з'єднання між цими поверхнями. Цей опис дає змогу класифікувати деталі за кількістю вершин і кількістю зв'язків у структурному графі, здійснювати аналіз і вибір вільних, наприклад, при складанні, поверхонь деталей, розв'язувати задачі ідентифікації.

## 5.6. Узагальнена концептуальна модель гнучкого виробництва

**Трактування поняття групової технології в умовах ГКІС.** Використання групової технології, яка полягає у формуванні номенклатури оброблюваних об'єктів виробництва (ОВ) із властивостями формоутворення, що забезпечуються наявними у виробничій системі обробляючими ресурсами, є обов'язковим компонентом вимог при створенні ГКІС. Інакше кажучи, склад обробляючих ресурсів системи повинен забезпечити весь набір технологічних операцій, які визначають виготовлення ОВ із заданими конструктивними параметрами та функціональними властивостями. Підготовка групового виробництва супроводжується побудовою так званого *віртуального ОВ*, який узагальнює в собі всі особливості технологічних операцій з виготовлення кожного з об'єктів, що включається в групу. В подальшому, в разі необхідності доповнення групи іншими ОВ або при формуванні нової групи об'єктів, які запускаються у виробництво, на основі матриць відповідності (в тому числі реляційних таблиць) або використання прийомів апарату теорії множин «просіюються» технологічні параметри усєї групи ОВ з визначенням їх відповідності наявним у системі ресурсам, які можуть формувати необхідний набір технологічних операцій, що забезпечують групу. Проте *загальна концепція створення ГКІС* при трактуванні поняття «*групова технологія*» ґрунтується на ширшому змісті, при якому до складу обробляючих ресурсів, крім власне формоутворюючих, додаються також ресурси *системи упорядкування середовища (СУС)*, які забезпечують упорядкування ОВ, тобто орієнтуючі, транспортні, завантажувальні та нагромаджувальні засоби, а також промислові роботи, які зокрема можуть виконувати й функції перелічених засобів СУС. Отже, група ОВ, яка виготовляється в ГКІС, повинна формуватися також і з урахуванням можливості їх обслуговування засобами СУС при переналадженні у межах технологічної і структурної гнучкостей ГКІС [19; 23].

Викладені нижче концептуальні засади щодо моделювання групового виробництва мають узагальнювальний характер, але, враховуючи особливості створення

ГКІС із широким трактуванням поняття «групова технологія», більша увага приділяється функціональному аналізу СУС.

За такої постановки на узагальненому *верхньому рівні* абстрагування концептуально *функція ГКІС* ( $\Phi_{\text{ГКІС}}$ -*функція*) (як і будь-якої організаційно-технічної системи) як *об'єкта керування* (ОК) подається декартовим добутком множин  $M, R, T$  та  $K$  [94] (рис. 5.9):

$$\Phi_{\text{ГКІС}} \in \{M, E, I\} \times R \times T \times K, \quad (5.34)$$

де  $\{M, E, I\}, R, T$  та  $K$  – відповідно підмножини об'єктів праці (де  $M$  – матеріали,  $E$  – енергія та  $I$  – інформація), способів їх реалізації (перетворення, або взаємодії знарядь і об'єктів праці), моментів часу (в які здійснюються відповідні перетворення об'єктів праці) та просторових координат, які займають об'єкти праці в робочій зоні ОК\*.

Залежність (5.34) відображає відомий постулат діалектичного матеріалізму щодо існування матерії у просторі та часі у своїх різноманітних формах прояву. Отже, як основа формалізації при кінематичному, геометричному, технологічному та структурному описах ГКІС як ОК може використовуватися *метод  $\Phi$ -функцій*, за якими встановлюється відповідність множин об'єктів праці, методів (технологій) і засобів реалізації їх перетворення, що відбувається у просторових координатах ОК у відповідні часові інтервали. Тоді метою етапів системного синтезу/аналізу ГКІС є визначення складу та закономірностей організації окремих компонентів у єдину систему – ОК з урахуванням того, що *функція ОК визначає його структуру*.

#### **Формалізований опис функціонального складу і структури робототехнічних комплексів і об'єктів роботизації**

Як основа формалізації при описі конструктивно-технологічної бази ОК – технологічної підсистеми ГКІС та її функціональних компонентів – *елементарних об'єктів* (ЕО) на виділених ієрархічних рівнях використовуються скінченні графи, що відображають властивості та взаємозв'язки ОК та ЕО [94].

При створенні та дослідженні ГКІС вирізняють дві постановки задачі:

- *пряму* (задача аналізу) – відомі структура та склад компонентів ГКІС. Треба визначити склад і напрямки руху матеріальних потоків;
- *обернену* (задача синтезу) – відомі склад і напрямки руху (технологічні маршрути виготовлення виробів) матеріальних потоків у часі. Треба визначити склад і структуру (розташування у просторі) складових компонентів – обробляючих ресурсів ГКІС.

Пряма задача виникає у процесі запуску нової номенклатури виробів в існуючому ТП ГКІС, а обернена – при створенні нової або суттєвій модернізації існуючого ТП ГКІС під певну номенклатуру виробів.

Отже, якщо загальна  $\Phi_{\text{ГКІС}}$ -*функція* ТП ГКІС як об'єкта керування може бути подана на вищому ієрархічному рівні абстрагування виразом (5.34), то підфункції – *виробничі процеси*  $\Phi_{\text{вир.пр}}$ -*процеси* ОК, призначені для зміни в часі предметів праці або їх просторових координат, мають відповідно вигляд:

$$\begin{aligned} \Phi_{\text{вир.пр1}}(R, K) &\subset M \times T; \\ \Phi_{\text{вир.пр2}}(T, K) &\subset M \times R; \\ &\dots\dots\dots; \\ \Phi_{\text{вир.пр14}}(M) &\subset R \times T \times K. \end{aligned} \quad (5.35)$$

Іншими словами, вирази (5.35) являють собою проекції функціональних залежностей, що описують  $\Phi_{\text{вир.пр}}$ -*процеси*, на координатні площини відповідно ТОМ,

\* Далі для скорочення запису будемо вживати позначення  $M$  як таке, що узагальнює енергетичні та інформаційні потоки ОВ.





ТОК, МОК, ..., MOR. Зокрема, серед виразів (5.35) є подання предметів праці і у просторовому, і в часовому розподілах відповідно:

$$\Phi_{\text{вир.пр}_i}(T) \subset (R \times M \times K), \quad (5.36)$$

$$\Phi_{\text{вир.пр}_i}(K) \subset (R \times M \times T). \quad (5.37)$$

З умов визначеності функціонального відображення (5.34) послідовність реалізацій  $\Phi_{\text{вир.пр}}$ -процесів описується орграфом\*  $c_1, c_2, c_3, c_4 \in M \times K$  (рис. 5.10, а), який являє собою **другий ієрархічний рівень** (процесний) подання функцій ОК. Однак при цьому не забезпечується однозначність технологій і технологічних процесів для різних партій об'єктів  $M$ , що обслуговуються.

Справді, якщо сутність технологічних процесів полягає у взаємодії знаряддя і предмета праці, то якісні зміни в їх розвитку, що характеризують перехід до нового класу процесів (до наступної стадії їх розвитку), полягають у докорінних змінах характеру цієї взаємодії.

Тому подальша деталізація функції (5.34) дає змогу виділити **третій рівень** – формального подання *технологій* ( $\Phi_{\text{техн}}$ -технології).

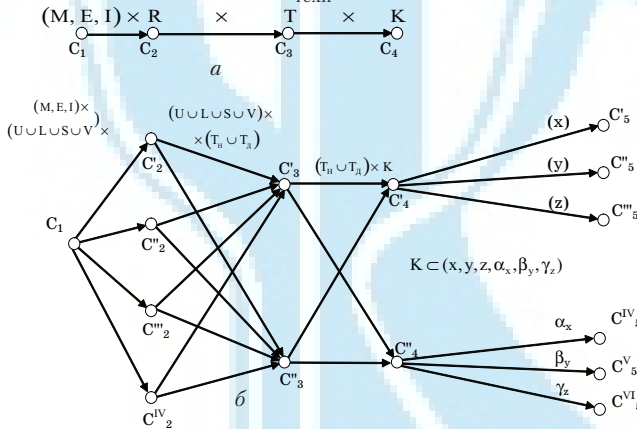


Рис. 5.10. Повний функціональний орграф виробничого процесу (а) та виробничої системи (б)

Для цього у виразі деталізується спосіб  $R$  взаємодії знарядь і об'єктів праці (рис. 5.10, б), який відображає якісні зміни в розвитку знарядь праці і визначає перехід до певного класу технологій (стадії їх розвитку).

Докорінні зміни *характеру взаємодії* описаних функціональних пристроїв/обробляючих ресурсів (П-ОР) та ОВ дають змогу виділити чотири класи технологій:

- **перший клас** (U-технології) характеризується точковим (індивідуальним) впливом  $R_T$  обробляючих ресурсів/пристроїв на окремі ОВ (наприклад, процеси точіння, хонінгування, поштучного завантаження та орієнтації тощо). Для операцій цього класу П-ОР характеризуються низькою продуктивністю;
- **другий клас** (L-технології) відображає лінійний вплив П-ОР на ОВ. При цьому одночасного впливу  $R_L$  зазнають декілька об'єктів, розташованих, наприклад, у ряд/лінію.

Прикладами таких технологій є фрезерування, волочіння, прокатування, безцентрове шліфування, одночасне орієнтування розташованих у ряд/стовпець

\* Орієнтованість орграфа може відображати послідовність виконання операцій і переходів на нижчих рівнях ієрархічного подання ГКІС.

ОВ тощо. Продуктивність у цьому разі збільшується пропорційно збільшенню кількості об'єктів, що зазнають одночасного впливу;

- для *S-технологій третього класу* характерним є поверхневий вплив  $R_{\Pi}$  функціональних пристроїв на ОВ — так званих засобів групового впливу (обслуговування) об'єктів, розташованих, наприклад, на поверхні П-ОР. Продуктивність підвищується пропорційно збільшенню робочої поверхні  $S$  обробляючих ресурсів/функціональних пристроїв (штамбування, пресування, лиття, плоскошліфувальні операції, технології з роторними лініями тощо);
- нарешті, *четвертий клас (V-технології)* характеризується просторовим впливом  $R_{\Pi P}$  функціональних пристроїв на ОВ, що обслуговуються. Прикладами можуть бути електрохімічна, термічна (сушіння, травлення, фарбування у ваннах) обробка, партійне завантаження тощо. Прикладом четвертого класу технологій служать пресо-штампувальні, ливарні та інші технології).

Для виділення підфункцій технологій — *технологічних процесів*  $\Phi_{\text{техн.пр}}$  -процесів на *четвертому ієрархічному рівні* необхідно деталізувати опис функції (5.34) за характером змін у часі — неперервним  $T_H$  або дискретним  $T_D$ , для чого у виразі (5.34) часова складова подається у вигляді  $T = T_H \cup T_D$  (рис. 5.10, б).

Описані вище класи технологій  $R \subset U \cup L \cup S \cup V$  не завжди забезпечують таке підвищення продуктивності, яке необхідне для ефективного застосування самих П-ОР. Це пов'язано не з технологією, а із засобами праці, якими реалізується та чи інша технологія. Основними суперечливими властивостями технологічної машини є *транспортні і технологічні рухи*: переміщення об'єкта через робочу зону машини і комплекс рухів (впливів), що супроводжують ці переміщення. Саме для третього  $S$  і, особливо, четвертого  $V$  класів технологій, які реалізуються відповідними П-ОР, забезпечується одночасний вплив на групу об'єктів, у зв'язку з чим зменшується операційний час і підвищується продуктивність процесу. Тому на *п'ятому рівні* виділяються *функції операцій* ( $\Phi_{\text{оп}}$ -операцій), для чого деталізуються координатні перетворення  $K$ , які необхідно подати у вигляді добутку  $K \subset \Pi \times P$  де  $\Pi$  — множина лінійних координат, а  $P$  — множина кутових (ротаційних) координат (рис. 5.10, б):

$$\Phi_{\text{оп}} \subset M \times (U \cup L \cup S \cup V) \times (T_H \cup T_D) \times (\Pi \cup P). \quad (5.38)$$

$$\left. \begin{aligned} \Phi_{\text{оп}}(P) &\subset (U \cup L \cup S \cup V) \times (T_H \cup T_D) \times M \times \Pi; \\ \Phi_{\text{оп}}(\Pi) &\subset (U \cup L \cup S \cup V) \times (T_H \cup T_D) \times M \times P; \\ \Phi_{\text{оп}}(M) &\subset (U \cup L \cup S \cup V) \times (T_H \cup T_D) \times \Pi \times P; \\ \Phi_{\text{оп}}(T) &\subset (U \cup L \cup S \cup V) \times M \times \Pi \times P; \\ \Phi_{\text{оп}}(M, P) &\subset (U \cup L \cup S \cup V) \times (T_H \cup T_D) \times \Pi; \\ &\dots; \\ \Phi_{\text{оп}}(U \cup L \cup S \cup V) &\subset (T_H \cup T_D) \times M \times (\Pi \cup P) \end{aligned} \right\}. \quad (5.39)$$

Одержана з виразу (5.38) система (5.39) визначає  $\Phi_{\text{оп}}$ -операції шляхом встановлення у часі відповідності: між елементами множин  $M$  і  $\Pi$  при несуттєвій зміні  $P$  або фіксованих його значеннях; між елементами множин  $M$  і  $P$  при несуттєвій зміні  $\Pi$ ; між лінійними  $\Pi$  та кутовими  $P$  координатами елементів тощо. Таким чином, множина (5.39)  $\Phi_5$ -операцій, утворюючи п'ятий ієрархічний рівень функцій ГКІС, визначає й типи функціональних П-ОР (зокрема, встановлення у часі відповідності кутових і лінійних координат, тобто *РТПР-операція* реалізується *РТПР-пристроєм*; *РТМП-операція* реалізується *РТМП-пристроєм* тощо), що належать виконавчій частині системи — ТП ГКІС. Порядок виконання  $\Phi_{\text{оп}}$ -операцій впливає з визначеності виразу (5.38) і подається орграфом  $c_1, c_2, \dots, c_5 \in M \times (U \cup L \cup S \cup V) \times (T_H \cup T_D) \times (\Pi \cup P)$ , зображеним на рис. 5.10, б, в якому

$$\begin{aligned}
 \Phi_1 &\subset (U \cup L \cup S \cup V) \times (T_H \cup T_D) \times M \times \Pi; \\
 \Phi_2 &\subset (U \cup L \cup S \cup V) \times (T_H \cup T_D) \times M; \\
 \Phi_3 &\subset [(U \cup L \cup S \cup V) \times (T_H \cup T_D) \times M] \cup [(U \cup L \cup S \cup V) \times (T_H \cup T_D) \times \Pi]; \\
 \Phi_4 &\subset [(U \cup L \cup S \cup V) \times (T_H \cup T_D)] \times \Pi \times R; \\
 \Phi_5 &\subset [M \times (U \cup L \cup S \cup V) \times (T_H \cup T_D)] \times \Pi \times R.
 \end{aligned}
 \tag{5.40}$$

Виділення п'ятого рівня ієрархії, що характеризує виконавчу частину ТП ГКІС, є надзвичайно важливим етапом системного аналізу/синтезу ТП ГКІС. Синтезовані на цьому етапі типи функціональних П-ОР є *необхідними і достатніми* для розв'язання прикладних задач побудови системи при обумовленому характері взаємодії R між П-ОР та ОВ.

**Шостий ієрархічний рівень** деталізації (уточнення) функції ГКІС як ОК відповідає *функціональним задачам-переходам* ( $\Phi_{\text{зад-задачах}}$ ), які реалізуються елементарними рухами робочих органів П-ОР. Множина ( $\Phi_{\text{зад-задачах}}$ ) формується з виразів (5.39) шляхом уточнення множин  $\Pi$  і  $P$ :

$$\Pi = X \times Y \times Z; \quad B = a \times b \times g,$$

де  $X, Y, Z$  – множини лінійних координат за відповідними осями;  $a, b, g$  – множини кутових координат відносно цих же осей.

Зокрема, в правій частині системи (5.38) при цьому утвориться декартовий добуток  $(T_H \cup T_D) \times [(X \times Y \times Z) \times (a \times b \times g)]$ , з якого можна одержати  $2^{n-1} - 1 = 2^{8-1} - 1 = 127$  фактор-множин, тобто 127 ( $\Phi_{\text{зад-задачах}}$ ), ( $n$  – кількість множників декартового добутку). Зокрема, множину  $\Phi_{\text{зад-задачах}}$  для РТПР-операції можна розбити на такі класи (за числами  $\Pi$  і  $P$ ):

$$\begin{aligned}
 &R \times T \times \Pi; \quad R \times T \times \Pi^2; \quad R \times T \times \Pi^3; \quad R \times T \times \Pi \times P; \quad R \times T \times \Pi^2 \times P; \\
 &R \times T \times \Pi^3 \times P; \quad R \times T \times \Pi \times P^2; \quad R \times T \times \Pi \times P^3; \quad R \times T \times \Pi^2 \times P^2; \\
 &R \times T \times \Pi^3 \times P^2; \quad R \times T \times \Pi^2 \times P^3; \quad R \times T \times \Pi^3 \times P^3; \quad R \times T \times P; \quad R \times T \times P^2; \\
 &R \times T \times P^3.
 \end{aligned}
 \tag{5.41}$$

Система (5.41) та рис. 5.10, б дають уявлення про *повний функціональний оргграф* (ПФО) ГКІС. Порівнюючи класи  $\Phi_{\text{зад-задачах}}$  (5.41) для  $\Phi_{\text{оп-операцій}}$  (5.39), можна, використовуючи кола Ейлера, побудувати розподіл (рис. 5.11), який показує, що у реальних технологічних системах деякі  $\Phi_{\text{оп-операцій}}$  можуть не виконуватися, а деякі ланцюжки  $\Phi_{\text{оп-операцій}}$  можуть виконуватися одним комбінованим функціональним пристроєм (наприклад, РТПР-*пристрої* можуть виконувати  $\Phi_{\text{зад-задачах}}$  РТП- і РТР-*пристроїв*, оскільки класи  $\Phi_{\text{зад-задачах}}$  у них перетинаються).

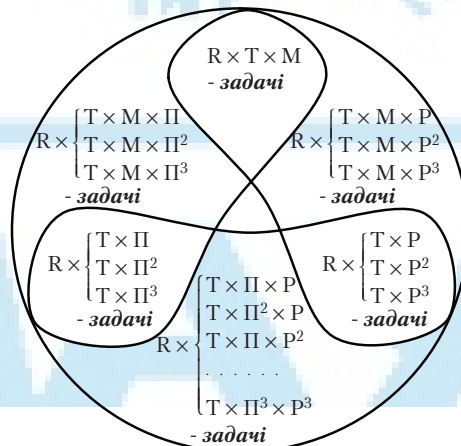


Рис. 5.11. Розподіл ейлерового простору  $\Phi_{\text{ГКІС-функції}}$  на підфункції  $\Phi_{\text{зад-задачах-переходів}}$

**Гіперпросторове подання процесу формування обробляючих ресурсів для групової технології**

Процедура формування складу технічних засобів ґрунтується на використанні *об'єктно-орієнтованого принципу групування (ООПГ)*, в основі якого лежить теоретико-множинний підхід [94]. Загальна схема формування групи ОВ спочатку полягає у визначенні типу розв'язуваної задачі та побудові інформаційних моделей або *пошукових образів об'єктів (ПОО)*: виробництва (ОВ); виробничих процесів ( $\Phi_{\text{вир.пр}}$ -процесів); технологій ( $\Phi_{\text{техн}}$ -технологій), що їх реалізують; технологічних процесів  $\Phi_{\text{техн.пр}}$ -процесів обробляючих ресурсів П-ОР тощо.

Формування ПОО здійснюється визначенням характеристик (ознак) об'єктів, необхідних для успішного виконання пошукових процедур. Необхідно зазначити, що при проектуванні ГКІС встановлюють і формалізовано подають лише ті властивості ОВ, які певним чином впливають на розроблювану систему.

Наведений функціонально-структурний аналіз встановлює багаторівневу ієрархічну структуру ТП ГКІС, а елементи на цих рівнях і закономірності утворення систем з цих елементів відповідають їх функціональним призначенням (рис. 5.12).

Наприклад, розглядаючи складальний виріб як механічну систему тіл і користуючись критерієм одноваріантності її поділу на елементи, можна виділити ряд рівнів цієї ієрархічної системи (рис. 5.13, а).

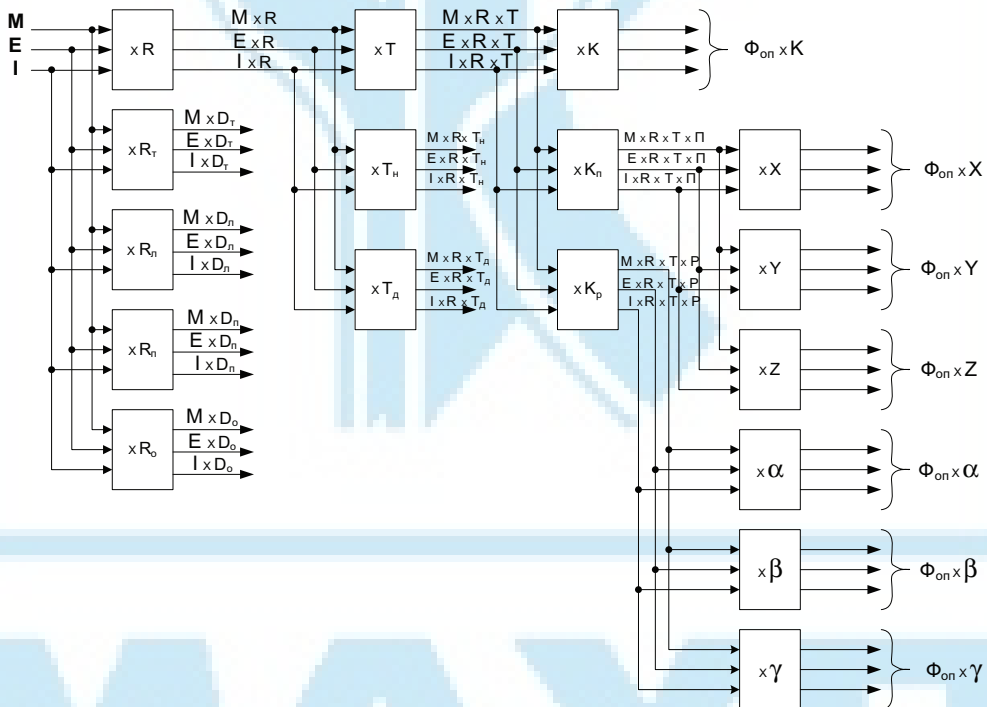


Рис. 5.12. Багаторівнева ієрархічна структура ТП ГКІС

При такому способі виділення ієрархічних рівнів складальні одиниці є підсистемами і залежно від завдання проектування можуть розглядатися або як вироби (об'єкти виробництва ОВ), або як деталі (елементарні об'єкти ЕО). Контури, тобто сукупності поверхонь, об'єднані певними властивостями, також розглядаються як підсистеми, тобто ЕО.

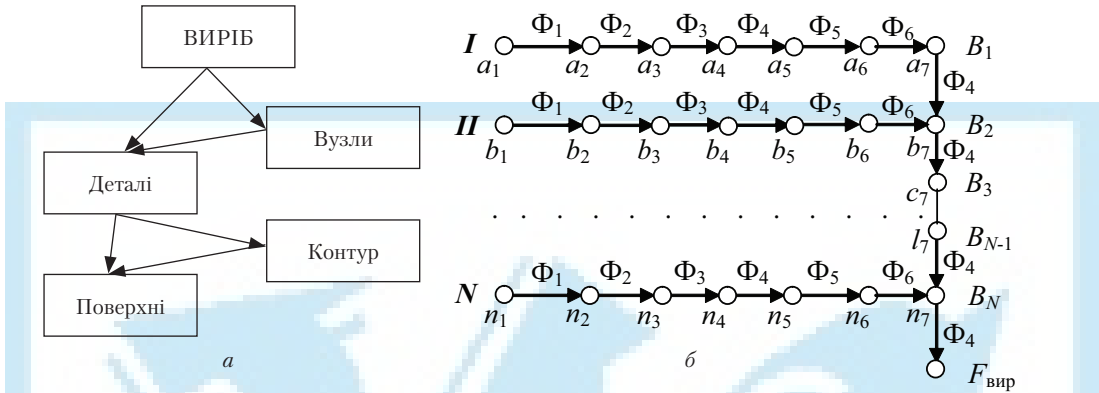


Рис. 5.13. Рівні ієрархії механічної системи – складальної одиниці (а) та її повний функціональний орграф (б)

З огляду на характер  $\Phi_{\text{оп}}$ -операцій, якими супроводжується  $\Phi_{\text{техн.пр}}$ -процес виготовлення виробу, ПФО (рис. 5.13, б) виробу включає обробку  $N$  комплектуючих складових з відповідними паралельними гілками  $a_1 - a_7; b_1 - b_7; \dots; n_1 - n_7$  з  $\Phi_{\text{оп}}$ -операцій  $\Phi_1 - \Phi_6$ , власне складання комплектуючих у вузли  $B_1 - B_N$  і, зрештою, останніх у виріб  $F_{\text{вир}}$  (вертикальна гілка  $\Phi_{\text{оп}}$ -операцій).

При цьому ПФО виробу можна розбити на дві частини: в одній зв'язки  $\Phi_1, \Phi_2, \Phi_3$ , а в іншій – решта зв'язків.

Сукупність  $\Phi_{\text{оп}}^1$ -операцій першої частини реалізує система упорядкування середовища (СУС), а сукупність  $\Phi_{\text{оп}}^2$ -операцій другої частини – гнучка складальна технологічна позиція. При використанні ПР як складального ТПВ-пристрою її називають роботизованою технологічною позицією (РТП).

Система СУС і РТП утворюють роботизовану технологічну одиницю (РТО), що виконує одну технологічну операцію у ГКІС.

Графічна інтерпретація формування об'єктно-орієнтованого складу технічних засобів ГКІС через перетин множин ознак  $OB - \Phi_{\text{техн.пр}}$ -процесів –  $\Phi_{\text{оп}}$ -операцій –  $\Phi_{\text{зад}}$ -задач – методів  $R$  обслуговування  $OB$  – що реалізують методи пристроїв П-ОР, може бути подана у вигляді гіперпросторової моделі [1] (на рис. 5.14 відображена тільки та частина узагальненої моделі, яка інтерпретує об'єктну орієнтованість ознак  $OB$  і властивості компонентів СУС, що обслуговують ці ознаки).

Тоді процедура прийняття рішень  $Z$ , під якими розумітимемо відповідний П-ОР, що є зіставленням множин  $OB, \Phi_{\text{техн.пр}}$ -процесів,  $\Phi_{\text{оп}}$ -операцій,  $\Phi_{\text{зад}}$ -задач,  $R, П-ОР$ , а також відшукуванням між ними відповідності  $C$ , математично може бути подана таким виразом:

$$C \subseteq OB \times \Phi_{\text{техн.пр}}\text{-процеси} \times \Phi_{\text{оп}}\text{-операції} \times \Phi_{\text{зад}}\text{-задачі} \times R \times П-ОР. \quad (5.42)$$

Алгоритмічно процес вибору типу П-ОР здійснюється в два етапи. На початку першого етапу відбувається формування маскуючих масивів, структура яких аналогічна кортежним відношенням П-ОР у базі даних. Для цього ПОО, який являє собою набір

$$ПОО := \langle OB_p, \Phi_{\text{техн.пр}_p}\text{-процеси}, \Phi_{\text{оп}_k}\text{-операції}, \Phi_{\text{зад}_l}\text{-задачі}, R_g \rangle, \quad (5.43)$$

згідно із зазначенням пріоритетом ознак перетворюється у набір

$$ПОО_n := \langle OB_{i_{\text{пр}}}, \Phi_{j_{\text{пр}}}\text{-процеси}, \Phi_{k_{\text{пр}}}\text{-операції}, \Phi_{l_{\text{пр}}}\text{-задачі}, R_{g_{\text{пр}}} \rangle \cap$$

$$\cap \langle \text{ОВ}_{i_{\text{пр}}}, \Phi_{j_{\text{пр}}}\text{-процеси} - \Phi_{j_{\text{пр}}}\text{-процеси}, \Phi_k\text{-операції} - \Phi_{k_{\text{пр}}}\text{-операції}, \Phi_{\Gamma}\text{-задачі} - \Phi_{\Gamma_{\text{пр}}}\text{-задачі}, R_g - R_{g_{\text{пр}}} \rangle,$$

де  $\langle \text{ОВ}_{i_{\text{пр}}}, \Phi_{j_{\text{пр}}}\text{-процеси}, \Phi_{k_{\text{пр}}}\text{-операції}, \Phi_{\Gamma_{\text{пр}}}\text{-задачі}, R_{g_{\text{пр}}} \rangle$  – множина таких ознак.

На наступному кроці відбувається порівняння пріоритетних ознак з відношеннями підмножини декартового добутку

$$\text{ОВ}_m \times \Phi_n\text{-процеси} \times \Phi_p\text{-операції} \times \Phi_{\Gamma}\text{-задачі} \times R_d \times \text{П-ОР}_h,$$

який відображає взаємозв'язок властивостей П-ОР і множини значень технологічної ситуації, де П-ОР<sub>h</sub> – пристрій з h-ю властивістю.

В результаті реалізації цього кроку формується множина маскувальних кортежів у вигляді обслуговуючих властивостей П-ОР стосовно відповідних ознак ОВ.

У подальшому здійснюється почергове зіставлення маскувальних кортежів із наборами відношення, яким подаються різні моделі П-ОР, та формування масивів, що містять інформацію про П-ОР<sup>f</sup>, які задовольняють функціональні, технологічні та інші вимоги. Розв'язок записується як перетин цих множин для пріоритетних (основних) ознак:

$$\text{П-ОР}_{\text{пр}}^f = \bigcap_{i=1}^G \text{П-ОР}_{i_{\text{пр}}}^f.$$

Аналогічна процедура застосовується до групи непріоритетних (додаткових) ознак, і остаточна відповідь визначається перетином множин:

$$\text{П-ОР}^f = \text{П-ОР}_{i_{\text{пр}}}^f \cap \text{П-ОР}_{i_{\text{пр}}}^f.$$

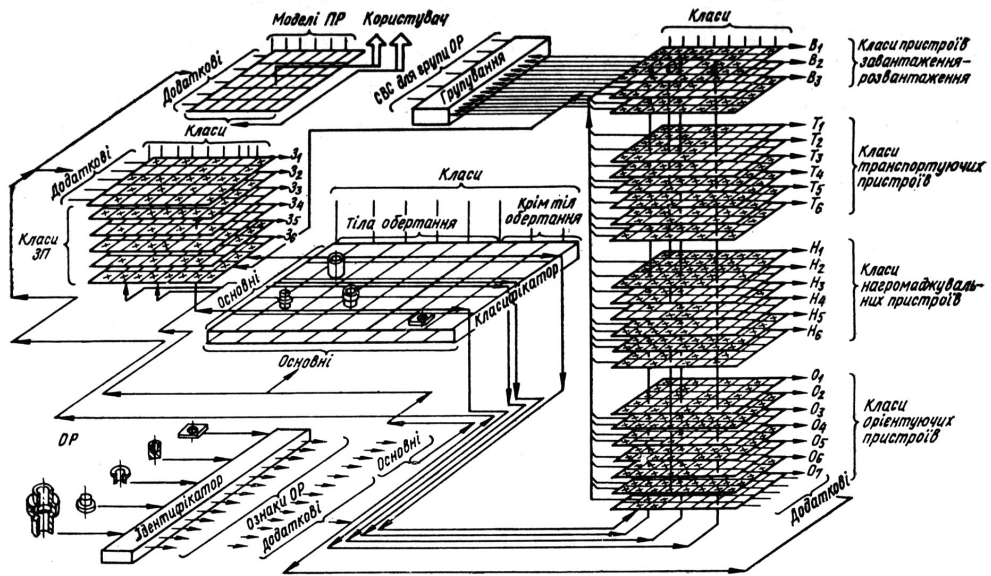


Рис. 5.14. Приклад відображення процесу формування складу технічних засобів

Як приклад відображення процесу формування групи ОВ, які обслуговуються складом технічних засобів СУС при використанні ООПГ, можна скористатися наведеною на рис. 5.15 схемою, яка наочно відтворює процедуру формування і зістав-

лення маскувальних кортежів у вигляді обслуговуючих властивостей П-ОР та відповідних ознак ОВ [1; 19].

Формування складу технічних засобів для обслуговування комплексу деталей складального виробу (відображення процесу формування наведено на рис. 5.15) здійснюється з урахуванням аналізу їх технологічності.

Деталі, які комплектують виріб, після механічної або термічної обробки подаються на ідентифікатор —  $n \times m$ -вимірну матрицю (наприклад, фотометричну) або задаються оператором у вигляді  $n \times m$ -вимірного опису, на виході яких формується склад основних і допоміжних ознак деталей [1]. Ці ознаки застосовуються для класифікації або ототожнювання властивостей деталей, що надійшли на вхід ідентифікатора, з властивостями еталонних описів, розміщених у класифікаторі для кожного класу. В свою чергу, для кожного з еталонів деталей у БД орієнтуючих, транспортувальних, завантажувальних, нагромаджувальних пристроїв і пристроїв поштучного відокремлення і завантаження деталей на робочу позицію, а також захватних та інших робочих пристроїв промислових роботів визначаються типові пристрої перелічених класів, які можуть обслуговувати еталони ідентифікованих деталей.

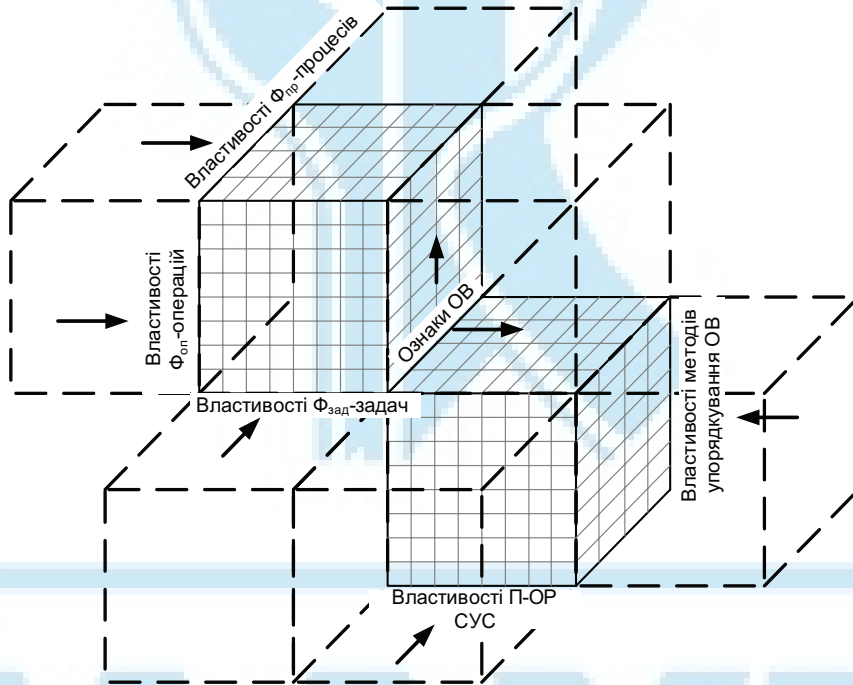


Рис. 5.15. Відображення процесу формування групи ОВ

Далі, використовуючи додаткові ознаки (наприклад, фізико-механічні, магнітні та електропровідні властивості, наявність та якість виготовлення базових поверхонь для накладання утримуючих зв'язків при захопленні деталей захватними пристроями тощо), про які інформує ідентифікатор або оператор, уточнюються конкретні моделі пристроїв для обслуговування всієї номенклатури деталей, що комплектують складальний виріб.



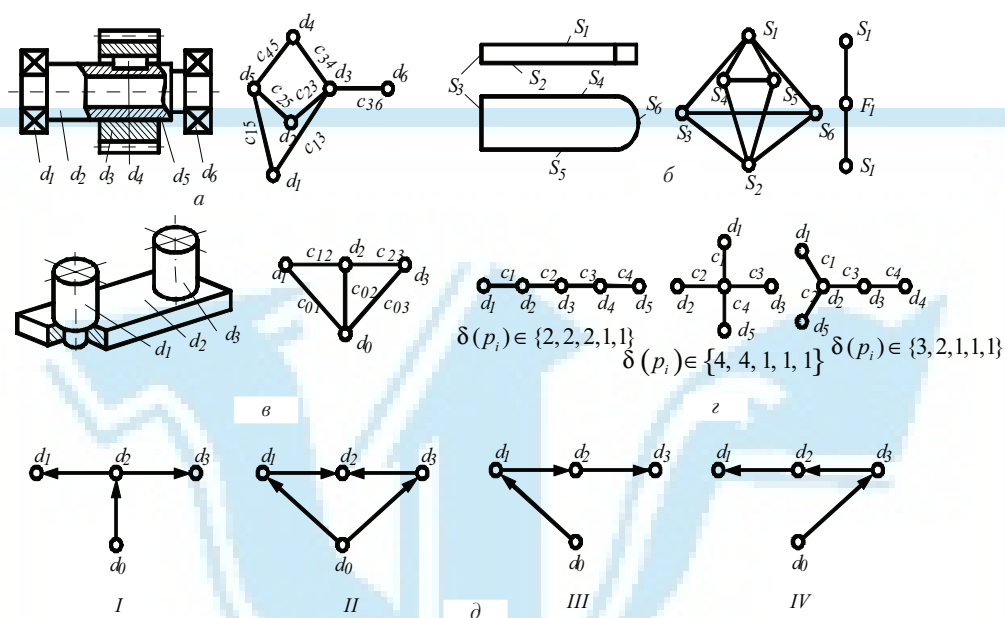


Рис. 5.16. Формалізований опис у вигляді скінченних графів структури виробів (а), деталей (б), геометричних образів об'єктів роботизації (в) та варіантів маршрутів складання виробу (z, д)

Структурні властивості об'єктів керування (РТК, їх елементів, виробів як об'єктів роботизації) на виділених ієрархічних рівнях зручно описувати скінченними графами. Так, на рис. 5.16, а показано структурний граф виробу, що відображає взаємозв'язок його елементів.

Приклад формалізованого опису деталей показано на рис. 5.16, б. Об'єднання поверхонь  $S_3, S_4, S_5, S_6$  у контур  $F_1$  дає змогу одержати граф деталі. Формалізований опис структури виробів дає змогу класифікувати їх за кількістю вершин і кількістю зв'язків у структурному графі виробу.

Для графа, що має  $p$  вершин, мінімальна кількість зв'язків дорівнює  $q_{\min} = p - 1$ , а максимальна кількість зв'язків  $q_{\max} = \frac{p!}{2!(p-2)!}$ .

Деякі  $pq$ -графі можуть мати різні графічні зображення, наприклад, 54-граф має три графічні образи (рис. 5.16, z). Тому для кожного геометричного образу необхідно вказувати ступені вершин або кількість ребер, зв'язаних з кожною вершиною.

Розглянемо як приклад синтез множини варіантів маршрутів складання виробу.

Запропонований метод синтезу відрізняється від описаного раніше (див. рис. 5.13, б) доповненням структурного графа вершиною  $d_0$ , що має зв'язки з усіма вершинами графа.

На рис. 5.16, в показано виріб і його структурний граф, доповнений вершиною  $d_0$ , що береться за базову. Всі інші вершини можуть бути сполучені з першою одна за одною, а кожна з тих вершин, що залишилися, може бути сполучена з отриманою раніше сукупністю одна за одною і т. д. Оскільки перебору підлягають тільки вершини  $d_1, d_2$  і  $d_3$ , то кількість перестановок дорівнює  $3! = 6$ . Запишемо можливі маршрути:

- 1)  $d_0, d_1, d_2, d_3$ ; 2)  $d_0, d_1, d_3, d_2$ ; 3)  $d_0, d_2, d_1, d_3$ ; 4)  $d_0, d_2, d_3, d_1$ ;  
 5)  $d_0, d_3, d_1, d_2$ ; 6)  $d_0, d_3, d_2, d_1$ .

Отримані маршрути оцінюються за умовами доступу, визначеності базування, характеристиками сполучення і кількістю одночасно встановлюваних зв'язків. Оскільки конструкція виробу за умовами доступу і повноти базування допускає всі маршрути, то за третьою і четвертою умовами можна утворити такі групи маршрутів: № 3 і № 4 з однаковими рухами сполучення і з послідовно встановлюваними зв'язками (поз. I на рис. 5.16, д); № 2 і № 5 із двома одночасно встановлюваними зв'язками (поз. II на рис. 5.16, д); № 1 і № 6 з різними рухами сполучень деталей  $d_1$  і  $d_3$  (поз. III та IV на рис. 5.16, д).


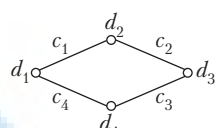
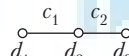
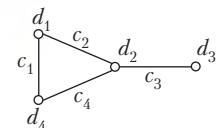
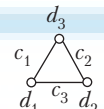
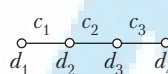
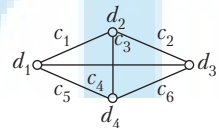
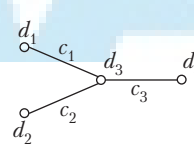
Оптимізація груп маршрутів сполучення деталей визначається можливостями їх реалізації в процесі виконання транспортних та орієнтуючих рухів ПР, тобто можливостями кінематичної структури і системи керування, що обслуговує складальний процес ПР.

Розглянуті положення дають змогу упорядкувати процес проектування РТК на етапі структурно-функціонального синтезу.

У табл. 5.2 наведено структурні граfi виробів, що складаються з двох, трьох і чотирьох деталей.

Таблиця 5.2

Структурні граfi виробів

$p\left(\frac{q_{\min}}{q_{\max}}\right)$	$q$	Степені вершини	Структурний граф	$p\left(\frac{q_{\min}}{q_{\max}}\right)$	$q$	Степені вершини	Структурний граф
$2\left(\frac{1}{1}\right)$	1	(1,1)			4	(2,2,2,2)	
$3\left(\frac{2}{3}\right)$	2	(1,2,1)		4	$\left(\frac{3}{6}\right)$	(3,2,2,1)	
	3	(2,2,2)				5	(3,3,2,2)
$4\left(\frac{3}{6}\right)$	3	(1,1,2,2)		6	(3,3,3,3)		
		(3,1,1,1)					

### 5.7. Формалізований опис ознак об'єктів та оперативний аналіз сенсорної інформації

Ефективність процесів розпізнавання об'єктів роботизації (ОР) визначається організацією процедур обробки зорової інформації, які охоплюють: формування інформаційної моделі середовища; оперативний аналіз сенсорної інформації; взаємодію сенсорних даних з моделлю середовища; формування формалізованих даних про ОР та прийняття рішень для створення керуючих команд. Найбільш відповідальним етапом побудови моделі середовища є процес вибору системи ознак для опису ОР.

**Методи формалізації опису ознак об'єктів ГКІС.** Вирізняють п'ять основних груп ознак об'єктів роботизації: 1) матричні характеристики — периметр, площа та інші геометричні оцінки форми; 2) моментні інваріанти — інтегральні характеристики; 3) характеристики аналітичного перетворення зображень — виділені гармоніки з дискретного спектра періодичного сигналу, що відповідає розгортанню контуру проекції об'єкта на часовій осі; 4) структурні і топологічні ознаки — подання за допомогою графів технологічного зв'язку елементів тривимірного об'єкта (для бінарних зображень об'єктів встановлення топології полягає в побудові зв'язків довільної точки з точками заданого растру); 5) логічні ознаки — логічні функції подання типу й орієнтації об'єкта на розпізнавальній площині.

Найбільш природним є опис *геометричних величин* об'єктів (зокрема, див. підрозд. 5.5). При мінімізації матричних ознак об'єкта часто використовують функціонально незалежні характеристики  $c_i$ , інваріантні до перетворення подібності:

$$c_1 = l_1 l_2 / s; \quad c_2 = l_1 l_2; \quad c_3 = p / l,$$

де  $l_1, l_2$  — більша та менша сторони мінімального за площею прямокутника, описаного навколо зображення об'єкта;  $s, p$  — площа та периметр об'єкта.

Існує понад 15 ознак, за допомогою яких можна скласти набір, необхідний для розпізнавання групи об'єктів. Позитивна якість матричних характеристик полягає в тому, що вони інваріантні до зсувів і поворотів об'єктів, а також дзеркальних відображень їх проекційних зображень. Суттєвий недолік цієї системи ознак об'єктів полягає у вимогах високої роздільної здатності для одержання достовірних оцінок.

Найчастіше при розпізнаванні орієнтації об'єктів використовують *моментні інваріанти*, які дають змогу визначати координати «центра ваги» зображення, напрямки головних осей інерції та ін. Для аналізу багатоградаційних за яскравістю зображень об'єктів обчислюють моменти, які єдиним способом визначають функцію яскравості  $B_{ij}$ , таким чином, дають повний опис об'єкта.

Момент  $J_{n,m}$  порядку  $(n, m)$

$$J_{n,m} = \sum x_i^n y_j^m B_{ij}; \quad n, m = 0, 1, 2 \quad (5.44)$$

задається в площі зображень  $(x, y)$ , причому момент нульового порядку визначає площу фігури, а моменти  $J_{01}$  та  $J_{10}$  характеризують положення її центра ваги.

Моменти другого порядку

$$\begin{cases} J_{20} = \sum_{ij} x_i^2 B_{ij} - (1/s) J_{10}^2; \\ J_{02} = \sum_{ij} y_j^2 B_{ij} - (1/s) J_{01}^2; \\ J_{11} = \sum_{ij} x_i y_j B_{ij} - (1/s) J_{10} J_{01} \end{cases} \quad (5.45)$$

використовують для визначення ознак  $c_1, c_2$ , інваріантних до зсувів і повороту фігури:

$$c_1 = [1/(2s)] \left\{ J_{01} + J_{10} + [(J_{01} + J_{10})^2 + 4J_{11}^2]^{1/2} \right\}; \quad (5.46)$$

$$c_2 = [1/(2s)] \left\{ J_{02} + J_{20} + [(J_{02} + J_{20})^2 + 4J_{11}^2]^{1/2} \right\}. \quad (5.47)$$

Орієнтація об'єкта визначається значенням кута  $\varphi = \frac{1}{2} \arctg \frac{2J_{11}}{J_{20} - J_{02}}$ , який вказує напрямок осі з більшим моментом відносно координати  $i$ . Для бінарних зображень об'єктів аналогічні інваріанти обчислюються як числа матриці моментів другого порядку.

Розпізнавання об'єктів та їх положення за допомогою моментних характеристик виконується з високою точністю, і в цьому полягає перевага використання методу. Проте, як і для матричних характеристик, тут також ставляться високі вимоги щодо ступеня роздільної здатності зображень і потрібне багаторазове сканування усіх рецепторних точок.

При формуванні *характеристик аналітичних перетворень* (див. підрозд. 5.5) дискретний сигнал  $F_2(t)$  апроксимує форму  $F_1(t)$  контуру об'єкта з похибкою  $e(t)$ :

$$\left. \begin{aligned} F_1(t) &\approx K_{12} F_2(t); \\ e(t) &= F_1(t) - g_{12} F_2(t); \\ g_{12} &= \left[ \int_T F_1(t) F_2(t) dt \right] / \left[ \int_T F_2^2(t) dt \right] \end{aligned} \right\}. \quad (5.48)$$

Інформацію про зображення об'єкта можна подати у вигляді масиву координат  $F, t$  апроксимуючого сигналу. Тоді функція, що характеризує контур, задається відношенням  $\rho(\varphi) = r / [r(\varphi)]$ , яке може бути подано рядом Фур'є:

$$\rho(\varphi) = C_0 / 2 + \sum_{i=1}^{\infty} C_i \sin(i\varphi + \alpha_i), \quad (5.49)$$

де  $r$  — периметр контуру;  $r(\varphi)$  — полярне рівняння контуру;  $C_0, C_i$  — коефіцієнти розвитку (значення амплітуд). Ці коефіцієнти, а також фази  $\alpha_i$  можуть бути ознаками опису об'єкта, бо при  $i \rightarrow \infty$  вони однозначно визначають контур.

Інваріантність аналітичних характеристик перетворень зображень об'єктів до зміщень забезпечується тим, що контур описується відносно центра ваги. Значення кутового зміщення визначається коефіцієнтами  $\alpha_i$ , а введення безрозмірного відношення  $r(\varphi)$  забезпечує інваріантність до масштабного фактора.

Графи об'єктів і топологічні схеми [23] при застосуванні методу **топологічних і структурних ознак** дають повне уявлення про зображення об'єктів. Вони інваріантні до його афінних перетворень, але мають значну надмірність через багаторазове врахування топологічних зв'язків. Відмінність між топологічними та структурними ознаками полягає в тому, що перші дають змогу розпізнавати об'єкти не за кількісними, а за якісними оцінками їх конфігурації, а це переважно збільшує точність і вірогідність ототожнення об'єктів.

Коли об'єкт займає в робочій зоні фіксовану кількість положень, економічними засобами опису його типу й орієнтації є **логічні ознаки**. Для задавання моделі рухомих об'єктів набір логічних ознак можна дістати синхроімпульсним методом

з виділенням характерних точок (областей), суть якого полягає в знаходженні точок зображення об'єкта, що однозначно характеризують його положення відносно деякої синхроточки. Надання чутливості цим точкам за допомогою рецепторів дає змогу однозначно визначити тип об'єкта та його орієнтацію. Користуючись логічними ознаками об'єкта, можна здійснювати контроль практично ненадлишкової інформації про нього, а інформативна місткість системи залежить від кількості розпізнавальних класів об'єктів та кількості допустимих помітних їх положень у полі сенсора [18].

**Вимоги до ідентифікації об'єктів ГКІС.** При роботизації ТП основну увагу приділяють найбільш трудомістким їх елементам. Одним з таких елементів (зокрема для складальних процесів) є орієнтування об'єктів у просторі та відносне орієнтування їх і робочих органів устаткування (або спряжуваних при складанні компонентів). Трудомісткість таких операцій значною мірою визначається можливостями розпізнавання образу об'єкта та його положень у просторі чи на робочій позиції, що зумовлено як характеристиками самого об'єкта моделювання (ОМ), так і існуючими методами та засобами розпізнавання.

Цій вимозі найбільшою мірою відповідають системи технічного зору (СТЗ), використання яких у ГКІС пояснюється їх універсальністю, багатofункціональністю, порівняно простими засобами спряження з обчислювальними засобами та елементами ТО. На рис. 5.17 подано класифікацію СТЗ, де використовуються найбільш уживані в ГКІС ознаки.

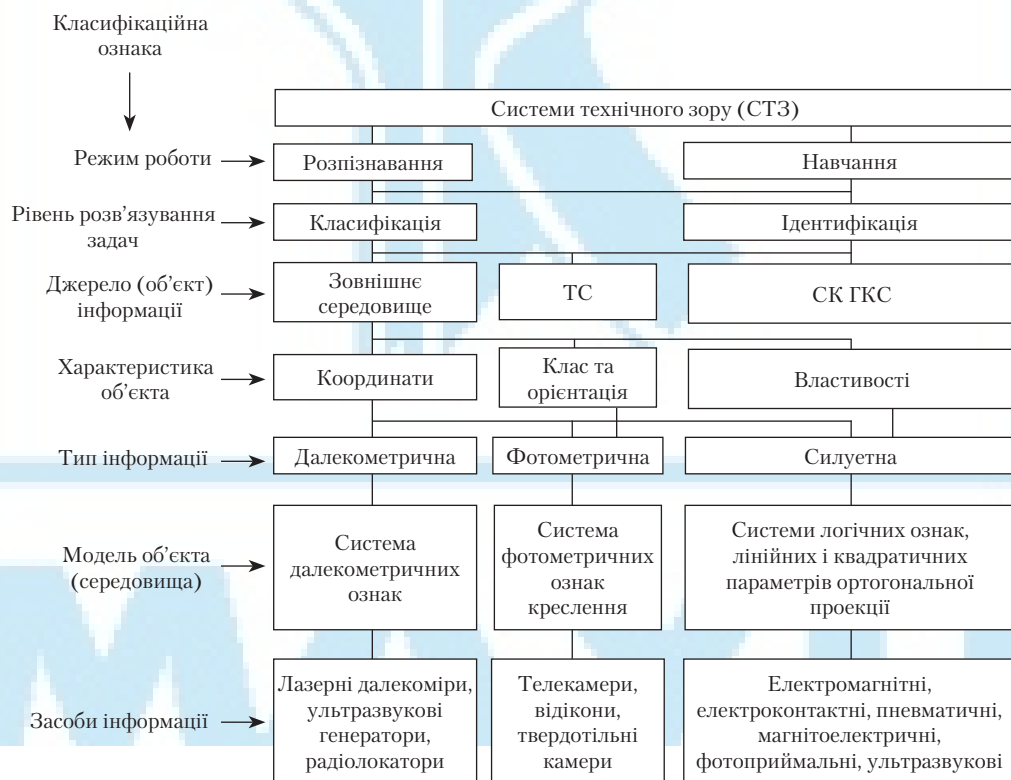


Рис. 5.17. Класифікація систем технічного зору

Традиційно схема роботи СТЗ містить таку послідовність етапів: **навчання** (подання, опис об'єкта у термінах заданого простору та запис цих даних у пам'ять ЕОМ як *модель середовища*) → **розпізнавання** (сенсорне сприйняття об'єкта та його опис у термінах прийнятої системи ознак, порівняння одержаних даних з еталонними записами моделі, прийняття рішення про тип об'єкта) → **самонавчання** (аналіз якості ідентифікації та корекції моделі середовища).

**Задача автоматичного розпізнавання об'єктів має два аспекти**, що впливають з таких особливостей: об'єкт займає нескінченну множину положень на робочій позиції (позиції); в полі зору виділяються різноманітні за формою об'єкти. Завдяки *першій* особливості виникає необхідність *ідентифікації об'єктів* — *приписування кожному з них імені з набору імен прототипів, опис яких зберігається в пам'яті ЕОМ*. Невизначеність ідентифікації породжується великою кількістю зображень одного об'єкта, частина з яких може з певним наближенням збігатися із зображеннями інших, подібних до нього об'єктів. Саме ця обставина призводить до невизначеності при виділенні класу об'єкта серед інших у загальній їх *класифікації*. Тому при ідентифікації об'єктів виділяються специфічні їх ознаки, які дають змогу виявити властивості об'єкта серед інших конкретних об'єктів того ж класу, і таким чином на підкласи розбивається вся множина проєкцій об'єкта, відображених на площину. При цьому використовують *інваріантні ознаки, чутливі до повороту об'єкта*.

З урахуванням викладеного автоматичне розпізнавання здійснюється розв'язанням двох задач: **класифікації та ідентифікації**. При *класифікації вся різноманітність об'єктів групується за кількома класами, які відрізняються цільовими та узагальненими ознаками* (наприклад, тіла обергання характеризуються наявністю осі обергання, а плоскі об'єкти — малою товщиною тощо). Отже, до кожного класу входить група об'єктів з деякими схожими цільовими ознаками, і після розв'язування задачі класифікації об'єкт зараховується до певного класу й розпізнається. При цьому при розв'язанні задач класифікації об'єктів намагаються знайти такі їхні ознаки, які є *інваріантними до його плоскопаралельного переносу та повороту*.

Вимоги, що ставляться до СТЗ, полягають у забезпеченні високої надійності ідентифікації, прийнятті рішення в реальному масштабі часу, визначенні місця та орієнтації об'єкта, розташування можливої області захвату об'єкта, здійсненні функції його якісного контролю. Виконання останньої вимоги в спеціалізованих СТЗ (зокрема бінарних) не обов'язкове, бо ці функції можуть надаватися ЗП роботів, а СТЗ при цьому можуть приймати рішення, використовуючи зображення з низькою роздільною здатністю. Такий підхід до поділу функцій між СТЗ та чутливим ЗП робота сприяє розв'язанню задачі розпізнавання об'єкта в реальному масштабі часу.

Економічність опису об'єкта пов'язана з його кількісними параметрами і характеризується витратами машинного часу, що залежить як від потужності ЕОМ, так і від самого об'єкта. У потоково-конвеєрних ГКІС існують певні умови організації праці, які дають змогу набором нескладних технічних засобів (упорядкування) первинно орієнтувати об'єкти (див. розд. 7 [23]) та розпізнавати їх на основі *детермінованих методів*. Тоді задача ідентифікації об'єктів має скінченну кількість значною мірою визначених відповідей, що істотно зменшує невизначеність при прийнятті рішення.

**Постановка задачі розпізнавання.** З урахуванням введення обмеження на скінченну кількість помітних положень об'єкта можна сформулювати таку постановку задачі їх розпізнавання. Нехай є скінченна кількість класів  $K_{ij}$ , що об'єднують усю номенклатуру об'єктів, які потрібно розпізнати. Кожен з них відображається на площину зображення в обмеженій кількості проєкцій  $P_{ij}$ . Тоді на основі креслень конструкторської документації на номенклатуру об'єктів можна знайти логічні змінні як ознаки, що повністю характеризують цей об'єкт серед інших, а також однозначно описують його положення в просторі. При цьому з метою впорядкування об'єкта різноманітність їх класів  $K_{ij}$  розбивається на скінченну кількість підкласів, кожний з яких є конкретним об'єктом, який має скінченну множину помітних положень.

Залежно від обчислювальних можливостей ЕОМ *навчаючою послідовністю* для такої системи розпізнавання є або ряд образів  $R_{ij}$  конкретних об'єктів, які подають пристрою введення для зображення в потрібному (еталонному) положенні  $P_{et}$ , або класифікатор еталонних ознак.

Основне завдання полягає в тому, щоб вибрати таку систему бінарних ознак і розв'язувальних правил  $RP$ , які дають змогу автоматично класифікувати об'єкт, тобто розпізнати серед певної кількості інших та ідентифікувати його положення мінімальними за обсягом технічними засобами при задовільній швидкодії. Остання повинна бути близькою до сприйняття людиною, а капітальні витрати на виготовлення та впровадження системи мають забезпечувати її комерційну доступність при автоматизації більш простих операцій, коли необхідно розпізнавати обмежену кількість об'єктів. Якщо ж при розширенні класу об'єктів виникає ситуація, за якої система бінарних ознак буде недостатньо ефективною, вводиться додаткова система ознак з підвищеною універсальністю.

## 5.8. Методи і засоби розпізнавання об'єктів роботизації

**Метод ідентифікації об'єктів з використанням інформаційних поверхонь.** Великого поширення в ГКІС набули СТЗ, дія яких ґрунтується на методах ідентифікації проєкцій об'єктів з використанням інформаційних поверхонь. Суть розпізнавання об'єктів із використанням чутливої поверхні полягає в тому, що матриця чутливих елементів (останні можуть бути виконані з пневмодатчиками, високочастотними джерелами та приймачами, оптоелектронними елементами з бінарними структурами або напівпровідниковим запам'ятовуючим шаром з великою кількістю дискретних станів тощо), завдяки чому з матриці знімається інформація, що характеризує клас і положення об'єкта в полі матриці і використовується для формування керуючих дій на виконавчі пристрої ПР або інших модулів ГКІС.

Задача ідентифікації об'єктів у такій інформаційній системі зводиться до поділу вибраної сукупності об'єктів на класи в процесі зіставлення апріорної інформації про кожен їх клас у вигляді набору ознак з інформацією про невідомий об'єкт, одержаною «опитуванням» чутливої поверхні. Перевагою таких інформаційних систем є можливість розв'язування задач ідентифікації плоских об'єктів порівняно нескладної геометричної форми за короткий час і, таким чином, керування адаптивними виконавчими пристроями модулів ГКІС у реальному масштабі часу.

Перспективним для логічного розпізнавання об'єктів є метод, пов'язаний із застосуванням інформаційних поверхонь, розташованих у робочій зоні робота [94]. Суть його полягає у використанні чутливої поверхні, оснащеної матрицею чутливих елементів із двома граничними станами: «1» — якщо розпізнаваний об'єкт накриває цей елемент, «0» — у протилежному разі.

Потрапляючи на чутливу поверхню, об'єкти ніби активізують різні сполучення її чутливих елементів. Інформація, що знімається з них, після обробки використовується для ідентифікації об'єктів або їх розташування на поверхні з метою формування необхідних керуючих впливів на виконавчі пристрої промислових роботів або інших елементів робототехнічної системи.

Задача ідентифікації у такій інформаційній системі зводиться до поділу вибраної сукупності об'єктів на класи в процесі зіставлення апріорної інформації про кожний клас об'єктів у вигляді набору ознак з інформацією про невідомий об'єкт, отриманої «опитуванням» чутливої поверхні. Перевагою таких інформаційних систем є можливість розв'язання задачі ідентифікації плоских об'єктів щодо нескладної геометричної форми за невеликий час, а отже, керування адаптивним роботом у реальному масштабі часу.

**Синхроімпульсний метод ідентифікації об'єктів.** В умовах часто переналагоджуваного виробництва найбільш ефективним є синхроімпульсний метод контролю і метод послідовних перерізів [94; 136].

Сутність *синхроімпульсного методу контролю* положення об'єкта полягає в тому, що в процесі переміщення транспортними пристроями робототехнічної системи зчитується інформація одночасно по всій проекції об'єкта за характерними точками його контуру, причому синхронізуючий момент зчитування задається самим об'єктом, що рухається.

**Метод контролю положення деталі за поперечними перерізами.** Модифікацією синхроімпульсного методу є *метод контролю положення деталі за поперечними перерізами*, що передбачає розміщення контрольних точок  $X_1 \dots X_n$  поперечно в одну лінію. Деталь, що рухається, проглядається контрольними точками своєї вимірювальної «лінійки» вроздріб. Ділянки розбивки кодуються розташованими на шляху деталі синхроточками. Вибір довжини закодованих ділянок забезпечується відповідним зсувом синхроточок відносно «лінійки» або одна від одної по ходу руху деталі. Кількість синхроточок завжди на одиницю менша за кількість послідовно контрольованих ділянок деталі. У межах розбивки вказаним вище способом здійснюється пошук мінімальної кількості й місця розташування синхроточки і контрольних точок «лінійки».

Синхроімпульсний метод розпізнавання і логіко-програмного керування маніпуляційними системами можна використовувати не тільки для розпізнавання положення об'єкта в просторі, а й для активно-примусового переорієнтування об'єктів за найкоротшим шляхом. Синтез, конструкцію й роботу маніпуляторів із логічною системою керування можна розглянути на прикладі розпізнавання положення, орієнтування й відбраковування найпростіших деталей з несиметрично розташованими на їхніх поверхнях наскрізними отворами. Отвори можуть займати на поверхні транспортної системи (наприклад, *лотка*) помітні положення, що відрізняються розташуванням отворів. Відсутність отвору в деталі фіксується логічною схемою як брак виробництва [94].



Поштучно подавані по лотку (рис. 5.18) деталі приводяться в задане положення на чутливій поверхні маніпулятора, і пристрій розпізнавання у вигляді маски визначає положення деталі за станом восьми фотодіодів  $A, B, C, \dots, H$ , а логічна схема за результатами контролю вибирає найкоротший шлях розв'язання поставленої задачі. Командою на виконання чергової операції є закінчення попередньої.

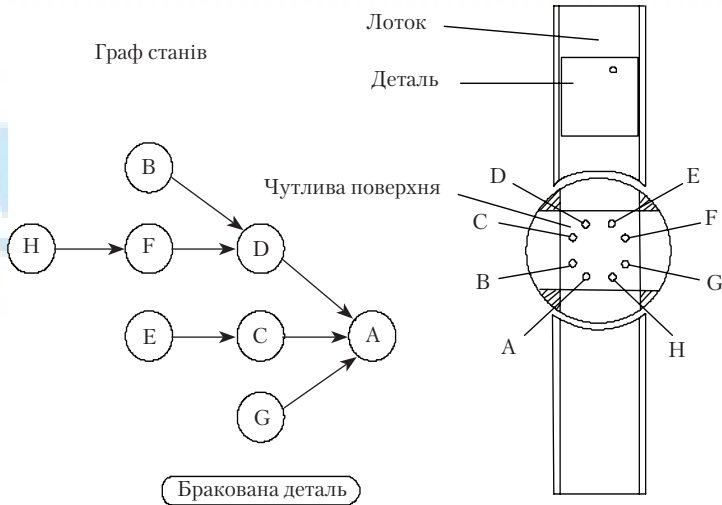


Рис. 5.18. Граф подання процесу орієнтування

На рис. 5.18 у вигляді графа подано алгоритм роботи фотодіодів розпізнавальної маски, що керують процесом розпізнавання, орієнтування і відбраковування. Освітлений стан одного фотодіода при затемненні інших однозначно характеризує кожне з можливих положень деталі. Затемнений стан усіх восьми фотодіодів означає появу на чутливій поверхні бракованої деталі.

Стани фотодіодів із відповідними схемами посилення і формування в розпізнавальному пристрої маніпулятора характеризуються відповідними логічними функціями  $P1 = C \vee E \vee F \vee H$ ;  $P2 = B \vee G$ ;  $P3 = D$ ;  $P4 = A$ .

**Ідентифікація об'єктів довільного опуклого контуру з використанням ланцюгового коду Фрімена.** Задача ідентифікації об'єктів за допомогою інформаційних чутливих поверхонь може бути розширена до розгляду плоских фігур довільного опуклого контуру при одночасному скороченні часу на обробку інформації. Обробка здійснюється на етапі виділення контуру об'єкта за допомогою відповідних алгоритмів ідентифікації [94], наприклад, таких, що ґрунтуються на використанні ланцюгового коду Фрімена (див. підрозд. 5.4), у яких машинний час визначається не кількістю накритих об'єктом точок чутливої поверхні, а кількістю точок, що лежать на його контурі. Практично така задача розв'язувалася й у випадку розглянутого *синхронізованого методу контролю* положення об'єктів. Однак кількість і розташування контрольних точок при цьому обумовлювалися заздалегідь відомим класом розпізнаваних об'єктів, а задача полягала в розпізнаванні орієнтації об'єкта. Відмінність у цьому випадку полягає в тому, що об'єкт може займати довільне положення на чутливій поверхні, причому клас об'єкта заздалегідь не відомий.

Блок-схему алгоритму наведено на рис. 5.19.

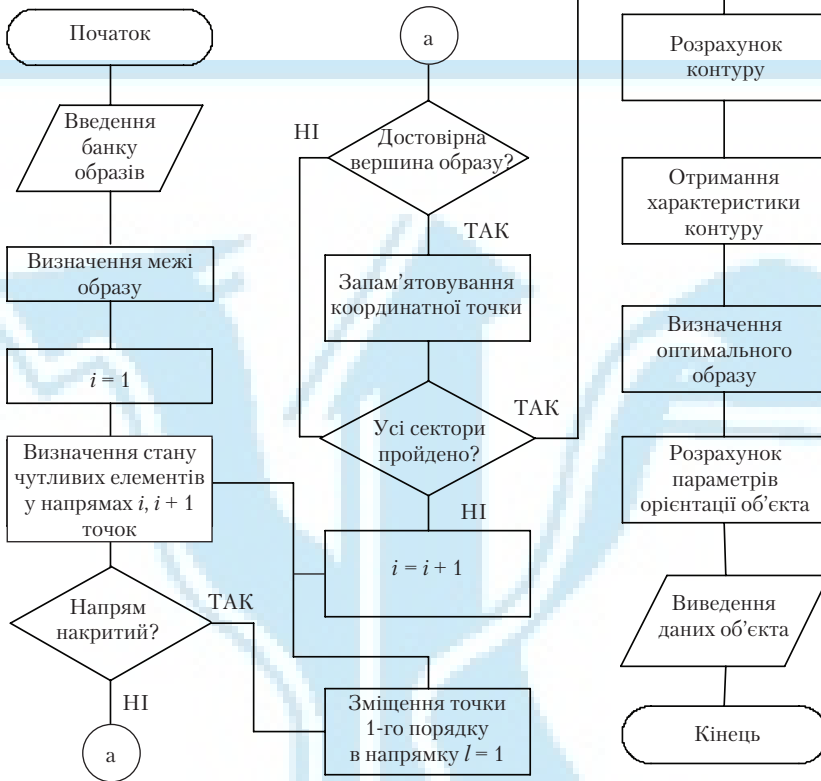


Рис. 5.19. Блок-схема алгоритму ідентифікації об'єктів на чутливій поверхні

Вихідним є банк даних з еталонними ознаками розпізнаваних об'єктів. Задача полягає в знаходженні серед цих об'єктів такого, який за своїми ознаками був би найближчим до розташованого на чутливій поверхні об'єкта, а також знаходження розташування центра мас об'єкта і параметрів його орієнтації на поверхні.

Нехай як об'єкт розглядається довільний чотирикутник  $abcd$ , розташований на чутливій поверхні (рис. 5.20), де траєкторію перегляду дискретної матриці — чутливої поверхні показано пунктирною лінією.

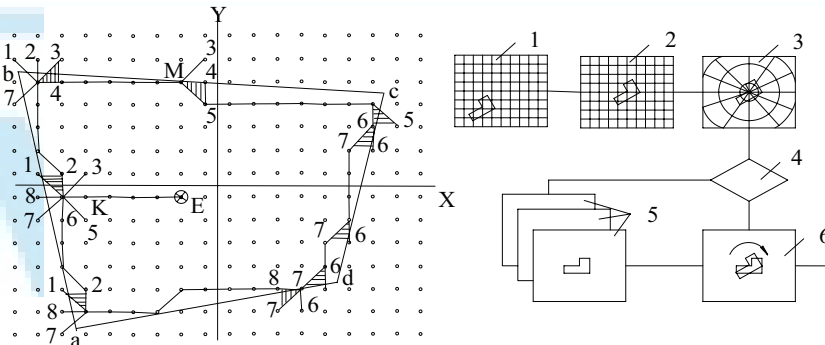


Рис. 5.20. Схема послідовності перетворення зображень

Спочатку при запуску інформаційної системи здійснюється сканування чутливих елементів поверхні з великим кроком до моменту, доки хоча б один з елементів не набуде стану «1» (нехай такою вихідною точкою є точка  $E$  на рис. 5.20). Далі, переміщуючи по заздалегідь обраному напрямку (наприклад, ліворуч), можна переглянути елементи чутливої поверхні, розташовані на цьому рядку матриці. Поява елементів із станом «0» свідчить про знаходження межі об'єкта, причому центр мас об'єкта розташовується праворуч від межі. Останній елемент із станом «1» маркується на поверхні як точка 1-го порядку (точка  $K$ ). Точка 1-го порядку має вісім напрямків (див. рис. 5.5, підрозд. 5.4, код Фрімена), причому кожні два сусідніх напрямки утворюють сектор  $45^\circ$ . У разі прямолінійності контуру, що відстежується, об'єкт завжди перебуватиме в одному секторі.

Зокрема, у розглянутому прикладі об'єкт обмежений прямою  $ab$ , що має кут нахилу  $90^\circ \leq \varphi \leq 135^\circ$ . Тому розглянута пряма завжди лежить у заштрихованому секторі  $1 - K_n - 2$  точки і принаймні одна з точок 1 чи 2 матиме стан «1» (на рис. 5.20 стан «1» має точка 2). Таким чином, визначається наступна точка контуру, в яку переміщується початкова позиція наступного кроку пошуку.

Надалі цикл, що включає аналіз стану множини точок 1 і 2, повторюється багаторазово доти, доки покроковий рух по контуру не переривається на кроці, у якому стани точок 1 і 2 одночасно стають нульовими. Це свідчить про наявність зламу в контурі об'єкта, що відстежується, у межах робочого сектора  $1 - K_n - 2$  точки першого порядку (тут  $n$  – номер кроку, на якому зустрівся злам). Абсолютні координати центра точки  $K_n$  запам'ятовуються й приймаються за координати вершини об'єкта ( $b^1$ ). Запам'ятовується також номер вектора точки  $K_n = b^1$ , що надалі використовується для визначення кінця обходу контуру.

Обертанням за годинниковою стрілкою в точці  $K_n$  знаходимо наступний сектор точки першого порядку, у якому хоча б одна з точок із чутливим елементом має стан «1» (на рис. 5.20 сектор  $3 - b^1 - 4$ ). Цю процедуру подано на схемі рис. 5.19 як цикл із параметром 1, що дорівнює номеру вектора напрямку точки  $K_n$ . Таким чином, пряма, що належить до контуру об'єкта, лежить у секторі  $3 - b^1 - 4$ , і подальшого аналізу відповідно до описаної методики зазнають точки 3 і 4. Можливе одержання помилкових вершин (наприклад, точка  $M$ ), якщо пряма, що утворює контур об'єкта, має напрямку, близький до напрямку одного з векторів поточної точки першого порядку та істотно віддалена від попередньо знайденої вершини. Для прямої контуру  $bc$  робочим сектором у вершині  $b^1$  є сектор  $4 - b^1 - 5$ , однак внаслідок зазначеного помилково був обраний сектор  $3 - b^1 - 4$ , що призвело до відшукування помилкової вершини (точка  $M$ ). При визначенні помилкової вершини здійснюється перехід у наступний сектор (тобто сектор  $4 - b^1 - 5$ ), а координати помилкової вершини стираються.

Таким чином, здійснюється обхід усього контуру, координати вершин  $abcd$  запам'ятовуються, а обхід контуру закінчується при зустрічі точки, з якої почалося відстеження. При цьому автоматично витримується умова замкненого контуру.

Описаний алгоритм дає змогу обходити не тільки прямолінійні, а й довільно опуклі криволінійні контури, одержуючи їх відображення, апроксимоване відрізками прямих.

Точність апроксимації криволінійних контурів можна підвищити введенням точки 2-го порядку із шістнадцятьма напрямками і більше, для чого необхідно використовувати чутливе поле з більшою роздільною здатністю або меншою дискретністю розташування чутливих елементів. Підвищення точності апроксимації обмежується конструктивними особливостями використаних чутливих елементів.

**Ідентифікація об'єктів у матеріальних потоках.** Швидкодія та якість процесу ідентифікації визначаються ступенем розподілу процедурних функцій СТЗ між режимами навчання (самонавчання) та розпізнавання ОР. Урахування специфіки керування адресуванням матеріальних потоків ОР передбачає скінченну точність процесу ідентифікації для різноманітних ОР при різних вимогах реального виробництва. Це зумовлено ось чим: кількість матеріальних потоків  $W_i (i = \overline{1, l})$  менша, ніж кількість об'єктів  $R_j (j = \overline{1, a}; l < a)$ ; формування керуючих команд подається оператором зарахування довільного  $R_j$  до одного з матеріальних потоків  $W_i$ .

У результаті ідентифікації однозначно визначаються групи ОР: для  $W_1 \rightarrow K_{1W_1}, K_{2W_1}, \dots, K_{aW_1}$ ;  $W_2 \rightarrow K_{1W_2}, K_{2W_2}, \dots, K_{aW_2}$  і т. д. Кожній групі ОР відповідає набір ідентифікаційних характеристик  $SP_{11}, SP_{21}, \dots, SP_{1j}, SP_{2j}, \dots, SP_{ij}$ , рівня  $ij$ , і в разі їх виявлення система розпізнавання формує команду керування.

Вибір ненадмірних і достатніх характеристик рівня  $ij$  передбачає можливість однозначного опису матеріальних потоків  $W_i$  і не зумовлюється необхідністю ідентифікувати тип кожного ОР окремо.

Залежно від вимог гнучкого виробництва поєднання ОР у матеріальні потоки може змінюватись як кількісно, так і якісно, що, у свою чергу, спричинює зміну вихідного набору ідентифікуючих характеристик. Для опису нових потоків  $W'_m (m = \overline{1, d})$  повинен дозволятися перехід до ознак  $SP'_{m,n} (n = \overline{1, z})$ , які розглядаються як ненадмірний і достатній набір характеристик другого ієрархічного рівня  $mn$ .

**Схема процесу ідентифікації ОР у матеріальних потоках.** Фактично має місце дворівнева організація процесу ідентифікації ОР (див. підрозд. 5.1): навчання та розпізнавання (відповідно позиції 9 і 10 на рис. 5.21). Верхній рівень містить операції роздрібнення сукупності розпізнавальних ОР на функціонально зв'язані множини — кластери, а нижній — операції одержання ідентифікуючих характеристик у вигляді формалізованого опису кластерів [99].

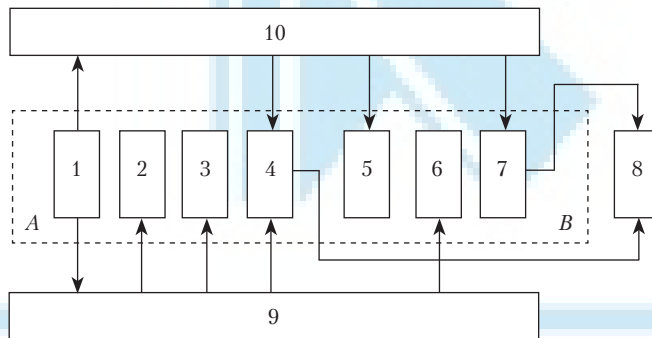


Рис. 5.21. Структурна схема дворівневої ідентифікації ОР

Процес ідентифікації ОР на верхньому рівні (А) складається з послідовно виконуваних етапів. *Перший етап (1)* передбачає дослідження виробничої дільниці, в контурі якої використовується СТЗ, а також визначення та аналіз номенклатури ОР. Залежно від характеру впорядкування матеріального потоку береться відповідна форма подання даних про ОР в моделі середовища, а на основі розмірних і якісних характеристик геометричної форми ОР виконується попередній вибір параметрів сенсорної системи: способу подання ОР (наприклад, у вигляді градієнтних або бінарних проєктивних зображень), роздільної здатності сенсора та яскравісних параметрів зображення ОР.

На *другому етапі* (2) здійснюється вибір алфавіту ознак для опису ОР (див. також підрозд. 5.2). При розв'язуванні задачі керування адресуванням матеріальних потоків в умовах гнучкого виробництва, коли конфігурація ОР задовольняє вимоги технологічності (симетричність форми, явно виражені конструктивні особливості та ін.), достовірне розпізнавання ОР може провадитися за допомогою сенсорних систем з низькою роздільною здатністю ( $\sigma \leq 64 \times 64$ ) при бінарній функції яскравості. Введення цих обмежень дає змогу вже на етапі сенсорного сприйняття значною мірою мінімізувати вхідну інформацію та звести задачу ідентифікації ОР до аналізу конкретних зображень за результатами сканування й формалізації перепадів яскравості, які характеризують контур ОР.

У процесі перетворення реальних зображень ОР за допомогою фотометричних сенсорних засобів контурним лініям проєкції ставляться у відповідність взаємно ортогональні відрізки різної довжини, орієнтовані за рядками і стовпцями сенсорної сітки. Як спосіб опису проєктивних зображень ОР вибирається скінченний ряд (алфавіт) конструктивних структур  $S_T$  — різних комбінацій ортогональних відрізків. Другим інформаційним параметром є характеристика симетрії зображення  $V$ , а додатковим параметром для опису бінарного проєктивного зображення — характеристика його «внутрішньої симетрії»  $V_{\text{вн}}$ . Цей показник характеризує розташування отворів різноманітної конфігурації в площині проєкції ОР.

З урахуванням викладеного узагальнюючий структурний запис проєктивного зображення ОР має вигляд

$$\theta = \{(S_T), (h_i), (\xi_{ij}), (V), (V_{\text{вн}})\},$$

де  $h_i$  — кількість появ  $i$ -го символу в контурі зображення ОР;  $\xi_{ij}$  — показник зв'язку між  $i$ -м та  $j$ -м символами, який характеризує черговість проходження  $i$ -го та  $j$ -го символів у контурі.

Подальша оптимізація вихідного алфавіту  $S_T$  ознак ОР передбачає скорочення їх кількості і підвищення інформативності. В разі формування зображень ОР за допомогою сенсорної системи матричного типу здійснюється прив'язка до прямокутної системи сканування контуру. Остання допускає однозначне задання не тільки осі, а й напрямку сканування ( $x^+$ ,  $x^-$ ,  $y^+$ ,  $y^-$ ). Додатковий фактор мінімізації структурного запису пов'язаний з геометричним аспектом сканування контуру зображень ОР, а саме: в замкнених контурах завжди спостерігається лінійна залежність між параметрами ( $x^+$ ,  $x^-$ ,  $y^+$ ,  $y^-$ ).

Узагальнюючий структурний запис довільного проєктивного зображення ОР зводиться до задання багатовимірної ознаки у вигляді системи рівнянь

$$\begin{cases} x'_i = (V, V_{\text{вн}}); \\ x''_i = (y^+, x^+, x^-). \end{cases}$$

На *третьому етапі* (3) процесу ідентифікації ОР на основі прийнятого алфавіту ознак визначається ненадмірна навчаюча множина ОР, яка є базисною при побудові моделі середовища. Розв'язування цієї задачі зводиться до розробки процедур аналізу та синтезу морфологічної моделі проєкційних зображень, яка є засобом узагальнення властивостей проєкції заданої вибірки ОР. Фізичний зміст цієї моделі полягає в одержанні комплексного «портрета» сукупності бінарних проєктивних зображень. За її допомогою реалізується аналіз проєкцій, побудованих на базі алфавіту  $S_T$ , які задовольняють такі вимоги технологічності: форма контуру проєктивного зображення ОР є порівняно простою, що передбачає об-

межену кількість символів  $y^+$ ,  $x^+$ ,  $x^-$ ; проекції мають якомога більше значення показника симетрії  $V$  і найменше значення параметра «внутрішньої симетрії»  $V_{\text{вн}}$ . Крім того, функція значень параметрів симетрії дає змогу організувати ряд окремих видів морфологічної моделі проекційних зображень, характеристики яких конкретизовані, завдяки чому можна задавати скінченний перелік проєктивних зображень ОР.

На *четвертому етапі (4)* формується модель середовища. За допомогою процедур кластер-аналізу виконується роздібнення ОР навчаючої множини на кластери, здійснюється оптимізація їх складу та формалізація параметрів. Визначальною задачею при впорядкуванні проєктивних зображень ОР в ознаковому просторі є обчислення значень аргументів керування для визначення  $x'_i$  та  $x''_i$ . Рівняння для запису  $x'_i$  дає змогу задавати функціональну область ознакового простору, в якому розташовано ОР, що мають відповідні параметри  $V$  та  $V_{\text{вн}}$ . Рівняння для  $x''_i$  є власне ідентифікаційним, і можна розділити функціональну область ознакового простору на кластери, які задають групи ОР за спільними конструктивними ознаками. Оскільки кластеризація здійснюється за поєднанням трьох ознак  $y^+$ ,  $x^+$ ,  $x^-$ , ознаковий простір є тривимірним. Довільна точка його відповідає деякому  $i$ -му ОР, а відстань між точками є тією мірою подібності, залежно від якої організуються кластери.

Процедури кластеризації розбиваються на дві послідовні задачі:

- обчислення міри подібності на основі модифікованої метрики в базовому і трансформованому ознакових просторах ґрунтується на модифікованій формулі евклідової метрики, якою враховують тільки нульові значення відстаней між ОР в ознаковому просторі;
- динамічне розширення кластерів при об'єднанні подібних ОР.

Обчислювальна суть останньої задачі полягає в тому, щоб «продивитись» первинні групи та поєднати всі взаємозв'язані ОР. Динамічність процедур досягається нарощуванням паралельних обчислень у міру збільшення кількості елементів, які входять до складу кластерів.

Після виконання зазначених процедур множина бінарних проєктивних зображень ОР подрібнюється на кластери, причому між елементами кластера існує ознаковий зв'язок, який більш вагомий, ніж фактор відстані в ознаковому просторі.

Завершальною ланкою побудови моделі технологічного середовища й запису її у пам'ять ЕОМ є оптимізація структури кластерів. Цільовою функцією оптимізації є сукупні вимоги: оптимізації загальної кількості та розмірності кластерів, розмірного розподілу кількісного складу їх елементів, збільшення щільності кластерів, виявлення ядра кластера, побудови відношень зв'язків між ядром та елементами кластера. Наслідком оптимізації структури кластерів є однозначний зв'язок елементів кластера з ядром, і вся корисна інформація зводиться до задавання значень координат ядра і розмірів кластера в ознаковому просторі. Це правило формалізації даних є загальним для всіх кластерів; обсяг запису не залежить від кількості ОР, які входять до складу кластера, що дає змогу за необхідності нарощувати кількість елементів у моделі середовища без збільшення пам'яті ЕОМ.

На *нижньому рівні (В)* для однозначного визначення належності конкретного ОР до певного кластера здійснюється порівняння сенсорної інформації про цей об'єкт з апіорними відомостями, записаними в моделі середовища. Взаємодія

«ОР – модель» складається з таких процедур: оперативного аналізу сенсорної інформації, відновлення елементарної структури кластерів, зіставлення апостеріорних та апіорних відомостей про ОР.

*Аналіз сенсорної інформації (етап 5)* здійснюється виконанням операції визначення інформативної області зображення ОР, параметрів його симетрії та виділенням контурних ознак.

Для відновлення елементарної структури кластера (етап 6) проводяться ітеративні процедури нарощування елементів навколо ядра.

Після зіставлення апостеріорної та апіорної сенсорної інформації здійснюється прив'язка ОР до імені кластера, яка є ідентифікуючою характеристикою верхнього рівня.

При ідентифікації ОР всередині кластера та конкретизації типу ОР (етап 7) виконуються операції: аналізу якісних і кількісних показників кластера, пошуку відокремлюючих ознак для ОР всередині кожного кластера, формалізації ідентифікуючих характеристик у вигляді відношення «ОР – кластер – тип».

За результатами формалізації типу ОР формуються керуючі команди (етап 8) для виконавчих пристроїв технологічної системи щодо обслуговування конкретного типу ОР.

**Розпізнавання рухомих об'єктів  $m$ -вимірними лінійними чутливими елементами.** Прикладом реалізації методу контролю положення деталі за поперечними перерізами може бути описана в роботі [96] фотоелектронна система для ідентифікації і класифікації рухомих деталей різної форми, яка забезпечує зорову чутливість ПР при обслуговуванні конвеєрів. Принцип розпізнавання об'єктів у такій системі (рис. 5.22) полягає у виділенні основної характерної ознаки об'єкта, зокрема площі його плоскої проекції, а також ряду допоміжних ознак для виключення однозначної класифікації за однаковими плоскими поверхнями об'єктів, але різних за конфігурацією – габаритних за висотою та шириною у площині перегляду системою. Позитивною стосовно розглянутого попередньо випадку (рис. 5.20), де розпізнавання відбувалося чутливою поверхнею у вигляді  $m \times n$ -розмірної матриці чутливих елементів, у цьому разі (рис. 5.22, а) використовується  $m$ -вимірні лінійка з чутливих елементів, тобто в  $n$  разів зменшуються витрати на її створення і в стільки ж разів збільшується швидкодія системи розпізнавання.

Подана на рис. 5.22, а схема компонування системи складається з таких основних компонент:

- імпульсного газорозрядного світловипромінювача (стробоімпульсного генератора) 4 з параметрами світлових імпульсів: початкова частота 100 кд/с і тривалість  $10^{-4}$  с, які можуть варіюватися залежно від характеристики опромінювання об'єктів (параметрів поверхні), віддаленості джерела від конвеєра, швидкості його руху і мінімальної відстані між об'єктами на конвеєрі;
- фотоприймача 1, що розташований вертикально в просторі і паралельно площині  $X - Y$  підвіски об'єктів 3 на відстані  $z_2 - z_1$  від підвіски конвеєра 2 і  $z_2$  від світловипромінювача 4 і являє собою лінійку з набором рівновіддалених фотодатчиків для прийому світлової інформації і формування електричних сигналів;
- оптичного пристрою на виході (випромінювач) і вході (фотоприймач) системи (на рис. 5.22, б не показані);
- пристрою первинної обробки інформації (ППОІ).

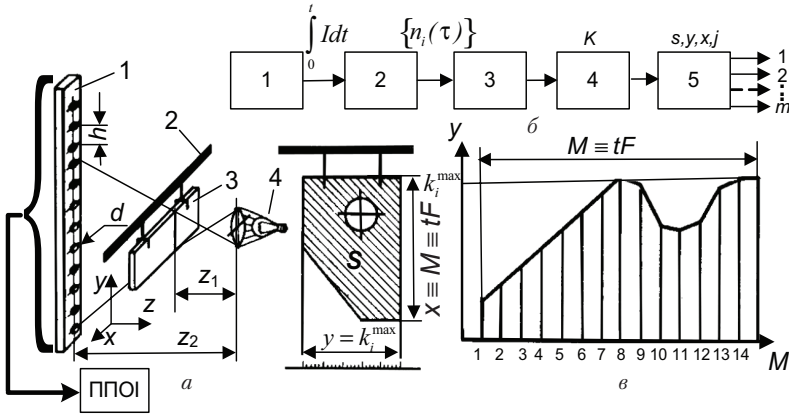


Рис. 5.22. Компонування системи розпізнавання рухомих об'єктів (а), структурна схема системи (б) та дискретна інтерпретація об'єкта (в)

Щільність розташування фотодатчиків визначається необхідною точністю роботи системи, бо характеризує зону нечутливості зорової чутливості ПР, і задається залежністю

$$V = K / L = 1(d + h), \tag{5.50}$$

де  $K$  – кількість фотодатчиків на лінійці;  $L$  – довжина лінійки;  $d$  – діаметр фотодатчиків;  $h$  – відстань між фотодатчиками.

Робота системи розпізнавання здійснюється у такій послідовності. В момент спалаху світловипромінювача на фотоприймачі утворюється сукупність сигналів  $\{n_i(\tau)\}$  ( $i = 1, 2, \dots, K$ ), яка являє собою дискретну апроксимацію фрагмента проекції деталі. Опитування усіх фотодатчиків здійснюється пристроєм ППОІ паралельно у часі із частотою надходження імпульсів від світловипромінювача. При цьому «образ» деталі формується фрагментами – миттєвими перетинами при її русі на конвеєрі площиною, що є перпендикулярною до площини підвіски деталі і проходить через фотоприймач. Кількість опитувань  $M$  залежить, таким чином, від частоти  $F$  спалахів світловипромінювача і часу  $t$  руху деталі в зоні датчика, тобто

$$M = tF. \tag{5.51}$$

Причому чим більше  $F$ , тим більша кількість фрагментів-перетинів деталі потрапить на реєстрацію, тим вища точність розпізнавання.

Пристрій ППОІ перетворює множину сигналів від фотодатчика у символи «1» (фотодатчик освітлений) і «0» (фотодатчик затемнений) і визначає кількість затемнених фотодатчиків, тобто визначає фрагменти освітленої частини деталі, а отже, і площу  $S$  площини проекції. Площа  $S$  деталі (рис. 5.22, в) визначається як сума розгорнень затемнених фотодатчиків за кількість спалахів світловипромінювача у процесі проходження деталі через зону дії датчика візуалізації:

$$S = \sum_{i=1}^M K_i, \tag{5.52}$$

де  $K_i$  – кількість затемнених фотодатчиків у момент  $i$ -го спалаху ( $i = 1, 2, \dots, M$ ).

Висота деталі  $y$  визначається найбільшою кількістю затемнених фотодіодів серед усіх  $K'_i$  дискретних затемнень:

$$y = K'_i. \tag{5.53}$$



Ширина деталі  $x$  дорівнює кількості спалахів світловипромінювача за час проходження деталі через зону огляду:

$$x \equiv M \equiv tF. \quad (5.54)$$

В момент виходу деталі із зони дії фотоприймача (усі фотодатчики освітлені) значення  $S$ ,  $y$ ,  $x$  паралельно в часі передаються у класифікатор, який працює за принципом кодування сигналів від пристрою ППОІ номером каналу. Таке кодування забезпечує генерування відповідного елемента класифікатора залежно від значення сигналу, а отже, від технологічного номера деталі.

Зрозуміло, що в разі необхідності ідентифікації об'єктів за їх об'ємними відображеннями, слід використовувати два  $m$ -вимірні лінійні чутливі елементи, розташовані в найбільш інформативних площинах проекції об'єктів.

Описані процедури ідентифікації (рис. 5.20 та рис. 5.22) можуть також використовуватися в системах автоматизованого проектування РТК як при розв'язуванні задачі формування банку даних про номенклатуру об'єктів, які зазнають роботизованого виготовлення, так і при оберненій задачі — введенні до САПР даних про конкретний об'єкт з метою визначення номенклатури функціональних пристроїв, що повинні в складі проектуваного РТК обслуговувати об'єкт.

**Використання перетворювачів вал-число для ідентифікації положення об'єктів у процесі їх орієнтування.** Як відомо, для дискретного відображення орієнтування (відносного або просторового) об'єктів при виконанні цієї процедури промисловими роботами використовують сенсорні пристрої внутрішньої інформації ПР, призначені для вимірювання лінійних і кутових переміщень їх робочих органів. Найчастіше цю групу пристроїв утворюють *лічильно-імпульсні* та *кодові датчики*, тобто датчики, в яких вихідна величина характеризується кількістю імпульсів або певним набором сигналів. У тих випадках, коли треба знайти напрямок зміни вимірюваної величини, застосовують двофазну систему сприймаючих елементів.

Кодові фотоелектричні датчики кутових переміщень здійснюють перетворення кута повороту виконавчих механізмів маніпуляційної системи (МС) ПР у послідовність імпульсів — *унітарний код* або певний набір сигналів — наприклад, *циклічний код Грея* або *звичайний двійковий код*. Варто зазначити, що найчастіше використовують саме перетворювач вал-число з циклічним кодом Грея, бо при цьому похибка відліку не перевищує «ваги» однієї одиниці кутового значення через те, що відбувається зміна коду не більше ніж в одному розряді, що й є особливістю цього коду. В подальшому для використання в системах керування цей код Грея перетворюється у звичайний двійковий код, на якому і працюють числові системи керування ПР, за схемою перетворення, яка як приклад наведена в табл. 5.3 для чотирирозрядного диска.

**Приклад 5.10.** Розглянемо особливості перетворення циклічного коду Грея і звичайного двійкового коду. Як видно з табл. 5.3, в циклічному коді Грея відбуваються зміни при переході на крок не більше ніж в одному розряді, тоді як у звичайному двійковому коді зміни можуть досягати максимальної величини, наприклад, як це в положеннях з 3 на 4, або з 7 на 8, або з 15 на 0.

Як бачимо, при використанні перетворювача вал-число зі звичайним двійковим кодом при неперервних змінах кута повороту вала можуть виникати невизначеності в положенні кодового диску, причому похибка може досягати максимального значення у фазах переходу дискретного стану (наприклад, див. рис. 5.23, б при переході від положення 31 до положення 0). Ця обставина накладає надзвичайні вимоги щодо точності виготовлення кодових дисків, технологія якого складна і витратна.

Відповідність коду Грея звичайному двійковому та десятковому

Циклічний код Грея	Звичайний двійковий код	Десятковий код
0000	0000	0
0001	0001	1
0011	0010	2
0010	0011	3
0110	0100	4
0111	0101	5
0101	0110	6
0100	0111	7
1100	1000	8
...	...	...
1011	1101	13
1001	1110	14
1000	1111	15

Фотоелектричний перетворювач кутових переміщень показано на рис. 5.23. Принцип його дії полягає в тому, що на кодовий диск 4 (рис. 5.23, а) падає світло від освітлювача (наприклад, газонаповненої лампи) 1, яке проходить через оптичну систему 3. При обертанні вала 2, а разом з ним і диска 4 останній «проглядається» вузьким паралельним пучком світла, що проходить через прозорі сектори і потрапляє через фокусуючу систему 5 на фіксуючий пристрій 6. Диск складається з кількох концентричних кілець (рис. 5.23, б), утворених прозорими і непрозорими секторами, які чергуються в певній послідовності (коді). На рисунку наведено диск із звичайним двійковим кодом.

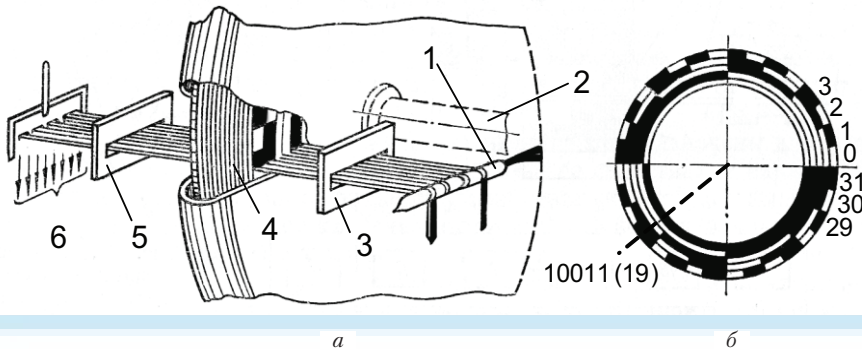


Рис. 5.23. Загальне компонування фотоелектричного перетворювача кутових переміщень (а) і диск з двійковим кодом (б)

Проходження світла через прозорий сектор сприймається фотолічильним фіксуючим пристроєм 6 у двійковій системі відліку як наявність сигналу, тобто «1», а затримка світла непрозорим сектором — як відсутність сигналу, тобто «0». Розряд визначається номером концентричного кільця від периферії до центра диска. Положення останнього, при якому паралельний промінь світла від оптичної системи 5 розташовується по штриховій лінії на рис. 5.23, б, відповідає числу 19 (10011). Зображений на рисунку п'ятирозрядний диск може займати (фіксувати) 31 положення.

### Контрольні завдання

1. Розкрити етапність розв'язання задачі розпізнавання.
2. Охарактеризувати поняття «клас» та «об'єкт» з позицій теорії розпізнавання.
3. Навести основні властивості класів.
4. Проаналізувати постановку задачі і дати тлумачення типової послідовності процедури розпізнавання.
5. Охарактеризувати задачу розпізнавання як задачу прийняття рішень.
6. Дати загальну характеристику основних методів розпізнавання.
7. Охарактеризувати дискримінантні методи розпізнавання.
8. Пояснити, що таке простір ознак, і перелічити відомі вам типи ознак.
9. Охарактеризувати байєсівські методи розпізнавання і дати означення функції щільності розподілу випадкової величини.
10. Дати тлумачення абстрактної моделі розпізнавання.
11. Розкрити особливості та навести етапність проектування об'єкта на чутливу поверхню.
12. Пояснити, що являє собою та як підвищується роздільна здатність чутливої поверхні.
13. Сформулювати основні операції перетворення та дії над зображеннями об'єкта.
14. Розкрити сутність і призначення попередньої обробки сигналів і зображень.
15. Охарактеризувати процедуру дискретизації зображень і сформулювати вимоги до вибору кроку дискретизації на основі теореми Котельникова.
16. Пояснити, що являє собою код Фрімена і показати типову схему такого кодування.
17. Розкрити загальні принципи параметризації неперервних функцій на основі лінійних ортогональних перетворень.
18. Охарактеризувати принципи інтегрального перетворення Карунена — Лоева.
19. Пояснити, в чому полягає оперативний аналіз сенсорної інформації та яку роль при цьому відіграє фільтрація зображень об'єкта.
20. Дати загальну характеристику методів формалізованого подання об'єктів ГКІС.
21. Розкрити сутність і прокоментувати основні принципи створення формалізованих технологічних мов.
22. Дати загальну характеристику методів аналітичного подання об'єктів.
23. Розкрити сутність методу R-функцій Рвачева та окреслити сферу його застосування.
24. Пояснити, в чому полягають наближені методи подання об'єктів й охарактеризувати їх переваги й недоліки при застосуванні.
25. Навести чотири складові визначеності формальної системи і проаналізувати відповідність їм методу координатних моделей подання об'єктів.
26. Сформулювати необхідний і достатній перелік основних типів елементарних координатних моделей і розкрити їх зміст у послідовності подання об'єктів ГКІС.
27. Розкрити повний зміст поняття «групова технологія» в умовах ГКІС.
28. Пояснити, що являє собою «віртуальний» об'єкт виробництва в умовах ГКІС.

29. Розкрити функцію і пояснити зміст її складових на верхньому узагальненому рівні подання ГКІС як об'єкта дослідження, а також навести повний функціональний оргграф виробничої системи на цьому рівні.
30. Пояснити, що являє собою метод  $\Phi$ -функцій як основа формалізованого подання об'єктів керування.
31. Прокоментувати взаємозалежність складових концептуальної моделі ГКІС в умовах групової технології.
32. Охарактеризувати пряму (аналізу) та обернену (синтезу) задачі створення та дослідження ГКІС.
33. Розкрити підфункцію виробничих процесів  $\Phi_{\text{вир.пр}}$  на другому рівні абстрагованого подання ГКІС, в тому числі й у вигляді повного функціонального оргграфа виробничого процесу.
34. Дати визначення третього рівня абстракції ГКІС – формального подання технології  $\Phi_{\text{техн}}$  та охарактеризувати чотири класи технологій.
35. Пояснити, як на четвертому рівні абстрагованого подання ГКІС формується виділення підфункцій технологій – технологічних процесів  $\Phi_{\text{техн.пр}}$ .
36. Дати визначення шостого ієрархічного рівня деталізації функцій ГКІС як об'єкта керування – функціональних задач-переходів  $\Phi_{\text{зад}}$  і пояснити, чим реалізуються ці задачі.
37. Пояснити, в чому полягає об'єктно-орієнтований принцип групування (ООПГ) в задачах формування складу технічних засобів ГКІС.
38. Розкрити особливості побудови інформаційних моделей – пошукових образів об'єктів (ПОО) як основи формування групової технології у ГКІС.
39. Охарактеризувати процес визначення відповідності обслуговуючих обробних ресурсів/пристроїв властивостям об'єктів виробництва як алгоритмічну покрокову процедуру.
40. Прокоментувати схематично відображення процесу формування групи об'єктів виробництва при використанні ООПГ з побудовою інформаційної моделі ПОО.
41. Пояснити, що являють собою і з якою метою будуються скінченні графи об'єктів і технологічних процесів.
42. Розкрити методи формалізації опису ознак об'єктів ГКІС.
43. Дати характеристику методів розпізнавання об'єктів ГКІС: синхроімпульсного, за поперечними перерізами, статичного на чутливій поверхні та динамічного за перетином фотоелектронної лінійки.
44. Розкрити зміст етапів формування набору кластерних характеристик дворівневої ідентифікації об'єктів у задачах керування адресуванням матеріальних потоків.
45. Охарактеризувати принцип роботи перетворювачів вал-число і пояснити відмінності в роботі дискових датчиків, що ґрунтуються на циклічному коді Грея та на звичайному двійковому коді.

## КОМБІНОВАНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

---

---

*Дурні ніколи не прощають і не забувають;  
наївні — прощають і забувають;  
мудрі прощають і не забувають.  
На цих складових будуються наші взаємовідносини.*

Томас Шаш

*Я ніколи не стою, якщо можна сидіти,  
і я ніколи не сиджу, якщо можна лежати.*

Уїнстон Черчіль

### 6.1. Методи математичного моделювання та управління виробничими процесами

**Основні визначення.** Під *оптимальним* розуміють таке управління, яке забезпечує найкраще значення певного критерію, що характеризує ефективність управління тими чи іншими процесами при заданих обмеженнях. *Узагальненим критерієм оптимальності* виробничих процесів є мінімум витрат суспільної праці; в технологічних процесах цей критерій виражається через *показники оптимальності*.

**Означення 6.1.** *Оптимальними системами (процесами)* є такі, в яких забезпечується оптимальне значення якогось основного показника якості системи (процесу): перехідних характеристик, продуктивності виконуваної операції, собівартості продукції, споживаної потужності, точності відпрацювання керуючої дії тощо.

Дослідження умов оптимальності виробничих процесів на практиці виконують *фізичним, математичним, імітаційним та електронним* (на ЕОМ) моделюванням.

**Означення 6.2.** *Фізичне моделювання* — це моделювання, яке здійснюється на реальному фізичному об'єкті або на його масштабованому аналозі, в якому відтворюються основні функціональні можливості реального об'єкта.

До такого підходу вдаються тоді, коли вкрай бажаною є необхідність високої точності збігу результатів моделювання з реальними процесами в об'єкті. Як неоліки необхідно відзначити, по-перше, високу вартісну складову такого моделювання через необхідність використовувати обробні ресурси реальної системи, а по-друге, призупинення на період моделювання штатних режимів експлуатації реального об'єкта і виключення його з експлуатаційного циклу за призначенням.

**Означення 6.3.** *Математичними моделями* називаються комплекси математичних залежностей і знаково-логічних виразів, що відображають суттєві характеристики досліджуваного явища.

Оскільки математичні моделі є найбільш абстрактними, вони досить поширені у системних дослідженнях. Розробляючи математичну модель, потрібно уважно перевірити, чи вона адекватно описує об'єкт.

Математичні моделі базуються на *аналітичних залежностях*, коли процеси функціонування складної системи записуються у вигляді деяких функціональних відношень (алгебраїчних, інтегральних, диференціальних та ін.) або логічних умов.

Отже, при побудові математичної моделі залежно від виду керованих процесів і задач дослідження може використовуватися математичний апарат різної складності: у простих випадках — *алгебраїчні рівняння*, у більш складних (коли необхідно розглядати явище в динаміці) — *диференційні рівняння* (як звичайні, так і в частинних похідних), а в найбільш складних випадках (коли розвиток процесів залежить у часі від великої кількості складно переплених між собою випадкових факторів) застосовуються методи *статистичного моделювання*. Їх ідея полягає в тому, що процес немовби «копіюється», відтворюється на ЕОМ з усіма супровідними випадковостями. Таким чином, будується один екземпляр (одна реалізація) випадкового процесу з випадковим перебігом і наслідками.

Сама по собі одна така реалізація не дає підстав для вибору рішень, але, діставши множини таких реалізацій, ми обробляємо їх як звичайний статистичний матеріал (звідси і термін «статистичне моделювання»), виводимо середні характеристики на основі множини реалізацій і формуємо уявлення про те, як у середньому впливають умови та визначальні параметри на хід і результати перебігу процесу.

Статистичні моделі процесів порівняно з аналітичними є більш точними й детальними, не потребують грубих припущень і дають змогу враховувати велику (в теорії — необмежену) кількість факторів. Однак і вони мають недоліки: порівняну громіздкість, значні витрати машинного часу, погану наочність одержуваних результатів і утрудненість їх осмислення. А головне — вкрай утруднене знаходження оптимальних рішень, які при статистичному моделюванні процесів доводиться шукати «на дотик», шляхом здогадок і спроб.

Ось чому при оптимізації рішень дуже корисною є попередня їх оптимізація на аналітичних моделях процесів, що потім при використанні точніших статистичних моделей дає змогу шукати оптимальні рішення не наздогад, а в обмежених областях, які містять рішення, близькі до оптимальних в аналітичних моделях.

При дослідженні *стохастичних процесів*, на перебіг яких впливають випадкові фактори, застосовуються як аналітичні, так і статистичні їх моделі: перші є більш грубими, в них враховується менша кількість факторів порівняно із статистичними моделями, але вони неминуче потребують якихось припущень і спрощень, описують явища тільки наближено, схематично, хоча результати такого моделювання є більш наочними і виразніше відображають властиві для явища закономірності. А головне — аналітичні моделі більше пристосовані до пошуку оптимальних рішень при застосуванні сучасних математичних методів. В аналітичному моделюванні забезпечується подібність характеристик об'єкта і моделі. Аналітичну модель можна досліджувати або *аналітично* (в загальному вигляді отримують явні залежності для шуканих величин), або *числовим методом* (якщо немає рішень у загальному вигляді, застосовують обчислювальну техніку, щоб отримати числові результати за конкретних числових даних), або *якісно* (якщо немає розв'язку в явному вигляді, можна знайти його деякі властивості, наприклад, оцінити стійкість розв'язку).

Для того щоб процес моделювання був зручним і природним, усі матмоделі повинні задовольняти три вимоги: точність, економічність і універсальність:

- *Точність* моделі — це властивість, що відображає ступінь збігу прогнозованих з її допомогою значень параметрів об'єкта з істинними значеннями. Істинні значення параметрів об'єкта ототожнюють з отриманими експериментально.
- *Економічність* математичної моделі визначається передусім витратами машинного часу або пам'яті ЕОМ на програмну реалізацію моделі. Показником економічності математичної моделі може також бути кількість використаних внутрішніх параметрів.

- *Ступінь універсальності* математичної моделі визначається можливістю застосовувати її до аналізу групи однотипних об'єктів в одному або багатьох режимах функціонування.

**Означення 6.4.** *Імітаційне моделювання* — таке моделювання, при якому відтворюється фізичний експеримент із забезпеченням подібності процесів, що відбуваються як у моделі, так і в реальному об'єкті, а також є можливість отримати наочні результати моделювання.

Перевагою імітаційних моделей є можливість моделювання навіть у тих випадках, коли аналітичні моделі або відсутні, або через складність системи не дають корисних з практичного погляду результатів. Імітаційне моделювання також дає змогу врахувати вплив великої кількості випадкових і детермінованих факторів, а також складних залежностей при введенні в модель відповідних елементів та операцій. Іноді застосовуються імітаційні моделі, що відображають тільки *структурно-топологічні* властивості об'єкта (тобто об'єкт задається за допомогою матриці, графа та ін.) або аналізуються тільки *функціональні* залежності.

**Означення 6.5.** *Електронне (машинне) моделювання* — це моделювання, яке відтворюється за допомогою апаратних, інструментальних і програмно-алгоритмічних засобів електронно-обчислювальної техніки (ЕОТ). Власне, на сучасному етапі розвитку науки і техніки всі експерименти з моделювання обробляються й узагальнюються за допомогою ЕОТ. Проте самі моделі, як вже йшлося вище, можуть мати свою фізичну сутність. Тому в цьому разі електронне моделювання насамперед асоціюється зі створенням моделі фізичного об'єкта або процесів, що відбуваються в ньому під час функціонування, на машинних засобах ЕОТ.

Усі задачі з дослідження процесів поділяються на *прямі* та *обернені*. Перші відповідають на запитання: що буде, коли за заданих умов приймається якесь рішення? Зокрема, чому дорівнюватиме при заданому розв'язку вибраний показник ефективності  $J$ . *Обернені* задачі відповідають на запитання: як слід вибирати суттєві параметри  $x_1, x_2, \dots$  процесу для того, щоб показник ефективності  $J$  прямував до максимуму (мінімуму чи якогось іншого наперед заданого значення)?

Отже, у найзагальнішому вигляді *постановку задачі оптимізації* можна сформулювати так: є деякий процес, на якість перебігу якого можна деякою мірою впливати, вибираючи тим чи іншим способом залежні параметри  $x_1, x_2, \dots$ , а ефективність процесу характеризується показником  $J$ , який необхідно перетворити в максимум (мінімум).

Нехай пряму задачу розв'язано, і математична модель процесу дає змогу обчислити значення  $J$  за будь-яким вибраним розв'язком і для будь-якої сукупності умов  $a_1, a_2, \dots$ . Таким чином, заздалегідь відомо фактори (умови)  $a_1, a_2, \dots$ , на які впливати не можна (зокрема, відомо обмеження, що накладаються на розв'язок), а також є залежні фактори (параметри)  $x_1, x_2, \dots$ , які в певних межах можна вибирати. До речі, фактори  $a_1, a_2, \dots$  та  $x_1, x_2, \dots$  не обов'язково є числами; вони можуть бути, наприклад, функціями чи зв'язками, що накладаються на параметри.

Показник ефективності процесу запишемо у вигляді

$$J = J(a_1, a_2, \dots; x_1, x_2, \dots). \quad (6.1)$$

Вважається, що вид залежності відомо, і за допомогою такої математичної моделі можна обчислити при будь-яких заданих значеннях  $a_1, a_2, \dots; x_1, x_2, \dots$  (тобто пряма задача є розв'язуваною).

Обернену задачу сформулюємо так: за заданих умов  $a_1, a_2, \dots$  треба знайти такі параметри  $x_1, x_2, \dots$  процесу, які спрямовують показник  $J$  в максимум (мінімум). Це типова математична задача, яка належить до класу *варіаційних задач*. Методи їх розв'язання дуже поширені в математиці й добре розроблені. Найпростіші з них дуже відомі: вони потребують для визначення екстремуму функції диференціювання її за аргументами, прирівнювання похідних до нуля та розв'язування одержаної системи рівнянь.

Однак коли аргументів багато, задача пошуку екстремуму стає складнішою, та й екстремум часто досягається зовсім не в точці, де похідна перетворюється в нуль (такої точки може й не існувати), а десь на межі області змін аргументів. Виникають специфічні труднощі багатовимірної варіаційної задачі за наявності обмежень, яку, через її складність, не під силу розв'язати іноді навіть сучасній ЕОМ.

Крім того, якщо функція  $J$  взагалі не має похідних, а є цілочисловою або заданою тільки при цілочислових значеннях її аргументів, то вибираються *методи цілочислового програмування*. Наприклад, якщо функція  $J$  лінійно залежить від параметрів  $x_1, x_2, \dots$  процесу, а обмеження, що накладаються на ці параметри, мають вигляд лінійних рівностей або нерівностей, виникає *задача лінійного програмування*. Якщо ж  $J$  є опуклою функцією, то застосовуються спеціальні *методи опуклого програмування* з їх різновидом — *методом квадратичного програмування*.

Для оптимізації управління багатоетапними процесами можна застосувати *метод динамічного програмування*. Зазначимо також, що існує цілий набір числових методів відшукування екстремумів функцій багатьох аргументів, спеціально пристосованих для реалізації на обчислювальних машинах.

Досі йшлося про дуже прості задачі, в яких фігурують тільки дві групи факторів: умови  $a_1, a_2, \dots$  і параметри (елементи розв'язку)  $x_1, x_2, \dots$  процесу. Реальні ж задачі, що розв'язуються під час моделювання виробничих систем, частіше зводяться до таких, де наявна третя група факторів — невідомі фактори  $f_1, f_2, \dots$ , значення яких не можна завбачити:

$$J = J(a_1, a_2, \dots; x_1, x_2, \dots; f_1, f_2, \dots). \quad (6.2)$$

Тоді задачу оптимізації процесу можна сформулювати так: за заданих умов  $a_1, a_2, \dots$  з урахуванням невідомих факторів  $f_1, f_2, \dots$  знайти такі значення параметрів (елементів розв'язку)  $x_1, x_2, \dots$  процесу, які, якщо це можливо, забезпечують максимальне (мінімальне) значення показника ефективності  $J$ .

Це вже не суто математична задача, оскільки наявність невідомих факторів переводить її в нову якість: вона перетворюється в *задачу про вибір розв'язку в умовах невизначеності*. Наведемо приклад такої задачі. Нехай на токарному верстаті планується обробляти заготовки. Параметри режимів обробки обмежено можливостями верстата (наприклад, його приводів), вимогами щодо якості оброблюваної деталі (температурні обмеження в зоні різання, силові деформації тощо), тобто існують граничні умови  $a_1, a_2, \dots$ . Постає запитання: яке співвідношення параметрів  $x_1, x_2, \dots$  треба підтримувати, щоб за наявності випадкових факторів  $f_1, f_2, \dots$  (зокрема нерівномірності твердості заготовок, розбіжності розмірів їх уздовж твірної обробки, ріжучих здатностей інструмента тощо) забезпечити необхідне значення показника ефективності  $J$  обробки (наприклад, мінімальної собівартості, максимальної продуктивності і т. ін.)?

Найпростішим випадком невизначеності є той, коли невідомі фактори  $f_1, f_2, \dots$  — випадкові величини (або випадкові функції), статистичні характеристики яких,



скажімо закон розподілу, відомо або в принципі їх можна одержати. Такі задачі дістали назву *стохастичних* (відповідно — *стохастичних процесів*), а притаманна їм невизначеність називається *стохастичною*.

Тепер розглянемо стохастичну задачу, невідомі фактори  $f_1, f_2, \dots$  в якій є звичайними випадковими величинами з деякими (в принципі відомими) імовірнісними характеристиками. Тоді показник ефективності  $J$ , що залежить від цих факторів, також буде величиною випадковою, і перше, що спадає на думку, — це вибір як показника ефективності не самої величини  $f$ , а її середнього значення (математичного сподівання  $M$ )

$$J_M = M [J(a_1, a_2, \dots; x_1, x_2, \dots; f_1, f_2, \dots)]. \quad (6.3)$$

і вибір таких параметрів  $x_1, x_2, \dots$  (та їх співвідношень), за яких це середнє значення прямує до максимуму (мінімуму).

Такий підхід, коли вибирається розв'язок, що перетворює на максимум (мінімум) середнє значення  $J_M$  показника ефективності  $J$ , є цілком виправданим, якщо йдеться про *повторювані процеси* (наприклад, про визначення середнього значення продуктивності за певний проміжок часу тощо), оскільки програш в одному випадку компенсується виграшем в іншому. Якщо ж йдеться про *унікальну операцію* (процес), ставиться вимога, щоб імовірність одержання неприпустимо малого (чи великого) значення  $J$  (скажімо,  $J < J_0$ , чи  $J > J_0$ , чи  $J = J_0$ ) була досить малою:

$$P(J \leq J_0, J \geq J_0) \begin{matrix} \leq \\ > \end{matrix} \gamma,$$

де  $\gamma$  — якесь мале число, настільки мале, що подію з імовірністю  $\gamma$  можна вважати практично неможливою.

Цю умову-обмеження можна враховувати при розв'язуванні задачі поряд з іншими обмеженнями (умовами). Тоді відшукується розв'язок задачі, який перетворював би в максимум (мінімум) середнє значення  $J_M$ , але з урахуванням додаткової, «перестраховальної» умови, яку наведено вище.

Ще гірші справи з розв'язуванням задачі, коли невідомі фактори  $f_1, f_2, \dots$  неможливо описати статистичними методами (або коли невідомі розподіли ймовірностей і їх не можна одержати або коли вони не існують взагалі). Це «*погана невизначеність*». Тоді йдуть таким шляхом: задаючи якість значень факторів  $f_1, f_2, \dots$ , знаходять для них умовно-оптимальний розв'язок, який, проте, не буде оптимальним для реальних умов. Тому вибирають не оптимальний для якихось умов розв'язок, а деякий компромісний розв'язок, який не є оптимальним для жодних умов, але все-таки буде сприятливим у всьому їх діапазоні. До таких умов наближається і *метод експертних оцінок*.

Розглянемо ще один підхід до вибору розв'язку в умовах невизначеності — застосування *адаптуючих алгоритмів управління*: спочатку вибирається будь-який (напевно, не найкращий) алгоритм і в міру застосування раз від разу він «поліпшується», тобто йде робота за принципом «учитися на помилках». Важливим тут є тільки те, щоб досліджуваний процес (операція) належав до категорії неодноразово повторюваного.

Нарешті, може виникнути не «погана», а «*ворожа невизначеність*», що буває в конфліктних ситуаціях з різними інтересами. Тоді використовується спеціальний розділ математики — *теорія ігор* (наприклад, несумісними є вимоги до задоволення мінімальної собівартості при максимальній продуктивності за умови забезпечення найвищої точності обробки). Тоді, скориставшись теорією статистичних розв'язків,

математично обґрунтовуються раціональні розв'язки в умовах невизначеності. При цьому слід пам'ятати про зауваження: при обґрунтуванні розв'язку в умовах невизначеності, що б ми не робили, елемент невизначеності залишається. Через те вимагати точного розв'язання таких задач навряд чи розумно, і замість того, щоб намагатися дістати однозначно один єдиний, оптимальний з якоїсь точки зору розв'язок, краще виділити цілу область прийнятних розв'язків, які будуть несуттєво гіршими від інших, хоч би якою точкою зору ми користувалися.

**Багатокритеріальні задачі.** І все ж таки, як бути, коли використовується не один єдиний показник ефективності  $J$  для оцінювання виробничого (технологічного) процесу? Така ситуація в машино- та приладобудуванні зустрічається на кожному кроці (згадаймо хоча б наведений вище приклад): деякі з показників бажано спрямувати до максимуму, а інші — до мінімуму. Постає запитання: чи можна знайти розв'язок, який би одночасно задовольняв усі ці вимоги? Звичайно, відповіддю буде — ні. Тож законним є таке формулювання: досягнення заданого ефекту при мінімальних витратах або ж досягнення максимального ефекту при заданих витратах.

На практиці застосовується такий спосіб складання узагальненого показника ефективності: беруть окремі показники і приписують їм якісь ваги ( $b_1, b_2, \dots$ ); перемножують кожен показник на відповідну вагу і додають; ті показники, що мають бути максимізовані, входять до суми зі знаком «плюс», а мінімізовані — зі знаком «мінус»:

$$J_{yz} = b_1A + (-)b_2B.$$

Однак вагові коефіцієнти  $b_1$  і  $b_2$  не є сталими, а залежать як від величин  $A$  та  $B$ , так і від обставин. Вихід можна знайти, якщо спочатку розв'язати пряму задачу й одержати статистичну оцінку впливу коефіцієнтів  $b_1, b_2$  на складові узагальненого показника  $J_{yz}$ , а потім, розв'язуючи обернену задачу, використати бажану оцінку в окремих випадках. Цей підхід не дає змоги знайти найефективніший розв'язок багатокритеріальних задач, але інших поки що немає, оскільки застосування будь-якого з методів розв'язання дає змогу знайти тільки розв'язок, оптимальний за одним — скалярним критерієм  $J$ .

За цих обставин людина мислить не формально, а узагальнено, інтуїтивно оцінюючи ситуацію в цілому, відкидаючи несуттєві дрібниці та використовуючи підсвідомо весь свій досвід стосовно якщо й не буквально таких, то аналогічних ситуацій, і приймає рішення. При цьому (неформальному) виборі компромісного рішення людині може суттєво допомогти математичний апарат, надаючи можливість задалегідь відкинути невдалі варіанти рішень, над якими немає сенсу думати. Продемонструємо один з таких прийомів попереднього відбраковування невдалих варіантів рішень, який дуже легко реалізується машинними засобами.

Нехай потрібно зробити вибір між кількома варіантами розв'язків  $X_1, X_2, \dots, X_n$  (кожен з варіантів є вектором, складові якого — параметри досліджуваного процесу). Ефективність процесу оцінюється двома показниками: продуктивністю  $N$  і вартістю  $C$ . Перший показник бажано максимізувати, а другий — мінімізувати.

Кожний з варіантів рішення приводить до деяких значень показників  $N$  і  $C$ . Зобразимо для наочності кожен з цих варіантів у вигляді точки  $X_i$  на площині  $NOC$ , координати якої характеризують значення показників  $N_i$  та  $C_i$  (рис. 6.1, *a*). Маємо сукупність точок на площині  $NOC$ , і ця сукупність займає якусь область. Відразу видно, що деякі з варіантів рішення свідомо не вигідні і їх необхідно відкинути.

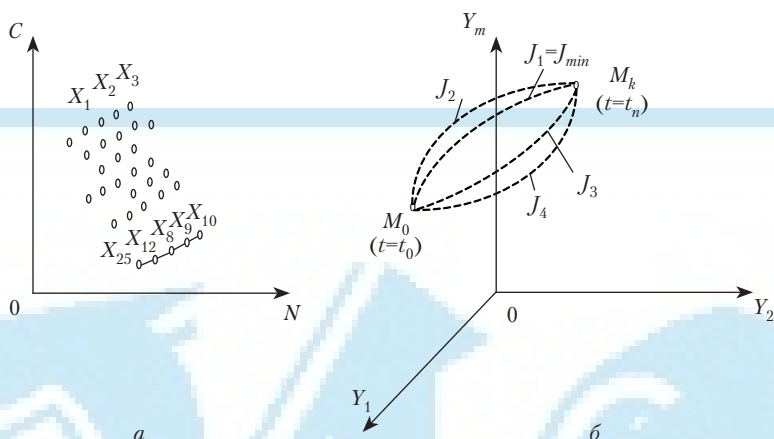


Рис. 6.1. Зображення варіантів розв'язків двокритеріальної задачі (а) та геометричне інтерпретування у фазовому просторі оптимального управління (б)

Справді, ті варіанти, що мають перевагу в продуктивності, повинні лежати праворуч від межі області можливих варіантів. Ті ж варіанти, які за однакової продуктивності мають меншу вартість, повинні розташовуватись на нижній межі цієї області. Отже, перевагу треба віддавати тим варіантам рішень, що лежать відразу і на правій, і на нижній межах області (на рис. 6.1, а це ряд  $X_{25} - X_{12} - X_8 - X_9 - X_{10}$ , який позначено кривою).

Таким чином, надалі залишається проаналізувати тільки п'ять з розташованих на граничній кривій варіантів рішення. З них  $X_{10}$  є найбільш продуктивним, але порівняно дорогим, а  $X_{25}$  — найдешевшим, однак не таким продуктивним рішенням.

Аналогічно виконується відкидання непридатних варіантів рішень і тоді, коли показників ефективності процесу не два, а більше (проте якщо їх більше ніж три, втрачається наочність геометричного інтерпретування, але суть процедури вибору рішення залишається тією самою, а кількість конкурентоспроможних варіантів рішення різко скорочується). Що ж стосується остаточного вибору того чи іншого варіанта рішення, то це залишається прерогативою людини. І тут на допомогу приходять *евристичні методи прийняття рішень*.

**Загальні принципи оптимального управління виробничими процесами.** Процес оптимізації залежить від умов (обмежень)  $a_1, a_2, \dots$ , а також факторів (наприклад, зовнішнє збурення)  $f_1, f_2, \dots$ , які діють на нього. Для управління ним служать фактори (параметри управління)  $x_1, x_2, \dots$ . Тому задачу оптимального управління виробничими процесами можна поділити на три етапи: *перший*, пов'язаний з обґрунтуванням вибору значень параметрів управління  $x_i$  (режимів обробки, складання тощо, які задовольняють заданий показник якості); *другий*, спрямований на визначення умов (обмежень)  $a_j$ , що накладаються на процес (через обмеження значень параметрів управління); *третій* передбачає управління вихідною величиною  $Y$  (наприклад, зусиллям різання чи складання, споживаною потужністю устаткування тощо) з метою підтримки оптимального значення показника якості в умовах впливу на процес збурюючих факторів і за умови обмежень регулювання.

Як уже гадувалось, окремим випадком оптимального управління є екстремальне управління, що найчастіше реалізується *самоналагоджуваними* або *адап-*



## 6.2. Деякі підходи до моделювання в комбінованих інтелектуальних системах

До основних задач моделювання в комбінованих інтелектуальних системах можна зарахувати:

- системний аналіз взаємодії змінних у складних об'єктах з метою визначення множини характеристичних змінних і розподілу її на підмножини вхідних та вихідних (незалежних і залежних) змінних;
- структурну й параметричну ідентифікації об'єктів, що моделюються, з уже відомими моделями взаємозмінних, які їх характеризують;
- якісне та кількісне прогнозування процесів з різною попередністю;
- планування, управління та прийняття рішень.

При цьому існує дещо принципова відмінність двох підходів до моделювання: а) базування тільки на досвіді спостережень за діями об'єкта, поданого у вигляді уявлень користувача, що склалися в нього про об'єкт; б) порівняння ряду розрахунків на основі дослідних даних як результатів проведення активного чи пасивного експерименту. *Перший* супроводжується використанням принципів *імітаційного моделювання*, а *другий* — *експериментального методу самоорганізації моделей* [35].

Уразливим місцем імітаційних методів є те, що користувач загальної моделі змушений зазначити закономірності для всіх елементів досліджуваної системи, в тому числі й для тих з них, які він знає гірше від інших або які взагалі найменшою мірою піддаються імітації. Тому найкраще застосовувати *комбінований метод*, у якому використовуються обидва підходи і який забезпечує найвищу ефективність дослідження складних об'єктів. При цьому моделі погано вивчених елементів формуються за допомогою експериментальних методів самоорганізації моделей, тоді як відомі елементи задаються безпосередньо користувачем моделі.

У цьому разі побудова динамічної моделі процесу зводиться до виконання таких етапів:

- виявлення змінних, які характеризують процес, і планування експериментів;
- реєстрування даних під час спостережень за об'єктом при проведенні експериментів (зміни параметрів слід проводити в однакові інтервали часу та кроків у просторі вимірювань; максимально припустимий крок таблиці треба визначати за теоремою відліків Котельникова — Шеннона);
- вибір фізичного закону, який доцільно покласти в основу моделі (елемент імітаційного моделювання). Якщо в основу моделювання той чи інший фізичний закон покласти важко, застосовується *індуктивний підхід*, згідно з яким *структура моделі оптимальної складності підбирається випробуванням кількох моделей-претендентів за критеріями селекції*. При цьому в просторі критеріїв кожній моделі відповідає певна характеристична точка. Відповідно до принципу самоорганізації систем вибирається деяка кількість точок (отже, й моделей), найближче розташованих до початку координат. Процедура до-визначення моделі полягає в тому, що з відібраних кращих моделей береться одна за головним критерієм:
- заміна безперервного математичного опису його дискретним аналогом (різницеvim рівнянням);
- оцінювання й адаптування (облік за всіма даними) коефіцієнтів різницевого рівняння (елемент самоорганізації моделей).

Самоорганізація (тобто перебирання) моделей, так само як регресійний аналіз, належить до експериментальних методів моделювання, оскільки вона ґрунтується на обробці *пасивного* (спостереження за нормальною роботою об'єкта, як правило, при ручному керуванні ним) або *активного* (зміна зовнішніх дій здійснюється за спеціальною програмою) експериментів. Але якщо в регресійному аналізі структура моделі задається, то самоорганізація моделей передбачає визначення *моделі оптимальної складності*, коли при поступовому ускладненні структури моделі (наприклад, при збільшенні кількості членів і степені поліноміальної моделі) значення зовнішніх критеріїв спочатку зменшуються, а потім збільшуються, тобто є їх мінімум, який визначає модель оптимальної складності.

Загалом процес самоорганізації подібний до роботи *генетичних алгоритмів*, які починають свою роботу з деякого набору даних і процедур і намагаються комбінувати їх різним способом (відбувається *схрещування*). Далі з цих комбінацій відбираються найкращі за певним критерієм (*селекція*), і саме вони беруть участь у подальших схрещуваннях. Інколи до даних і процедур, які залучені до еволюційного відбору, вносяться певні випадкові зміни (відбуваються *мутації*). Генетичні алгоритми добре зарекомендували себе для розв'язання ряду погано формалізованих та важкорозв'язуваних задач.

Оскільки аналітична залежність того чи іншого критерію від складності моделі невідома, задача пошуку його мінімуму розв'язується *методом індукції*, тобто шляхом повного або неповного спрямованого перебирання багатьох моделей-претендентів. У програмах, що реалізують самоорганізацію моделі, передбачаються блок генерування множини моделей-претендентів, блок обчислення значень певного критерію та вихідний блок вибору моделі оптимальної складності за мінімумом цього критерію.

**Особливості використання методу групового врахування аргументів.** Алгоритми самоорганізації моделей можуть бути однорядними (кількість членів повного опису  $m < 20$ ) або багаторядними ( $m < 1000$ ), причому як перші, так і другі реалізують *метод групового врахування аргументів* (МГУА) [34].

*Метод повної математичної індукції* (тобто повного перебирання всіх варіантів моделі, які можна одержати із заданого повного опису) реалізується в *однорядних* (комбінаторних) *алгоритмах* МГУА, що застосовуються для розв'язання визначених і перевизначених задач моделювання з кількістю членів повного опису  $m$  (найчастіше полінома), яка є меншою від кількості точок у таблиці дослідних даних  $M$  (тобто  $m < M$ ).

Для розв'язання недовизначених задач ( $M < m$ ) створено багаторядні алгоритми МГУА, що є засобом раціонального перебирання моделей, за яким кількість випробуваних моделей значно зменшується з дуже малим ризиком втрати найоптимальнішої моделі (тут кількість моделей, що перебираються ЕОМ, збільшується не в степеневій, а в поліноміальній залежності від кількості змінних).

При переході від однорядних алгоритмів до багаторядних можлива як втрата найоптимальнішої моделі, так і невелика «похибка багаторядності» (аналогічна статичній похибці систем управління), яку для поліноміальних алгоритмів можна визначити як різницю прогнозувань (рішень), одержаних за одно- та багаторядними алгоритмами. Багаторядний алгоритм, так само як система стабілізування, збігається до стану рівноваги — «точки згущення».

Для багаторядних алгоритмів селекції моделі в алгоритмах самоорганізації дуже корисним є *принцип неостаточних рішень* [110], за яким з ряду в ряд селекції пере-

дається не одна, а кілька кращих за критеріями моделей, тобто на кожному з рядів зберігається «свобода вибору», і тільки на останньому ряді вибирається єдина (для кожного ансамблю критеріїв) модель оптимальної складності.

Проте вибір критеріїв з ансамблю все ще є великою майстерністю, й тому метод самоорганізації моделей дістав назву *евристичного*. Для забезпечення самоорганізації моделі можна, наприклад, включити до складу ансамблю *критерій мінімуму зсуву*

$$m_{\text{зс}}^2 = \sum_1^M (\hat{g}_A - \hat{g}_B)^2 / \sum_1^M g_{\text{ф}}^2 \rightarrow \min, \quad (6.7)$$

де  $\hat{g}_A$ ,  $\hat{g}_B$  — виходи моделей однакової структури, коефіцієнти яких обчислено на частинах  $A$  та  $B$  вибірки;  $g_{\text{ф}}$  — фактичні (табличні) дані;  $M = M_A + M_B$  — множина точок початкових даних, поділена на дві частини:  $M_A \approx 70\%$  і  $M_B \approx 30\%$ .

Можна також використати *критерій балансу змінних*

$$H = \sum_1^M h_i^2 \rightarrow \min, \quad h_i = f_1(g_i) - f_2(g_i), \quad (6.8)$$

де  $h_i$  — непогодженість допустимого зв'язку функцій  $f_1$  і  $f_2$ .

Перший критерій рекомендується застосовувати тоді, коли необхідно визначити фізичну модель, а другий — при розв'язуванні задачі довизначення моделі на довготерміновому прогнозуванні.

**Однорядні (комбінаторні) алгоритми.** В комбінаторних алгоритмах виконується перебирання всіляких окремих моделей із заданого базису (поліномного чи гармонійного) з вибором найкращої за заданим критерієм селекції. При перебиранні складність окремих моделей (тобто кількість аргументів або частот) поступово нарощується від 1 до  $m$  ( $m$  — кількість аргументів базисного набору функцій).

Отже, загальна схема комбінаторного алгоритму зводиться до виконання таких операцій: за *методом найменших квадратів* визначаються коефіцієнти всіх окремих моделей при складності  $K=1, m$ ; для кожної з них обчислюється значення зовнішнього індивідуального чи комбінованого критерію селекції; єдина модель оптимальної складності вибирається за найменшим значенням вибраного критерію.

Таким чином, комбінаторний алгоритм МГУА ґрунтується на методі повної математичної індукції, оскільки при цьому не упускається жоден з можливих варіантів моделі, закладених у початковому повному базисі.

**Приклад 6.1.** Розглянемо кілька прикладів. Нехай задано повний квадратичний поліном для випадку трьох змінних

$$g = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_1^2 + b_5x_2^2 + b_6x_3^2 + b_7x_1x_2 + b_8x_1x_3 + b_9x_2x_3. \quad (6.9)$$

Маємо 10 членів полінома ( $m = 10$ ). У будь-якому разі повний поліном є лінійним за коефіцієнтами, для визначення яких використовується метод найменших квадратів. При цьому степені та коваріації початкових змінних розглядаються як еквівалентні перепозначені лінійні аргументи. Тоді відповідно до наведеної вище схеми алгоритм працює так. Спочатку визначаються всі моделі при  $K = 1$  (тобто такі, що складаються лише з одного аргументу):  $g_1 = b_0$ ;  $g_2 = b_1x_1$ ;  $g_3 = b_2x_2$ ; ...;  $g_{10} = b_9x_2x_3$  (отже, таких моделей  $C_{10}^1 = 10$ ).

Далі розглядаються всілякі моделі при  $K = 2$  (що складаються з двох аргументів):  $g_{11} = b_0 + b_1x_1$ ;  $g_{12} = b_0 + b_2x_2$ ; ...;  $g_i = b_0 + b_9x_2x_3$ ;  $g_j = b_1x_1 + b_2x_2$ ; ...;  $g_k = b_1x_1 + b_9x_2x_3$ ; ...;  $g_{45} = b_8x_1x_3 + b_9x_2x_3$ , яких уже буде  $C_{10}^2 = 45$ . Аналогічно одержуємо  $C_{10}^3$  (для  $K = 3$ ),  $C_{10}^4$  (для  $K = 4$ ) і так далі до  $C_{10}^{10} = 1$  (для  $K = m$ ), тобто до повного перебирання варіантів моделі.

Загальна кількість  $N_m$  окремих моделей, побудованих з  $m$  аргументів (доданків) повного полінома методом повного комбінаторного перебирання, обчислюється за формулою

$$N_m = \sum_{K=1}^m C_m^K = 2^m - 1. \quad (6.10)$$

Зі збільшенням  $m$  швидко збільшується  $N_m$ , і тому можливості комбінаторного перебирання обмежені, а його максимальна складність  $K_{\max}$  задається виходячи з можливостей ЕОМ (як правило,  $m \leq 18$ )

Якщо йдеться про гармонічний аналіз полінома (6.9) з дискретним набором гармонік, тобто

$$x_i = E_i + L_i \sin \omega_i t + R_i \cos \omega_i t, \quad i = \overline{1, m}, \quad (6.11)$$

то коефіцієнти  $E_i, L_i, R_i$  визначаються за методом найменших квадратів на всіх  $M$  точках початкових даних. При цьому, починаючи з  $K = 2$ , обчислюються коефіцієнти в моделях відносно вже визначених гармонічних аргументів  $g_i = b_0 + b_1 x_i + b_2 x_j$ . Обсяг перебирання становить  $N_m = 2^m - 1$ , і на такий алгоритм накладаються ті самі обмеження, що й на поліномний.

Гармонічні алгоритми з аналітичним визначенням або з перебиранням частот процесу, що визначені аналітично, теж мають ті самі обмеження за швидкодією, але вони принципово відмінні від звичайного гармонічного алгоритму, оскільки тут частоти не призначаються, а визначаються аналітично, і тому вони максимально наближені до істинних гармонік коливального процесу.

**Багаторядні алгоритми.** Такі алгоритми можуть не передбачати обчислення залишків (рис. 6.2, а) або передбачати їх (рис. 6.2, б) [36]. Спочатку розглянемо перший випадок, вважаючи, що початковий склад аргументів, з якого починається процедура багаторядної селекції моделі процесу, будується на нульовому ряді алгоритму, який по-різному організовується в поліномних і гармонічних алгоритмах. Наприклад, у класі алгебраїчних функцій найбільш загальною моделлю є поліном Колмогорова — Габора, що містить  $v$  змінних:

$$g = b_0 + \sum_{i=1}^v b_i \alpha_i + \sum_{i=1}^v \sum_{j=1}^v b_{ij} \alpha_i \alpha_j + \sum_{i=1}^v \sum_{j=1}^v \sum_{l=1}^v b_{ijl} \alpha_i \alpha_j \alpha_l + \dots \quad (6.12)$$

і є сумою лінійних, квадратичних, кубічних та інших членів.

Після перетворення всіх його складових дістанемо лінійний поліном  $g = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_m x_m$ , члени якого і складають початковий набір аргументів, побудований на нульовому ряді багаторядного алгоритму.

У випадку різницевих моделей до аналогічного вигляду зводиться модель, яку можна одержати перепозначенням усіх початкових змінних, їх запізнювань і заданих нелінійних функцій.

Для багаторядних гармонічних алгоритмів (як з обчисленням залишків, так і без їх обчислення) початковий склад аргументів повинен будуватися на нульовому ряді так само, як і в комбінаторному гармонічному алгоритмі з перебиранням частот, визначених аналітично: спочатку одержуємо  $l$  моделей у вигляді суми однієї, двох, трьох і так далі до  $m$  гармонік з некрatними частотами, а потім всі наявні в цих моделях  $m = \frac{1}{2}l(l+1)$  гармонічних складових перепозначимо як нові аргументи:

$$x_i = L_i \sin W_i t + R_i \cos W_i t, \quad i = \overline{1, m}.$$

Багаторядні алгоритми, як правило, працюють за такою загальною схемою:

- *перший ряд* — на основі даних таблиці спостережень будуються окремі описи всіх попарних комбінацій початкових (перепозначених) аргументів з розташуванням за методом найменших квадратів вихідної змінної  $y$ :



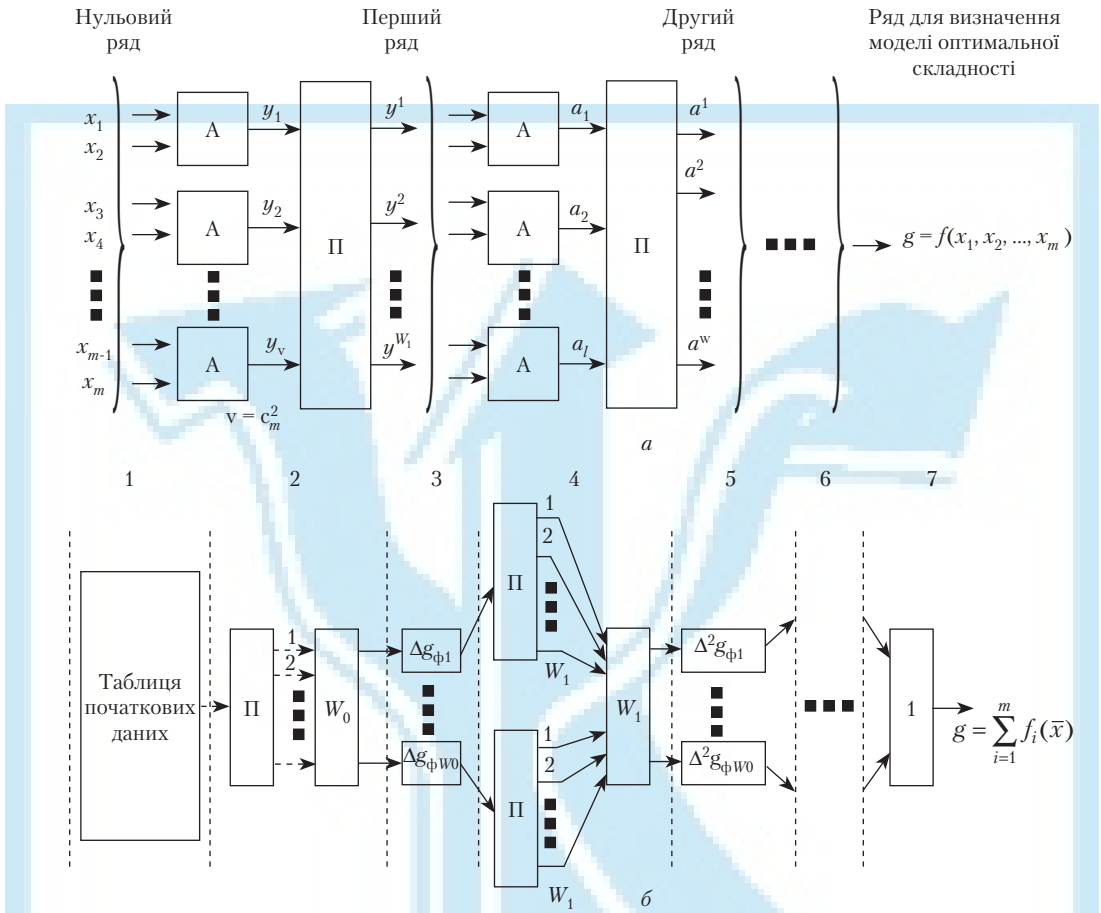


Рис. 6.2. Багаторядні алгоритми МГУА (А – алгоритм попарної обробки аргументів; П – пороговий відбір  $W$  найкращих моделей)

$$y_1 = f_1(x_1, x_2); y_2 = f_2(x_1, x_3); \dots; y_v = f_v(x_{m-1}, x_m),$$

і з цих  $v = C_m^2$  моделей відокремлюється деяка кількість  $W_1$  найкращих за зовнішнім критерієм;

- *другий ряд* – одержані змінні беруться як аргументи другого ряду і знову будуються всі окремі моделі двох аргументів:

$$a_1 = \xi_1(y_1, y_2); a_2 = \xi_2(y_1, y_3); \dots; a_l = \xi_l(y^{W_1-1}, y^{W_1}); l = C_{W_1}^2. \quad (6.13)$$

З них за зовнішнім критерієм відокремлюються  $W_2$  найкращих моделей як змінні для наступного ряду і т. д. Ряди нарощуються доти, поки знижується значення зовнішнього критерію.

Кожний такий опис може бути лінійною  $f = b_0 + b_1x_i + b_2x_v$  чи нелінійною  $f = b_0 + b_1x_i + b_2x_v + b_3x_ix_v$  (або  $f = b_0 + b_1x_i + b_2x_v + b_3x_ix_v + b_4x_i^2 + b_5x_v^2$ ) функцією двох змінних, коефіцієнти яких можна визначити за методом найменших квадратів, маючи відповідно три, чотири (або шість) точок спостережень у навчальній послідовності. Виключивши проміжні змінні після призупинення алгоритму, дістанемо мо-

дель, кількість коефіцієнтів якої значно перевищує кількість точок спостережень. Таким чином, у багаторядних алгоритмах МГУА відтворюється *схема масової селекції*.

Структура багаторядних алгоритмів МГУА з обчисленням залишків (рис. 6.2, б) містить: 1 – таблицю початкових даних; 2 – нульовий ряд, на якому за комбінаторним алгоритмом МГУА здійснюється вибір  $W_0$  найкращих моделей; 3 – обчислення перших різниць (залишків)  $\Delta g_{\text{ф}} - g^*$  (тут  $g_{\text{ф}}$  – фактичні дані таблиці спостережень;  $g^*$  – вихід моделі), які утворюють  $W_0$  таблиць з даними перших залишків; 4 – перший ряд селекції, на якому за комбінаторним алгоритмом МГУА визначаються  $W_0 N_m$  моделей, з яких вибираються  $W_1$  найкращих; 5 – обчислення других різниць (залишків)  $\Delta^2 g_{\text{ф}} = \Delta g_{\text{ф}} - \Delta g^*$  (тут  $\Delta^2 g_{\text{ф}}$  – фактичне значення першого залишку;  $\Delta g^*$  – вихід моделі); 6 – наступні ряди селекції й обчислення залишків; 7 – останній ряд, на якому здійснюється вибір моделі оптимальної складності.

При нарощуванні рядів селекції такого алгоритму комбінований критерій селекції  $\rho^2 = m_{\text{зс}}^2 + \Delta^2(D)$  монотонно зменшується і селекція зупиняється зовсім, щойно системний критерій починає збільшуватись. Тому сигналом для припинення лічби є досить мале зменшення системного критерію, коли об'єктивний характер вибору моделі втрачається.

Справді, поблизу мінімуму кривої комбінаторного критерію селекції розміщується послідовність характеристичних точок, які відповідають найкращим моделям. Таким чином, не обов'язково вибирати модель, що відповідає найнижчій точці на цій кривій. Порівнявши за вторинним критерієм (наприклад, точністю прогнозування, балансом змінних тощо) кілька найкращих моделей, можна вибрати оптимальну модель за двома критеріями.

Якщо будуються моделі на довгостроковий прогноз, то замість пари критеріїв  $m_{\text{зс}}^2$ ,  $\Delta^2(D)$  потрібно взяти пару критеріїв  $m_{\text{зс}}^2$ ,  $J_2(M)$ . Останнє стосується тільки поліномних різницевоїх моделей, оскільки аналог критерію  $J_2(M)$  для алгебраїчних і гармонічних моделей поки що добре ще не розроблено.

**Структура комбінованого алгоритму.** Як уже згадувалось, у структурі кожного з алгоритмів МГУА можна виділити три основні блоки [34]:

- перетворення початкових даних відповідно до вибраної системи опорних (базисних) функцій, в якій відшукується модель (базис може бути поліномним, тригонометричним, у вигляді різницевоїх рівнянь тощо);
- генерування (перебирання) повної або неповної множини ускладнюваних окремих моделей у вибраному базисі;
- обчислення значень деякого критерію селекції з властивостями зовнішнього доповнення та послідовного відбирання найкращих за цим критерієм моделей.

Відмінність усіх алгоритмів МГУА полягає в структурі «генератора моделей», тобто схемі обчислень, яка реалізується в другому блоці.

За цією ознакою алгоритми МГУА поділяються на *багаторядні, комбінаторні* та *мішані*, для яких характерними є специфічні схеми нарощування складності окремих описів. Нижче як приклад детальніше розглядаються обчислювальні особливості комбінаторного алгоритму МГУА для кожного з його основних блоків.

**Блок формування базису.** Розглянемо випадок поліномних опорних функцій. Якщо задано значення деяких вхідних змінних  $a_1, \dots, a_e$  об'єкта моделювання (ОМ) і максимального степеня полінома, то кількість доданків  $t$  у повному поліномі степеня  $v_{\text{max}}$  при  $e$  змінних визначається однозначно:

$$m = \prod_{j=1}^e \frac{v_{\max} + j}{j} = \frac{(v_{\max} + e)! - (v_{\max})!}{v!}. \quad (6.14)$$

Повний поліном при цьому записується у вигляді

$$y = \sum_{i=1}^m b_i \prod_{j=1}^e a_j^{v_{ij}} = \sum_{i=1}^m b_i x_i, \quad (6.15)$$

де кожний узагальнений лінійний аргумент  $x_i$  є нелінійною функцією початкових змінних  $a_j$ :

$$x_i = \prod_{j=1}^e a_j^{v_{ij}}. \quad (6.16)$$

Степені  $v_{ij}$  аргументів можуть набувати цілі значення  $0, 1, \dots, v_{\max}$  такі, що сума їх не перевищує значення  $v_{\max}$ , тобто  $\sum_{j=1}^e v_{ij} \leq v_{\max}$ . Для формування степенів аргументів зручно організувати процедуру одержання послідовності  $r$ -розрядних чисел  $v_i = (v_{1i}, \dots, v_{ri})$  з основою  $1, 2, \dots, v_{\max} + 1$  і відбирати з них ті, що задовольняють вказане обмеження. Наприклад, при  $r = 2$  (два аргументи) і  $v_{\max} = 2$  маємо послідовність  $m = 6$  чисел  $v_i$ ,  $i = \overline{1, 6}$ , а саме: 00, 10, 01, 20, 11, 02, що відповідає повному поліному

$$y = b_1 + b_2 a_1 + b_3 a_2 + b_4 a_1^2 + b_5 a_1 a_2 + b_6 a_2^2.$$

Отже, члени  $x_i$  повного полінома є базисним набором опорних функцій для комбінаторного алгоритму МГУА. Наведена процедура дає змогу сформувати матрицю вимірювань узагальнених аргументів  $X[M \times m]$ , де  $M$  — кількість точок вимірювань. Нехай ці точки розподілено (підряд) на три послідовності: навчальну  $A$  завдовжки  $M_A$ , перевірну  $B$  завдовжки  $M_B$  та екзаменаційну  $D$  завдовжки  $M_D$ , причому  $M_A + M_B + M_D = M$ .

Значимо, що для переходу до другого базису, наприклад різницевого чи тригонометричного, досить обчислити відповідно аргументи  $a$  згідно з (6.13) і потім скористатися процедурою формування матриці  $X$ .

*Блок генерування (перебирання) окремих моделей.* До основних операцій, що виконуються цим блоком, належать: формування структури чергової окремої моделі; формування відповідної нормальної системи рівнянь; розв'язування одержаної системи рівнянь (оцінювання коефіцієнтів моделі).

Формування структур окремих моделей формалізується за допомогою двійкового структурного вектора  $u = (u_1, u_2, \dots, u_m)$ . Якщо елемент  $u_i$  цього вектора набуває одиничного значення, то відповідний  $i$ -й аргумент включається в окрему модель, а якщо нульового — не включається ( $i = \overline{1, m}$ ).

Найпростішим способом організації зміни стану вектора  $u$  є такий, за яким спочатку одержують усі можливі варіанти розміщення у векторі  $u$  однієї одиниці (загалом  $C_m^1 = m$  варіантів), потім двох одиниць (загалом  $C_m^2 = \frac{1}{2}m(m+1)$  варіантів) і так далі аж до  $m$  одиниць ( $C_m^m = 1$  варіант). Загальна кількість варіантів згідно з формулою (6.10) становитиме  $N_m = 2^m - 1$  різних структур (повне перебирання). Проте програмна реалізація цього способу є далеко не швидкодіючою, і таке перебирання структур доцільно застосовувати тільки в алгоритмах їх неповного перебирання.

Більш компактною й універсальнішою є схема зміни двійкового вектора за принципом роботи двійкового лічильника, до останнього розряду якого додається одиниця. Цей спосіб алгоритмічно є дуже простим і зручним саме при повному перебиранні, незважаючи на те, що кількість і склад аргументів в окремих моделях увесь

час змінюються (наприклад, при  $m = 3$  послідовно формуються структури 001, 010, 011, 100, 110, 111). До того ж він дає змогу навіть організувати рекурентну процедуру перебирання моделей.

Для формування нормальної системи рівнянь, що відповідає черговому структурному вектору, можна піти формальним шляхом: із стовпців повної матриці  $X$ , зазначених одиничними елементами  $u$ , складається окрема матриця  $X_b$ , а потім обчислюються елементи  $X_l^T, X_l, X_l^T y$  нормальної матриці. Проте при повному перебиранні цей формальний шлях є найгіршим, оскільки призводить до багаторазового обчислення тих самих скалярних добутоків. Наприклад, для структур 010, 011, 110 тричі обчислюється та сама величина  $x_2^T x_2 = \sum_{j=1}^M x_{2j}^2$ . Тому в комбінаторному алгоритмі МГУА досить тільки один раз обчислити матриці повної нормальної системи, яка містить елементи всіх окремих нормальних систем:

$$X^T X = \begin{pmatrix} x_1^T x_1 & x_1^T x_2 & \dots & x_1^T x_m \\ x_2^T x_1 & x_2^T x_2 & \dots & x_2^T x_m \\ \dots & \dots & \dots & \dots \\ x_m^T x_1 & x_m^T x_2 & \dots & x_m^T x_m \end{pmatrix}, \quad X^T Y = \begin{pmatrix} x_1^T y \\ x_2^T y \\ \dots \\ x_m^T y \end{pmatrix}. \quad (6.17)$$

*Блок відбирання окремих моделей за зовнішнім критерієм.* Обчислення в цьому блоці, як і в попередньому, організовується з урахуванням застосовуваних зовнішніх критеріїв. Якщо критерій є несиметричним (типу регулярності  $\Delta^2(B)$ ), то оцінювання коефіцієнтів  $b_l$  виконуються на одній послідовності, а значення критерію відшукується на іншій (що й забезпечує його зовнішній характер):

$$\Delta_l^2(B) = \left( y_B - X_{lB} \hat{b}_l(A) \right)^T \left( y_B - X_{lB} \hat{b}_l(A) \right). \quad (6.18)$$

Якщо критерій є симетричним, то для тієї самої структури  $u$ , коефіцієнти  $b_l$  обчислюються на обох послідовностях ( $A$  та  $B$ ), а потім визначається, наприклад, критерій мінімуму зсуву

$$m_{зс}^2 = \left( X_l \hat{b}_l(B) - X_l \hat{b}_l(A) \right)^T \left( X_l \hat{b}_l(B) - X_l \hat{b}_l(A) \right). \quad (6.19)$$

Селекція (відбирання) найкращих моделей за заданим критерієм  $J$  зазвичай виконується не наприкінці їх перебирання (коли сформовано всі окремі моделі), а в його процесі. З цією метою заповнюються значення критерію  $J$  для заданої кількості  $W$  перших моделей, після чого значення критерію  $J_l$  для кожної наступної моделі порівнюються з найгіршим  $J_{\max}$  із  $W$  значень. Якщо  $J_l < J_{\max}$ , то нова модель (структура, оцінки коефіцієнтів і критерій) запам'ятовується замість найгіршої; якщо ж  $J_l \geq J_{\max}$ , то ця модель (тобто  $J_l$ ) з подальшого аналізу виключається. Після закінчення перебирання моделей ті  $W$  з них, які залишилися, і є найкращими з усіх  $N_m$  за заданим критерієм.

В алгоритмах самоорганізації моделей застосовується також відбирання їх за двома й більше критеріями. Проте, наприклад, використання *комбінованих критеріїв*  $d^2 = \beta m_{зс}^2 + (1 - \beta) \Delta^2(B)$ , що складаються із суми двох наведених критеріїв, утруднюється складністю вибору вагового коефіцієнта  $\beta$ . Тому зазвичай кілька критеріїв використовуються послідовно — за ступенем важливості тієї чи іншої властивості моделей, що оцінюються за певним критерієм. Наприклад, за критерієм  $m_{зс}^2$  відбираються  $W_1$  найменш зсунених моделей, потім з них за критерієм  $\Delta^2(B)$  здійснюється вибір  $W < W_1$  найточніших на перевірній послідовності.

Для підвищення достовірності та завадостійкості відбирання моделей застосовується інформація третьої частини  $D$  вибірки. При цьому спочатку виконується етап адаптації коефіцієнтів — перелік оцінювань  $W_1$  моделей, що були відібрані на попередньому етапі, за точками двох перших вибірок  $A \cup B$ . Потім з  $W_1$  моделей відбираються  $W < W_1$  за новим критерієм регулярності  $\Delta^2(D)$ . Наприклад, двоетапне відбирання ( $m_{zc}^2 \rightarrow \Delta^2(D)$ ) моделей набагато підвищує завадостійкість моделювання за комбінаторним алгоритмом МГУА.

В алгоритмах МГУА зазвичай виконується ще один етап обчислень — оцінювання якості відібраних найкращих моделей. При цьому обчислюються, наприклад, квадратична похибка апроксимації  $\delta^2(A + B)$  (точність оцінювання на  $A \cup B$ ) і похибка екстраполяції (прогнозування)  $\Delta^2(D)$ .

### 6.3. Гібридні інтелектуальні системи

**Загальні принципи побудови.** Функціонування *гібридних інтелектуальних систем* (ГІС) ґрунтується на взаємодії компонентів, які відображають технологію ШІ (головно ЕС), з моделями та методами, що стосуються інших, як правило, суміжних галузей знань (зокрема дослідження операцій), в тому числі з алгоритмами числової оптимізації, ІМ. Доцільність такого функціонування ГІС полягає у взаємному доповненні переваг різних підходів, у спробі покласти на один з них розв'язання задач, які є складними для іншого підходу.

Взаємодія ЕС і зовнішніх компонентів у ГІС може бути непрямою і безпосередньою. При *непрямій взаємодії* функціями ЕС можуть бути: вибір одного з методів пошуку рішень, реалізованого алгоритмічно й поданого з наявного банку алгоритмів; аналіз умов задачі й зарахування її до одного з типових класів (або розпізнавання її типу); перевірка несуперечності умов; зведення обмежень до вигляду, що є необхідним (стандартним) для застосування вибраного розв'язувального алгоритму; формування (синтез) стратегії пошуку із застосуванням кількох процедур, які є у банку алгоритмів.

При *безпосередній взаємодії* функціонування ЕС і зовнішніх компонентів може бути рівноправним або підпорядкованим. Однак такий розподіл функціонування є досить умовним, оскільки ґрунтується він на відомостях про джерела ініціалізації окремих модулів і впливає більше з концептуальних уявлень про стратегію пошуку, ніж з реалізації конкретного механізму управління взаємодією.

Якщо йдеться про *рівноправне функціонування* ЕС і зовнішніх компонентів, то дії ЕС та алгоритмічних модулів відповідають одному рівню в логічній структурі розв'язання задачі. Такі дії можуть бути окремими послідовностями етапів деякого загального методу. Так, в оптимізаційній задачі спочатку ЕС вибирає деяку точку як початкове наближення, а потім виконується числовий алгоритм, за яким і визначається точка оптимуму. Ініціювання окремих компонентів (тобто ЕС та алгоритмічних модулів) у цьому разі має фіксовану черговість, що не залежить від проміжних результатів і може виконуватись деяким зовнішнім механізмом синхронізації.

При *підпорядкованому функціонуванні* ЕС і зовнішніх компонентів одна з них розв'язує задачу, що є підзадачею щодо дій іншої (провідної) компоненти. Остання ініціює виконання підпорядкованої компоненти, до того ж робить це або в разі необхідності з урахуванням проміжних (своїх) результатів, або обов'язкові точки виклику підпорядкованої компоненти можуть задаватися початково, виходячи з логіки дій, які реалізуються провідною компонентою.

Наведемо два варіанти підпорядкованого функціонування ЕС і зовнішніх компонент.

1. Імітаційна модель (ІМ) виробничої системи звертається до ЕС, яка вирішує конфліктні ситуації в модельованому процесі, наприклад, вибирає один з кількох претендентів на обслуговування транспортним засобом (тут провідна компонента — ІМ, необхідність звернення до ЕС визначається в процесі роботи у разі появи конфліктних ситуацій).

2. Модуль числової оптимізації на кожній ітерації звертається до ЕС, яка пропонує напрямок наступною кроку (провідна компонента — модуль числової оптимізації як такий, що реалізує основний алгоритм; точки ініціалізації ЕС визначаються апіорі в схемі алгоритму, що реалізується).

Очевидно, ті самі компоненти в різних ситуаціях можуть бути на різних рівнях підпорядкованості. Так, ЕС, здійснюючи початкову локалізацію інтервалу пошуку, виконує перший етап оптимізації і, таким чином, є рівноправною з подальшою числовою процедурою пошуку екстремуму. Коли ж ця процедура потребує звернення до ЕС для оцінювання довжини чергового кроку, то ЕС відіграє підпорядковану роль.

Таким чином, характеризуючи взаємодію елементів ПС з точки зору умов ініціалізації ЕС, підпорядкованість слід зараховувати скоріше не до самих компонентів ПС, а до конкретних підзадач, які ними розв'язуються.

При визначенні способу виконання алгоритмічних модулів враховується факт звернення їх до ресурсів ЕС. Відсутність цього звернення кваліфікується як повністю числова (алгоритмічна) реалізація. До таких моделей можна зарахувати програми статистичної обробки даних, прості оптимізаційні процедури тощо; при цьому модулі можуть самі виконувати основні функції або ініціюватись ЕС (тобто розв'язувати підпорядковані задачі).

Організація звернень числових алгоритмів до ресурсів ЕС має на меті підвищення ефективності пошуку (скорочення кількості ітерацій, запобігання призупиненню на локальних екстремумах тощо) за допомогою більш гнучкого та потужного засобу, ніж просте включення в програму жорстких і фіксованих евристик.

При визначенні місця інтелектуальної компоненти в моделі ПС під час пошуку враховується наявність в ЕС здатності міркувати про сам метод пошуку. Так, наявність метарівня означає зокрема, що ЕС може синтезувати структуру алгоритмічного модуля з набору наявних базових блоків (фрагментів). Завданням такого синтезу може бути побудова алгоритму, придатного для існуючої постановки задачі, або формування такої його модифікації, за допомогою якої найкраще враховується характер наявних умов тощо.

У разі відсутності метарівня міркування інтелектуальної компоненти ЕС обмежується межами методу розв'язання поточної задачі. Наприклад, ЕС призначає початкове налагодження верстатів для ІМ виробничої системи (метарівень передбачав би зокрема синтез самої моделі за допомогою ЕС).

**Активізація компонентів та інформаційний обмін.** Побудова ПС може здійснюватися двома способами:

- разом з ЕС використовується готовий пакет прикладних програм;
- засоби ЕС та алгоритмічні модулі розробляються спільно.

У свою чергу, на *першому* шляху можуть бути два випадки: а) пакет передбачає звернення до програми тільки в діалозі через інтерфейс користувача; б) пакет може бути бібліотекою незалежно використовуваних програмних модулів. У *першому випадку* реалізація ПС можлива тільки за наявності допоміжної програми,

що емулює термінал користувача. Експертна система взаємодіє з пакетом через цю програму, відіграючи, таким чином, роль користувача. В *другому випадку* немає необхідності мати програму-емулятор, хоча може бути потрібен програмний інтерфейс, якщо в програмному середовищі ЕС не передбачено можливості звернення до зовнішніх процедур. Однак в обох випадках ЕС виконує, як правило, рівноправну або провідну функцію. Підпорядковану функцію вона реалізує тільки тоді, коли пакет програм орієнтований на діалогове розв'язання задач.

Що стосується *другого* шляху (тобто спільного розроблення ЕС та алгоритмічних модулів), то, коли останні дістають повністю числову реалізацію, результат буде схожий (з точки зору взаємодії, а не ефективності рішень) на згаданий вище другий випадок побудови ГІС першим шляхом. Якщо ж розроблена алгоритмічна процедура передбачає звернення до ЕС, то у відповідних ситуаціях остання відіграватиме підпорядковану роль.

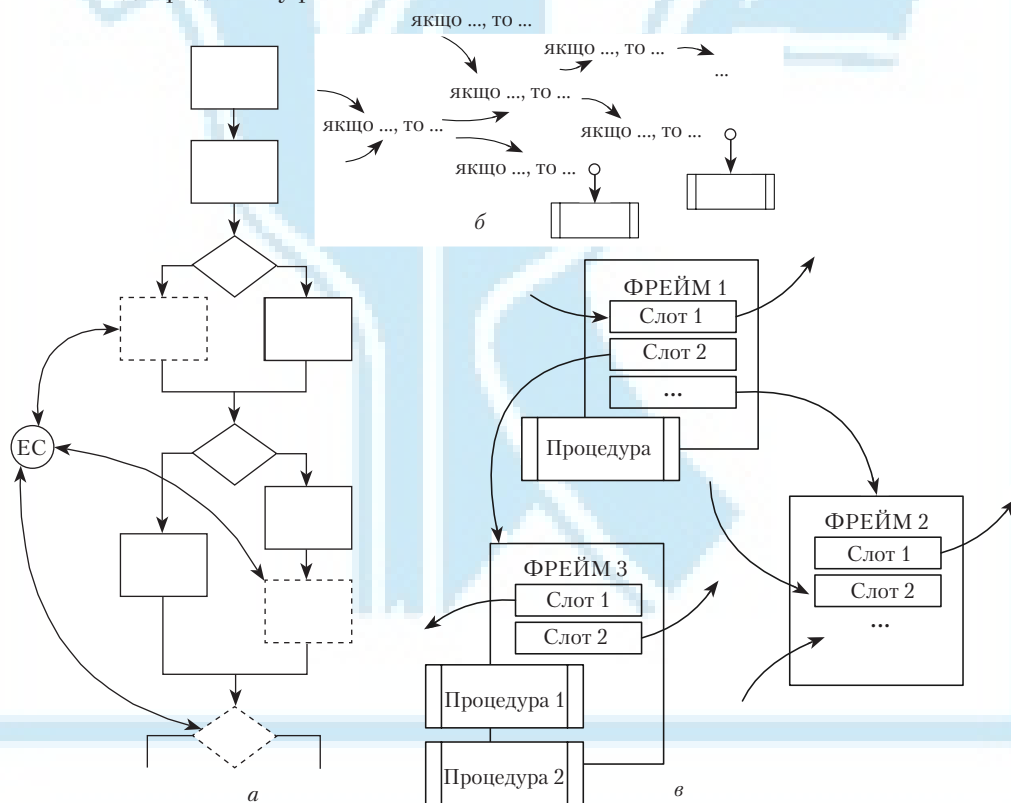


Рис. 6.3. Варіанти умовної структури взаємодії компонентів ГІС при їх підпорядкованому функціонуванні

У загальному випадку застосування готових пакетів прикладних програм у ГІС є доцільним при розв'язанні задач, що пов'язані з різноманітними трудомісткими розрахунками, не потребують відповіді в реальному часі і для яких існують добре зарекомендовані програмні засоби. До таких задач зокрема належать задачі проектування, календарного планування, статистичної обробки даних тощо. В разі ж комбінованого підходу до побудови ГІС, що працюють у режимі реального часу,

доцільно розробити як інтелектуальну компоненту, так і алгоритмічні модулі з орієнтацією на їх спільне функціонування. До можливих застосувань таких систем можна зарахувати задачі диспетчеризування виробництва, моніторингу, керування рухами робочих органів устаткування тощо.

Варіанти умовної структури взаємодії компонентів ПС при їх підпорядкованому функціонуванні показано на рис. 6.3, де рис. 6.3, *а* відповідає випадку, коли провідна компонента — алгоритмічний модуль, а рис. 6.3, *б, в* — коли провідною компонентою є ЕС; до того ж рис. 6.3, *б* є випадком подання знань у вигляді продукційних правил, а рис. 6.3, *в* — у вигляді фреймів.

Найбільш природний варіант програмної реалізації звернення алгоритмічних процедур до ЕС (див. рис. 6.3, *а*) полягає у формуванні та застосуванні відповідних інтерфейсних функцій (функцій узгодження), аргументи яких передаються в ЕС для виконання виведення. У загальному випадку таке звернення може мати вигляд:

ім'я функції  $(x_1, \dots, x_n; c_1, \dots, c_k; a_1, \dots, a_m)$ ,

де  $x_1, \dots, x_n$  — параметри, значення яких використовуються як умови (посилання) при виведенні в ЕС;  $c_1, \dots, c_k$  — керуючі параметри, що задають режими виведення (прямий або зворотний, стратегії вибору правил, покажчики наборів продукцій тощо);  $a_1, \dots, a_m$  — змінні, що містять допоміжну інформацію (ідентифікатори точок звернення тощо).

Аналогічно може бути реалізована ініціалізація числових розрахунків ЕС. Зокрема ініціалізація їх ЕС продукційного типу досягається за допомогою правил, консеквентами яких служать оператори звернення (імена) до відповідних процедур, тобто загальний вигляд цих правил є таким:

ЯКЩО <умова>, ТО <ім'я процедури (аргументи)>.

Звернення до алгоритмічних модулів від інтелектуальних компонентів, які реалізовано на основі фреймового підходу, досягається використанням механізму приєднаних процедур (див. підрозд. 2.4), за допомогою яких визначаються необхідні програмні модулі. Виконання цих процедур здійснюється активізацією відповідних слотів фрейму.

Розглянемо такий приклад. Гібридна інтелектуальна система (ПС), що використовує фреймове подання знань, формує виробничі завдання в ГВС і розподіляє їх за модулями технологічного обладнання.

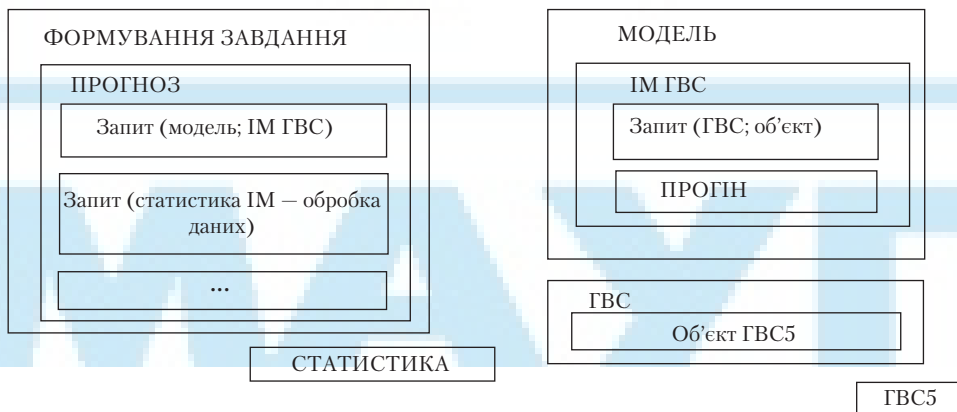


Рис. 6.4. Приклад можливої структури міжфреймових обмінів даними



Одним з етапів прийняття рішень є прогнозування їх можливих наслідків, яке виконується за допомогою імітаційного моделювання. Таким чином, ЕС реалізує провідну функцію, а ІМ — підпорядковану. При цьому прогін моделі розглядається як приєднана процедура.

Приклад можливої структури міжфреймових обмінів даними наведено на рис. 6.4. Значеннями більшої частини слотів фрейму «ФОРМУВАННЯ ЗАВДАНЬ» є приєднані процедури, що реалізують окремі етапи, пов'язані з таким формуванням. Процедура «ПРОГНОЗ» складається з формування функції «Запит»: перший запит стосується звернення до ІМ ГВС, другий — до модуля статистичної обробки даних. Службова функція «Запит» призначена для обміну повідомленнями між фреймами, її перший аргумент вказує ім'я фрейму-адресата, другий — ім'я слота.

Значеннями слотів фрейму «МОДЕЛЬ» є приєднані процедури, що реалізують моделювання окремих складових виробництва: ГВС, ЕРК, маніпуляторів тощо. Приєднана процедура «ІМ ГВС» містить встановлення початкових значень параметрів конкретного ОМ (за допомогою службової функції «Запит»), а також прогін моделі.

Фрейм «ГВС» містить слот «Об'єкт», значенням якого є показчик фрейму конкретної ГВС, яка аналізується. В результаті виконання функції «Запит (ГВС; об'єкт)» формується показчик фрейму «ГВС5», значення слотів якого є даними для прогону моделі.

Спрощену структуру звернень показано на рис. 6.3, а штриховою лінією (відсутні можливі звернення фреймів для одержання випадкової інформації).

**Взаємодія ЕС і числових алгоритмів.** Комбінований гібридний підхід є ефективним при спільному використанні ЕС, з одного боку, та методів математичного програмування, диференціального й інтегрального числень, розв'язання алгебраїчних, диференціальних та інтегральних рівнянь і їх систем, статистичної обробки даних — з іншого. При цьому ЕС може виконувати підготовчу і/або заключну функцію, а також брати участь у реалізації окремих кроків алгоритмів.

Враховуючи особливе значення оптимізаційних задач у плануванні та управлінні, розглянемо їх докладніше. Зокрема, підготовчими функціями ЕС при розв'язанні задач оптимізації можуть бути: розпізнавання класу задач за видом цільової функції та обмежень; вибір методу розв'язання; зведення постановки задачі до стандартного вигляду, необхідного для застосування конкретної програмної реалізації методу; компонування алгоритму з готових бібліотечних модулів.

На етапі оптимізації виконання процедур пошуку ЕС може використовуватись для: вихідної локалізації області пошуку; вибору початкової точки; вибору кроку; вибору напрямку пошуку; прогнозування положення екстремуму; пошуку множин Парето-оптимальних рішень тощо. При цьому застосування ЕС може бути виправдане в тих ситуаціях, коли ефективними є евристичні прийоми розв'язання важкоформалізованих ситуацій, що виникають у процесі пошуку ЕС, або коли область пошуку дуже значна, а розрахунок цільової функції потребує багато часу, або коли функція відклику має складну форму, розриви, стрибки, численні локальні екстремуми тощо.

Прикладами правил на підготовчому етапі можуть бути такі:

ЯКЩО	матриця коефіцієнтів обмежень є розрідженою,
ТО	перетворити її подання до розрідженого формату.
ЯКЩО	задачею є лінійне програмування
І	умови задано в розрідженому форматі,
ТО	застосувати модифікацію симплекс-методу до розрідженого формату.

ЯКЩО	область пошуку ЕС є одновимірною
I	цільова функція унімодална,
ТО	використати метод Фібоначчі.

Очевидно, для встановлення істинності відправлень таких правил, як і для реалізації приписок у висновках, повинні виконуватися відповідні процедури узгодження.

Початкова локалізація області пошуку ЕС і вибір початкового наближення здійснюються на основі апріорної інформації про положення оптимуму, вплив на нього деяких факторів тощо. Наприклад, може бути відомо, що цільова функція монотонно збільшується по  $X_2$  при будь-яких фіксованих інших параметрах (рис. 6.5). У цьому разі можна застосувати таке правило:

ЯКЩО	цільова функція збільшується по $X_2$ ,
ТО	її екстремум розташовується на межі області пошуку, орієнтованої в бік осі $X_2$ .

Цій межі відповідає лінія  $AB$  на рис. 6.5, *a*. При програмній реалізації в антецеденті правила замість символічного рядка «екстремум розташовується...» записується виклик відповідної процедури локалізації області пошуку.

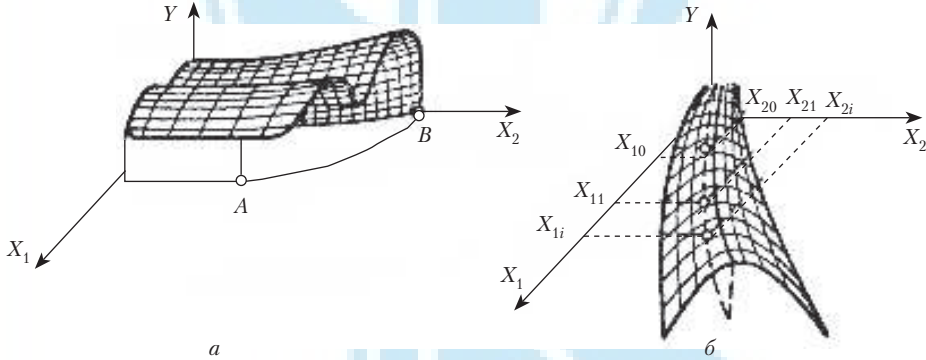


Рис. 6.5. Поверхні відклику з локалізацією екстремумів на межі області пошуку ЕС (*a*) та ілюстрація набору початкового наближення (*б*)

З накопиченої інформації може, наприклад, впливати, що зі збільшенням  $X_1$  екстремум цільової функції монотонно зміщується в напрямку збільшення  $X_2$  (рис. 6.5, *б*). В цьому разі може бути корисним правило:

ЯКЩО	зі зростанням $X_1$ екстремум цільової функції зміщується уздовж осі $X_2$
I	задано $X_1 = X_{12}$ ,
I	відстань уздовж осі $X_1$ до найбільшого відомого екстремуму є порівнянною з розмірами області пошуку,
ТО	для початкового наближення $X_2$ вибрати значення, зміщене уздовж осі $X_2$ пропорційно зміщенню уздовж осі $X_1$ .

Згідно з цим правилом початковим наближенням по  $X_2$  буде:

$$X_{21} = X_{20} + \left( \frac{X_{21} - X_{20}}{X_{11} - X_{10}} \right) (X_{12} - X_{10}),$$

де  $(X_{10}, X_{20})$  і  $(X_{11}, X_{21})$  – координати двох відомих точок екстремуму в уже дослідженій області.

#### 6.4. Інтелектуальні системи моделювання

**Загальні принципи побудови.** За традиційною схемою організації моделювання, коли інформаційними потоками обмінюються дослідник та ІМ, зворотний зв'язок за результатами моделювання здійснює зовнішня відносно системи імітаційного моделювання ланка — людина із залученням допоміжних засобів програмного забезпечення [66]. Таким чином, і дослідник у системах ІМ, і оператор у системах з автоматичного керування ОМ відіграють своєрідну роль експерта з функціями інтерпретатора поточної ситуації з прийняттям відповідних рішень щодо подальших керуючих дій, коректність яких визначається кваліфікацією дослідника (оператора) і зволіканням у прийнятті рішень. Тому дуже важливу гілку напрямку з розширення апарату СП відіграють його модифікації, пов'язані з удосконаленням власне теорії автоматизованого моделювання [25], а також інтелектуалізації СП-моделювання [43; 88; 89; 102; 109].

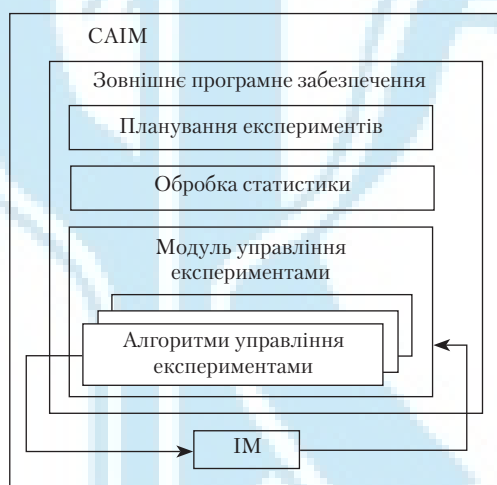


Рис. 6.6. Уявлення про структуру автоматизованого управління експериментами

Спрямовані серії експериментів відповідно до заданої мети і з урахуванням обмежень зміни параметрів організують модулі, що спеціально вводяться до складу зовнішнього програмного забезпечення (рис. 6.6). У загальному випадку ці модулі повинні задавати набори початкових даних, ініціювати прогони моделі, обробляти результати та приймати рішення про подальший перебіг експериментів відповідно до реалізованого алгоритмом управління моделюванням. Такий алгоритм, спрямовуючи експерименти, в області допустимих значень параметрів здійснює пошук такого їх поєднання, що забезпечує оптимум заданого показника якості, тобто по суті розв'язує задачу оптимізації  $f(\bar{x}) \rightarrow \max$ , де  $f$  — цільова функція, подана алгоритмічно ІМ;  $\bar{x} = (x_1, x_2, \dots, x_n)$  — вектор параметрів ОМ;  $X$  — множина допустимих значень параметрів.

**Твердження 6.1.** Систему автоматизованого ІМ (CAIM) утворює сукупність алгоритмічних і програмних засобів, які забезпечують процес автоматизованого моделювання з розв'язанням задачі оптимізації  $f(\bar{x}) \rightarrow \max(\min) | \bar{x} \in X$ , де  $f$  — цільова функція, подана алгоритмічно;  $\bar{x} = (x_1, x_2, \dots, x_n)$  — вектор параметрів ОМ;  $X$  — множина допустимих значень параметрів [125].

Оскільки користувач не вводить кожний набір початкових даних  $W_{вх}$  для чергового прогону ІМ і тільки зазначає мету (критерій) та область варіювання параметрів  $D_{w_{2E}}$ , тоді як пошук прийнятних розв'язків задач моделювання виконується за допомогою САІМ автоматично, є всі підстави застосувати до неї термін *інтелектуальної системи моделювання*. Наприклад, функції САІМ в контурі управління ГКІС полягають в аналізі альтернативних варіантів поведінки останньої після прийняття того чи іншого можливого рішення щодо диспетчеризування, оперативного планування тощо.

Ефективність реалізації цих функцій зумовлена заміною жорсткої логіки управління, що передбачає використання в окремих ситуаціях заданих і фіксованих евристик, гнучким і динамічним механізмом, який забезпечує прийняття рішень не тільки на основі докладного аналізу поточного стану ОМ, а й з урахуванням перспектив його розвитку. При цьому поєднуються висока вірогідність результатів і можливість автоматичного прийняття рішень у режимі реального часу.

Проблема автоматизації управління експериментами може бути подана логічною структурою її поетапного розв'язання (рис. 6.7).

На *першому* етапі автоматизації управління експериментами розв'язуються дві задачі: виконується структурно-алгоритмічна побудова саме модуля управління; формуються змістові основи та формальні вимоги до організації інформаційного обміну з ІМ.

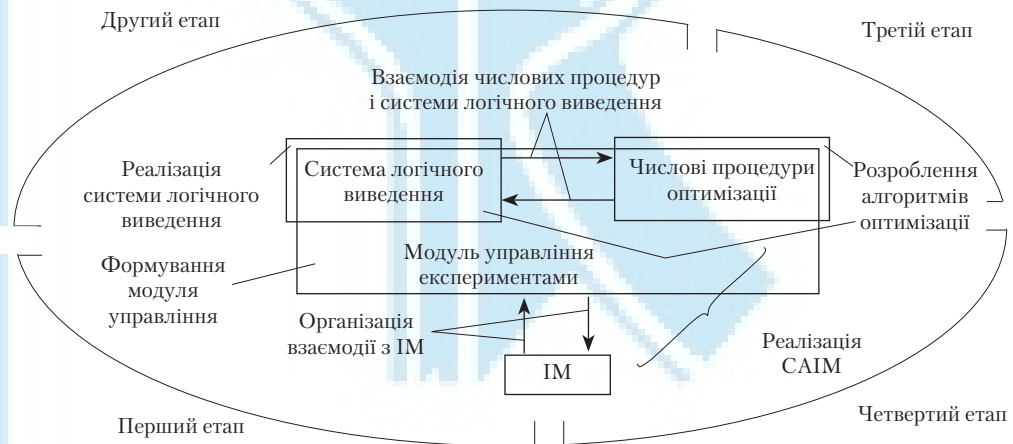


Рис 6.7. Зміст етапів автоматизованого моделювання об'єктів

Задача структурно-алгоритмічної побудови модуля управління експериментами розв'язується в такій послідовності: визначаються склад і структура модуля (розробляються умови взаємодії його компонентів і місце в загальній структурі САІМ).

При виконанні серії прогонів ІМ відбувається цілеспрямоване варіювання значень параметрів, яке може впливати на значення цільової функції не тільки через пряму дію на показники функціонування ОМ, а й побічно через інші спряжені елементи об'єкта. Як наслідок змінюються вартісні та інші його показники. Водночас на значення деяких параметрів можна накласти обмеження, зокрема частину їх зафіксувати (задати директивно). Це має відобразитися при формуванні наборів початкових даних.

Розглядаючи ІМ як засіб цілеспрямованого перетворення інформації відповідно до певної системи приписок, є сенс говорити про *алгоритм* ІМ (АлгІМ). Тоді формальну інтерпретацію розглянутих вимог можна записати так:

$$\left\{ \begin{array}{l} W_{\text{вх}} \in Q \\ W_{\text{вх}} \in C \end{array} \right\}, \quad (6.20)$$

де  $W_{\text{вх}}$  — вхідне слово АлгІМ;  $Q$  — множина значень допустимих наборів параметрів ОМ;  $C$  — область визначення АлгІМ.

Вхідне слово  $W_{\text{вх}}$  задає набір початкових даних для конкретного експерименту, тобто  $W_{\text{вх}} = (\xi_1, \xi_2, \dots, \xi_n)$ , де кожна величина  $\xi_i$  відповідає деякому значенню певного параметра ОМ. Область  $C$  визначається програмною реалізацією АлгІМ, а утворюється вона множиною  $D_{\text{вх}}$  наборів елементів вхідного алфавіту АлгІМ. Всі величини  $\xi_i$ , що мають практично здійснювану реалізацію в ОМ, утворюють множину  $U$ .

Таким чином, вирази (6.20) відповідно визначають умови узгодження  $W_{\text{вх}}$  з ІМ і з АлгІМ. Задача полягає в розробленні апарату формального аналізу несуперечності змін семантично взаємозв'язаних параметрів (для забезпечення варіювання, що не порушує вимоги (6.20) і не призводить до зміни фіксованих параметрів).

На *другому* етапі автоматизації управління експериментами розв'язуються такі задачі: розроблення принципів взаємодії системи логічного виведення і числових оптимізаційних процедур; вибір (розроблення) апарату реалізації логічного виведення в комбінованих алгоритмах управління експериментами; розроблення на основі запропонованого апарату ефективних алгоритмів виведення з урахуванням специфіки взаємодії з числовими процедурами оптимізації.

На *третьому* етапі автоматизації управління експериментами розв'язується задача розроблення оптимізаційних процедур, які використовуються в плануванні екстремальних експериментів. При цьому питання розроблення алгоритмічного забезпечення повинно розглядатися для специфічних аспектів моделювання об'єкта, а саме:

- автоматизації управління експериментами в умовах лінгвістичної невизначеності параметрів (наприклад, «ступінь схожості/відмінності...» не має натурального числового вимірювання);
- організації екстремальних експериментів у задачах однопараметричної дискретної оптимізації при суттєво нерівномірному розташуванні точок усередині інтервалу (приклад — унімодальна функція відклику);
- скорочення часу пошуку при значних тривалостях прогону ІМ (що є важливим при багатоітераційних алгоритмах пошуку, повному перебиранні варіантів тощо).

**Четвертий** етап автоматизації управління експериментами пов'язаний з особливостями реалізації САІМ у конкретній сфері практичного використання.

**Реалізація інтелектуальної компоненти.** Функціональну складову САІМ, що організовує певну послідовну серію експериментів з ІМ відповідно до конкретної стратегії, називатимемо *пошуковим модулем* (ПМ). Остання реалізує деякий метод пошуку, для якого ІМ виступає як засіб розрахунку значень заданих показників. Умовно структуру функціонування контуру ПМ — ІМ показано на рис. 6.8, а. ПМ і ІМ, що утворюють замкнений контур, називатимемо *автоматизованою ІМ* (АІМ).

Умовно зображена на рис. 6.8, б АІМ у вигляді пари ПМ — ІМ служить основою для формування складних ієрархічних структур модуля управління імітаційними

експериментами. Використовуваний при цьому програмно реалізований евристичний алгоритм є по суті ПМ, але в інших задачах він сам може розглядатись як ІМ наближеного міркування.

Таким чином, функціональний розподіл ПМ – ІМ визначається проблемною галуззю. Різні варіанти вкладеності, що виникають при цьому, утворюються такими основними типами охоплення моделей:

(...(ІМ (ІМ (...)))...). У складі ІМ виділено підмодель. Приклад: модель обробно-го центру в складі ІМ ГВС.

(...(ІМ(ПМ (...)))...). ІМ містить підмодель, «охоплену» ПМ. При цьому ПМ формує набори початкових даних тільки для підмоделі, прогін якої здійснюється з використанням проміжних даних поточного експерименту з усією ІМ і не потребує зміни всього набору початкових даних. Приклад: при обробці кожного з варіантів компоновки гнучкої виробничої дільниці шляхом прогону всієї її моделі організовується «внутрішня» серія експериментів з підмоделлю роботизованої комірки (що входить до складу дільниці), спрямована на пошуки оптимальних режимів її роботи.

(...(ПМ (ІМ (...)))...). Відповідає розглянутому вище уявленню про АІМ. На відміну від п. 2 в цьому разі ПМ керує всією ІМ.

(...(ПМ (ПМ (...)))...). Декілька ПМ керують експериментами з однією ІМ.

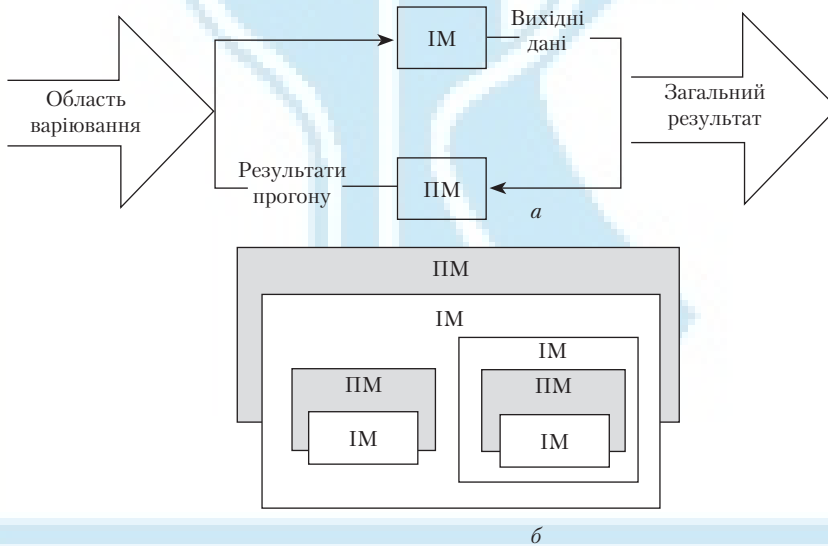


Рис. 6.8. Замкнений контур ПМ – ІМ (а) та узагальнена ієрархічна структура модуля управління експериментами (б)

Функціонування ПМ у цьому разі може бути: підпорядкованим (див. підрозд. 4.1), коли ПМ нижчого рівня організує експерименти з ІМ у межах набору початкових даних, то визначаються ПМ верхнього рівня (приклад: ПМ верхнього рівня аналізує вплив пріоритетних правил диспетчеризування, а ПМ нижчого рівня для кожного з них організує серію експериментів за різними варіантами виробничого завдання); рівноправним (паралельним), коли ПМ самостійно задають значення різних параметрів і обробляють результати чергових прогонів з гарантією незалежності змінюваних параметрів.

Узагальнену ієрархічну структуру модуля управління експериментами можна зобразити умовно (рис. 6.8, б), а визначається вона системою моделювання та конкретними цілями експериментів.

При формуванні багаторівневих систем управління імітаційними експериментами необхідно враховувати такі вимоги: при підпорядкованому функціонуванні параметри, що задаються ПМ верхнього рівня, для ПМ нижчого рівня є фіксованими; повинна передбачатись можливість реалізації управління ІМ відповідно до протилежних локальних цілей (приклад – мінімаксна задача: при заданому режимі роботи транспортної системи ПМ нижнього рівня, формуючи різні варіанти виробничого завдання, відшукує максимальні значення простоїв, тоді як ПМ верхнього рівня, варіюючи режими роботи транспортної системи, мінімізує можливі найгірші простой).

З урахуванням викладеного можна запропонувати такий склад засобів автоматизованого управління експериментами, умовну структуру їх взаємодії та місце в узагальненій САІМ (рис. 6.9).

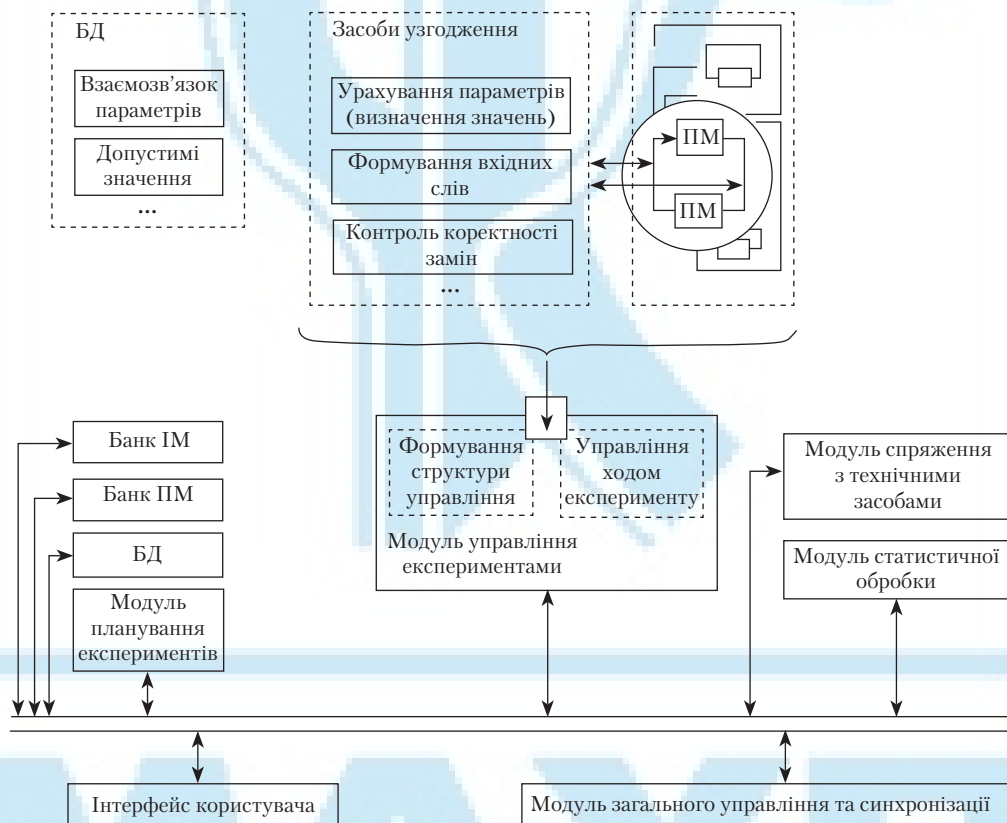


Рис. 6.9. Склад засобів автоматизованого управління експериментами з умовною структурою їх взаємодії та місцем в узагальненій САІМ

**Формальні основи організації експериментів.** Автоматизованій ІМ відповідає деякий узагальнений алгоритм – алгоритм автоматизованого імітаційного моделювання (АлгАІМ), що є композицією АлгІМ та алгоритму ПМ (АлгПМ).

На вхід АлгАІМ надходить слово  $W_{\text{вх}}$ , яке відповідає множині  $D_{W_{\text{вх}}}$  що є областю варіювання параметрів (на практиці може задаватись непрямим способом – за допомогою обмежень). Вихідне слово  $W_{\text{вих}}$  визначає основні результати експерименту.

**Твердження 6.2.** *Необхідними умовами функціонування АІМ є: існування непорожньої множини  $D_0 = D_{\text{вх}} \cap D_{\text{вих}} \neq \emptyset$  збіжність алгоритму автоматизованого ІМ (Алг ІМ) в  $D_0$ ; несуперечність варіювання вхідних параметрів для ІМ (тут  $D_{\text{вх}}, D_{\text{вих}}$  – відповідно області варіювання вхідних параметрів  $D_{\text{вх}}$  і результатів  $D_{\text{вих}}$  експерименту).*

Для урахування взаємозв'язку параметрів вводяться спеціальні формалізми (характеристичні множини та графи параметрів ОМ) [140; 148] і на цій основі визначаються вимоги до основних компонентів організації експерименту, пов'язані з варіюванням значень параметрів і задаванням вхідного слова  $D_{\text{вх}}$ . Кожному з елементів  $\xi_i \in W_{\text{вх}}$  відповідає впорядкована пара  $(x^i, a_i^l)$ , де  $x^i$  – ім'я (ідентифікатор) параметра;  $a_i^l$  – його значення. Множину  $X_{\text{ОМ}}$  всіх  $x^i$  назовемо *характеристичною множиною* ОМ.

Параметри  $x^i$  можуть бути як початковими для конкретних досліджень (наприклад, кількість деталей у партії), так і похідними від інших параметрів (наприклад, варіант виконання автоматизованої транспортно-складської системи (АТСС), що визначається значеннями параметрів «кількість транспортних модулів», «координати позицій завантаження – розвантаження» тощо). Кожному довільному параметру  $x^k \in X_{\text{ОМ}}$  відповідає деяка множина параметрів  $X_{\text{ОМ}}^k \in X_{\text{ОМ}}$ , якими він повністю визначається; у свою чергу, множина  $X_{\text{ОМ}}^k$  може містити як початкові, так і похідні параметри.

За допомогою множин  $X_{\text{ОМ}}^k$  введемо на множині  $X_{\text{ОМ}}$  транзитивне бінарне відношення залежності  $R = \{(x^i, x^j) | x^j \in X_{\text{ОМ}}^i\}$ . Подальші міркування є справедливими для довільно сформованих множин  $X_{\text{ОМ}}^k$ : при  $(x^i \in X_{\text{ОМ}}^j) \cap (x^j \in X_{\text{ОМ}}^k)$  можна задавати як  $(x^i \in X_{\text{ОМ}}^k)$ , так і  $(x^i \notin X_{\text{ОМ}}^k)$ .

Структуру змістового взаємозв'язку параметрів подамо за допомогою оргграфа  $G_{\text{ОМ}}(X_{\text{ОМ}}, L)$ , де  $X_{\text{ОМ}}$  – множина вершин (вона ж характеризує множину ОМ);  $L$  – множина дуг, причому  $L = \{(x^j, x^i) | x^i R x^j\}$ .

Оргграф  $G_{\text{ОМ}}$  називається *графом параметрів* ОМ. Зі скінченності  $X_{\text{ОМ}}$  випливає і скінченність  $G_{\text{ОМ}}$ . Приклад фрагмента графа параметрів при реалізації автоматизованої транспортно-складської системи показано на рис. 6.10.

Розглянемо вхідне слово  $W_{\text{вх}} = (\gamma_1, \dots, \gamma_z)$ . З (6.20) випливає, що кожному з елементів  $\xi_i \in W_{\text{вх}}$  можна поставити у відповідність елемент  $x_{W_{\text{вх}}}^i \in X_{\text{ОМ}}$ , причому  $(\gamma_i = \xi) \rightarrow (x_{W_{\text{вх}}}^i = x^j)$ . Вхідному слову  $W_{\text{вх}}$  відповідає множина  $X_{W_{\text{вх}}}^0 = \{x_{W_{\text{вх}}}^i | \xi_i \in W_{\text{вх}}\}$ .

Сформулюємо множину

$$X_{W_{\text{вх}}}^1 = X_{W_{\text{вх}}}^0 \cup \left\{ x^k | \left( X_{\text{ОМ}}^k \subseteq X_{W_{\text{вх}}}^0 \right) \cup \left( \exists x^l \in X_{W_{\text{вх}}}^0 : x^k \in X_{\text{ОМ}}^l \right) \right\}, \quad (6.21)$$

потім множину

$$X_{W_{\text{вх}}}^2 = X_{W_{\text{вх}}}^1 \cup \left\{ x^k | \left( X_{\text{ОМ}}^k \subseteq X_{W_{\text{вх}}}^1 \right) \cup \left( \exists x^l \in X_{W_{\text{вх}}}^1 : x^k \in X_{\text{ОМ}}^l \right) \right\}, \quad (6.22)$$

і так далі.



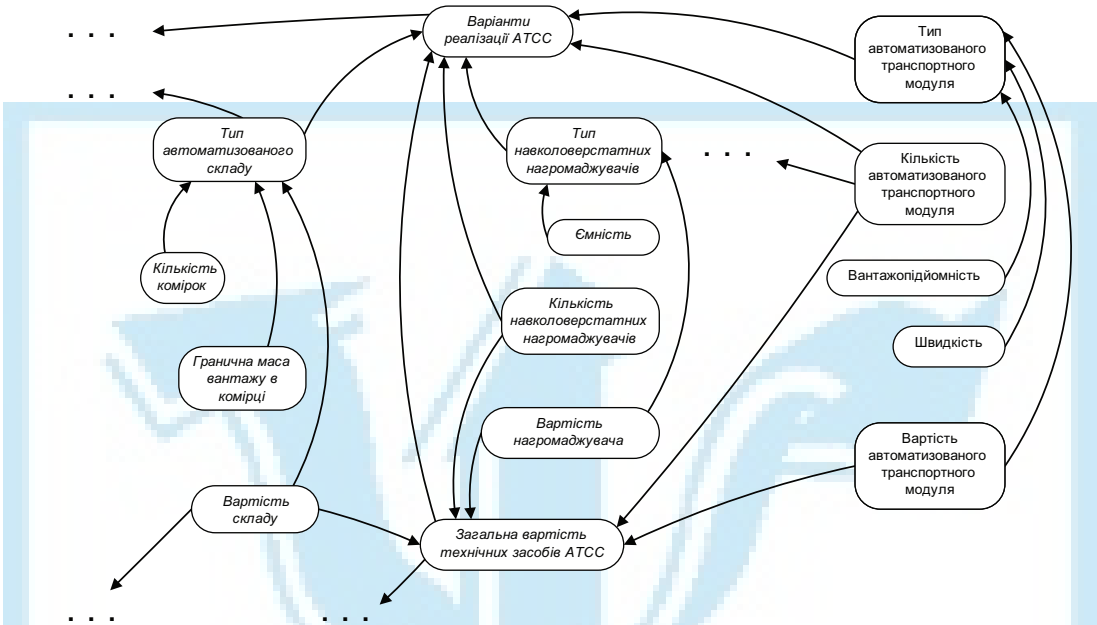


Рис. 6.10. Фрагмент графа параметрів при реалізації АТСС

Послідовність множин  $X_S^i$  збігається (що впливає зі скінченності  $X_{OM}$ ) до деякої множини  $X_S$ , яку назовемо *характеристичною множиною вхідного слова*  $W_{BX}$ . Остання об'єднує параметри, безпосередньо подані у вхідному слові.

Визначимо граф параметрів вхідного слова  $G_{W_{BX}} = (X_{W_{BX}}, L_{W_{BX}})$ , де  $X_{W_{BX}}$  – множина вершин,  $L_{W_{BX}}$  – множина дуг, причому

$$L_{W_{BX}} = \left\{ (x_{W_{BX}}^i, x_{W_{BX}}^j) \mid (x_{W_{BX}}^i, x_{W_{BX}}^j \in X_{W_{BX}}) \cap (x_{W_{BX}}^i R x_{W_{BX}}^j) \right\}.$$

Аналогічно на основі множини  $D_{BX}$  визначаються вхідна характеристична множина моделі  $X_{IM_{BX}}$  і граф вхідних параметрів моделі  $G_{IM_{BX}}$ , а на основі вихідного алфавіту АлгІМ ( $D_{ВИХ}$ ) – вихідна характеристична множина моделі  $Y_{ВИХ} = \{y_j\}$ , яка об'єднує параметри, що визначаються з урахуванням вихідних слів\*.

Дійсними є такі співвідношення:  $X_{IM_{BX}} \subset X_{OM}$ ;  $Y_{ВИХ} \subset X_{OM}$ .

До основних вимог автоматизованої *заміни параметрів* ОМ (зокрема, при їх варіюванні) належать виконання умов узгодження (6.20) і незмінність значень його фіксованих параметрів.

Кожному з параметрів  $x^i$  відповідає скінченна множина його допустимих значень:  $A_i = \{a_i^1, a_i^2, \dots, a_i^r\}$ . Конкретне значення  $a_i^*$  визначається комбінацією значень  $a_j^*$  всіх параметрів  $x^j \in X_{OM}$ .

У загальному випадку кожному  $a_i^*$  можна поставити у відповідність деяку множину  $C_i^*$  таких комбінацій.

Розглянемо властивості графа  $G_{W_{BX}}$ . Кожна його вершина відповідає деякому вхідному параметру. Приписуючи кожній вершині  $x_{W_{BX}}^i$  її конкретне значення з  $A_i$ , дістанемо позначений граф  $\tilde{G}_{W_{BX}}$  (див. підрозд. 2.3), приклад якого ілюструє рис. 6.11, а.

\* Наведені міркування та висновки справедливі і для незорієнтованих графів, а конкретний механізм реалізації одержаних результатів у САІМ може будуватися як на поданому варіанті графів параметрів, так і на інших модифікаціях СС, фреймових структурах тощо.

**Означення 6.2.** Вершина  $x_{w_{\text{BX}}}^i$  графа  $\tilde{G}_{W_{\text{BX}}}$ , що має значення  $a_i^1$ , вважається не-суперечною, якщо значення  $a_i^m$  усіх вершин  $x_s^j \in X_{\text{OM}}^i$  утворюють комбінацію, яка належить  $\tilde{C}_i^l$ .

**Означення 6.3.** Граф  $\tilde{G}_{W_{\text{BX}}}$  вхідного слова  $W_{\text{BX}}$  називається не-суперечним, якщо всі його вершини є не-суперечними.

Несуперечність графа  $\tilde{G}_{W_{\text{BX}}}$  є необхідною, але не достатньою умовою узгодження вихідних даних ОМ та ІМ. Вона забезпечує допустимість значень кожного вхідного параметра, але не гарантує виведення їх недопустимих комбінацій.

У характеристичній множині  $X_{W_{\text{BX}}}$  кожного недопустимого вхідного слова  $W_{\text{BX}}^i$  відокремимо підмножину  $P_i$ , що складається з параметрів, значення яких утворюють заборонені сполучення. Сформуємо множину  $P' = \cup P_i$ , увівши для неї в графі  $\tilde{G}_{W_{\text{BX}}}$  допоміжну вершину  $\tilde{x}'_{W_{\text{BX}}}$  і дуги  $\tilde{x}'_{W_{\text{BX}}}$ ,  $\tilde{x}'_{W_{\text{BX}}}$  для всіх  $x_{W_{\text{BX}}}^i \in P'$ . Тоді фіктивний параметр  $x'_{W_{\text{BX}}}$  матиме множину визначальних параметрів  $X'_{\text{OM}} = P'$ .

Присвоїмо вершині  $\tilde{x}'_{W_{\text{BX}}}$  єдине значення  $a'$  і припустимо, що цьому значенню відповідає множина  $C'$  усіх допустимих сполучень значень параметрів множини  $X'_{\text{OM}}$  (на практиці перевага віддається не переліку всіх значень і їх комбінацій, а поданню допустимих сполучень інтервалів варіювання, заданих граничними точками). В результаті дістанемо розгорнутий граф вхідного слова (рис. 6.11, б)  $\tilde{M}_{W_{\text{BX}}}(X_{W_{\text{BX}}}, L_{W_{\text{BX}}})$ , де  $\tilde{X}_{W_{\text{BX}}} = X_{W_{\text{BX}}} \cup \tilde{x}'_{W_{\text{BX}}}$ ,  $\tilde{L}_{W_{\text{BX}}} = L_{W_{\text{BX}}} \cup \{(x_{W_{\text{BX}}}^j, x'_{W_{\text{BX}}})\}$ .

**Твердження 6.3.** Необхідною і достатньою умовою узгодження даних ОМ та ІМ є не-суперечність графа  $\tilde{M}_{W_{\text{BX}}}$ .

Справедливість твердження 6.3 впливає із способу побудови графа. Вершини графа  $\tilde{M}_{W_{\text{BX}}}$ , що відповідають фіксованим параметрам, називаються *фіксованими*, а параметри та вершини, які не є фіксованими, дістали назву *вільних*.

**Означення 6.4.** Коректною заміною (підставлянням) значення параметра є таке, яке не призводить до суперечності жодної з вершин графа  $\tilde{M}_{W_{\text{BX}}}$  і не потребує змін значень фіксованих параметрів.

Згідно з цим означенням і врахуванням твердження 6.3 поняття *коректності* охоплює основні вимоги (6.20) до змін значень параметрів.

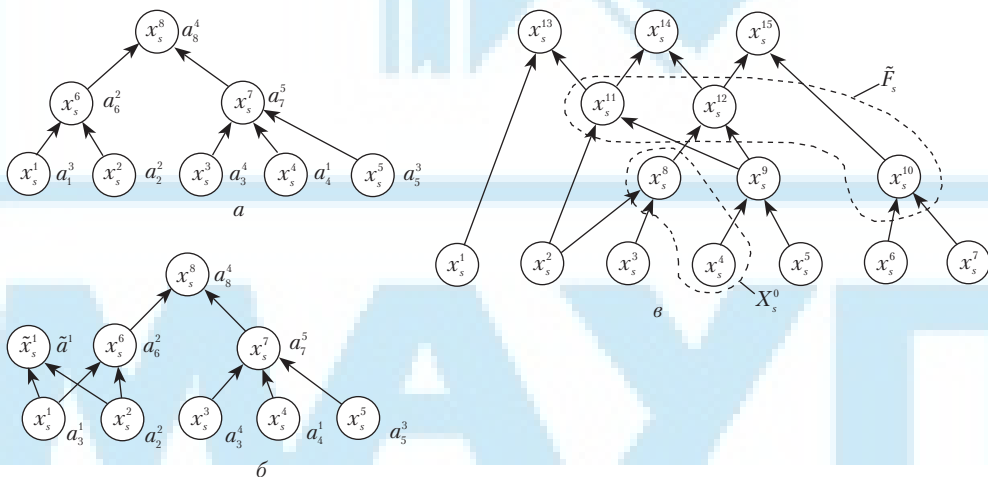


Рис. 6.11. Позначений (а), розгорнутий (б) та розгорнутий з виділенням підграфа (в) графі вхідного слова

Тепер розглянемо умови *коректності замін* параметрів ОМ. Нехай  $\tilde{M}_{W_{\text{BX}}}^0(X_{W_{\text{BX}}}, L_{W_{\text{BX}}})$  – початковий несуперечливий розгорнутий граф вхідного слова;  $\tilde{X}_{W_{\text{BX}}}^0 \subseteq \tilde{X}_{W_{\text{BX}}}$  – множина вершин, значення яких необхідно змінити;  $\tilde{F}_X$  – множина фіксованих вершин, до складу якої входить і вершина  $\tilde{x}_{W_{\text{BX}}}^i$ , що відповідає фіктивному параметру.

Сформуємо множину

$$\tilde{X}_{W_{\text{BX}}}^1 = \tilde{X}_{W_{\text{BX}}}^0 \cup \left\{ x_{W_{\text{BX}}}^i \mid \left( x_{W_{\text{BX}}}^i \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^0 \right) \left( \exists x_{W_{\text{BX}}}^j \in \tilde{X}_{W_{\text{BX}}}^0 \right) \times \right. \\ \left. \times \left[ \left( \left( x_{W_{\text{BX}}}^i, x_{W_{\text{BX}}}^j \right) \in \tilde{L}_{W_{\text{BX}}} \right) \vee \left( \left( x_{W_{\text{BX}}}^j, x_{W_{\text{BX}}}^i \right) \in \tilde{L}_{W_{\text{BX}}} \right) \right] \right\},$$

потім множину

$$\tilde{X}_{W_{\text{BX}}}^2 = \tilde{X}_{W_{\text{BX}}}^1 \cup \left\{ x_{W_{\text{BX}}}^i \mid \left( x_{W_{\text{BX}}}^i \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^1 \right) \left( \exists x_{W_{\text{BX}}}^j \in \tilde{X}_{W_{\text{BX}}}^1 \right) \times \right. \\ \left. \times \left[ \left( \left( x_{W_{\text{BX}}}^i, x_{W_{\text{BX}}}^j \right) \in \tilde{L}_{W_{\text{BX}}} \right) \vee \left( \left( x_{W_{\text{BX}}}^j, x_{W_{\text{BX}}}^i \right) \in \tilde{L}_{W_{\text{BX}}} \right) \right] \right\}$$

і так далі (кожна множина  $\tilde{X}_{W_{\text{BX}}}^{i+1}$  містить множину  $\tilde{X}_{W_{\text{BX}}}^i$ , а також усі вершини, що є суміжними з вільними вершинами множини  $\tilde{X}_{W_{\text{BX}}}^i$ ).

З урахуванням скінченності множини  $\tilde{X}_{W_{\text{BX}}}^z$  послідовність множин  $\tilde{X}_{W_{\text{BX}}}^i$  зводиться до деякої множини  $\tilde{X}_{W_{\text{BX}}}^z$ , на якій позначимо підграф  $\tilde{M}_{W_{\text{BX}}}^z \subseteq \tilde{M}_{W_{\text{BX}}}$ , причому  $\tilde{M}_{W_{\text{BX}}}^z = (\tilde{X}_{W_{\text{BX}}}^z, \tilde{L}_{W_{\text{BX}}}^z)$ , де (див. рис. 6.11, в)

$$\tilde{L}_{W_{\text{BX}}}^z = \left\{ \left( x_{W_{\text{BX}}}^i, x_{W_{\text{BX}}}^j \right) \mid \left( \left( x_{W_{\text{BX}}}^i, x_{W_{\text{BX}}}^j \right) \in \tilde{L}_{W_{\text{BX}}} \wedge \left( x_{W_{\text{BX}}}^i, x_{W_{\text{BX}}}^j \right) \in \tilde{X}_{W_{\text{BX}}}^z \right) \right\}.$$

**Твердження 6.4.** *Необхідною та достатньою умовою коректності зміни параметрів ОМ є коректність у межах підграфа  $\tilde{M}_{W_{\text{BX}}}^z$ .*

Необхідність очевидна, тому розглянемо виконання достатньої умови. Нехай  $x_{W_{\text{BX}}}^i \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^z$  і  $(\exists x_{W_{\text{BX}}}^j) \left[ \left( \left( x_{W_{\text{BX}}}^i, x_{W_{\text{BX}}}^j \right) \in \tilde{L}_{W_{\text{BX}}} \right) \vee \left( \left( x_{W_{\text{BX}}}^j, x_{W_{\text{BX}}}^i \right) \in \tilde{L}_{W_{\text{BX}}} \right) \right]$ . Доведемо, що тоді  $(x_{W_{\text{BX}}}^j \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^z) \vee (x_{W_{\text{BX}}}^j \in \tilde{F}_x)$ . Для цього припустимо супротивне, тобто  $x_{W_{\text{BX}}}^j \in \tilde{X}_{W_{\text{BX}}}^z \setminus \tilde{F}_x$ . Однак тоді за способом формування підграфа  $\tilde{M}_{W_{\text{BX}}}^z$  вершина  $x_{W_{\text{BX}}}^i$  повинна була б потрапити у множину  $\tilde{X}_{W_{\text{BX}}}^z$ , що суперечить припущенню. Таким чином:

$$\left[ \left( x_{W_{\text{BX}}}^i \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^z \right) \left( \exists x_{W_{\text{BX}}}^j \in \tilde{X}_{W_{\text{BX}}}^z \right) \left( \left( \left( x_{W_{\text{BX}}}^i, x_{W_{\text{BX}}}^j \right) \in \tilde{L}_{W_{\text{BX}}} \right) \vee \left( \left( x_{W_{\text{BX}}}^j, x_{W_{\text{BX}}}^i \right) \in \tilde{L}_{W_{\text{BX}}} \right) \right) \right] \rightarrow \\ \rightarrow \left( \left( x_{W_{\text{BX}}}^j \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^z \right) \vee \left( x_{W_{\text{BX}}}^j \in \tilde{F}_x \right) \right). \quad (6.23)$$

З (6.23) випливає, що вершини  $x_{W_{\text{BX}}}^i \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^z$  можуть мати суміжними в підграфі  $\tilde{M}_{W_{\text{BX}}}^z$  тільки фіксовані вершини. Отже, зміни в  $\tilde{M}_{W_{\text{BX}}}^z$  не впливають на позначення вершин  $x_{W_{\text{BX}}}^i \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^z$  і відповідні параметри можуть зберігати свої позначення. Враховуючи те, що початковий граф несуперечливий, позначення вершин  $x_{W_{\text{BX}}}^i \in \tilde{X}_{W_{\text{BX}}} \setminus \tilde{X}_{W_{\text{BX}}}^z$  не порушує умови коректності замін. Оскільки ж ця умова виконується і в підграфі  $\tilde{M}_{W_{\text{BX}}}^z$ , можна зробити висновок, що ця умова виконується в усьому графі  $\tilde{M}_{W_{\text{BX}}}$ .

Таким чином, на основі твердження 6.4 практичне вирішення питання про коректність заміни параметрів ОМ досягається виконанням формальних алгоритмів пошуку допустимого позначення вільних вершин підграфа  $\tilde{M}_{W_{\text{BX}}}^z$ . Ураховуючи те, що стосовно значної кількості задач для таких складних об'єктів, якими є ГКІС, граф параметрів є деревом, цей пошук значно полегшується.

Тепер розглянемо, як позначаються на взаємозв'язках параметрів з урахуванням виконання необхідних умов автоматичного функціонування ІМ **механізми задавання вхідного слова**. Параметри, значення яких у явному вигляді фігурують при виконанні АлгІМ, утворюють множину  $B_{ІМ}$ . Значення параметрів, що належать множині  $X_{ІМ}^z = B_{ІМ} \cap X_{ІМ_{вх}}$ , повинні встановлюватися з урахуванням вхідного слова перед прогоном ІМ. Найбільш натуральним і зручним є їх безпосереднє подання у вхідному слові. В цьому разі  $X_{ІМ}^z = X_{W_{вх}}^0$ , і встановлення значень параметрів відбувається шляхом прямого їх присвоювання, що не потребує перетворень.

Якщо ж  $X_{ІМ}^z \neq X_{W_{вх}}^0$ , то потрібно мати засоби (їх умовно можна зарахувати до АлгІМ) для відтворення значень параметрів множини  $X_{ІМ}^z \setminus X_{W_{вх}}^0$ . Наприклад, використовується ІМ, яка оперує партіями деталей як єдиним цілим з урахуванням сумарної тривалості їх обробки. Параметри, що варіюються, можуть бути, наприклад, тривалостями обробки однієї деталі і/або кількостями деталей у партії. Тоді ці значення як початкові дані надходять на вхід АлгІМ, і вже на їх основі розраховується сумарна тривалість обробки деталей.

**Твердження 6.5.** Якщо АлгІМ може бути застосований до вхідного слова  $W_{вх}$ , то  $X_{W_{вх}} \subseteq X_{ІМ_{вх}}$  і  $G_{W_{вх}} \subseteq G_{ОМ}$  (це твердження випливає з умови (6.20), а також способу побудови характеристичних множин і графів параметрів).

Що стосується **врахування параметрів** при автоматичному виконанні основних компонентів організації експерименту, то таке врахування може бути *повним* і *частковим*.

**Означення 6.5.** Повне врахування параметра  $x_{ОМ}^i$  вважається забезпеченим, якщо в процесі експерименту вихідні дані (або результат) однозначно ідентифікують його значення.

Тоді очевидна справедливість наступного твердження.

**Твердження 6.6.** Необхідною умовою повного врахування впливу цього параметра  $x_{ОМ}^i$  (або на цей параметр  $x_{ОМ}^i$ ) є умова  $x_{ОМ}^i \in X_{W_{вх}}$  (або  $x_{ОМ}^i \in Y_{вих}$ ). Достатні ж умови визначаються видом взаємозв'язків, можливостями зростаючого та спадного встановлення значень параметрів.

При зростаючому встановленні значення  $a_i^j$  похідного параметра  $x_{W_{вх}}^i$  легко визначається з належності множині  $C_i^j$  комбінацій таких значень  $a_i^m$ , що  $x_{W_{вх}}^i \in x_{ОМ}^i$ . При спадному встановленні одному значенню  $a_i^j$  параметра  $x_{W_{вх}}^i \in X_{W_{вх}}^0$  можуть відповідати кілька альтернативних комбінацій з  $C_i^j$ , що однозначно не визначає набір усіх початкових даних на рівні подання в ІМ. Для однократного прогону ІМ вхідне слово має містити тільки таку комбінацію значень похідних параметрів, яка забезпечує однозначну ідентифікацію значень параметрів  $x_{W_{вх}}^i \in X_{ІМ}^z$  нижчого рівня.

Розглянуті вище формалізми та вимоги стосуються насамперед задач, які пов'язані з повним урахуванням параметрів.

Суть *часткового врахування параметра* ґрунтується на визначенні належності його значень до деяких підмножин. Часткове врахування параметра  $x_{ОМ}^i$  може спостерігатися: якщо  $x_{ОМ}^i \in X_{ІМ_{вх}}$  (або  $x_{ОМ}^i \in Y_{вих}$ ) і набір початкових даних не дає змоги визначити його значення однозначно; якщо

$$\begin{aligned} & (x_{ОМ}^i \in X_{W_{вх}}) \& (x_{ОМ}^i \cap X_{W_{вх}} \neq \emptyset) (x_{ОМ}^i \setminus X_{W_{вх}} \neq \emptyset) \\ & \text{(або } (x_{ОМ}^i \in Y_{вих}) \& (x_{ОМ}^i \cap Y_{вих} \neq \emptyset) (x_{ОМ}^i \setminus Y_{вих} \neq \emptyset)). \end{aligned}$$

Так, у другому випадку часткове врахування параметра здійснюється з урахуванням значень параметрів з  $x_{ОМ}^i \cap X_{W_{вх}}$  (або  $x_{ОМ}^i \cap Y_{вих}$ ).

Кожне таке сполучення параметрів може входити до комбінації, що належить одній або кільком множинам  $C_i^j$  і, таким чином, відповідати одному або кільком значенням  $a_i^j$  параметра  $x_{OM}^i$ . Подальше прийняття рішень на основі часткового врахування параметрів потребує, як правило, неформалізованих дій і цілеспрямованого використання елементів III в задачах автоматизації моделювання об'єктів (детальніше див. у розд. 12).

### 6.5. Загальні основи моделювання складних комп'ютерно-інтегрованих систем

**Гнучка КІС як об'єкт інтелектуалізованого моделювання, планування і керування.** Інтеграція різних матеріальних, енергетичних та інформаційних процесів, що відбуваються в гнучких комп'ютерно-інтегрованих системах (ГКІС), до того ж таких, що характеризуються багатомономенклатурністю оброблюваних виробів, дають змогу розглядати такі системи як складні, динамічні, паралельні з циклами, які повторюються. До найбільш ефективних засобів подання та дослідження процесів ГКІС можна зарахувати графоаналітичні методи, в тому числі й ті, що базуються на теорії сіток Петрі (СП-методи) [54; 74]. Перевагою цієї теорії є те, що вона дає можливість моделювати динаміку ГКІС на різних рівнях абстракції, а також на різних рівнях інтегрованості ГКІС і їх окремих підсистем, функціональних модулів, елементів і вузлів технологічного обладнання тощо [139].

З іншого боку, ГКІС можна розглядати як логістичну систему. Типова логістична система складається з потоків товарів (вантажів) і послуг, а також спостереження (вивчення) і керування (регулювання) цими потоками [181]. Серед складових процесів ГКІС найбільш типовими є транспортування, розподілення і власне виготовлення виробів. Координація цих процесів і складових є головним об'єктом керування логістичної системи, тобто мета полягає в досягненні добре синхронізованого режиму динамічної взаємодії компонент. Серед багатьох якісних показників (таких як «вузькі місця» або простоювання), які визначають режим функціонування системи, блокування процесу (тупикові ситуації) через наявність «критичного циклу» в послідовності дій [178], а також ймовірність виникнення нештатних режимів роботи системи внаслідок впливу на неї надзвичайних факторів [130] відіграють найсуттєвішу роль.

У межах цього розділу можна виокремити чотири основні задачі, осмислення і комплексне урахування яких певною мірою сприятиме розв'язанню загальної проблеми підвищення ефективності створення, дослідження і верифікації ГКІС, а саме:

- розкриття визначальних властивостей процесів ГКІС і методів формалізованого її подання як об'єкта інтелектуалізованого моделювання, планування і керування;
- аналіз методів і засобів моделювання ГКІС;
- обґрунтування апарату сіток Петрі як базового для ідентифікації і верифікації процесів;
- аналітична систематизація еволюції розвитку основних модифікацій базового СП-апарату.

Структурно розділ побудований таким чином, щоб послідовно висвітлити шляхи розв'язання зазначених задач із застосуванням для цього нових підходів.

Сучасна ГКІС\* являє собою складний науково-виробничий комплекс, який реалізує комплексно-автоматизоване виробництво, що оперативнo переналагоджується у певному асортименті продукції. Така система характеризується технологічною, структурно-організаційною, параметричною гнучкістю та гнучкістю потужності [23; 142]. Термін «інтегрована» передбачає узгодження складових систем: необхідного і достатнього за своїм технологічним призначенням для певної номенклатури виробів основного устаткування; засобів сервісного обслуговування промислових роботів та упорядкування середовища під групову технологію; матеріальних потоків і засобів їх нагромадження, перевантаження і розподілення за обробляючими ресурсами; підсистем автоматизованого проектування керуючих програм, технологічного оснащення і типових технологічних процесів виготовлення виробів і, передусім, власне об'єктів виробництва певної номенклатури.

Отже, є сенс говорити про ГКІС, коли різнономенклатурні матеріальні потоки, відмінні за класом і призначенням устаткування, підсистеми підготовки і керування виробництвом функціонують, а диспетчеризація матеріальних потоків та інформаційних сигналів і команд здійснюється за допомогою засобів комп'ютерної підтримки і забезпечення складових гнучкості.

Для всіх стадій виробничого циклу в ГКІС властиві три рівні в ієрархії керування: стратегічний; тактичний; виконавчий. Підрозділи ГКІС на стадіях підготовки, освоєння та виготовлення виробів мають близький набір задач: при *плануванні* — оптимальний розподіл обсягів продукції на календарний період часу за обробляючими ресурсами; при *керуванні* — диспетчеризування в реальному часі інформаційних і матеріальних потоків (заготовок, напівфабрикатів, готових виробів, керуючих програм, засобів технічного забезпечення і завдання операторам та лінійному персоналу).

Оперативне керування і планування виробництва становлять тактичний і стратегічний рівні керування.

Зокрема, на *виконавчому* рівні ставляться задачі оптимізації функціонування обробляючих ресурсів, а на *тактичному* і *стратегічному* рівнях, відповідно, — задачі диспетчеризування матеріальних та інформаційних потоків і оптимального планування робочих ресурсів. В результаті розв'язання задач диспетчеризування на тактичному рівні план адаптується до зміни виробничої ситуації. Для підвищення надійності функціонування ГКІС у цілому необхідно автоматично перерозподіляти матеріальні/інформаційні потоки, що пов'язано з побудовою алгоритмів керування цими потоками у ГКІС.

Загальноприйнятого визначення диспетчерського керування немає, проте ним успішно користуються розробники автоматизованих систем керування виробництвом. Зокрема, для ГКІС на рівні дільниці задачею диспетчерського керування є координація роботи складових обробляючих модулів такої системи між собою і з АТСС. Розгляд шляхів розв'язання цієї задачі із загальних позицій дає змогу створити єдиний підхід до розробки різних автоматизованих систем диспетчерського керування в ГКІС, які забезпечують найвищу ефективність їх функціонування (максимальне використання обробляючих ресурсів і завантаження компонентів АТСС, а також відсутність простоювань компонентів ГКІС через блокування, пов'язане з нераціональним використанням ресурсів).

\* Тут і надалі йдеться про системи виробничого призначення. Проте всі міркування і викладені далі підходи щодо аналізу і синтезу складних систем можуть використовуватися для ГКІС будь-якої природи (організаційних, економічних, екологічних, технічних тощо).

**Скінченно-автоматна інтерпретація ГКІС.** Наявність функціональної та організаційної аналогії у процедурах оперативного керування процесами на всіх рівнях інтеграції і в складових ГКІС дає підстави для використання єдиної методологічної основи формального опису їх функціонування. Методи доступу до внутрішніх накопичувачів інструменту, заготовок, що реалізуються в ГКІС, можна формально подати методами доступу до довгочасної (автоматизовані склади (АСк)) та оперативної (міжопераційні нагромаджувачі) пам'яті ЕОМ у складі автоматизованої системи керування. Інтегрування окремих гнучких виробничих одиниць в ГКІС організовано за принципом мультипроцесора, коли транспортна сітка, яка зв'язує окремі складові ГКІС з АСк, легко асоціюється з транспортним каналом «загальна шина». Виробничі підрозділи ГКІС за допомогою такого каналу інтегруються за принципом «загальна пам'ять», якою є АСк заготовок, інструменту, оснащення і готових виробів, роль інтерфейсу введення/виведення інформації в системі відіграють пристрої завантаження/розвантаження об'єктів виробництва, в тому числі промислові роботи (ПР).

Отже, всі перелічені компоненти ГКІС з'єднані між собою за допомогою шин: адреси і даних (потоки запитів, відповідей, повідомлень); даних (потоки об'єктів виробництва); керування (потік команд диспетчеризування). Все це дає змогу координувати їх спільну роботу.

Усе викладене вище дає змогу сформулювати поняття про *технологічний процесор* ГКІС і розкрити уявлення про *технологічне мультипроцесорне середовище* ГКІС, згідно з яким процеси в ній подібні до обчислювальних процесів у мультипроцесорних структурах [99]. Водночас систему керування будь-яким компонентом (або в цілому) ГКІС можна розглядати як *скінченний автомат* (СкА) з мікропроцесорним керуванням. Тоді в загальному вигляді підхід до розгляду керування ГКІС з точки зору теорії СкА полягає в описі всіх функціональних модулів (у тому числі ПР) як СкА і в побудові для керування ними відповідного пристрою (або кількох пристроїв), який також є СкА. Такий підхід дає змогу формалізувати опис функціонування ГКІС у вигляді логічної сітки, що складається із скінченних автоматів, і подати роботу будь-якого елемента ГКІС на логічному рівні керування також у вигляді сітки автоматів, об'єднаних загальними входами та виходами. Тоді, виходячи з позиції теорії СкА, передбачається, що процес переходу обладнання (обробних ресурсів у більш загальному трактуванні для всіх інтегрованих у ГКІС основних і забезпечувальних підсистем) з одного стану до іншого при надходженні керуючої дії (тобто виконання певних умов) не контролюється, а фіксується лише факт переходу після досягнення бажаного результату. Наведені міркування є основним висновком, що свідчить на користь застосування СП-методів як базового апарату для формального подання процесів ГКІС.

**Особливості функціонування ГКІС.** Високою функціональною та структурною складністю зумовлені деякі особливості функціонування ГКІС, які необхідно враховувати при моделюванні. До таких особливостей можна зарахувати [95; 125; 126; 139]:

1. *Циклічність.* Практично всі процеси в ГКІС циклічні, тому що її функціонування пов'язане з багаторазовим виконанням одного й того самого технологічного процесу, що змінюється досить рідко.

2. *Дискретність.* При роботі ГКІС постійно відбуваються деякі, добре помітні події (наприклад, вмикання або вимикання устаткування, початок і кінець перемі-

щення або обробки деталі і т. ін.), які являють собою передумови для подальших дій, виконуваних системою; тому можна говорити про дискретний характер функціонування ГКІС.

3. *Асинхронність*. Події, що виникають у ході роботи ГКІС, відбуваються в невизначені моменти часу, отже, ГКІС є асинхронною системою.

4. *Паралельність*. Гнучкі виробничі системи характеризуються наявністю безлічі матеріальних і нематеріальних потоків, що переміщуються одночасно – переміщення заготовок, готових виробів, інструментів, відходів виробництва, виконання технологічних і допоміжних операцій, передача інформації про керуючі впливи, стан устаткування й об'єктів обробки (складання) і т. ін.; відповідно й функції, що керуються або керують цими потоками, виконуються одночасно, що зумовлює паралельність процесів, які відбуваються в ГКІС.

5. *Наявність черг*. Робота транспортно-складської системи, що нагромаджує і розподіляє заготовки, вироби та інструменти, неминуче призводить до виникнення черг на обслуговування або складування.

6. *Конкуренція за ресурси*. Така конкуренція зумовлюється можливою наявністю декількох одиниць устаткування, що виконують аналогічні дії, або неоднозначністю технологічного процесу.

Отже, виробничий процес в ГКІС складається з окремих асинхронних й узгоджених технологічних процесів, що відбуваються одночасно на різних обробляючих ресурсах. Використання паралельних процесів дає змогу легко структурувати систему, а кожен процес подати як самостійну програмну одиницю. При цьому ГКІС легко піддається ідентифікації і дослідженню.

**Формування вимог до моделі ГКІС.** Для того щоб можна було здійснювати ефективні дослідження, модель ГКІС повинна задовольняти певні вимоги до неї [108]:

1. *Адекватність*. Модель повинна описувати все різноманіття властивостей функціонування ГВС і мати ефективні засоби відображення паралелізму, альтернативи, асинхронності, конвеєрності, циклічності, ієрархічності й конкуренції.

2. *Універсальність*. Модель повинна подавати як матеріальні процеси, так і процеси керування, а також моделювати інформаційні потоки між технологічною системою і системою керування.

3. *Високий ступінь формалізації*. Модель повинна мати низьку трудомісткість, наочність й аналітичність формалізованого опису.

4. *Можливість оптимізації*. Повинна бути забезпечена можливість розв'язання на моделях оптимізаційних завдань (аналітично або засобами імітаційного моделювання).

5. *Гнучкість*. При зміні станів зовнішнього середовища повинна існувати можливість здійснення структурних і параметричних перетворень моделі.

6. *Коректність*. Повинні існувати методи перевірки правильності взаємодій і відсутності блокувань ресурсів за формальним описом ОМ.

**Аналіз методів моделювання ГВС. Методи розв'язання задач оперативного керування процесами в ГКІС.** Центральною підсистемою автоматизованого оперативного керування є підсистема *оперативно-календарного планування* (ОКП), яка задає динаміку ГКІС, визначає взаємодію сукупності її елементів, тобто узгоджений у часі і в маршрутно-технологічному орієнтованому просторі (підрозділи, ділянки, робочі місця) рух матеріальних та інформаційних потоків.



Основним методом розв'язання задач, що входять до ОКП, є *формування розкладу роботи обробляючих ресурсів* (ОБР). Проте специфіка дрібносерійного та одиничного виробництва (що є найхарактернішою сферою застосування ГКІС) призводить до того, що розклад роботи ОБР неможливо скласти на тривалий час — реальний процес швидко змінюється, відхиляючись від запланованого, в тому числі і у зв'язку з надходженням директивних пріоритетних завдань.

Питання, пов'язані з розробкою алгоритмів формування розкладів, вивчаються в *теорії розкладів*. По суті, формування розкладу є задачею упорядкування скінченної множини робіт при заданих обмеженнях, яка зводиться до визначення такого упорядкування, яке оптимізує деякий критерій. Можна виокремити три шляхи розв'язання задачі упорядкування: скінченне перебирання можливих варіантів (потребує значних часових і машинних ресурсів); математичного моделювання (характеризується значною трудомісткістю і неможливістю враховувати реальні виробничі обмеження і, як наслідок, можливістю формувати розклад на невеликі проміжки часу); евристичні методи (не забезпечують строго оптимальних рішень, проте дають змогу формувати розклад на досить великий термін).

Для евристичних методів характерним є поняття дисципліни призначень (наприклад, першим прийшов — першим обслуговується, правило найкоротшої або найдовшої операції, правило «розшивання» «вузьких місць» у технологічному маршруті тощо).

Задача упорядкування є *NP*-складною задачею теорії розкладу [51]. Через її значну розмірність перевагу віддають використанню наближених методів розв'язання, в основі числового алгоритму яких лежить *метод розв'язувальних (евристичних) правил*. Ідея цього методу полягає в послідовній, крок за кроком, побудові розкладу призначенням операцій (з технологічно підготовлених) на вільний на момент розгляду обробний ресурс. Обмеженість ресурсів і черга операцій, що виникає при цьому, потребують прийняття рішень щодо того, яким операціям надавати ресурси в першу чергу на основі правил пріоритету. При цьому остаточний вибір здійснюється або за формальним критерієм (наприклад, за мінімумом часу реалізації програми випуску продукції), або на основі переваг, наданих користувачем.

І нарешті, якщо при реалізації поточного плану ОКП наявне незавершене виробництво, то задачу можна сформулювати так: узгодити план випуску з незавершеним виробництвом так, щоб загальна напруженість була найменшою (аналогічна ситуація спостерігається і при надходженні пріоритетних замовлень). Її можна звести до задачі *цілочислового лінійного програмування* [31].

**Аналітичне й імітаційне моделювання.** Вибір апарату моделювання, який враховував би всі особливості функціонування ГКІС і задовольняв усі вимоги, викладені вище, можливий тільки на основі аналізу існуючих формальних підходів і методів побудови моделей.

Для моделювання ГКІС використовуються різноманітні методи, які можна розділити на два класи: аналітичні й імітаційні [59; 100].

**Аналітичне моделювання.** Аналітичне моделювання базується на описі об'єкта, що моделюється, за допомогою набору рівнянь, шляхом аналітичного розв'язання яких можна обчислити необхідні характеристики досліджуваного об'єкта. Найістотніша характеристика аналітичних моделей полягає в тому, що модель не є структурно подібною до об'єкта моделювання. Під структурною подібністю мається на увазі однозначна відповідність елементів і зв'язків моделі елементам і зв'язкам

об'єкта. Процеси, що відбуваються на об'єкті, не мають прямих аналогів у моделі. До аналітичних зараховують моделі, побудовані на основі апарату математичного програмування, кореляційного, регресійного аналізу. Такі моделі завжди являють собою формальну конструкцію, яку можна розв'язати й проаналізувати математичними засобами, і є ефективним інструментом для розв'язання задач оптимізації або обчислення характеристик виробничих систем.

*Імітаційне моделювання.* Імітаційне моделювання полягає у проведенні численних експериментів на основі програмно реалізованої моделі об'єкта з метою дослідження її характеристик. Особливістю таких моделей є структурна подібність об'єкта і моделі. При побудові імітаційної моделі описуються закони функціонування кожного елементу об'єкта і зв'язків між ними. Процес, що відбувається в моделі в ході експерименту, подібний до процесу в реальному об'єкті, отже, дослідження об'єкта зводиться до вивчення характеристик процесу, що відбувається в моделі.

Імітаційна модель дає змогу відображати особливості функціонування системи з будь-яким ступенем деталізації (зазвичай за рахунок збільшення складності й розмірів моделі).

**Формалізоване подання об'єктів керування.** Моделлю об'єкта керування (ОК) є формальний опис, яким задається подальша реалізація алгоритму керування програмним способом. Ефективність керування на всіх стадіях виробничого циклу в ГКІС залежить насамперед від коректності побудови моделі ГКІС як ОК. Найширшого застосування на практиці набули математичні методи теорії масового обслуговування [47], імітаційного моделювання [101; 112; 140; 195; 200], теорії графів і сіток [80; 83; 113; 179].

При застосуванні *методів теорії масового обслуговування* для моделювання процесів ГКІС остання розглядається як система керування, яка реагує на асинхронні події усередині самої ГКІС. При цьому в основному передбачається дослідження залежності кількісних показників обслуговування від характеристик потоку заявок і параметрів в АТСС (продуктивність ГКІС, кількість і потужність ресурсів певного класу тощо). Проте модель, побудована методами теорії масового обслуговування, не відображає паралельності поточних процесів, які конкурують у доступі до спільних ресурсів, тому спрощено подає динаміку реального ГКІС. Таке моделювання можна використовувати для попереднього оцінювання альтернативних варіантів моделей ГКІС.

*Імітаційні моделі* є ефективним інструментом дослідження і врахування таких особливостей ГКІС, як багатоваріантність розподілу потоків завдань між ресурсами, стохастичний характер виконання завдання кожним ресурсом та ГКІС загалом, можливість надходження непередбаченої планом заявки, наявність альтернативності в послідовності виконання окремих етапів завдань, і дають змогу простежити динаміку процесу функціонування ГКІС, виявити «вузькі місця», якісні та кількісні взаємозв'язки між параметрами ГКІС. Проте можливості ІМ обмежені задачами аналізу ситуацій, що виникають, збирання статистики і не дають змоги ефективно розв'язувати задачі синтезу, формувати керуючі алгоритми, які наперед володіють потрібними властивостями. Тому ІМ може бути додатковим засобом розв'язання задач формалізованого подання процесів керування і планування при дослідженні ГКІС.

Залежно від класу системи, ступеня деталізації її структури, характеру досліджуваних проблем застосовують *сіткові моделі і графові методи* для вивчення

властивостей дискретних процесів. До найбільш уживаних для цієї мети належать марковані (сигнальні) графи, графи обчислень,  $P/V$ -системи та системи з повідомленнями, автоматні системи взаємозв'язаних графів (АСВГ), граф-схеми взаємодіючих паралельних процесів, діалогічні графи (*UCLA*-графи), системи складання і заміщення векторів, сітки Петрі та їх розширення [54; 83; 113; 125; 139; 179].

З огляду на викладене можна стверджувати:

- *Перший напрямок* у розвитку методів абстрактного математичного моделювання, пов'язаний з дослідженням абстрактних автоматів, логічних сіток та рекурсивних функцій, спрямований головню на розв'язання принципів питань абстрактної математичної логіки функціонування дискретних систем.
- *Другий напрямок* пов'язаний насамперед з теорією і практикою імітаційного моделювання, а також з поданням складних систем і процесів, що в них відбуваються, формальною мовою високого рівня.

Неузгодженість цих напрямків полягає в тому, що:

- абстрактність теоретичних досліджень та узагальнень першого напрямку утруднює використання їх методології і результатів при розв'язанні задач не тільки структурного синтезу й аналізу складних систем, а й визначення і перевірки кількісних та якісних характеристик останніх;
- водночас обмеженість підходів і можливості використання результатів другого напрямку розвитку моделювання виявляється, якщо необхідно розкрити структуровану сутність об'єкта досліджень, а також супроводжується значно гіршим, з точки зору наочності, відображенням логічних (і функціональних) зв'язків та особливостей взаємодії компонентів об'єкта досліджень.

Ось чому формальний апарат моделювання, який базується на *сіткових і графоаналітичних методах*, займає проміжну нішу між наведеними вище напрямками.

Нижче наведено короткий огляд найпоширеніших підходів до моделювання дискретних, безперервних і комбіновано-гібридних систем.

**Моделювання дискретно-подійних систем.** Існує ряд різних формалізмів і підходів до моделювання *дискретно-подійних систем* (ДПС). Застосування того чи іншого апарату зумовлюється його аналітичними можливостями, моделюючою потужністю, обчислювальною складністю, прозорістю відображення в ньому процесів, що відбуваються в системі тощо.

**Теорія скінченних автоматів.** Дано означення дискретно-подійної системи.

**Означення 6.6.** *Дискретно-подійна система* в автоматному поданні розглядається як п'ятірка  $D = (\Sigma, Q, \delta, q_0, Q_m)$ , де  $\Sigma$  – це множина (алфавіт) подій,  $Q$  – множина станів,  $\delta: \Sigma \times Q \rightarrow Q$  – функція переходів (часткова),  $q_0 \in Q$  – початковий стан,  $Q_m \subseteq Q$  – множина позначених станів [168; 188; 189].

Динаміка системи трактується в термінах здійснених переходів у такому автоматі та їхніх послідовностей і описується словами з мови  $L$ , породженої  $D$ :

$$L(D) = \{w \mid (w \in \Sigma^*) \wedge \delta(w, q_0)\}.$$

Існують модифікації автоматів, які додатково відображають специфічні характеристики систем: стохастичні, часові, нечіткі, гібридні та ін. [171; 175; 194].

Теорія автоматів забезпечує розвинений апарат опису й аналізу переходів між станами системи, однак не відображає причинно-наслідкових відношень на рівні внутрішніх процесів, а також паралелізму, що впливає з часткової впорядкованості подій у ГВС; окрім того, композиція моделей і їх ієрархічний опис у рамках апарату дещо ускладнені.

**Системи масового обслуговування (СМО).** Методи і результати теорії масового обслуговування [67; 68; 82] використовуються при розв'язуванні задач аналізу процесів функціонування ГКІС як системи, у якій виконуються однорідні операції обслуговування. Така система зазвичай складається із сервера (у загальному випадку — обробляючого ресурсу) й черги, що складається з потоку заявок. Потік являє собою множину випадкових моментів надходження заявок. Залежно від функції розподілу ймовірностей виокремлюють різні види потоків (наприклад, пуасонівський, ерлангівський).

За наявності декількох серверів, які працюють паралельно, обслуговуючи вхідний потік заявок, така система називається багатолінійною; якщо сервери послідовно обслуговують заявки, то це багатофазна система. Кілька серверів, зв'язаних потоками заявок, утворюють мережу черг.

Найважливішими характеристиками, розглянутими в теорії масового обслуговування, є показники зайнятості окремих серверів/обробляючих ресурсів, середній час очікування, інтенсивності потоків та ін.

Найпростіший і вивчений клас СМО — марковські процеси, що дають змогу одержати характеристики СМО у вигляді аналітичних виразів і дослідити більш загальні моделі шляхом зведення їх до марковських.

**Означення 6.7.** Марковським процесом називається випадковий процес  $v(t)$ ,  $t \in T$  зі значеннями з фазового простору  $X$ , що задовольняє таку властивість: для кожного набору моментів часу  $t_1 < t_2 < \dots < t_n < \dots$  з  $T$  і будь-якої множини  $B \subset X$  є справедливим

$$P(v(t_{n+1}) \in B | v(t_1), \dots, v(t_n)) = P(v(t_{n+1}) \in B | v(t_n)).$$

Прикладом марковського процесу є процес із пуасонівським потоком подій, який називається *пуасонівським*. Марковська властивість цього процесу впливає із властивості показового розподілу. Марковський процес  $X(t)$ ,  $t \in T$ , у якого  $X \subset \{1, 2, \dots\}$ , називається *ланцюгом Маркова*.

Апарат теорії марковських процесів дає змогу розраховувати ймовірності знаходження системи в конкретному стані, інтенсивності переходів і т. ін. Окремим напрямком є апарат так званих напівмарковських процесів, який базується на розгляді спеціально введених зовнішніх подій, що спонукають систему до переходу.

Таким чином, теорія СМО враховує стохастичний аспект поведінки системи й дає можливість розрахувати його числові характеристики щодо певних проміжків часу. Водночас СМО абстрагуються від структури й внутрішньої логіки функціонування системи, потребують певних обмежувальних припущень щодо характеру потоків, унеможливають розгляд на рівні обробки індивідуальних завдань.

**Сітки Петрі.** Ефективним інструментом моделювання дискретних процесів є сітки Петрі (СП) [54; 83; 191]. Їхні основні особливості — можливість відображати паралелізм, асинхронність, ієрархічність об'єктів, що моделюються, простішими засобами, аніж при використанні інших засобів моделювання [83; 116; 117; 134; 139; 150; 151; 197].

Сітки Петрі розроблялися спеціально для побудови системних моделей процесів, установ і систем, в яких значну роль відіграють матеріальні та особливо інформаційні регульовані потоки.

Загалом порівняно з іншими системними моделями для СП характерно [192]:

1. Причинно-наслідкові зв'язки всередині множини подій подаються явно. Незалежні одна від одної події не проектуються на часову вісь; для них вводять-

ся відношення паралельності або конкуренції. Ці поняття є основними для загальної теорії сіток.

2. Системи можуть подаватися на різних рівнях абстрагування без зміни мови подання. Ці рівні абстрагування дають змогу досліджувати складні системи з різним ступенем деталізації.
3. Сіткове подання дає можливість досліджувати властивості систем і доводити коректність процесів, використовуючи для цього формальні засоби теорії сіток.

**Означення 6.8.** *Сітка Петрі (СП) з маркуванням  $\mu$*  — це п'ятірка  $N = (P, T, F, H, \mu)$ , де  $P$  — скінченна множина позицій;  $P = \{p_i\}$ ,  $i = 1, n$ ;  $T$  — скінченна множина переходів;  $T = \{t_j\}$ ,  $j = 1, m$ ; множини  $P$  і  $T$  не перетинаються;  $P \cap T = \emptyset$ ;  $F: P \times T \rightarrow \{0, 1, 2, \dots\}$  — вхідна функція (відображення з переходів у позиції);  $H: T \times P \rightarrow \{0, 1, 2, \dots\}$  — вихідна функція (відображення з позицій у переходи);  $\mu$  — маркування СП (відображення множини позицій у множину невід'ємних цілих чисел  $\mathbf{N}$ ), а саме  $\mu: P \rightarrow \mathbf{N}^n$ ,  $n = |P|$ .

Граф СП — дводольний орієнтований мультиграф, у якому позиції позначають кружками, переходи — планками, а вхідні й вихідні функції — орієнтованими дугами, кратність яких відповідає значенню функції. Функція  $\mu$  визначає для кожної позиції  $p_i$  СП кількість фішок (маркерів) у цій позиції, яка дорівнює  $\mu(p_i)$ . Функціонування СП відображається шляхом зміни її маркування.

На базі СП створено множину модифікацій, покликаних збільшити їхні моделюючі можливості: часові СП, інгібіторні СП, розфарбовані СП, навантажені СП та ін. [54; 83; 156]. Загалом слід відзначити придатність СП до усіляких модифікацій, що робить їх зручним інструментом для аналізу систем зі специфічними особливостями поведінки. Сітки Петрі набули широкого застосування в таких галузях, як дослідження і проектування систем логічного керування технологічними процесами (у працях О. О. Таля, С. А. Юдицького [116; 117; 134], В. О. Горбатова [107], А. В. Борусана, К. Б. Остапченко, А. А. Лаврова, Л. С. Ямпольського [62; 74; 125]), моделювання асинхронних процесів в обчислювальних системах (у працях В. Е. Котова [53; 54], О. Л. Бандман [69]), задачі схемотехніки для дискретної автоматики (в працях Б. І. Варшавського, Л. Я. Розенблюма, Б. С. Цирліна [2; 7; 104]), проектування робототехнічних систем (у працях В. М. Назаретова, А. В. Кульби [100; 103], Б. Ф. Фоміна, А. Н. Плоткіна [124]).

Сітки Петрі ефективно відображають дійсний паралелізм і внутрішню логіку функціонування системи; крім того, вони мають потужні засоби аналізу поведінки. До недоліків можна зарахувати недостатні комбінаційні властивості, обмежені можливості синтаксичного опису, швидке збільшення розмірності при збільшенні масштабів системи.

**Різницеві рівняння.** Різницеві рівняння описують систему у вигляді моделі

$$\bar{x}_0, \bar{x}_{n+1} = f(\bar{x}_n, \bar{u}_n), \quad \sigma_n = \varphi(\bar{x}_n, \bar{u}_n),$$

де  $\bar{x}$  — вектор фазових координат,  $\bar{u}$  — вектор керування,  $\sigma$  — вихід системи.

Різницеві рівняння є формою подання, ефективною при заданій дискретизації часу й відомих послідовностях подій; крім того, вони досить прості й зручні при їх комп'ютерній реалізації для числових розрахунків. Однак у складних, особливо паралельних і розподілених ДПС, відповідне аналітичне подання в різницевих рівняннях і його аналіз ускладнені. Наявність факторів недетермінованого вибору, потенційних тупикових ситуацій, пов'язаних з паралельними процесами, надзвичайно ускладнюють і унеможливають використання цього апарату.

**Інші методи.** Крім розглянутих вище, найчастіше вживаних для моделювання ДПС, також використовуються такі методи.

**Процесні алгебри (Бокс-алгебри)** [147; 166; 180] – методи, що забезпечують подання процесів (поведінки системи) алгебраїчними виразами, утвореними з процесних термів за допомогою комбінованих операторів. Найважливішими операторами є: паралельна композиція, композиція вибір, послідовна композиція, префікс-атомарна дія. Аналіз особливостей процесів здійснюється дослідженням еквівалентностей виразів, через використання аксіоматичних законів та операційних правил. Алгебри процесів забезпечують розвинені набори операцій композиції, компактне синтаксичне подання, розвинені методи синтаксичного аналізу; водночас вони не здатні відображати дійсного паралелізму, ускладненого аналізу динамічних особливостей з погляду використання структурних методів.

**Діодні алгебри (Макс-плюс алгебри)** [10; 98; 157] – алгебри, які часто використовують при дослідженні поведінки дискретних систем, на відміну від неперервних, тільки дві операції – максимум (мінімум) і суму. За допомогою цих операцій визначається математичний апарат, що дає змогу досліджувати системи з досить складною структурою. Макс-плюс алгебра дає змогу формулювати певні рівняння стану дискретної системи й робити деякі висновки щодо її поведінки, однак потребує досить специфічного подання системи.

Можливий опис систем на базі різних логік (логіки предикатів, модальної логіки), однак такі моделі більше придатні для специфікації вимог до системи й опису таких її особливостей, як живучість і досяжність, аніж для відображення логіки роботи системи.

**Моделювання неперервних систем.** У той час, як для моделювання ДПС існує множина різноманітних методів при відсутності одного загальнозживаного, для моделювання безперервних систем використовують практично тільки *диференційні рівняння*. При цьому зміна стану системи в часі задається за допомогою аналітичної функції, наприклад

$$\frac{d(\bar{x}(t))}{dt} = A(t)\bar{x}(t) + B(t)\bar{u}(t),$$

де  $\bar{x}(t)$  – вектор змінних стану системи,  $\bar{u}(t)$  – вектор керуючих впливів.

Розроблено множину методів аналітичного й числового розв'язання диференційних рівнянь і систем на їх основі. За допомогою цього підходу можна визначати якість і тривалість перехідного процесу, наявність і параметри сталих режимів роботи і т. ін.

Крім того, використання диференційних рівнянь дає змогу синтезувати системи із наперед заданими характеристиками функціонування.

**Моделювання компонентів складних динамічних систем.** Для побудови тематичної моделі компонентів складної динамічної системи, якими у загальному випадку є устаткування, що входить до складу ГПС, застосовують три групи методів, доведених до повної автоматизації; це методи, що ґрунтуються на *теоремах механіки* [16; 17; 57; 186]; *рівняннях Лагранжа* [15; 24; 84; 167]; *«енергії» прискорення* [15; 84].

При використанні методів, що ґрунтуються на *теоремах механіки*, задача динамічного аналізу розділяється на три етапи. *Перший етап* – кінематичний аналіз, у ході якого розраховуються положення, переносні швидкості й прискорення центрів ваги, а також кутові швидкості й прискорення всіх ланок механізму. *На другому етапі* за допомогою знайдених величин кутових швидкостей і прискорень ланок

знаходять головний вектор і головний момент інерційних сил. *Третій етап* — статичний аналіз, на основі якого вирішується задача рівноваги механізму під дією зовнішніх та інерційних сил.

У методах, що ґрунтуються на *рівняннях Лагранжа*, математичну модель одержують у вигляді

$$\frac{d}{dt} \left( \frac{\partial \mathbf{E}}{\partial \dot{\mathbf{q}}} \right) \cdot \left( \frac{\partial \mathbf{E}}{\partial \mathbf{q}} \right) = \mathbf{Q},$$

де  $\mathbf{E}$  — вектор кінетичної енергії;  $\mathbf{q}$  — вектор узагальнених координат;  $\mathbf{Q}$  — вектор зовнішніх сил.

Методи, що ґрунтуються на «енергії» прискорення, базуються на функції Гіббса

$$G = \frac{1}{2} \sum_{i=1}^n m_i \left[ \mathbf{W}_{i1}^2 + \mathbf{W}_{i2}^2 + \mathbf{W}_{i3}^2 \right]^T,$$

де  $m_i$  — маса  $i$ -ї ланки;  $\left[ \mathbf{W}_{i1}^2 + \mathbf{W}_{i2}^2 + \mathbf{W}_{i3}^2 \right]^T$  — вектор прискорень  $i$ -ї ланки. Рівняння Гіббса — Аппеля визначає закон руху системи  $\left( \frac{\partial G}{\partial \ddot{\mathbf{q}}_i} \right) = \mathbf{Q}_i$ .

Всі зазначені методи широко застосовуються на практиці, однак найчастіше для опису динаміки складних механічних систем використовують рівняння Лагранжа II роду.

**Моделювання комбіновано-гібридних систем.** Системи, функціонування яких поєднує в собі властивості дискретності й безперервності, можна умовно зарахувати до комбіновано-гібридних інтелектуальних систем (КГІС). Розробка формальних засобів опису КГІС, їхнього аналізу й синтезу не є актуальною задачею. Переважна більшість існуючих підходів не оперує якимсь універсальним метаформалізмом, що дає змогу описати всі аспекти поведінки системи, а спрямовані на комбінування взаємодії неперервних і дискретних компонентів.

Найпоширенішим методом моделювання КГІС є метод, що ґрунтується на застосуванні *карт станів* (statecharts) [163]. Карта станів являє собою орієнтований граф, вершини якого відображають стани системи, а дуги — переходи з одного стану в інший. У загальному випадку кожен перехід характеризується запускаючою подією  $A$ , запобіжним предикатом  $B$  і дією  $C$ . Семантика спрацьовування переходу з вершини  $P_1$  у вершину  $P_2$  така: якщо система перебуває в стані  $P_1$  і при настанні запускаючої події  $A$  запобіжний предикат  $B$  дійсний, то виконується дія  $C$  і система переходить у стан  $P_2$ .

Існують модифікації карт станів, у яких кожній вершині приписується неперервна поведінка, тобто перебуваючи в певному стані, система функціонує за певним неперервним законом ( $B$ -charts) [48].

Хоча цей підхід і дає змогу створювати моделі КГІС, з погляду моделювання дискретних аспектів поведінки він має недоліки, характерні для теорії автоматів.

### Контрольні завдання

1. Дати означення оптимальних процесів (систем).
2. Охарактеризувати методи фізичного і математичного моделювання.
3. Розтлумачити сутність багатокритеріальних задач.
4. Охарактеризувати основні задачі моделювання складних об'єктів.
5. Розкрити відмінність основних підходів до моделювання об'єктів.

6. З'ясувати, в чому полягає індуктивний підхід до синтезу структури моделі оптимальної складності.
7. Розкрити особливості використання методу групового врахування аргументів у плануванні, проведенні та інтерпретації результатів експериментів.
8. Окреслити галузі застосування одно- та багаторядних алгоритмів МГУА.
9. Пояснити, в чому полягає ідея масової селекції при прогнозуванні процесів і явищ з використанням МГУА.
10. Навести структуру комбінованого алгоритму МГУА.
11. Пояснити особливості використання комбінованих критеріїв у алгоритмах самоорганізації моделей об'єктів.
12. Розкрити загальні принципи побудови гібридних інтелектуалізованих систем.
13. Навести особливості взаємодії ЕС і числових алгоритмів у комбінованих інтелектуальних системах.
14. Дати означення САІМ і розкрити етапи автоматизації управління експериментами.
15. Розкрити можливості реалізації інтелектуальної компоненти САІМ – пошукового модуля.
16. Назвати формальні основи організації експериментів з ІМ.
17. Дати поширене обґрунтування механізму задавання вхідного слова в автоматизованій ІМ.
18. Розкрити зміст поняття «алгоритм ІМ».
19. Охарактеризувати ГКІС як об'єкт інтелектуалізованого модулювання, планування та керування.
20. Розкрити особливості скінченно-автоматної інтерпретації ГКІС.
21. Охарактеризувати особливості функціонування, притаманні ГКІС.
22. Перелічити і обґрунтувати вимоги до моделі ГКІС.
23. Дати загальну оцінку основних методів моделювання ГКІС з урахуванням властивостей останніх.
24. Дати і прокоментувати означення дискретно-подійної системи.
25. Охарактеризувати клас комбіновано-гібридних систем і розкрити особливості їх моделювання.





## СІТКИ ПЕТРІ ЯК БАЗОВИЙ АПАРАТ МОДЕЛЮВАННЯ І ПРОЕКТУВАННЯ ПРОЦЕСІВ ГКІС

---

---

*Математика – найбільш довершений спосіб  
водити самого себе за ніс.*

Альберт Ейнштейн

### 7.1. Класична модель сіток Петрі

**Класифікація та структура процесів ГКІС.** Виробничі процеси дискретної ГКІС різняться виконуваними функціями (проекткування, нормування робіт, розробка технологій виготовлення виробів, складання, транспортування, обробка інформації тощо), використанням технологій (ручна або машинна обробка документів, шліфування, фрезерування, лиття та ін.), а також організацією виконання (базується на застосуванні потокових методів індивідуальної обробки/обслуговування, групової технології і т. ін.). Спільним у таких процесах є наявність упорядкованих у просторі і часі наборів технологічних операцій, прийомів, необхідних для підготовки, освоєння та випуску окремих видів виробів.

У зв'язку з багатостадійністю і складністю опису процесів ГКІС доцільно їх моделювання зводити до декомпозиції до такого рівня, на якому кожен елемент можна адекватно описати у вигляді типових блоків. При цьому ієрархічні структури складних ГКІС можуть об'єднувати математичні моделі типових елементів процесів ГКІС.

Таким чином, при декомпозиції складні процеси зручно розподіляти на блоки, які являють собою подібні (тобто типові) підпроцеси, критерієм виділення яких є спільність подання матеріальних зв'язків операцій. Тоді блоки та взаємодії між ними повинні подаватися так, щоб кожен структурний елемент у моделі являв собою сукупність інших елементів, пов'язаних деякими відношеннями із системи типових матеріальних зв'язків. Необхідність синтезу безтупикових алгоритмів керування, ефективність подання таких особливостей функціонування ГКІС, як дискретність, асинхронність, паралельність, а також можливість пониження вимірності моделей, що застосовуються, передбачають використання для цих цілей сіткових методів, зокрема, апарату сіток Петрі (СП). Проте СП загального вигляду непридатні для розв'язання поставлених задач у запропонованій інтерпретації через те, що методи аналізу властивостей досліджуваних сіткових моделей за деревом досяжності та рівнянням стану досить трудомісткі і в цьому випадку неефективні (особливо при значних розмірностях отримуваних моделей). Тому для наведених задач формалізації процесів і синтезу алгоритмів керування диспетчеризуванням матеріальних та інформаційних потоків у ГКІС необхідно застосовувати поширення апарату СП (або відповідні модифікації СП) для [74]:

- ефективного відображення проходження різноманітних матеріальних та інформаційних потоків;

- можливості проведення декомпозиції сіткового опису у блочно-ієрархічну структуру;
- можливості поблокового опису типових підпроцесів;
- формалізованого відображення конвеєрності, циклічності, конкуренції обробляючих ресурсів ГКІС та врахування впливу виробничих операцій на стан ресурсів;
- можливості синтезу моделей процесів ГКІС та алгоритмів керування із заданими властивостями відсутності тупиків або використання формальних методів перевірки моделей та механізмів коригування їх опису;
- надання моделі властивості гнучкого коригування часового навантаження та відображення динамічного режиму функціонування процесів ГКІС.

Процес функціонування ГКІС доцільно подавати ієрархією типових підпроцесів, вид яких визначається наявністю відповідних типових відношень. Тоді за заданими відношеннями зв'язку і способу проходження цього процесу можна ідентифікувати ГКІС будь-якої складності.

Усі типи відношень можна розподілити на дві групи: ті, що накладають, і такі, що не накладають обмежень на порядок реалізації операцій в процесі у вигляді причинно-наслідкових зв'язків [125]. *Перша група* поділяється на відношення, в яких реалізація однієї операції виключає можливість реалізації іншої (відношення *альтернативи* –  $R_A$ ), і такі, в яких не виключає, проте визначено строгий порядок реалізації операцій (відношення *наступності* –  $R_H$ ).

*Друга група* розподіляється на відношення, які допускають одночасне виконання операцій (відношення *паралелізму* –  $R_P$ ), та будь-який порядок реалізації, окрім одночасного (відношення *конкуренції* –  $R_K$ ).

Тоді виділяють такі процеси: *послідовні лінійні* – ЛП<sub>С</sub>; *паралельні лінійні* – ЛП; *альтернативні* – А; *паралельні з конкуренцією* – ПК.

Крім того, залежно від способу перебігу та організації можна виділити *циклічні* Ц та конвеєрні К процеси.

На базі цих понять запропоновано класифікатор складальних процесів, наведений у табл. 1.3 [125], для яких можна сформулювати концептуальну модель такого вигляду:

$$\Pi_{\text{ГКІС}} = (D, M, O, G, \tau, S_D, S_M, TM),$$

де  $D = \{D_6, D_K, D_T, D_V\}$  – множина об'єктів виробництва (базові деталі, комплектуючі, тара та інструмент);  $M = \{M_{\text{СК}}, M_{\text{ТР}}, M_{\text{ЗБ}}\}$  – множина обробляючих ресурсів ГКІС (складання, транспортування, зберігання);  $O = \{O_{\text{СК}}, O_{\text{НЛ}}, O_{\text{ТР}}, O_{\text{ЗБ}}\}$  – множина операцій процесу (складання, налагодження, транспортування, зберігання);  $\tau = \{\tau_{\text{СК}}, \tau_{\text{НЛ}}, \tau_{\text{ТР}}, \tau_{\text{ЗБ}}\}$  – часові характеристики відповідних операцій;  $S_D : D \rightarrow M$  – стан об'єктів виробництва;  $S_M : M \rightarrow D$  – стан обробляючих ресурсів;  $G = (O, W)$  – граф структурної організації процесу;  $W = \{R_H, R_A, R_P, R_K\}$  – система типових відношень зв'язку операцій;  $TM : \prod_{j \in J} O_{\text{СК}} \rightarrow D_6$  – технологічні маршрути виготовлення (в цьому разі складання), причому  $J = \{a_j \mid a_j \in \{1, 2, 3, \dots\}, j = 1, |D_6|\}$ .

Виходячи з наведеного встановлюються межі формалізації предметної області ГКІС (рівень деталізації подається сітковою моделлю процесу) при проектуванні алгоритму управління; визначається склад елементів, які входять у модель; конкретизується зміст та ідентифікується належність процесу, що моделюється, до типового.

Чи не найпоширенішими на сьогодні і не найефективнішими й потужними в ній наведених завдань є формалізми сіток Петрі і їх розширення (модифікації) як універсального, так і конкретного (прикладного) застосування. Справді, останні займають особливе, проміжне положення між методами абстрактного (скінченні автомати, машини Тьюринга і Поста [70], логічні та семантичні сітки тощо) і прикладного (системи GPSS [195], VHDL [200]) моделювання. Це пояснюється тією обставиною, що класичний апарат СП є еквівалентним скінченним автоматам і логічним сіткам, тоді як різноманітні модифікації у вигляді ієрархічних [70], розфарбованих [173], безконтактних [192] СП з універсальними властивостями поширюються на клас об'єктів моделювання (ОМ) різної природи. Останніми роками теорія СП набула подальшого розвитку у вигляді об'єктно-орієнтованих модифікацій через необхідність адекватної інтерпретації явищ, які відбуваються при функціонуванні ГКС, і відображення ієрархічності побудови інтегрованих в них підсистем (технологічної, керуючої, підготовки виробництва, планувальної та інформаційної), а також паралельності процесів, які відбуваються в ГКІС. Основними модифікаціями СП при цьому є інгібіторні [139], часові самонавантажувальні [125; 140], прості і нечіткі логічні [140], інтелектуалізовані [172] сітки.

Отже, різноманітні розширення класичних СП-методів дають змогу досліджувати саме кількісні і якісні характеристики складних систем, у тому числі розв'язувати питання підвищення ефективності функціонування через механізми прийняття рішень щодо спрямованості дії на входи останніх.

Розглянемо ретроспективно деякі основні модифікації СП-методів, які спрямовані, з одного боку, на адаптацію їх до ефективного розв'язання прикладних задач моделювання ГКІС, а з іншого — на підвищення власне їх виразних спроможностей та моделюючої потужності.

**Класична (базова) СП.** В СП як базова інформація використовуються дані про логічний взаємозв'язок подій, а причинно-наслідковий зв'язок подій задається множиною відношень виду «умова (в СП, наприклад, це позиція  $P$ ) — подія (наприклад, перехід  $T$ )». Послідовності подій відображаються спрацьовуванням переходів. Виконання якоїсь умови пов'язане з появою маркера у відповідній позиції. Погодження про правила спрацьовування переходів є засобом відображення концепції причинно-наслідкових зв'язків між умовами та подіями в ОМ.

Для ілюстрації можливостей подання процесів в об'єктах, що моделюються СП, як приклад розглянемо структуру деякого *тестового гнучкого виробничого модуля* (ТГВМ) у складі ГКІС, який містить (рис. 7.1, а): нагромаджувач заготовок (НЗ); нагромаджувач виробів (НВ); промисловий робот (Р1), який виконує операції завантаження-розвантаження; двопозиційний стіл-маніпулятор виробів (МВ) з позиціями П1 і П2; робот-складальник (Р2), який виконує складання виробів; нагромаджувач деталей (НД). Умовно припустимо, що всі нагромаджувачі мають ємність, що дорівнює одиниці. Відповідний структурі ТГВМ граф моделюючої СП з випадковими навантаженнями позицій зображено на рис. 7.1, б, а формальне визначення при випадковому навантаженні, зображеному на графі СП, подано на рис. 7.1, в.

**Означення 7.1.** Сітка Петрі — це четвірка [54; 139]

$$N = (P, T, F, H), \quad (7.1)$$

де  $P = \{p_1, p_1, \dots, p_n\}$ ,  $T = \{t_1, t_2, \dots, t_m\}$  — скінченні непорожні множини позицій (станів, місць) і переходів (подій) відповідно, причому  $P \cap T = \emptyset$ ;  $F: P \times T \rightarrow \{0, 1, 2, \dots\}$ ,  $H: T \times P \rightarrow \{0, 1, 2, \dots\}$  — вхідна та вихідна функції відповідно (це функції інцидентності) (рис. 7.1), причому  $F$  і  $H$  називають інцидентними функціями.

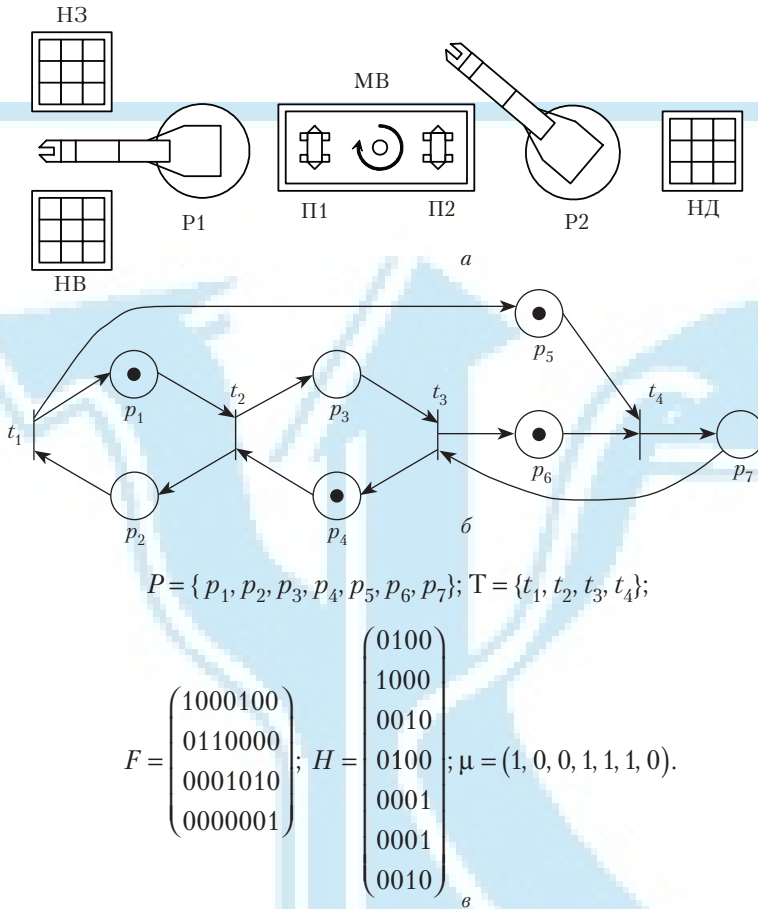


Рис. 7.1. Тестовий гнучкий виробничий модуль (а), граф відображаючої сітки Петрі  $N = (P, T, F, H)$  з маркуванням  $\mu$  (б) і формальне означення СП (в)

Вхідні позиції переходу  $t_j$  об'єднуються у множину його попередників  $PRE(t_j) = \{p_i \in P \mid H(p_i, t_j) \geq 1\}$ . Вихідні позиції переходу  $t_j$  об'єднуються у множину його послідовників  $POST(t_j) = \{p_i \in P \mid I(t_j, p_i) \geq 1\}$ .

Дослідження об'єкта моделювання за СП значною мірою ґрунтується на формальному визначенні (рис. 7.1, в), проте для ілюстрацій понять теорії СП набагато зручніше використовувати їх графічне зображення (граф рис. 7.1, б).

Графічним поданням СП є дводольний орієнтований граф, що має два типи вузлів — позицій і переходів, які спрямовано з'єднуються дугами таким чином, що дуга від  $p_i$  до  $t_j$  визначає цю позицію як вихідну. В СП допускається існування кратних вхідних і вихідних дуг (відповідно,  $\#(p_i, F(t_j))$  та  $\#(p_i, H(t_j))$ ). Функції входу і виходу задаються множиною одинарних дуг-ребер, що зв'язують вершини мультиграфа, а саме:  $(t_j, p_i)$  та  $(p_i, t_j)$ .

**Означення 7.2.** Граф сітки Петрі — дводольний орієнтований мультиграф, в якому позиції позначаються колами, переходи — планками, а вхідні та вихідні функції — орієнтованими дугами, кратність яких відповідає значенню функції (рис. 7.1, в).

**Означення 7.3.** Маркування сітки Петрі  $N = (P, T, F, H)$  є функція, яка відображає множину позицій  $P$  у множину невід’ємних цілих чисел  $\mathbf{N}$ ;  $\mu : P \rightarrow \mathbf{N}^n$ ,  $n = |P|$ .

Тобто маркування в СП може зображатися  $n$ -вимірним вектором  $\mu = (\mu(p_1), \mu(p_2), \dots, \mu(p_n))$ , де  $\mu(p_i)$  — маркування  $i$ -ї позиції.

**Означення 7.4.** Маркована сітка Петрі (МСП) — це сукупність СП  $N = (P, T, F, H)$  і маркування  $\mu$ , яка може бути записана у вигляді  $N_\mu = (P, T, F, H, \mu)$ .

Маркування СП, яке показують графічно маркерами-фішками в позиціях, може змінюватися в результаті спрацьовування (запуску) переходу. Останній визначається запущеним, якщо він є дозволеним, тобто коли  $\mu(p_i) \geq \#(p_i, F(t_j))$  для всіх  $p_i$ , які входять в  $F(t_j)$ . Результатом запуску будь-якого дозволеного переходу  $t_j$  є утворення нового маркування  $\mu'$ . Рух маркерів в СП розпочинається за наявності бодай одного збудженого переходу в початковому маркуванні. Якщо в СП є декілька одночасно збуджених переходів, то порядок їх спрацьовування є невизначеним, і тоді можна сформулювати кілька послідовностей спрацьовування переходів. Тому дуже важливим для аналізу роботи СП є поняття *досяжності* маркування  $\mu'$  безпосередньо з маркування  $\mu$  (тобто  $\mu \xrightarrow{t_i} \mu'$ ), що є наслідком дії, в результаті якої після спрацьовування переходу  $t_i$  з усіх його позицій виключається  $F(p_i, t_j) = \#(p_i, F(t_j))$ , а в кожен вхідну позицію додається  $H(t_j, p_i) = \#(p_i, H(t_j))$  маркерів [125; 139].

**Означення 7.5.** Безпосередньо досяжне маркування  $\mu'$  для сітки  $N = (P, T, F, H)$  з маркуванням  $\mu$  — це таке маркування, для якого  $\exists t_j \in T : \delta(\mu, t_j) = \mu'$ . Якщо  $\mu'$  безпосередньо досяжна з  $\mu$ , а  $\mu''$  — з  $\mu'$ , то кажуть, що  $\mu''$  досяжна з  $\mu$ .

Отже, функціонування СП — це послідовна зміна маркувань при спрацьовуванні збуджених переходів, що визначається *функцією наступності*  $z(\mu, t_i)$ , яка вказує на можливість безпосереднього переходу від  $\mu$  до  $\mu'$ . Можливі варіанти функціонування СП описує *граф досяжності* зі спрямованою дугою  $\mu \xrightarrow{t_i} \mu'$ , де  $t_j \in T$ . Якщо ж для деякого маркування жоден з переходів спрацювати не може, то таке маркування називається *тупиковим*. Перевірка *умов досяжності* (або коректності складення СП) є дуже трудомісткою задачею, яка і за умови її комп’ютерного розв’язання потребує значних часових і машинних витрат. Деякі з наведених далі розширень СП-апарату стосуються саме проблеми створення коректних СП-моделей.

**Означення 7.6.** Множина *досяжності*  $R(N, \mu)$  для СП  $N = (P, T, F, H)$  з маркуванням  $\mu$  є найменшою множиною маркувань, визначених таким чином:  $\mu \in R(N, \mu)$ ; якщо  $\mu' \in R(N, \mu)$  і  $\exists t_j \in T : \delta(\mu', t_j) = \mu''$ , то  $\mu'' \in R(N, \mu)$ .

Детальніше особливості побудови дерева досяжності СП і можливості дослідження властивостей керування об’єктами, що моделюються, розглянуто у підрозд. 7.8.

Інші модифікації, що розглядаються далі, є проблемно-орієнтованими і спрямовані на дослідження властивостей ОМ в умовах обмеженості ресурсів або їх неефективного використання, а також за наявності елементів  $X = \{\mu_i | i = \overline{1, m}\}$ ;  $Y = \{y_j | j = \overline{1, n}\}$  невизначеності.

Не вдаючись у детальний опис особливостей формалізованого подання процесів у ГКІС за допомогою модифікацій СП-апарату, як приклад коротко розглянемо спрощену формальну постановку задачі керування в ГКІС. Нехай модель ГКІС задано у вигляді СП, поданої залежністю (1), стан якої визначається маркуванням  $\mu$ . Керування в ГКІС можна задати функцією стану  $Y = f(X)$ , де  $X$  і  $Y$  — множини станів ОМ і операторів керування переходами сітки залежно від маркування її позицій відповідно. Тоді алгоритм керування в ГКІС можна задати у формі модифікованої СП, а саме:  $N_a = (N, \mu, \Phi, Y, f)$  та умов:

$$\forall p \in P, \forall \mu \in R(N): \mu(p) \leq 1;$$

$$\forall \mu_i, \mu_j \in R(N): \mu_i \rightarrow \mu_j;$$

$$\forall t \in T, \exists \mu_k, \mu_i \in R(N): \mu_k \xrightarrow{t} \mu_i,$$

причому  $N$  – є СП, що визначає структуру алгоритму керування;  $\mu$  – множина станів ОМ;  $\varphi: T \rightarrow 2^\mu$  – функція помітки переходів, яка ставить в однозначну відповідність підмножину станів ОМ;  $2^\mu$  – множина всіх підмножин  $\mu$ ;  $Y$  – множина операторів керування;  $f: \mu \rightarrow Y$  – функція, яка кожному маркуванню ставить в однозначну відповідність оператор або сукупність операторів алгоритму керування.

Перевагою такого підходу є можливість формалізації процесів планування і керування на кожному з рівнів ієрархії структури задач ГКІС на єдиній методологічній і математичній основі.

Перед розкриттям сутності нових модифікацій зупинимось детальніше на деяких інтерпретаціях класичного СП-методу.

**Відображення властивостей об'єкта керування в термінах класичної СП.** Для дослідження процесу функціонування об'єктів шляхом визначення властивостей СП-моделі і розв'язання задач, пов'язаних з їх функціонуванням, необхідно застосовувати методи аналізу СП. Існує два основних методи [54; 83]:

- аналіз СП за деревом досяжності;
- аналіз СП із застосуванням матричних рівнянь.

**Побудова дерева досяжності.** Розглянемо детальніше перший метод як більш придатний для розв'язання задач, пов'язаних з пошуком критичної операції і визначенням найбільш ефективної послідовності спрацьовування переходів.

**Означення 7.7.** *Дерево досяжності* – це спрямований граф, вершини якого є маркуваннями СП, а дуги – переходами, в результаті спрацьовувань яких одне маркування змінюється на інше.

Дерево досяжності може містити вершини чотирьох типів:

- кореневу – вершину, що відповідає початковому маркуванню СП;
- внутрішню – вершину, що має вхідну і хоча б одну вихідну дугу;
- термінальну – вершину, що відповідає тупиковому маркуванню;
- дублюючу – вершину, для якої відповідне їй маркування дорівнює маркуванню, що відповідає іншій вершині дерева досяжності. При побудові дерева досяжності використовується ще поняття граничної вершини – вершини, для якого ще не побудовані вихідні дуги.

Маркування, що відповідає вершині дерева досяжності  $x$ , визначається як  $\mu[x]: P \rightarrow N_\omega^n$ ,  $n = |P|$ , а кількість фішок у позиції  $p_i \in P$  при маркуванні  $\mu[x]$  позначається як  $\mu[x]_i$ .

**Означення 7.8.** *Множина*  $N_\omega = N \cup \{\omega\}$  – розширена множина натуральних чисел, де  $\omega$  – елемент, що задовольняє для будь-якого  $n \in N$  такі властивості:

$$\omega > n;$$

$$n + \omega = \omega + n = \omega + \omega = \omega - n = \omega - \omega = \omega;$$

$$\omega \cdot n = n \cdot \omega = \omega;$$

$$\omega \times 0 = 0 \times \omega = 0.$$

Алгоритм побудови дерева досяжності, наведений нижче, має ітераційну структуру і ґрунтується на тому, що для кожного маркування шукаються всі маркування, які з нього безпосередньо досяжні.

**Алгоритм 7.1:**

*Крок 1.* Оголосити початкове маркування СП  $\mu$  кореневою вершиною.

*Крок 2.* Взяти як поточну вершину  $x$  кореневу  $\mu[x] = \mu$ .

*Крок 3.* Якщо дерево має вершину  $y$ , що не є граничною, для якої виконується умова  $\mu[x] = \mu[y]$ , то вершина  $x$  — дублююча.

*Крок 4.* Якщо  $x$  — дублююча вершина, перейти до кроку 9.

*Крок 5.* Якщо для маркування  $\mu[x]$  виконується умова  $\forall t_j \in T : \delta(\mu, t_j) = \emptyset$ , тобто жоден перехід не є збудженим, то  $x$  — термінальна вершина.

*Крок 6.* Якщо  $x$  — термінальна вершина, перейти до кроку 9.

*Крок 7.* Для кожного дозволеного переходу  $t_j \in T, \exists \mu' : \delta(\mu[x], t_j) = \mu'$  створити нову вершину  $z$  дерева досяжності. Маркування  $\mu' = \mu[z]$ , пов'язане з цією вершиною, для всіх  $i = 1, \dots, n, n = |P|$  визначається в такий спосіб:

- якщо  $\mu[x]_i = \omega$ , то  $\mu[z]_i = \omega$ ;
- якщо на шляху від кореневої вершини до  $x$  існує вершина  $y$  така, що  $\mu[y] < \delta(\mu[x], t_j)_i$  і  $\mu[y]_i < \delta(\mu[x], t_j)_i$ , то  $\mu[z]_i = \omega$ ;
- інакше  $\mu[z]_i = \delta(\mu[x], t_j)_i$ .

*Крок 8.* Вершина  $x$  оголошується внутрішньою.

*Крок 9.* Якщо дерево досяжності містить граничні вершини, то одна з них вибирається як поточна вершина  $x$ , інакше перейти до кроку 11.

*Крок 10.* Перейти до кроку 3.

*Крок 11.* Закінчити роботу алгоритму.

Доведено [83], що дерево досяжності скінченне для скінченної СП і, отже, описаний алгоритм закінчує роботу.

Символ  $\omega$  у вершинах дерева досяжності вказує на необмежені позиції.

Приклад дерева досяжності СП-моделі ТГВМ (рис. 7.1, а) наведено на рис. 7.2.

**Аналіз СП за допомогою дерева досяжності.** Використовуючи дерево досяжності, можна визначати такі властивості СП, як безпека, обмеженість, консервативність, живучість, розв'язувати задачу покривності і в деяких випадках розв'язувати задачу власне досяжності.

**Умови безпеки та обмеженості.** При моделюванні процесу функціонування ГКІС необхідно уникати конфліктів, пов'язаних з перевищенням обмежень на ресурси, які використовуються в системі. Наприклад, міжопераційний нагромадjuвач не може містити кількість деталей, більшу за його місткість, ПР не здатен маніпулювати декількома об'єктами в один і той самий час, обчислювальний пристрій може оперувати числами, що не перевищують заданої кількості розрядів і т. ін.

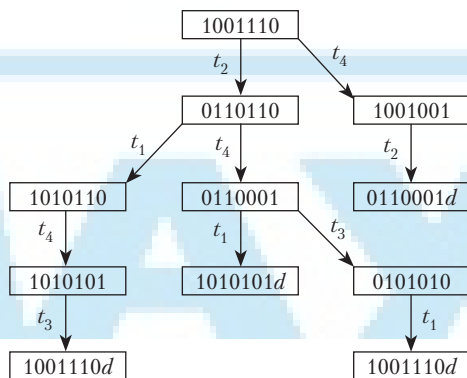


Рис. 7.2. Дерево досяжності СП, зображеної на рис. 7.1. Тут  $d$  — дублюючі вершини

Виявити наявність подібних конфліктних ситуацій можна шляхом дослідження СП-моделі ГКІС на володіння властивостями *обмеження й безпеки*.

**Означення 7.9.** *Позиція*  $p_i \in P$  сітки  $N = (P, T, F, H)$  з маркуванням  $\mu \in$  обмеженою, якщо  $\exists k \in N$ , для якого  $\forall \mu' \in R(N, \mu) : \mu'(p_i) \in \{0, 1, \dots, k\}$ , причому в цьому випадку позиція називатиметься  $k$ -обмеженою. Сітка Петрі є обмеженою, якщо обмежена кожна з її позицій.

**Означення 7.10.** *Сітка Петрі*  $k$ -обмежена, якщо кількість фішок у будь-якій позиції ніколи не перевищує  $k$ .

**Висновок 7.1.** *Сітка Петрі називається безпечною*, якщо вона є 1-обмеженою.

**Означення 7.11.** *Сітка Петрі є обмеженою тоді і тільки тоді*, коли у вершинах дерева досяжності відсутній символ  $\omega$ .

Для того щоб визначити межу обмеженості, необхідно перебрати скінченну кількість вершин (тому що дерево досяжності скінченне) і визначити найбільше значення серед компонентів маркувань, що визначають кількість маркерів у позиції. Якщо ця межа дорівнює 1, то СП – безпечна.

Отже, властивість безпеки є окремим випадком обмеженості при  $k = 1$ .

Інакше кажучи, СП є безпечною, якщо в жодній з позицій при будь-якому маркуванні не з'явиться більш як однієї фішки. У загальному випадку не можна адекватно промоделювати реальний об'єкт необмеженою СП, бо це означатиме відсутність обмежень на ресурси в системі.

У прикладі, наведеному на рис. 7.1, *а*, коректна робота ТГВМ може бути забезпечена тільки тоді, коли його СП-модель (рис. 7.1, *б*) є безпечною, оскільки на кожній позиції маніпулятора МВ не може перебувати більше одного об'єкта, а місткість усіх накопичувачів дорівнює одиниці.

Отже, кожна «позиція-ресурс»  $p_r \in \bigcup_{i=0}^{\rho} P_{ri}$  і пов'язані з нею через вхідні та вихідні переходи «позиції-операції»  $p_c \in \tilde{P}_c$  утворюють у *типовій розфарбованій сітці Петрі* (ТРСП) замкнені кільцеподібні підсітки, які є обмеженими внаслідок виконання умов:

$$\Delta = \sum_{\omega \in \Omega} \sum_{p \in P_c \cup \{p_r\}} \mu(p, \omega) = \sum_{\omega \in \Omega} \sum_{p \in P_c \cup \{p_r\}} \mu_c(p, \omega).$$

Якщо  $\Delta = 1$ , то позиція  $p_c \in \tilde{P}_c$  є безпечною. Тому для забезпечення першої умови коректності необхідно для відповідних «позицій-операцій» (складання, налагоджування, транспортування тощо) ввести «позиції-ресурси» з безпечним початковим маркуванням тільки за одним з кольорів, а для решти «позицій-операцій» (зокрема зберігання) – з обмеженням за загальною кількістю маркерів початковим маркуванням.

**Умова консервативності.** При моделюванні ГКІС за допомогою СП можна інтерпретувати фішки як різні ресурси. В цьому разі корисним може виявитись дослідження моделі на консервативність. Зазначимо спочатку, що СП не може бути консервативною, якщо вона необмежена, тобто в якійсь із вершин відповідне маркування містить символ  $\omega$ .

**Означення 7.12.** *Сітка Петрі є консервативною*, якщо вона не втрачає і не породжує фішок, а просто пересуває їх.

У разі обмеженої СП за допомогою дерева досяжності легко визначити властивість консервативності стосовно вектора зважування  $\mathbf{w}$ . Необхідно для кожного маркування визначити зважену суму фішок  $s_l = \sum_{i=1}^n w_i \mu_l(p_i)$ ,  $n = |P|$ ,



$l = 1, \dots, r$ ,  $r = |R(C, \mu)|$ . Якщо виконується умова  $\forall l, m : s_l = s_m$ ,  $l, m = 1, \dots, |R(C, \mu)|$ , то СП консервативна стосовно вектора зважування  $\mathbf{w}$ .

Дерево досяжності можна також використати для визначення вектора зважування  $\mathbf{w}$ , при якому СП є консервативною. Для цього складається система лінійних алгебраїчних рівнянь  $s_l = \sum_{i=1}^n w_i \mu_l(p_i)$ , обмежена умовами  $w_i \geq 0$ ,  $i = 1, \dots, n$ , що складається з  $r$  рівнянь з  $n + 1$  невідомими, у результаті розв'язання якої можна одержати шуканий вектор зважування  $\mathbf{w}$ . У загальному випадку компоненти знайденого вектора  $\mathbf{w}$  будуть раціональними числами, але для одержання цілочислового розв'язку їх можна помножити на таке число  $w$ , для якого виконуватиметься умова  $\forall i : w \cdot w_i \in N$ .

Через те що в сітці може одночасно перебувати декілька видів ресурсів, до того ж ті самі ресурси можуть перетворюватися в інші, необхідно також визначити властивість консервативності щодо вектора зважування.

**Означення 7.13.** Сітка Петрі  $N = (P, T, F, H)$  з маркуванням  $\mu$  є строго консервативною щодо вектора зважування:

$$W = (w_1, w_2, \dots, w_n), n = |P|, w_i \geq 0, \text{ якщо } \forall \mu' \in R(N, \mu) : \sum_{i=1}^n w_i \mu'(p_i) = \sum_{i=1}^n w_i \mu(p_i).$$

Таким чином, якщо, наприклад, заготовка, що відповідає фішці в позиції 1, у процесі виготовлення виробу розрізється на дві частини, що відповідають фішкам у позиції 2, то, застосувавши вектор зважування  $(2, 1, \dots)$ , можна говорити про збереження властивості консервативності.

Строго консервативні СП є консервативними стосовно вектора зважування  $(1, 1, \dots, 1)$ .

У розглянутому прикладі (рис. 7.2), оскільки виріб виготовляється шляхом складання заготовки і деталі, СП повинна бути консервативною щодо вектора зважування  $(1, 2, 1, 2, 1, 1, 2)$ . З дерева досяжності видно, що умова консервативності виконується, тому що  $\forall i : s_i = 5$ .

**Умова покривності.** Задача покривності — це остання задача, однозначно розв'язувана із застосуванням дерева досяжності.

**Означення 7.14.** Маркування  $\mu''$  покриває маркування  $\mu'$ , якщо виконується умова  $\mu'' \geq \mu'$ .

Розв'язується задача простим перебиранням усіх вершин. Маркування, що відповідають вершинам, порівнюються з маркуванням  $\mu'$ , для якого зважується задача покривності. При цьому слід пам'ятати, що  $\forall n \in N : \omega > n$ . За деревом досяжності також може бути розв'язана задача пошуку послідовності запуску переходів, що приводить СП до маркування, яке покриває маркування  $\mu'$ .

**Умова досяжності.** У загальному випадку дерево досяжності не можна використовувати для розв'язання задачі досяжності й інших задач, що впливають з неї. Пов'язано це з існуванням символу  $\omega$ , який не дає змоги довідатися про реальну кількість фішок у маркуванні, що відповідає вершині дерева досяжності, а отже, задачі досяжності певного маркування розв'язати неможливо.

Проте існують окремі випадки СП, для яких задача досяжності є розв'язуваною при використанні дерева досяжності. *По-перше*, це випадок, коли СП обмежена, а, як відомо, дерево досяжності обмеженої СП не має символу  $\omega$  в маркуваннях, що відповідають вершинам, і, отже, всі маркування визначені, що й дає змогу однозначно розв'язувати задачу досяжності. *По-друге*, це випадок, коли шукане маркування зустрічається в дереві досяжності в явному вигляді без символу  $\omega$ .

Крім того, за допомогою дерева досяжності можна зробити висновок про те, що задача досяжності заданого маркування не має розв'язку. Для цього необхідно довести, що для цього маркування не має розв'язку і задача покриття.

З викладеного можна зробити висновок: через те, що реальні системи зазвичай моделюються обмеженими СП, то для моделей таких систем (у тому числі розглянутого ТГВМ) задачу досяжності здебільшого можна розв'язати, використовуючи дерево досяжності.

**Умова живучості.** Визначити, чи може при функціонуванні ГКІС виникнути тупикова ситуація (спрощений приклад тупикової ситуації наведено на рис. 7.3), можна, дослідивши СП-модель на володіння властивістю живучості. Властивість живучості показує, чи може в ході функціонування СП виникнути стан, при якому якийсь перехід виявиться «непотрібним» для її подальшої роботи.

**Означення 7.15.** Перехід  $t_j \in T$  СП  $N = (P, T, F, H)$  з маркуванням  $\mu$  називається живучим, якщо  $\forall \mu' \in R(C, \mu), \forall p_i \in P, \exists \mu'' \in R(C, \mu') : \mu(p_i) \geq I(p_i, t_j)$ .

**Означення 7.16.** Сітка Петрі називається живучою, якщо живучі всі її переходи.

**Означення 7.17.** Тупиковим маркуванням СП  $N = (P, T, F, H)$  з маркуванням  $\mu$  називається маркування  $\mu' \in R(N, \mu) : R(N, \mu') = \{\mu'\}$ .

З означень 7.15 і 7.17 випливає, що необхідною умовою живучості СП є відсутність тупикових маркувань.

Очевидно, що в загальному випадку для нормального функціонування ГКІС її СП-модель повинна бути живучою.

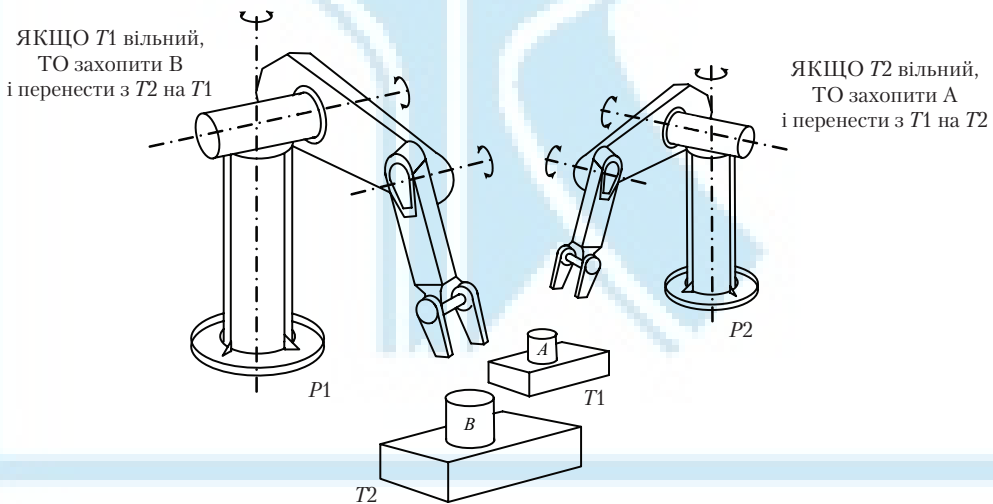


Рис. 7.3. Приклад тупикової ситуації

**Умова пріоритетності в СП.** Пріоритетні СП призначені для моделювання розв'язання конфліктних ситуацій, що виникають у системі, шляхом введення пріоритетів спрацьовування переходів.

**Означення 7.18.** Пріоритетні СП визначаються як  $N_{pr} = (N, pr)$ , де  $N = (P, T, F, H)$  – сітка Петрі, а  $pr : T \rightarrow \mathbf{N}$  – функція, яка надає переходам пріоритети їх спрацьовування.

Для спрацьовування з множини збуджених переходів  $\tilde{T}$  вибирається  $t_j$ , для якої справедливою є нерівність  $\forall t_i, t_j \in \tilde{T} : pr(t_j) > pr(t_i)$ .

Правило спрацьовування переходу пріоритетної СП збігається з правилом спрацьовування звичайної СП.

Таким чином, наведені в розд. 7 модифікації структурних перетворень СП-апарату і їх адаптація щодо подання процесів і можливості керування ними дають змогу будувати коректно сформовані СП-моделі, які задовольняють умови живучості і безпеки. Іншими словами, з впровадженням розширень класичного апарату СП отримано:

- **засіб моделювання**, універсально більш досконалий і продуктивний до ОМ різної природи;
- **набір формалізмів**, добре адаптований до інтерпретації явищ і подій, що відбуваються в ГКІС як на всіх стадіях їх життєвого циклу, так і на ієрархічних рівнях функціонування системи.

**Інгібіторні СП.** Для подання за допомогою СП процесів ГКІС необхідно мати засіб, який забороняє повторне ініціювання операцій під час їх виконання. Сітки, які володіють такою властивістю, називають безконтактними [192]. Ми скористаємося поняттям інгібіторних (заборонних) дуг, які перевіряють сітку на нульове маркування, а сітки з такими дугами мають назву *інгібіторних* [54; 139].

**Означення 7.19.** *Інгібіторна сітка Петрі (ІСП)*  $N_{\text{ін}} = (N, \bar{H}, \tau, \gamma)$  являє собою СП  $N = (P, T, F, H)$ , доповнену спеціальною функцією інцидентності  $\bar{H} : P \times T \rightarrow \{\emptyset, 1\}$ , яка вводить інгібіторні дуги для пар  $(p, t)$ , в яких  $\bar{H}(p, t) = 1$ .

Тоді правило спрацьовування переходу в ІСП має вигляд:

$$\forall p \in \bullet t : \mu(p) \geq H(p, t) \wedge \mu(p) \bar{H}(p, t) = 0, \quad (7.2)$$

де  $\bullet t$  — множина вхідних позицій переходу  $t$ .

**Розфарбовані СП.** В моделях реальних систем часто буває необхідно промоделювати об'єднання, дроблення та синхронізацію різних процесів, які відбуваються в системі. Крім того, при поданні сітками Петрі процесів ГКІС необхідно розрізняти маркери, що відповідають об'єктам матеріальних та інформаційних потоків у системі (дані, деталі, інструмент тощо). Для цього запроваджуються СП з множинами колірної досяжності, тобто промарковані фішками різних кольорів. Такі СП називаються *розфарбованими* СП [54; 139; 173].

**Означення 7.20.** *Розфарбована СП (РСП)* — це набір вигляду  $N_p = (N, \Omega, \lambda, \varphi, \mu_0)$ , де  $\Omega = \{\omega\}$  — непорожня скінченна множина кольорів маркерів;  $\lambda : (P \times \Omega) \times T \rightarrow \{0, 1\}$  та  $\varphi : T \times (P \times \Omega) \rightarrow (P \times \Omega)$  — функції розподілу кольорів маркерів за вхідними позиціями переходів сітки і вихідними позиціями сітки відповідно. Ці функції відповідно описують умови спрацьовування переходів і нове маркування сітки після цього спрацьовування.

Умова спрацьовування переходу  $t \in T$  за деяким маркуванням  $\mu$ :

$$\forall p_{i_k} \in \bullet t, \exists \omega_{j_k} \in \Omega : \mu(p_{i_k}, \omega_{j_k}) \geq H(p_{i_k}, t). \quad (7.3)$$

До РСП-сіток застосовні поняття *безпеки*, *досяжності* та *живучості*.

## 7.2. Об'єктно-орієнтоване на подання складних ОМ розширення базового апарату СП

Як уже зазначалося, при моделюванні складних ОМ типу ГКІС за допомогою СП-підходів виникають труднощі з перевіркою умов коректності побудови відображаючої властивості об'єкта сіткової моделі, причому задачі синтезу й аналізу СП-моделей великої розмірності можна зарахувати до задач на повне перебирання,

тобто до *NP*-складних (дерево досяжності експоненціально збільшується). Це, у свою чергу, утруднює виявлення і запобігання тупиковим ситуаціям (або «вузьким» місцям) у ГКІС. Для розв'язання цієї проблеми було запропоновано три основні групи розширень (модифікації) базового апарату СП, які полягають у впровадженні:

- правил структурного перетворення сіток на основі *типових сіткових блоків* (ТСБ) [139];
- типових складових елементів процесів (ТЕП);
- сіткових конструкцій (СК) [125].

Суть *першої групи модифікації* полягає у виділенні окремих фрагментів типових сіткових блоків (ТСБ), що описують послідовні, паралельні, альтернативні та безумовні виконання дискретних процесів. Далі, на основі обмеженої кількості ТСБ, в аналізі коректності функціонування (або планування чи керування) процесів ГКІС у сітковій інтерпретації за допомогою запропонованого підходу здійснюється перетворення сіткової структури моделі в ієрархічну систему ТСБ. Через коректність ТСБ за означенням аналіз СП-моделі у цілому полягає в побудові дерева досяжності маркувань для вершин ієрархії, що значно скорочує його тривалість.

*Друга група модифікації* має на меті розширити межі об'єктної орієнтації до розв'язання задач щодо відображення та поліпшення якісних і кількісних показників ГКС, а саме, продуктивності і надійності [46; 47] і базується на типових складових елементах процесів ГКІС, які коректно за означенням інтерпретуються або безпосередньо складом ТСБ, або їх простими сполученнями [125]. Тоді процедура синтезу/аналізу будь-якого процесу ГКІС полягатиме у деталізації процесу та ідентифікації виділених ТЕП з подальшою їх інтерпретацією через ТСБ у висхідному (для задач синтезу) і низхідному (для задач аналізу) режимах. Введення ТЕП дає змогу не тільки значно скоротити часові і машинні витрати на верифікацію коректності СП-моделі, а й розширити можливості СП-підходів щодо розв'язання задач автоматизованого керування в ГКІС.

Основою нововведень *третьої групи модифікації* є запровадження СК для ієрархічних перетворень в СП-моделі, що створює можливості автоматизованого синтезу структур коректних алгоритмів планування, керування в ГКІС. Введенням предикатів розвитку цієї модифікації дає змогу формувати умови спрацьовування переходів СП-моделі при розв'язанні конфліктних ситуацій. Як результат, одержуємо сітку, яка адекватно відображає алгоритм взаємодії компонентів у ГКІС, тобто забезпечує реалізацію конкретної стратегії дій (при цьому дерево досяжності не будується, а отже, ще більшою мірою знижується трудомісткість задач синтезу/аналізу складних ОМ).

### 7.3. Загальні підходи до структурного перетворення СП

**Задача автоматизації наскрізного циклу синтезу керуючого алгоритму.** Сіткові методи аналізу та синтезу процесів ГКІС дають змогу розв'язувати задачу автоматизації наскрізного циклу синтезу керуючого алгоритму: визначення класу процесу — вибір (модифікація) сіткової моделі процедури керування — створення керуючого алгоритму. Реалізація такого циклу передбачає розв'язання декількох підзадач, найважливішими з яких є:

- розробка методів автоматизованого синтезу сіткової моделі процесів у формі, зручній для машинної обробки;

- створення ефективних методів синхронізації конкуруючих процесів, тобто методів, які становлять основу побудови коректних (наприклад, таких, що не дозволяють виникнення блокувань керуючої системи) процедур керування системами в режимі реального часу;
- підготовка засобів автоматизованої оцінки та відбирання правил вибору пріоритетів, які використовуються в процесі формування керуючих алгоритмів.

Кожну з підзадач ієрархії, у свою чергу, можна подати новою сукупністю підзадач. При цьому *моделюючи потужність* сіток Петрі повинна забезпечувати окремий (поблоковий) опис послідовних і паралельних процесів і відображувати їх ієрархічну структуру, тоді як *розв'язувальна спроможність* – ефективне розпізнавання безпеки та живучості. Отже, основним напрямком модифікації СП є запровадження поняття ієрархії в теорію сіток і створення відповідного математичного апарату. Крім того, важливе місце займають пофарбовані СП, теорію яких створив датський професор Курт Йенсен [173; 174]. Проте серйозною перешкодою для застосування пофарбованих СП є розмірність сітки – адже навіть для порівняно невеликих систем побудова сітки і особливо її аналіз є складною задачею, яка часто не має розв'язку. Проблема розмірності не є специфічною для СП, вона властива для всіх існуючих мов моделювання.

Розв'язання проблеми розмірності в СП стало можливим у результаті теоретичних розробок сіток високого рівня та розвитку програмного забезпечення для підтримки великих ієрархічних конструкцій [170], які дають змогу аналізувати структури великих пофарбованих СП як набір взаємопов'язаних підсіток.

Ієрархічні конструкції призначені насамперед для опису та аналізу процесів, які відбуваються у великих системах реального світу. Вони не розширюють моделюючої потужності пофарбованих СП і через це, на перший погляд, їх запровадження здається просто графічною зручністю. Проте з токи зору теорії це несправедливо. На практиці для побудови моделі великої системи необхідно розробляти структуровані пакети, які дають змогу враховувати проблему розмірності.

Ієрархічні сіткові моделі володіють такою важливою властивістю, як *виконувальність*. Це означає, що мова моделювання дає змогу підтримувати: зв'язок між складовими підсітками згідно із графом відношень; дві стратегії побудови сіткових моделей – зверху до низу та з низу доверху.

**Означення 7.21.** Сітковим блоком  $N_G$  називається РСП-сітка  $N_p$ , для якої  $T_c = T = T_c^* \cup T_c^{дж} \cup T_c^{ст}$  – множина переходів, причому  $T_c^* = \{t^* \mid \exists p_k, p_i \in P_c : F_c(p_k, t^*) = 1 \cap H_c(t^*, p_j) = 1\}$  – множина внутрішніх переходів сітки;  $T_c^{дж} = \{t^{дж} \mid \forall p_k \in P_c : F_c(p_k, t^{дж}) = \emptyset \cap H_c(t^{дж}, p_k) = 1\}$  – множина джерел-витоків;  $T_c^{ст} = \{t^{ст} \mid \forall p_k \in P_c, \exists p_k \in P_c : H_c(t^{ст}, p_k) = \emptyset \cap F_c(p_k, t^{ст}) = 1\}$  – множина стоків;  $\Omega_c = \Omega = \{\omega_l \mid l = \overline{1, r}\}$  – множина кольорів маркерів; функції  $F_c = F$ ,  $H_c = H$ ,  $\lambda_c = \lambda$ ,  $\varphi_c = \varphi$ ,  $\mu_0^c = \mu_0$  без змін стосовно розфарбованої СП.

Таким чином, є всі умови, щоб дати означення СП, яка зв'яже сіткові блоки (як такі можуть використовуватися будь-які СК, в тому числі і ТСБ) у загальну зв'язуючу сітку Петрі.

**Означення 7.22.** Зв'язуючою сіткою  $N_{зв}$  стосовно блоку  $N_G$  є така РСП, яка утворюється з блоку  $N_G$  додаванням до нього позиції  $p_0$ , що називається нульовою, та дуг, які ведуть з  $p_0$  в усі витоків і з усіх стоків в  $p_0$ . Отже, сітка  $N_{зв}$  не має жодного витоків  $t^{дж}$  і жодного стоку  $t^{ст}$ . Оскільки метою цих модифікацій є створення умов

для одержання структури СП-моделі, яка є безпечною, досяжною і живучою, тобто коректно сформованою, дамо означення коректно сформованого блоку.

**Означення 7.23.** Блок  $N_6$  є коректно сформованим, якщо:

- його зв'язуюча сітка  $N_{зв}$  при початковому нульовому маркуванні  $\mu(p_0, \omega)$  безпечна в цілому (за всіма кольорами), тобто  $\forall p \in N_{зв}, \forall \omega \in \Omega : \mu(p, \omega) \leq 1$ , а  $\forall t, \mu$  послідовність, яка реалізується в блоці з порожнього початкового маркування  $\mu(p_0, \omega)$ , закінчується порожнім маркуванням тільки за умови  $|t^{лж}| = |t^{ст}|$ ;
- як початкове маркування  $\mu_{поч}(p, \omega)$  використовуються тільки такі, для яких виконується співвідношення:

$$\begin{aligned} \forall \mu_{поч}(p, \omega) \in R(N_6) : \mu_0(p_0, \omega) \xrightarrow{t^{лж}} \mu_{поч}(p, \omega); \\ \forall p_0 \in \cdot t^{лж} \exists \lambda_{t^{лж}}(p_0, \omega) = 1, \end{aligned} \quad (7.4)$$

де  $R(N_6)$  – множина всіх маркувань, що є досяжними з початкового маркування.

**Висновок 7.2.** Локалізованим вузлом називається коректно сформований сітковий блок  $N_6$ , в якому кожний крайовий перехід є або витокком  $t^{лж}$ , або стоком  $t^{ст}$ , а серед внутрішніх переходів у блоці  $N_6$  немає ні витоків, ні стоків.

З урахуванням введених понять можна дати означення ієрархічної розфарбованої СП.

**Означення 7.24.** Ієрархічно-розфарбованою СП (ІРСП) є сітка, яка складається із скінченної множини коректно сформованих сіткових блоків  $\bar{N}_{i,p} = \{N_6\}$ .

Наприкінці огляду модифікацій *першого напрямку – структурних перетворень* узагальнимо зміст поняття коректної СП-моделі, який впливає з аналізу її властивостей. Він полягає в перевірці безпеки (обмеженості) позицій; правильності параметризації (розфарбованості дуг, переходів, маркерів); структуризації моделі (відсутності взаємоблокувань через конкуренцію). Ті ж самі вимоги справедливі щодо ОМ, які подаються СП-моделями. Отже, розглянуті умови, що дають можливість сформувати коректні моделі, є справедливими і стосовно процесів ГКІС, які успадковують властивості коректно сформованих їх моделей.

**Забезпечення умови живучості у ТРСП.** Якщо типові процеси мають власні ресурси ГКІС, тобто сіткова модель процесу задовольняє умову  $\forall N_i, N_j \in \bar{N} : N_i \cap N_j = \emptyset$ , то можна наперед синтезувати сітковий опис із властивістю живучості.

**Твердження 7.1.** Якщо всі сіткові  $N_i \in \bar{N}$  і старша компонента  $N_0$  є живучими, то живучою є і вся ТРСП-сітка (тут  $N_i$  –  $i$ -й типовий сітковий блок ТРСП-сітки  $\bar{N}$ ).

За наявності загальних ресурсів у типових процесах ГКІС, тобто коли сіткова модель задовольняє умову  $\forall N_i, N_j \in \bar{N} : N_i \cap N_j \subseteq \bigcup_{i=0}^p P_{r_i}$ , живучість сіткових компонентів не гарантує виконання другої умови коректності і необхідно перевірити параметризацію і структуризацію моделі загалом.

Правильність параметризації СП-моделі встановлюється за результатами аналізу процедур надання підпроцесам і операціям виробничого процесу загальних виробничих ресурсів.

Формування керування рухом матеріальних або інформаційних потоків за виробними ресурсами в ГКІС полягає у визначенні в конфліктних ситуаціях заявки-предмета виробництва і засобів її обслуговування (механізм розв'язання конфліктних ситуацій, що виникають при цьому, розглядається в підрозд. 7.7).

З огляду на те, що будь-який перехід СП-моделі може одночасно належати декільком різним типовим конфліктам, це відображення на множині переходів подається сукупністю предикатів  $PR$  умов спрацьовування, співвідношення для яких формується на базі ієрархічної системи розв'язання типових конфліктів («обслуговування заявки» – «надання ресурсу» – «вибір незалежних заявок»). Тобто на кожному рівні ієрархії типових конфліктів формується значення локального предикату умови спрацьовування переходу з урахуванням результатів розв'язання на попередніх рівнях. Подання конфліктів у ГКІС має відобразитись у відповідних властивостях СП.

Розглянувши відображення особливостей функціонування ГКІС у властивостях СП, повернемося до аналізу можливостей структуроутворення розфарбованих сіток. Для автоматизації процесу побудови ієрархічної сітки, а також спрощення аналізу такої сітки на живучість необхідно визначити форму подання ієрархічної СП. З цією метою введемо форму подання *ієрархічно-розфарбованої сітки з типових блоків* (ІРСТБ).

**Означення 7.25.** *Ієрархічно-розфарбованою сіткою з типових блоків* є розфарбована сітка з типових блоків, у термінах яких розв'язується задача аналізу та синтезу дискретних процесів у ГКІС.

Для визначення зазначених у цьому розділі особливостей функціонування ГКІС за властивостями її СП-моделі необхідно розв'язувати задачі аналізу СП (про що йшлося вище). Однією з основних є задача реалізації умови досяжності.

Визначити властивості обмеженості і безпеки можна, розв'язавши задачу покривності.

#### 7.4. Типові сіткові блоки як основа структурного перетворення СП (І модифікація)

Розфарбовані сітки, що складаються з типових блоків, будують на основі елементарних сіток (рис. 7.4, а) і типових переходів (рис. 7.4, б–е).

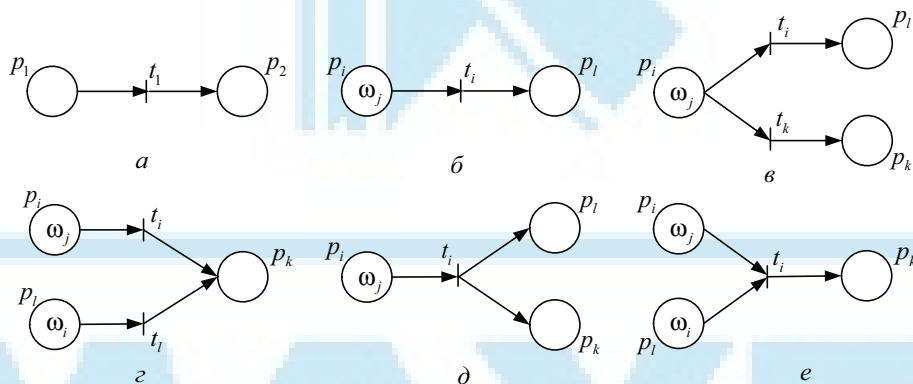


Рис. 7.4. Складові розфарбованої сітки, що складається з типових блоків: а – елементарна сітка; б–е – відповідно переходи типів А, У, Б, П та С

Введемо механізм маркування позицій у розфарбованих сітках, що складаються з типових блоків. Для цього використаємо поняття кольорів маркерів з теорії розфарбованих сіток. Оскільки позиції в РСТБ поставлені у відповідність до дій, то в позицію  $p$  помістимо маркер, що свідчить про виконання відповідної дії. При цьо-

му колір маркера позначає елемент матеріального або інформаційного потоку, над яким виконується дія (кольори маркера – це модельний спосіб трактування різних реальних елементів матеріальних та інформаційних потоків ГКІС).

У РСТБ використовують переходи п'яти типів  $T = \{A, Y, B, P, C\}$  (рис. 7.4), де А – автоматний; У – умовний; Б – безумовний; П – паралельний; С – синхронний переходи [125]. Правило спрацьовування будь-якого переходу із множини  $T$  має вигляд

$$\forall p_i \in \cdot t_i \exists \omega_j \in \Omega : \mu(p_i, \omega_j) \geq F(p_i, t_i), \lambda_{t_i}(p_i, \omega_j) = 1;$$

$$\mu'(p_i, \omega_j) = \mu(p_i, \omega_j) - \delta_{t_i}(\omega_j) \cdot F(p_i, t_i) + x_{t_i}(\omega_j) H(t_i, p_i),$$

де  $\mu'(p_i, \omega_j)$  – нове маркування;  $\delta_{t_i}(\omega_j)$ ,  $x_{t_i}(\omega_j)$  – відповідно функції вхідних і вихідних інцидентів кольору  $\omega_j$  (набувають значення 1, 0);  $\delta_{t_i}(\omega_j) = \lambda_{t_i}(p_i, \omega_j)$ ,

$$x_{t_i}(\omega_j) = \begin{cases} 1, & \text{при } \Psi_{t_i}(\cdot C_{t_i, \omega_j}) = C_{t_i, \omega_j}; \\ 0, & \text{у протилежному разі,} \end{cases}$$

де  $\lambda_{t_i}, \Psi_{t_i}$  – часткові функції: перша функція визначає з множини можливих вхідних маркувань  $L_{t_i}$  підмножину  $\cdot C_{t_i, \omega_j}$  допустимих маркувань  $\omega_j$  переходу  $t_i$ ; друга – допустимому розподіленню з  $\cdot C_{t_i, \omega_j}$  ставить у відповідність єдине вхідне розподілення маркерів певних кольорів з області  $\cdot C_{t_i}$ . При цьому  $\forall t_i \in T$  спрацьовує, якщо в кожній вхідній його позиції  $p_i$  перебуватиме маркер, що керує збудженням переходів. Такий маркер назвемо маркером кольорів спрацьовування  $\omega_j$  (МКС). Спрацьовування переходу  $t_i$  від маркерів у вхідних позиціях, відмінних від МКС, не залежить. У результаті спрацьовування із всіх вхідних позицій  $p_i \in \cdot t_i$  одержуються лише за одним МКС  $\omega_j = \Omega$  й один МКС додається до маркувань кожної вихідної позиції  $p_j \in t_i^*$ .

Зазвичай допускається сполучення деяких типів переходів, наприклад П- і С-переходів. На основі сполучення переходів визначимо коректно сформовані типові блоки СТБ.

**Означення 7.26.** *Сітковий блок називається автоматним (А-блоком), якщо  $\forall t \in T : |\cdot t| = |t^*| = 1$ , тобто кожен перехід має по одній вхідній і вихідній позиції; сітка блоку є сильно зв'язаною (для будь-якої пари позицій у ній є шлях, що веде з однієї в іншу); у початковому маркуванні блоку існує не більше як один МКС  $\omega_j \in \Omega$ , при цьому  $\lambda_{t_i}(p, \omega_j)$ .*

**Означення 7.27.** *Сітковий блок називається умовним (У-блоком), якщо серед внутрішніх переходів блоку сітки обов'язково знайдеться хоча б один У-перехід і хоча б один Б-перехід; у початковому маркуванні блоку існує не більше як один МКС  $\omega_j \in \Omega$  або  $\omega_i \in \Omega$ , причому для  $\forall t_i, t_j \in P^*$  є справедливим:*

$$\lambda_{t_i}(p, \omega_i) = 1; \lambda_{t_j}(p, \omega_j) = 1;$$

$$\lambda_{t_i}(p, \omega_j) = 0; \lambda_{t_j}(p, \omega_i) = 0.$$

**Означення 7.28.** *Сітковий блок називається паралельним (П-блоком), якщо серед переходів блоку є П-перехід, що є виток, і С-перехід, що є сток;  $\forall p \in P : |\cdot p| = |p^*| = 1$ , тобто в кожену позицію входить і з кожної позиції виходить по одній дузі; у початковому маркуванні є не більше як один МКС у кожній позиції, і справедлива умова*

$$\lambda_{t_i} s((p_i, \omega_j), \dots, (p_i, \omega_j)) = 1,$$

де  $n$  – кількість переходів у блоці.



**Означення 7.29.** Блок називається циклічним (Ц-блоком), якщо серед переходів є  $B$ - та  $U$ -переходи; зв'язуюча сітка є живучою, але не безпечною; у початковому маркуванні міститься не більше одного МКС  $\omega_j \in \Omega$  і справедливою є умова

$$\lambda_{t_{i_1}} = \lambda_{t_{i_2}} = \dots = \lambda_{t_{i_k}} = 1,$$

де  $k$  — кількість переходів у блоці.

Циклічний блок не є безпечним, тому не вважається коректно сформованим. Для забезпечення безпеки Ц-блоку слід провести інгібіторні дуги з усіх позицій у його витік [135; 139].

**Означення 7.30.** Конвеєрний блок (К-блок) — це Ц-блок, в який для забезпечення його безпеки з усіх позицій у його витік проведено інгібіторні дуги.

**Означення 7.31.** Ієрархічно-розфарбована сітка Петрі називається розфарбованою сіткою, що складається з типових блоків (СТБ), якщо всі складові її сітки (компоненти), крім старшої, є А-блоками, У-блоками, П-блоками або К-блоками, тобто

$$N_i = (P, T, F, H, \Omega, \lambda, \psi, \mu_0, B),$$

де  $B = \{b\}$  — скінченна множина типів блоків (рис. 7.5),  $d$  — позиція-дублер блоку сітки  $N_i$  будь-якого типу  $b_i \in B$ , що міститься в блоці сітки  $b_j \in B$  іншого типу, причому  $b_i \neq b_j$ .

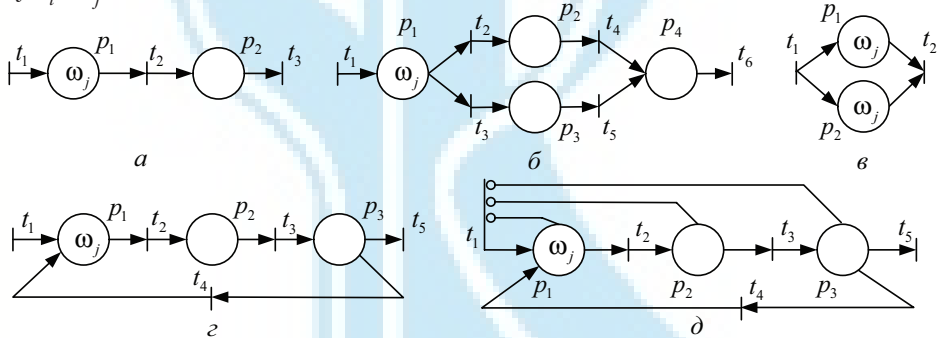


Рис. 7.5. Приклади типів блоків (а–д) відповідно А, У, П, Ц та К

Приклад розфарбованої сітки, що складається з типових блоків, показано на рис. 7.6, а, а дерево відношень цієї сітки — на рис. 7.6, б.

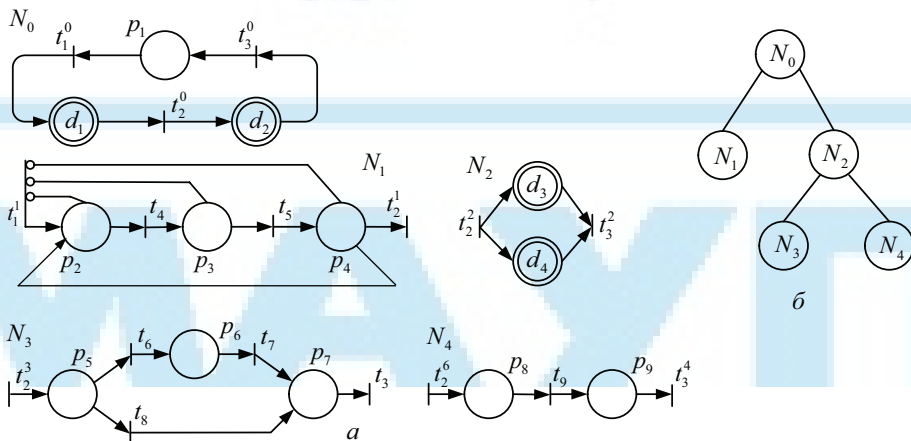


Рис. 7.6. Розфарбована сітка з типових блоків (а) і її дерево відношень (б)

Стандартною формою подання СТБ СП є нормалізована СТБ СП.

**Означення 7.32.** Нормалізованою розфарбованою сіткою, що складається з типових блоків, називається така СТБ, у якій старший компонент незведений, тобто серед локалізованих вузлів старшого компонента  $N_0$  не знайдеться жодного блоку  $N_i$  типу  $b_i \in B$ , що був би частиною (підграфом) якогось блоку  $N_j$  типу  $b_j \in B$ , при цьому  $b_i \neq b_j$ .

Для визначення ефективної галузі застосування розфарбованих сіток, що складаються з типових блоків, необхідно виявити їхні властивості й розробити методи аналізу, що дають змогу визначати коректність процесів, які моделюються за їх моделями у вигляді розфарбованих сіток, що складаються з типових блоків.

Дискретні процеси, що відбуваються у ГКІС, задовольняють умови коректності, якщо адекватні їм модифіковані моделі сітки є правильними, тобто живучими й безпечними.

Розглянемо властивості безпеки й живучості розфарбованих сіток, що складаються з типових блоків. Вирізняють безпечність розфарбованих сіток Петрі в цілому й за певним кольором  $\omega \in \Omega$  [139]. Для розфарбованих сіток, що складаються з типових блоків, також вирізняють безпеку за кольором та безпеку за будь-яким кольором, тобто в цілому.

**Означення 7.33.** Розфарбована сітка, що складається з типових блоків, називається безпечною за кольорами  $\omega \in \Omega$ , якщо

$$\forall p_i \in P \exists \omega \in \Omega : \mu(p_i, \omega) \leq 1;$$

$$\forall N_i \subseteq N \forall d_i \in N_i : \mu(d_i, \omega) \leq 1.$$

**Означення 7.34.** Розфарбована сітка, що складається з типових блоків, називається живучою, якщо:

$$\forall t \in T \exists \mu_k; \mu_l \in R(N) : \mu_k \xrightarrow{t} \mu_l; \quad \forall \mu_i, \mu_j \in R(N) : \mu_i \rightarrow \mu_j;$$

$$\forall \omega \in \Omega \exists p \in P \exists \mu(p, \omega) : \mu(p, \omega) \neq 0;$$

$$\forall N_i \subseteq N \forall \omega \in \Omega \exists d_i \in N_i \exists \mu(d_i, \omega) : \mu(d_i, \omega) \neq 0.$$

Сформулюємо необхідні й достатні умови живучості й безпеки розфарбованих сіток, що складаються з типових блоків.

**Твердження 7.2.** Для того щоб розфарбована сітка, яка складається з типових блоків, була живучою і безпечною, необхідно й достатньо, щоб її старша компонента була живучою та безпечною.

*Доведення.* Нехай існує розфарбована сітка, що складається з типових блоків  $N = \{N_i\}$ , де  $N_0$  — її старша компонента. За визначенням розфарбована сітка, що складається з типових блоків, складається з А-, У-, П- та К-блоків. Ці сіткові блоки є коректно сформованими, отже, безпечними. Замінімо типові сіткові блоки  $N_i$  зв'язуючими сітками  $N_S$ .

Якщо в будь-якому блоці сітки  $N_i$  відсутні маркери, то помістимо маркер у нульову позицію сітки  $N_S$ . У протилежному разі збережемо маркування блоку  $N_i$ . У результаті система сіток  $N' = \{N_S\}$  без врахування нульових позицій має ті самі властивості, що й  $N$ . Припустимо тепер, що сітка  $N$  живуча й безпечна. Тоді живуча й безпечна сітка  $N'$ , а отже, і  $N_0$ . Тепер покажемо, що якщо живуча й безпечна  $N_0$ , то живуча й безпечна  $N'$ , а отже, і  $N$ .

За визначенням при початковому маркуванні всі зв'язуючі сітки  $N_S$  коректно сформованих блоків є живучими й безпечними. Таким чином, якщо старша ком-

понента  $N_0$  є живучою й безпечною, то система  $N'$  буде живучою і безпечною за наявності збудженого переходу в будь-якому досяжному маркуванні. Причому, якщо цей перехід перебуває у зв'язці, тоді кожен з переходів зв'язування збуджений. Таке маркування існує відповідно до визначення коректно сформованих блоків. Необхідні й достатні умови твердження 7.2 доведено.

Нехай розфарбована сітка, що складається з типових блоків, задана набором

$$N = \{N'_j, b_i\}; 1 \leq i \leq n$$

( $b'_i$  – тип блоку  $j$ -ї компоненти сітки  $N$ ) і містить тільки елементи, що утворюють систему типових блоків (СТБ)  $B = \{b_1, \dots, b_m\}$ . Якщо хоча б один з блоків сітки  $b'_i$  не належить  $B$ , то виникає задача декомпозиції  $b'_i$  на такі складові  $b_p$ , що  $b_i \in B$ . У результаті декомпозиції (відповідно до процедури заміщення вузла) одержимо розфарбовану сітку, що складається з типових блоків вигляду

$$N = \{N_j, b_i\}, 1 \leq i \leq m.$$

Якщо при цьому старша компонента  $N_0$  буде незведеною, то така розфарбована сітка, що складається з типових блоків, буде нормалізованою. Будь-якій сітці Петрі відповідає множина еквівалентних їй розфарбованих сіток, що складаються з типових блоків, але в цій множині існує єдина нормалізована розфарбована сітка, що складається з типових блоків, яка має максимальну кількість блоків сітки з СТБ. Розглянемо правило перетворення сітки Петрі в нормалізовану розфарбовану сітку, що складається з типових блоків.

Кожен з блоків сітки, що входять у СТБ, характеризується типовими переходами. Процедура виділення блоку ґрунтується на розпізнаванні цих переходів.

1. Визначають всі А-переходи, що містяться в сітці Петрі.
2. Формують всі А-блоки й будують залишкову сітку.
3. Кожен перехід залишкової сітки перевіряють на належність до П- або С-переходів і виділяють їх.
4. Знаходять шлях з П- у С-перехід. Якщо таких шляхів не менше двох, то виділяють П-блок і будують залишкову сітку.
5. Кожен перехід залишкової сітки перевіряють на належність до У- або Б-переходів.
6. якщо будь-які У- і Б-переходи сполучені, виділяють У-блок. У протилежному разі переходять до наступного кроку.
7. Для кожного переходу перевіряють наявність інцидентних йому інгібіторних дуг і виділяють К-блок.
8. У всіх виділених блоках перевіряють наявність блоків будь-якого іншого типу. Якщо такі існують, то виділяють його.
9. Якщо залишкова сітка незведена, то процедура виділення блоку завершується. У протилежному разі переходять до кроку 1.

Приклад виділення блоку в сітці Петрі наведено на рис. 7.7.

У вихідній сітці (рис. 7.7, а) виділяють А-блоки  $N_1$  і  $N_4$  й будують залишкову сітку  $N_0$ . У результаті утвориться розфарбована сітка, що складається з типових блоків (рис. 7.7, б). Далі (рис. 7.7, в) виділяють П-блок  $N_2$  і знову будують залишкову сітку. Останнім кроком алгоритму буде виділення У-блоку й перевірка залишкової сітки. У результаті утвориться нормалізована розфарбована сітка, що складається з типових блоків.

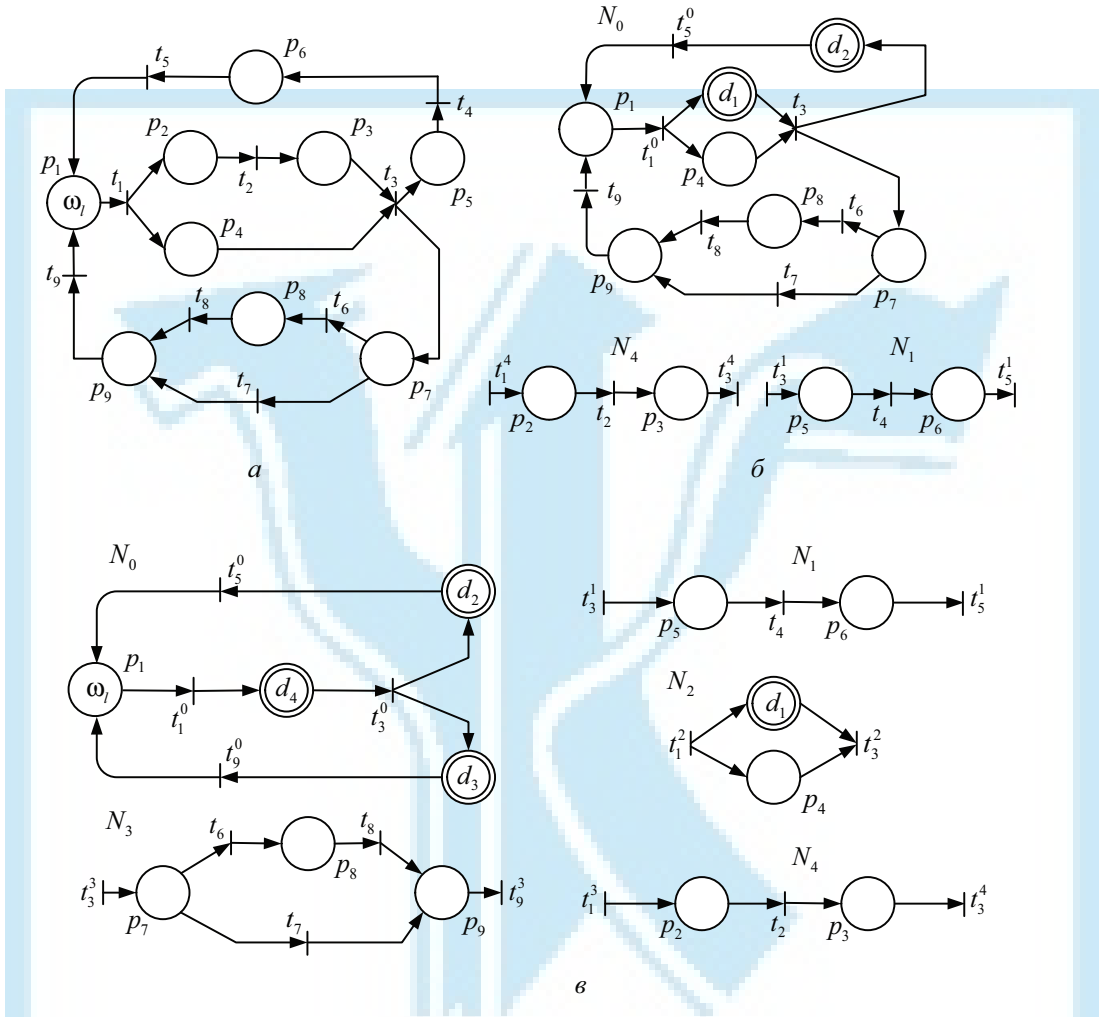


Рис. 7.7. Перетворення сітки Петрі в нормалізовану розфарбовану сітку, що складається з типових блоків: а – вихідна сітка Петрі; б – побудова залишкової сітки  $N_0$  при видаленні А-блоків  $N_1, N_4$ ; в – побудова залишкової сітки  $N_0$  при видаленні П-блоку  $N_2$

Нормалізовану розфарбовану сітку, що складається з типових блоків, можна одержати іншим способом. Для цього утворюють будь-яку розфарбовану сітку, що складається з типових блоків, еквівалентну вихідній сітці. Якщо отримана сітка містить блоки, що мають  $d$ -позиції блоків цього ж типу, то ці позиції необхідно замінити відповідними їм блоками сітки.

Таким чином, нормалізована розфарбована сітка, що складається з типових блоків, є стандартною формою подання розфарбованої сітки, що складається з типових блоків, і містить сіткові блоки чотирьох типів (А-, У-, П- та К-блоки).

## 7.5. Сіткові (блочні) моделі процесів ГКІС (II модифікація – перший етап проектування СК ГКІС)

Процес проектування системи керування складальною ГКІС можна подати у вигляді послідовності трьох основних етапів:

1. Формалізація об'єкта керування.
2. Дослідження властивостей об'єкта керування за його моделлю.
3. Синтез алгоритму керування.

Про **перший етап** проектування частково йшлося в розд. 7.1, коли процес функціонування виробничої системи поставав у формі ієрархії підпроцесів, вид яких визначався наявністю типових відносин.

Розглянемо інтерпретацію сітки виділених типових процесів і відносин. Багатономенклатурне складальне виробництво належить до класу складних систем із множиною різних динамічних об'єктів. Найчастіше зображені в сітці мітками об'єкти мають додаткові атрибути, які дають змогу розрізнити їх, а також використовувати ці розбіжності для керування функціонуванням системи.

Залежно від необхідного за концептуальною моделлю  $SP$  рівня детальності подання ієрархія в складних виробничих процесах (ВП) ГКІС задається структурою із множини більш простих підпроцесів, тип яких визначається наявністю тих чи інших співвідношень зв'язку виробничих операцій. Інакше кажучи, ВП складається з  $n$  ієрархічних рівнів, де деякі або всі операції підпроцесів  $i$ -го рівня ( $2 \leq i \leq n$ ) є процесами, що складаються з операцій, які належать до  $(i - 1)$ -го рівня. Старший рівень відповідає тільки одному підпроцесу.

Як модель ВП, що ефективно відображає встановлену ієрархію підпроцесів й адекватно подає властивості виділених типових відносин зв'язку операцій за матеріальним потоком в складальній ГКІС, пропонується модифікація ієрархічних розфарбованих сіток Петрі – типова розфарбована сітка ( $T$ -сітка).  $T$ -сітка будується із сіткових блоків, які формалізують встановлення типових відношень зв'язку операцій і подають типові підпроцеси в ієрархічній структурі.

**Означення 7.35.** Сітковим блоком у  $T$ -сітці називається розфарбована сітка  $N_i = (P_i, T_i, F_i, H_i, \Omega_i, \lambda_i, \Psi_i, \mu_i^0)$  така, що

$$P_i = P_{\text{оп}} \cup P_r; P_{\text{оп}} \cap P_r \neq \emptyset; |P_{\text{оп}}| \geq |P_r|;$$

$$T_i = T_i^* \cup T_i^I \cup T_i^S; T_i^* \cap T_i^I = \emptyset; T_i^* \cap T_i^S = \emptyset; T_i^I \cap T_i^S = \emptyset,$$

де  $P_{\text{оп}} = \{p_j \mid j = \overline{1, m}\}$  – множина позицій-операцій підпроцесу;

$P_r = \{p_r \mid \forall p_r (\exists \{p_j\} \subset P_{\text{оп}} : (p_j^* \cup \cdot p_j) \cap (p_r^* \cup \cdot p_r) \neq \emptyset) : p_j^* \subseteq \cdot p_r, \cdot p_j \subseteq p_r^*, p_r^* \cap \cdot p_r = \emptyset\}$  – множина позицій-ресурсів, на яких виконуються операції підпроцесу;

$T_i^* = \{t^* \mid \exists p_k, p_j \in P_{\text{оп}} : F_i(p_k, t^*) = 1 \wedge H_i(t^*, p_i) = 1\}$  – множина внутрішніх переходів, що визначають взаємозв'язок операцій у підпроцесі;

$T_i^I = \{t^I \mid \forall p_j \in P_{\text{оп}} \exists p_k \in P_{\text{оп}} : F_i(p_j, t^I) = 0 \wedge H_i(t^I, p_k) = 1\}$  – множина переходів-витоків, що визначають початкові операції підпроцесу;

$T_i^S = \{t^S \mid \forall p_j \in P_{\text{оп}} \exists p_k \in P_{\text{оп}} : H_i(t^S, p_j) = 0 \wedge F_i(p_k, t^S) = 1\}$  – множина переходів-стоків, що визначають фінальні операції підпроцесу.

Зазначимо, що конкуренція у ВП можлива як усередині найпростіших послідовних підпроцесів, так і між ними, що, у свою чергу, зумовлено дефіцитом виробничих ресурсів. Тому подання відношень конкуренції у сітці в загальному вигляді будується на основі ієрархії взаємопов'язаних підпроцесів та формалізується одним з варіантів  $K$ -блоку: лінійно-конкурентного ( $LK$ ); паралельно-конкурентного ( $PK$ ); альтернативно-конкурентного ( $AK$ ). Вибір варіанта  $K$ -блоку визначається типом підпроцесу (лінійного послідовного, паралельного, альтернативного), у якому розглядається конкуренція.

$T$ -сітка складається із скінченної множини сіткових блоків  $\bar{N} = \{N_i | i = \overline{0, \rho}\}$ , на якій визначено операцію підстановки, що зв'язує сіткові блоки в багаторівневе дерево  $\Gamma: \bar{N} \rightarrow N$ . Операція підстановки – це теоретико-множинна операція накладання структури сітки блоку на особливу позицію сітки ( $p_d$ -позицію), заміщення її вхідних і вихідних переходів і припустимих розподілів кольорів маркерів переходами (витоками, стоками блоку та їх розподілами). Для здійснення операції підстановки між елементами, які заміщаються, варто встановити однозначні відповідності.

У  $T$ -сітці використано таку інтерпретацію елементів: позиціям  $p_j$  поставлено у відповідність операції процесу, що моделюється, а також ресурси технологічної підсистеми, які їм надаються; переходам  $t_i$  – події зміни операцій; маркерам – одиниці матеріальних потоків та стани ресурсів. Тоді  $p_d$ -позиції набувають такого сенсу: кожній  $p_d$ -позиції відповідає макрооперація, яка є підпроцесом більш низького рівня складності подання, опис якого задається блоком сітки, що заміщається.

**Означення 7.36.** *Типовою розфарбованою сіткою* називається сітка вигляду  $N = (\bar{N}, \Gamma, B, \theta, Z)$ , де  $(\bar{N}, \Gamma)$  – ієрархічна розфарбована сітка;  $\theta: \bar{N} \setminus \{N_0\} \rightarrow B$ ,  $B = \{S, C, P, A, K\text{-блоки}\}$ ;  $\theta(N_0) \in \{S, C, \text{сітка або блок з автоматною структурою}\}$ ;  $\forall N_i, N_j \in \bar{N}: (N_i, N_j) \in \Gamma \Rightarrow \theta(N_i) \neq \theta(N_j)$ ,  $i \neq j$ ;  $Z: \bigcup_{i=0}^{\rho} P_i \setminus \left\{ \left\{ p_{d_j} | j = \overline{1, \rho} \right\} \cup P_{\bar{n}_i} \right\} \times \prod_{i=0}^{\rho} \Omega_i \rightarrow R_0$  – функція часових параметрів операцій.

Вважатимемо, що  $T$ -сітка  $N$  адекватна виробничому процесу, якщо між її елементами й атрибутами концептуальної моделі виробничого процесу встановлено взаємооднозначну відповідність (табл. 7.1).

**Означення 7.37.** *Сітковою моделлю виробничого процесу* будемо називати таку  $T$ -сітку  $N_{II}$ , для якої виконуються такі умови:

$$P_i = P_{ci} \cup P_{di} \cup P_{\bar{n}_i}, \quad P_{ci} \cup P_{di} = P_{\text{ов}}, \quad P_{ci} \cap P_{di} = \emptyset,$$

де  $P_{ci} = \{p_c | p_c = f_1(O_j), O_j \in O\}$ ;  $P_{di} = \{p_d | p_d = f_2(N_j), \Gamma(N_j) = N_i\}$  – множина підпроцесів  $i$ -го рівня ієрархії виробничого процесу;  $P_{\bar{n}_i} = \{p_r | p_r = f_3(m), m \in M\}$ ;

$$T_i = \{t | t = f_4(g), g = (O_i, O_j) \in R_i, R_i \in W\}, \quad \Omega_i = \Omega_{ci} \cup \Omega_{\bar{n}_i}, \quad \Omega_{ci} \cap \Omega_{\bar{n}_i} = \emptyset,$$

$$\Omega_{ci} = \{\omega_c | \omega_c = f_5(d), d \in D\},$$

$$\Omega_{\bar{n}_i} = \{\omega_r | \omega_r = f_6(s(m)), s(m) \in S_M\}, \quad \lambda_i = \{\lambda_t^i | t \in T_i, \lambda_t^i = f_7(TM)\},$$

$$\Psi_i = \{\psi_t^i | t \in T_i, \psi_t^i = f_8(TM)\},$$


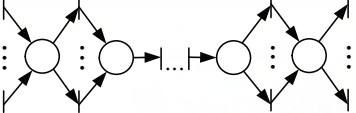
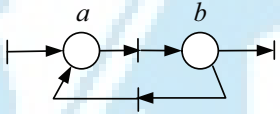
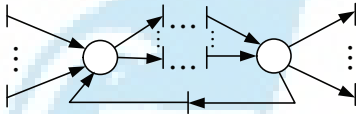
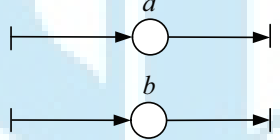
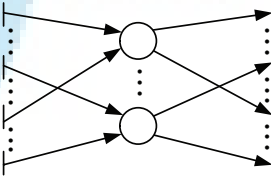
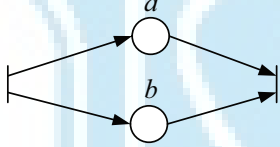
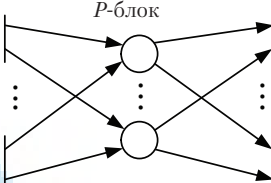
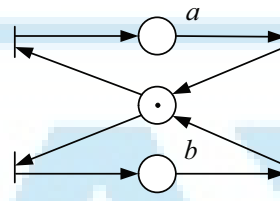
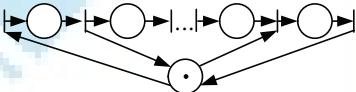
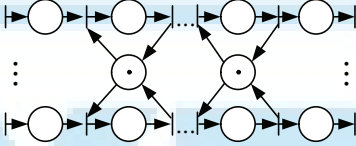
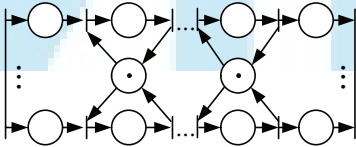
$$\mu_0^i(P_i \setminus P_{\bar{n}_i}) = \{\mu_0^i(p, \omega_c) | p \in P_i \setminus P_{\bar{n}_i}, \omega_c \in \Omega_{ci}, s(d) \in S_D, \mu_0^i(p, \omega_c) = f_9(s(d))\},$$

$$\mu_0^i(P_{\bar{n}_i}) = \{\mu_0^i(p_r, \omega_r) | p_r \in P_{\bar{n}_i}, \omega_r \in \Omega_{\bar{n}_i}, m \in M, \mu_0^i(p_r, \omega_r) = e(m)\},$$

$B = \{b_l | b_l = f_{10}(R_l), R_l \in W\}$ ,  $Z = \{z(p, \omega) | p \in P_{ci}, \omega \in \Omega_{ci}, z(p, \omega) = \tau\}$ ,  $e(m)$  – ємність засобу  $m$ .

Таблиця 7.1

Подання типових відношень сітковими блоками

Тип відношень	Подання відношень у сітці	Типовий сітковий блок
Прямуювання (строного порядку) $aR_{\Pi p}b$		S-блок 
Прямуювання (циклічного порядку) $R_{\Pi c} = R_{\Pi p}^+$ $aR_{\Pi c}b$		C-блок 
Альтернатива $aR_{\Lambda}b$		A-блок 
Паралелізм $aR_{\Pi}b$		P-блок 
Конкуренція $aR_{\kappa}b$		LK-блок 
		AK-блок 
		PK-блок 

За допомогою розглянутої моделі сітки можна формально подати ознаки типізації виробничих процесів і здійснити класифікацію моделей сітки. Так, у  $T$ -сітці тип виробничого процесу залежно від його структури зв'язку операцій визначається за наявністю сукупності блоків сітки  $N_i$  із системи типових блоків  $B$ , причому  $(\theta(N_i) \in B)$ , а вид подання кореневої вершини дерева відносин  $\Gamma$ , що належить сітці, визначає тип процесу залежно від характеру й способу реалізації. Сітка з автоматною структурою у кореневій вершині  $N_0$  відповідає циклічним процесам, а сітковий блок із системи  $B$  або з автоматною структурою — конвеєрним процесам. Таким чином, за результатами класифікації виробничих процесів можна побудувати аналогічний класифікатор відповідних моделей сітки.

### Формалізація опису паралельних виробничих процесів складальної ГКІС (ГКІСС)

**Приклад 7.1.** Розглянемо на прикладі деякої структури ГКІСС (рис. 7.8) спосіб формування сіткової моделі реалізованого в ній виробничого процесу, використовуючи розроблений апарат моделювання на базі  $T$ -сіток.

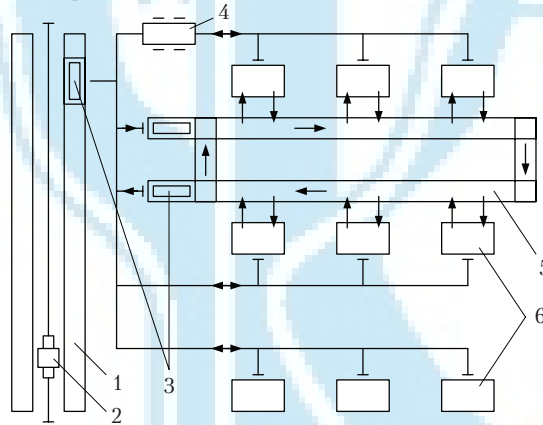


Рис. 7.8. Схема ГКІСС з комбінованим транспортуванням:

- 1 — автоматизований склад (АС); 2 — робот-штабелер (РШ);
- 3 — нагромаджувачі (Н); 4 — автоматизований транспортний модуль (АТМ);
- 5 — транспортний конвеєр (ТК); 6 — гнучкий виробничий модуль (ГВМ)

Оснoву цього способу і всієї процедури формалізації становить принцип відображення в моделі матеріально-транспортних зв'язків виробничих процесів при організації обслуговування роботи основного технологічного устаткування ГКІСС. Такий підхід до формалізації дає змогу сформувати структуру моделі, що буде інваріантною порівняно з технологіями виготовлення продуктів виробництва, тому що як параметри моделі враховуються стан устаткування, вплив операцій, що виконуються, на зміну цих станів, типи одиниць матеріальних потоків і технології. Крім того, у такому поданні можна типізувати не лише за структурою зв'язку операцій, а й за функціями підпроцесів, що виконуються, й виробничі операції.

У різних фазах функціонування ГКІСС одиниці матеріальних потоків подають (за призначенням) базові деталі й складальні вузли (БД), комплектуючі деталі й напівфабрикати (КД), тару (Т), інструмент й оснащення (І). Всі однотипні предмети групуються в партії й передаються у тарі. Можна виділити такі типові операції: зберігання одиниць матеріального потоку  $\alpha$ -го типу на  $\beta$ -му устаткуванні, що надійшли від  $\gamma$ -го виробничого засобу ( $Z\beta_{\beta}^{\alpha,\gamma} \in O_{\beta 0}$ ); транспортування одиниць матеріального потоку  $\alpha$ -го типу на  $\beta$ -му устаткуванні від  $\gamma$ -го до  $\rho$ -го виробничого засобу ( $Tr_{\beta,\alpha}^{\gamma,\rho} \in O_{C\kappa}$ ); складання ( $С\kappa_{\beta} \in O_{C\kappa}$ ) й налагодження ( $Н\lambda_{\beta} \in O_{Н\lambda}$ ) на  $\beta$ -му устаткуванні. На



сіткової моделі (рис. 7.9) для наочності подання упущено позиції, що відображають виробничі засоби відповідних операцій процесу.

Конвеєрний характер руху матеріальних потоків виробничими засобами, використання одного й того самого технологічного устаткування в різних маршрутах виготовлення виробів, а також наявність загальних транспортно-складських засобів обслуговування ГВМ дають підстави зарахувати виробничий процес, що відбувається в досліджуваній ГКІСС, до класу конвеєрних конкурентних з елементами паралелізму й альтернативи, а відповідну сіткову модель подати  $T$ -сіткою з кореневою вершиною у вигляді блоку сітки з автоматною конструкцією.

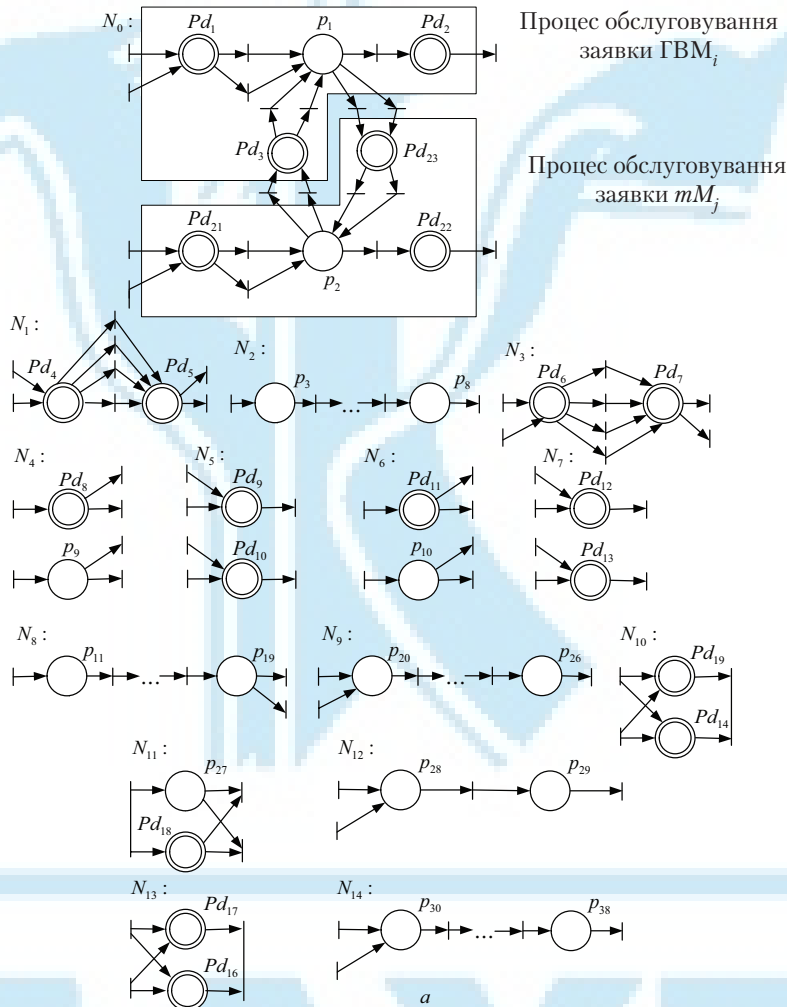


Рис. 7.9. Сіткова модель виробничого процесу:

- $N_1(N_3)$  – підготовка ГВМ<sub>i</sub> до обслуговування заявки від АС (ГВМ<sub>j</sub>);
- $N_2$  – доставка результатів складання на АС;  $N_4(N_6)$  – підготовка до налагодження ГВМ<sub>i</sub> на заявку від АС (ГВМ<sub>j</sub>);  $N_5(N_7)$  – підготовка до доставки об'єктів складання від АС (ГВМ<sub>j</sub>) до ГВМ<sub>i</sub>;  $N_8(N_{11})$  – налагодження ГВМ<sub>i</sub> на заявку від АС (ГВМ<sub>j</sub>);
- $N_9(N_{12})$  – доставка БД від АС (ГВМ<sub>j</sub>) до ГВМ<sub>i</sub>;  $N_{10}(N_{13})$  – доставка об'єктів складання від АС (ГВМ<sub>j</sub>) до ГВМ<sub>i</sub>;  $N_{14}$  – доставка КД до ГВМ<sub>i</sub> (БД, КД – базова й комплектуюча деталі відповідно)

На рис. 7.10 зображено дерево відношень цієї моделі сітки.

У досліджуваній ГКІСС виділяються такі схеми руху матеріальних потоків різних типів: 1) БД за схемою ГВМ-ГВМ або ГВМ-АС-ГВМ; 2) КД за схемою АС-АТМ-ГВМ; 3) Т за схемою ГВМ-АТМ-АС; 4)  $N$  за схемою АС-АТМ-ГВМ-АТМ-АС. Тоді виробничий процес функціонування ГКІСС можна подати у вигляді сукупності взаємопов'язаних окремих процесів обслуговування ГВМ, кожен з яких включає в себе різноманітні підпроцеси підготовки ГВМ до обслуговування заявок-виробів (підготовка до налагодження, налагодження, доставка об'єктів складання), самообслуговування заявок, які надходять на виконання складальної операції і доставку результатів складання на АС.

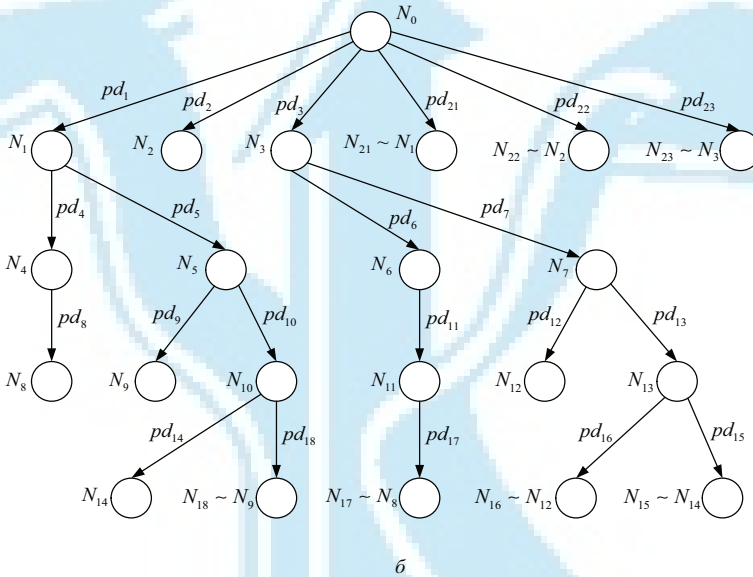


Рис. 7.10. Дерево відношень для сіткової моделі рис. 7.9

У табл. 7.2 наведено встановлені моделлю сітки відповідності між позиціями сітки й виробничими операціями, що виконуються над одиницями матеріальних потоків.

Таблиця 7.2

**Відповідність між позиціями моделі сітки та операціями виробничого процесу**

Позиція	Ідентифікатор операції	Зміст
1	2	3
$P_1$	$Ск_i$	Операція складання на ГВМ <sub>i</sub>
$P_2$	$Ск_j$	Операція складання на ГВМ <sub>j</sub>
$P_3(P_{29})$	$Зб_{ГВМ_i}^{БД, ГВМ}$	Зберігання на ГВМ <sub>i</sub> базових деталей БД, що надходять після операції складання
$P_8$	$Тр_{РШ, БД}^{АТМ, АС}$	Транспортування РШ* деталей БД, що надходять з АТМ на АС
$P_9$	—	Пропуск операції (очікування на АС)

\* РШ — робот-штабелер.

1	2	3
$P_{10}(P_{27})$	$Зб_{ГВМ_j}^{БД, ГВМ}$	Зберігання на ГВМ <sub>j</sub> деталей БД, що надходять після операції складання
$P_{11}$	$Тр_{РШ, I}^{АС, АТМ}$	Транспортування РШ інструмента I, що надходить з АС на АТМ
$P_{19}$	$Тр_{РШ, I}^{АТМ, АС}$	Транспортування РШ інструмента I, що надходить з АТМ на АС
$P_{20}$	$Зб_k^{БД, АС}$	Зберігання БД в комірці АС
$P_{26}$	$Зб_{ГВМ_i}^{БД, ТК}$	Зберігання БД у ГВМ <sub>i</sub> , що надходять із ТК на операцію складання
$P_{28}$	$Тр_{ТК, БД}^{j, i}$	Транспортування БД на ТК від ГВМ <sub>j</sub> до ГВМ <sub>i</sub>
$P_{30}$	$Зб_k^{КД, АС}$	Зберігання КД в комірці АС
$P_{33}$	$Зб_k^{Т, РШ}$	Зберігання Т в комірці АС, що надходить від РШ

Таким чином, з урахуванням наведеного у прикладі 7.1 процедура формування подання сітки виробничого процесу полягає у виконанні таких дій:

- формування описів матеріальних потоків і станів виробничих засобів;
- визначення схем руху предметів виробництва обробними ресурсами і виділення типових операцій і підпроцесів;
- ідентифікація досліджуваного процесу;
- структуризація повного опису з подання сітки підпроцесів;
- параметризація моделі згідно з існуючими технологіями виготовлення й допустимими схемами руху матеріальних потоків.

## 7.6. Аналіз умов організації виробничого процесу ГКІС за його сітковою моделлю (другий етап проектування СК ГКІС)

Після формалізації опису виробничого процесу переходять до *другого етапу* проектування системи керування ГКІС, метою якого є виявлення неприпустимих станів і визначення умов функціонування обробляючих ресурсів ГКІС, що не передбачає тупикового функціонування.

Процедура дослідження виробничого процесу за моделлю полягає у встановленні таких *умов коректності*:

1) операція процесу може повторно початися лише після свого завершення, тобто  $\forall O_j \in O_{Ск} \cup O_{Нл} \cup O_{Тр} : \tau_k^П(O_j) \geq \tau_{k-1}^З(O_j)$ ;

2) у ході виконання операції не повинно виникати тупикових станів і блокувань, при яких процес припиняє своє виконання або продовжує функціонування замкненим циклом, не досягнувши кінцевого стану, тобто  $\forall O_j \in O_{Ск} \cup O_{Нл} \cup O_{Тр} \cup O_{Зб}, \forall k \in K_j : \tau_k^П(O_j) - \text{існує, де } \tau_k^П(O_j), \tau_k^З(O_j) - k\text{-й момент початку/закінчення виконання операції } O_j; K_j - \text{множина моментів початку операції } O_j$ .

Аналогічно ієрархічним розфарбованим сіткам коректність виробничого процесу встановлюється виходячи з аналізу властивостей безпеки (обмеженості) і живучості сіткової моделі. Алгоритм аналізу властивостей Т-сіткової моделі полягає у перевірці безпеки (обмеженості) позицій-операцій виробничого процесу; живучості

сіткового опису виробничого процесу ГКІС; *правильності параметризації* (розфарбування дуг, переходів, міток) та *структуризації* моделі (відсутність взаємоблокувань через конкуренцію).

**Перевірка безпеки.** Кожна позиція-ресурс  $p_r \in \bigcup_{i=0}^p P_i$  і пов'язані з нею через вхідні й вихідні переходи позиції-операції  $p_c \in \tilde{P}_c$  утворюють у  $T$ -сітці замкнені кільцеподібні підсітки. Такі підсітки обмежені через те, що у виразі

$$\Delta = \sum_{\omega \in \Omega} \sum_{p \in \tilde{P}_c \cup \{p_r\}} \mu(p, \omega) = \sum_{\omega \in \Omega} \sum_{p \in \tilde{P}_c \cup \{p_r\}} \mu_0(p, \omega),$$

якщо  $\Delta = 1$ , тоді позиція  $p \in \tilde{P}_c$  є безпечною. Тому для забезпечення першої умови коректності необхідно для відповідних позицій-операцій (складання, налагодження, транспортування) ввести позиції-ресурси з безпечним початковим маркуванням лише за одним з кольорів, а для інших позицій-операцій (зокрема зберігання) — з обмеженням за загальною кількістю маркерів початковим маркуванням.

**Перевірка живучості.** Якщо типові підпроцеси мають власні ресурси технологічної підсистеми, тобто сіткова модель виробничого процесу задовольняє умову  $\forall N_i, N_j \in \bar{N} : N_i \cap N_j = \emptyset$ , тоді можна наперед синтезувати сітковий опис із властивістю живучості.

Так, якщо живучі всі сіткові блоки  $N_i \in \bar{N}$  і старша компонента  $N_0$ , тоді живуча й  $T$ -сітка в цілому. Таким чином, забезпечивши живучість усіх типових блоків сіткового подання й перевіряючи старшу компоненту, можна гарантувати виконання другої умови коректності виробничого процесу.

Якщо ж типові підпроцеси мають деякі загальні ресурси технологічної підсистеми, тобто сіткова модель задовольняє умову  $\forall N_i, N_j \in \bar{N} : N_i \cap N_j \subseteq \bigcup_{i=0}^p P_i$ , тоді живучість сіткових компонентів не гарантує другої умови коректності. У цьому випадку необхідно перевірити правильність параметризації і структуризації моделі за:

- адекватністю відображення на сіткову структуру технологій виготовлення й маршрутів руху матеріальних потоків. Перша умова правильності:

$$\forall p \in \bigcup_{i=0}^p P_i : \Omega_p^* = \cdot \Omega_p,$$

де

$$\Omega_p^* = \left\{ \omega \mid \forall t \in p^*, (p, \omega) \in (\cdot t, \Omega) : \lambda_t((\cdot t, \Omega)) = 1 \right\},$$

$$\cdot \Omega_p = \left\{ \omega \mid \forall t \in \cdot p, (p, \omega) \in (t^*, \Omega) : \Psi_t((t^*, \Omega)) = (t^*, \Omega) \right\};$$

- можливість здійснення в сітковій моделі режимів, що не призводять до відхилення від допустимого порядку організації процесу (наприклад, необхідний ресурс до виконання операції не підготовлений). Друга умова правильності:

$$\forall p_r \in \bigcup_{i=0}^p P_i, \forall t \in p_r^* : \cdot C_t = \prod_{p \in \cdot t} \Omega(p, t),$$

де  $\Omega(p, t) = \left\{ (p, \omega) \mid (p, \omega) \in (\cdot t, \Omega) : \lambda_t((\cdot t, \Omega)) = 1 \right\}$ .

**Перевірка правильності параметризації.** Невиконання першої умови правильності параметризації потребує перебудови сіткової моделі й перевірки вихідних даних; невиконання другої — доповнення сіткової моделі умовами блокування початкового переходу послідовності спрацьовуючих переходів  $v'$ , яке призводить

до маркування з неприпустимою розміткою позиції-ресурсу, що перевіряється, для спрацьовування її вихідного переходу. Крім того, ортогональною умовою повинен бути позначений початковий перехід послідовності спрацьовуючих переходів  $v'$ , який призводить до маркування, яке змінює неприпустиму розмітку позиції-ресурсу, що перевіряється, на допустиму за умови  $v' \neq v''$ .

**Перевірка правильності структуризації.** Правильність структуризації сіткової моделі встановлюється за результатами аналізу процедур надання підпроцесам й операціям у виробничому процесі ГКІС загальних виробничих ресурсів. Взаємні блокування підпроцесів можливі в сітковій моделі, якщо існують цикли, що проходять через одну або більше позицій-ресурсів. У такому випадку сіткову модель коригують, вводячи умови блокування надання ресурсів, що входять у цикл, якщо всі інші ресурси вже надані. Для цього використовують інгібіторні (забороняючі) дуги, визначені в праці [74].

### 7.7. Синтез алгоритмів керування виробничими процесами ГКІС (третій етап проектування СК ГКІС)

**Третій етап** проектування систем керування ГКІС полягає в одержанні обумовленої виконуваним виробничим процесом структури процесу керування, а потім у формуванні коректного алгоритму керування. Оскільки в отриманій моделі  $N_{ВП}$  можна реалізувати майже будь-яку послідовність рухів матеріального потоку, визначену в рамках сформованої концептуальної моделі ГКІС, то структура процесу керування вважається повністю заданою й адекватною стосовно зображеної на рис. 7.9  $T$ -сітки, якщо з кожною виробничою операцією зіставити оператор керування, що ініціює її виконання. Тоді задача синтезу алгоритму керування полягає в задаванні на множині переходів  $T$  моделі виробничого процесу  $N_{ВП}$  відображень  $\eta : P \times \Omega \rightarrow T$ , що адекватно визначають порядок реалізації операцій у конфліктних ситуаціях ( $P \times \Omega$ ), і введенні додаткових елементів, що моделюють інформаційний обмін керуючої підсистеми й об'єкта керування. Процес формування керування рухом матеріальних потоків, що обробляються, ресурсами у ГКІС полягає у встановленні в конфліктних ситуаціях обслуговуваної заявки-предмета, виробництва й засобу її обслуговування. Відповідно до такого подання в ході виконання виробничого процесу імовірно виникнення типових конфліктних ситуацій [125; 139]:

- обслуговування заявок тим самим засобом;
- надання засобів загальній заявці;
- виконання незалежних одна від одної операцій.

У сітковій моделі виробничого процесу виділені ситуації мають своє формалізоване подання у вигляді таких типових конфліктів на переходах  $T$ :

1. Конфлікт зв'язаних доступом до загального ресурсу заявок  $\omega_{ci}, \omega_{cj}$ :

а) конфлікт  $K_1$  заявок від однієї й тієї ж операції за виконання іншої загальної операції (рис. 7.11, а), де

$$K_1 = \bigcup_{t \in T} K_1(t); K_1(t) = \{ \cdot c_t \mid \cdot c_t \in \cdot C_t : |\cdot C_t| > 1 \}; T = \bigcup_{i=0}^p T_i;$$

б) конфлікт  $K_2$  заявок від різних операцій за виконання загальної операції (рис. 7.11, б), де

$$K_2 = \bigcup_{p_c \in P_c} K_2(p_c); K_2(p_c) = \{ t \mid t \in \cdot p_c \}; P_c = \bigcup_{i=0}^p P_{ci};$$

в) конфлікт  $K_3$  заявок за той самий ресурс (рис. 7.11, в), де

$$K_3 = \bigcup_{p_r \in P_r} K_3(p_r); K_3(p_r) = \left\{ t \mid \left( \forall t_i, t_j \in p_r : t_i \cap t_j = \emptyset \right) \wedge \right. \\ \left. \wedge \left( t_i \cap t_j \right) \setminus \{ p_r \} = \emptyset \left( \exists c_{t_i} \in C_{t_i}, c_{t_j} \in C_{t_j} : c_{t_i} = c_{t_j} \right) \right\}; P_r = \bigcup_{i=0}^p P_{ri}.$$

2. Конфлікт  $K_4$  різних ресурсів  $p_{ri}, p_{rj}$  за заявку  $\omega_{ci}$  (рис. 7.12, а), де

$$K_4 = \bigcup_{p_c \in P_c} K_4(p_c); K_4(p_c) = \left\{ t \mid \forall t_i, t_j \in p_c, \exists c_{t_i} \in C_{t_i}, c_{t_j} \in C_{t_j} : c_{t_i} = c_{t_j} \right\}.$$

3. Конфлікт  $K_5$  незалежно виконуваних заявок  $\omega_{ci}, \omega_{cj}$  (рис. 7.12, б), де

$$K_5 = \left\{ K_2(p_c), K_3(p_r), K_4(p_c) \mid p_c \in P_c, p_r \in P_r \right\} \cup \left( T \setminus (K_2 \cup K_3 \cup K_4) \right).$$

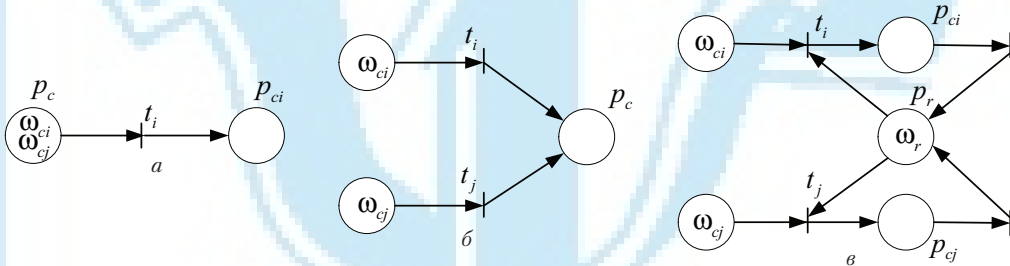


Рис. 7.11. Сіткове подання конфліктів заявок:

а, б – види конфліктів заявок за виконання загальної операції;

в – конфлікт заявок за той самий ресурс

Розгляд як конфлікту незалежно виконуваних заявок пов'язаний з орієнтацією сіткової моделі на опис паралельних підпроцесів в умовах послідовного прийняття керуючих рішень і моделювання на ЕОМ.

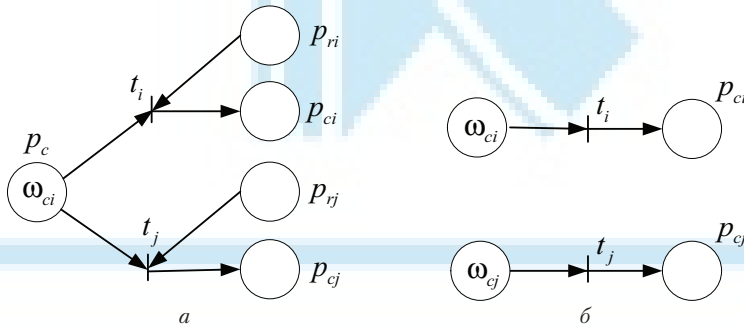


Рис. 7.12. Сіткове подання: а – конфлікту устаткування за обслуговування заявки;

б – незалежно виконуваних операцій

Для розв'язання конфліктів *першого типу* встановлюються пріоритети заявок  $N_1 : D \rightarrow \{1, 2, 3 \dots\}$ , що надходять на обслуговування у кожному конкретному випадку на основі таких правил переваги:

А. Найкоротшої операції:

$$\varphi = \min_{i \in I, j \in J} \tau_{ij},$$

де  $\tau_{ij}$  — час виконання  $j$ -ї за технологічним маршрутом складальної операції виготовлення виробу  $i$ -го типу.

*В. Мінімізації виробничого циклу:*

$$\varphi = \max_{i \in I} \sum_{j=k+1}^{J_i} \tau_{ij},$$

де  $k$  — остання операція, що виконується за технологічним маршрутом.

*С. Мінімальної трудомісткості операцій виготовлення, що залишились:*

$$\varphi = \min_{i \in I} \sum_{j=k+1}^{J_i} \tau_{ij}.$$

*Д. Призначення в порядку надходження:*

$$\varphi = \min_{i \in I, j \in J} \tau_{ij}^*,$$

де  $\tau_{ij}^*$  — момент надходження заявки на виконання  $j$ -ї складальної операції за технологічним маршрутом виготовлення виробу  $i$ -го типу.

*Е. Фіксованих пріоритетів виготовлених виробів:*

$$\varphi = \max_{d_i \in D} PN_1(d_i).$$

Тоді процедура вибору  $R_l$ ,  $l = \overline{1, 3}$  активізованої операції, що входить у конфлікт  $K_l$  першого виду, сформується на основі таких співвідношень:

$$\begin{aligned} & \forall \cdot c_t^i, \cdot c_t^j \in K_1(t) \exists d_i, d_j \in D: [N_1(d_i) > PN_1(d_j) \Rightarrow \\ & \Rightarrow u_t^1(\omega_{ci}) > u_t^1(\omega_{cj}) \Rightarrow R_1(\cdot c_t^i, \cdot c_t^j) = 1]; \cdot c_t^i = [(p_c, \omega_{ci})]; \\ & \forall t_i, t_j \in K_l(p_{h_l}) \exists d_i, d_j \in D: [N_1(d_i) > N_1(d_j) \Rightarrow \\ & \Rightarrow u_{t_i}^l(\omega_{ci}) > u_{t_j}^l(\omega_{cj}) \Rightarrow R_l(t_i, t_j) = 1], l = \overline{2, 3}, p_{h_2} \in P_c, p_{h_3} \in P_r. \end{aligned}$$

Для розв'язання конфлікту **другого типу**  $K_4$  встановлюються пріоритети на порядок надання устаткування  $N_2: M \rightarrow \{1, 2, 3 \dots\}$  на основі таких принципів адресування заявок до вільного технологічного устаткування: а) заявка направляється на найближчий за часом доставки ГВМ; б) заявка направляється на ГВМ із мінімальним поточним завантаженням; в) вибір ГВМ здійснюється випадково. Процедура вибору  $R_4$  активізованої операції, що входить у конфлікт  $K_4$  другого виду, сформується на основі такого співвідношення:

$$\begin{aligned} & \forall t_j, t_l \in K_4(p_c) (\exists m_j, m_l \in M \exists p_j = f_3(m_j), p_l = f_3(m_l): t_j \in p_{r_j}, \\ & t_l \in p_{r_l}): [N_2(m_j) > N_2(m_l) \Rightarrow u_{p_c}^4(t_j) > u_{p_c}^4(t_l) \Rightarrow R_4(t_j, t_l) = 1]. \end{aligned}$$

Конфлікт  $K_5$  **третього типу** розв'язується призначенням пріоритетів  $N_3: \{O_{CK}, O_{HL}, O_{TP}, O_{36}\} \rightarrow \{1, 2, 3, \dots\}$  виходячи з прийнятого першочергового порядку виконання незалежно активізованих операцій і підпроцесів у ВП. Процедура вибору  $R_5$  активізованої операції, що входить у конфлікт  $K_5$  третього виду, сформується на основі таких співвідношень:

$$\begin{aligned} & \forall t_j \in K_g(p_h) \subset K_5, \forall t_l \in K_k(p_i) \subset K_5 (\exists O_j, O_l \in O_{CK} \cup O_{HL} \cup O_{TP} \cup \\ & \cup O_{36} \exists p_{c_j} = f_1(O_j), p_{c_l} = f_1(O_l): t_j \in \cdot p_{c_j}, t_l \in \cdot p_{c_l}, h \neq i): [N_3(O_j) > \\ & > N_3(O_l) \Rightarrow u^5(t_j) > u^5(t_l) \Rightarrow R_5(t_j, t_l) = 1], g, k \in \{2, 3, 4\}. \end{aligned}$$

Враховуючи те, що будь-який перехід сіткової моделі може одночасно належати декільком різним типовим конфліктам, відображення  $\eta$  на множині переходів

подаємо у вигляді сукупності предикатів  $PR$  умов спрацьовування, співвідношення для яких формуються на основі ієрархічної системи розв'язання типових конфліктів («обслуговування заявок» – «надання устаткування» – «вибір незалежних заявок»). Іншими словами, на кожному рівні ієрархії типових конфліктів формується значення локального предиката умови спрацьовування переходу з урахуванням результатів розв'язання на попередніх рівнях. Тоді переходу  $\tilde{t}_j$  сіткової моделі можна поставити у відповідність узагальнений предикат умови спрацьовування такого вигляду:

$$B(\tilde{t}_j) = \bigwedge_{t_j^i \in j, \cdot c_{t_j^i} \in K_1(t_j^i)} q_l(\cdot c_{t_j^i}) \bigwedge_{l=1,5} (A_l(\cdot c_{t_j^i}) \vee \bar{q}_l(t_j^i)),$$

де  $\tilde{t}_j = \{t_j^i \mid t_j^i \in N_i\}$  – перехід-зв'язка;  $\bar{q}_l(t_j^i)$  – логічна умова входження переходу  $t_j^i$  у конфлікт  $l$ -го виду;  $A_l(\cdot c_{t_j^i}) = \bigwedge_{t_h^i \in K_l \setminus \{t_j^i\}} (q(\cdot c_{t_h^i}) \bigwedge_{k=1, l-1} (A_k(\cdot c_{t_h^i}) \vee \bar{q}_k(t_h^i)) \vee R_l(t_j^i, t_h^i))$  – локальний предикат умови спрацьовування переходу  $t_j^i$  за розподілом  $\cdot c_{t_j^i}$  в ієрархічній системі розв'язання конфлікту  $l$ -го виду,  $l = \overline{2, 5}$ ;  $q(\cdot c_{t_j^i})$  – логічна умова збудження переходу  $t_j^i$  за розподілом  $\cdot c_{t_j^i}$ ;  $A_1(\cdot c_{t_j^i}) = \bigwedge_{\cdot c_{t_j^i} \in K_1(t_j^i) \setminus \{\cdot c_{t_j^i}\}} (q(\cdot c_{t_j^i}) \vee R_1(\cdot c_{t_j^i}, \cdot c_{t_j^i}'))$ .

**Означення 7.38.** Сітковою моделлю керуючого процесу називається предикатна  $T$ -сітка такого вигляду:

$$N_K = (N_{\text{вп}}, PR),$$

де  $PR = \{pr_j \mid pr_j = B(\tilde{t}_j), j = \overline{1, |T|}\}$  – множина предикатів умов спрацьовування переходів.

При формалізації й побудові сіткових моделей алгоритмів керування (СМАК) на основі керуючого процесу (КП) необхідно забезпечити асинхронне узгодження взаємодій між операторами керування так, щоб моменти ініціювання виробничих операцій визначалися лише готовністю необхідних даних щодо об'єкта керування. У цьому випадку керуюча підсистема ГКІС формує керуючі впливи  $y_k$  у вигляді команд активізації виконання відповідних виробничих операцій в ОК, після чого він працює автономно, а керуюча підсистема очікує сигнал  $x_l$  від нього про закінчення виконання цих операцій. Тоді кожен оператор-дія, що виконує процедуру керування деякою виробничою операцією певної тривалості, у моделі виробничого процесу замінюється фрагментом сітки, зображеним на рис. 7.13, де  $w_i, v_i$  – зовнішні позиції, що відображають відповідно видачу керуючого впливу і прийом сигналу щодо стану устаткування. Таким чином, для опису СМАК на базі моделі КП  $N_K$  введемо розгортку  $T$ -сітки, у якій визначимо спеціальні множини позицій, що являють собою деякі вхідний і вихідний алфавіти, що характеризують асинхронні взаємодії з ОК.

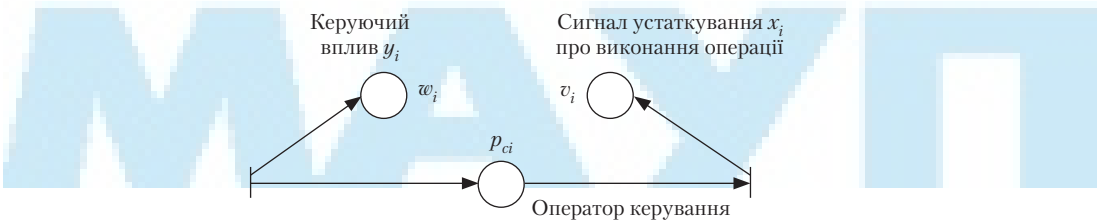


Рис. 7.13. Сіткове подання процедури керування виробничою операцією



**Означення 7.39.** Сітковою моделлю алгоритму керування називається розгортка предикатної  $T$ -сітки, що задається набором такого вигляду:

$$N_A = (N_K, X, Y, V, W, f, \varphi),$$

де  $X = \{x_l \mid x_l = (b_1, b_2, b_3, b_4), l = \overline{1, L}\}$  – множина сигналів щодо стану устаткування;  $b_i$  – логічні змінні, що відповідно вказують на несправність, зайнятість, налагодженість і стан роботи;  $Y = \{y_k \mid y_k = (d, par_1, par_2), k = \overline{1, K}\}$  – множина керуючих впливів;  $d \in D, par_1, par_2$  – керуючі параметри;

$$V = \left\{ v_j \mid \forall p_{cj} \in P_c \left( \exists \omega_l \in \bigcup_{i=0}^p \Omega_i : z(p_{cj}, \omega_l) \neq 0 \right) \exists v_j : v_j = p_{cj} \right\};$$

$$W = \left\{ w_j \mid \forall p_{cj} \in P_c \left( \exists \omega_k \in \bigcup_{i=0}^p \Omega_i : z(p_{cj}, \omega_k) \neq 0 \right) \exists w_j : w_j = p_{cj} \right\} \text{ – множини зовнішніх}$$

позицій;  $f : V \times \bigcup_{i=0}^p \Omega_i \rightarrow X$ ;  $\varphi : W \times \bigcup_{i=0}^p \Omega_i \rightarrow Y$ .

Сіткову модель алгоритму керування, сформовану на базі моделі виробничого процесу досліджуваної ГКІС, зображено на рис. 7.14, а функції однозначної відповідності  $f$  та  $\varphi$  подано в табл. 7.3 і 7.4.

Зазначимо, що для виділення рівнів керування обслуговуванням ГВМ структуру сіткової моделі процесу розподілено на два рівні з поданням моделей алгоритмів керування ГВМ у вигляді окремих блоків, у яких розміщено всі інформаційно-керуючі впливи.

Таблиця 7.3

**Відповідність зовнішніх позицій сіткової моделі операторам керування**

Позиція	Оператор керування	Зміст
$w_1$	$y_1 = (w_c^{бд}, l, k)$	Виконати складальну операцію на ГВМ <sub><i>i</i></sub> над <i>k</i> деталями виду $w_c^{бд}$ за керуючою програмою <i>l</i>
$w_8$	$y_8 = (w_c^{бд}, H_{РШ}, AC)$	Транспортувати за допомогою РШ готові вузли типу $w_c^{бд}$ з $H_{РШ}$ на АС
$w_{11}$	$y_{11} = (w_c^i, AC, H_{РШ})$	Транспортувати за допомогою РШ комплект інструменту типу $w_c^i$ з АС на $H_{РШ}$ для налагодження ГВМ <sub><i>i</i></sub> на заявку від АС
$w_{19}$	$y_{19} = (w_c^i, H_{РШ}, AC)$	Транспортувати за допомогою РШ використаний комплект інструменту типу $w_c^i$ з $H_{РШ}$ на АС після налагодження ГВМ <sub><i>i</i></sub> на заявку від АС
$w_{51}$	$y_{51} = (w_c^i, AC, H_{РШ})$	Транспортувати за допомогою РШ комплект інструменту типу $w_c^i$ з АС на $H_{РШ}$ для налагодження ГВМ <sub><i>i</i></sub> на заявку від ГВМ <sub><i>j</i></sub>
$w_{59}$	$y_{59} = (w_c^i, H_{РШ}, AC)$	Транспортувати за допомогою РШ використаний комплект інструменту типу $w_c^i$ з $H_{РШ}$ на АС після налагодження ГВМ <sub><i>i</i></sub> на заявку від ГВМ <sub><i>j</i></sub>

Згідно з принципом подій керування момент часу керування відповідає моменту надходження сигналу про готовність устаткування  $x_l$  (поміщення в позицію  $v_l$  мітки-сигналу).

Відповідність зовнішніх позицій сіткової моделі сигналам щодо виконання виробничих операцій

Позиція	Код сигналу $x = b_1, b_2, b_3, b_4$	Зміст
$v_1$	$x_1 = 1110$	Завершення складальної операції на ГВМ <sub>i</sub>
$v_8$	$x_8 = 1010$	Завершення транспортування за допомогою РШ готових вузлів до комірки АС
$v_{11}$	$x_{11} = 1011$	Завершення транспортування за допомогою РШ комплексу інструменту до $H_{РШ}$
$v_{19}$	$x_{19} = 1100$	Завершення транспортування за допомогою РШ комплексу інструменту до комірки АС
$v_{51}$	$x_{51} = 0111$	Завершення транспортування за допомогою РШ комплексу інструменту до $H_{РШ}$
$v_{59}$	$x_{59} = 0110$	Завершення транспортування за допомогою РШ комплексу інструменту до комірки АС

Тоді задача керування за сітковою моделлю полягає у виборі в цей момент однієї з множини готових до виконання операцій (вибрати за істинним значенням предиката  $pr_j$ ; відповідний перехід спрацьовування) і вказуванні шляху її реалізації через задавання керуючого впливу  $y_k$  (помістити в позицію  $w_k$  мітку керування). Множина предикатів сіткової моделі  $N_A$  формалізує стратегії керування, які вибираються на етапі імітаційного моделювання ГКІС.



Рис. 7.14. Сіткова модель алгоритму керування виробничим процесом

Таким чином, **третій етап проектування СК ГКІС – метод синтезу** сіткових моделей коректних алгоритмів керування на основі отриманого й перевіреного опису ВП можна подати у вигляді послідовності таких підетапів:

1. Сформувати структуру КП і визначити склад необхідних операторів керування операціями ВП.
2. За допомогою предикатів умов спрацьовування переходів усунути всі конфлікти, що виникають через загальні ресурси й альтернативний характер ВП.
3. Вибрати всі оператори-дії, для яких сформувати набори керуючих впливів і сигналів щодо стану відповідного устаткування.
4. Сформувати для кожного оператора-дії сіткове подання асинхронної взаємодії з виробничим устаткуванням.

У кожному конкретному випадку правила призначення пріоритетів під час формування множини предикатів вибираються за результатами імітаційних експериментів над сітковою моделлю ВП і порівняння за вибраним критерієм. Економіко-організаційній сутності такої задачі більшою мірою відповідає критерій мінімізації загального циклу виготовлення заданих виробничою програмою  $\Pi = \{ \Pi_i \mid i = 1, \overline{D_B} \}$  виробів, тому що він забезпечує безперервне й рівномірне завантаження ГВМ, мінімізацію загального строку виготовлення й розміру незавершеного виробництва, а також можливих відхилень від встановленого плану [20]. Крім того, використання правил віддавання переваги передбачає ефективну організацію керування згідно із ситуацією за умови прямого диспетчерського функціонування ГКІС.

Узагальнюючи наведене до використання сіток Петрі в задачах програмування та керування виробничими системами, можна констатувати, що *задача автоматизованого синтезу алгоритмів керування процесами у ГКІС* зазвичай розглядається як така, що складається з двох підзадач: *формування процедур керування, що відображають допустимі реалізації керованих процесів*, і *розробки алгоритмів керування, що відображають реалізації керованих процесів*. Сутністю першої задачі є розробка алгоритму, який за вихідними даними, тобто деякою специфікацією керованої системи, побудує динамічну модель поведінки системи керування; сутність другої – пошук стратегії керування, яка приводить до оптимальної реалізації керованого процесу, тобто реалізації, яка задовольняє задані критерії якості в термінах оцінки поведінки системи. Використовуючи деякі прийоми організації послідовності здійснення логічного аналізу опису для основних елементів системи, що зберігаються у БД, з подальшою їх інтерпретацією у сітках Петрі, а також перевіркою та усуненням суперечностей у поданні логіки керування, можна розробити методики синтезу алгоритмів керування для задач автоматизованого програмування виробничих контролерів і побудови інтерактивних систем проектування та планування виробничих процесів.

### Контрольні завдання

1. Обґрунтувати доцільність використання апарату СП для моделювання процесів ГКІС.
2. Навести класифікацію та охарактеризувати структуру процесів ГКІС.
3. Дати означення класичної (базової) СП і розкрити складові четвірки СП.
4. Навести визначення графа і означити функцію маркування СП.
5. Охарактеризувати основні властивості СП: досяжність, обмеженість, безтупиковість, безпечність.
6. Розкрити особливості побудови дерева досяжності та процедури аналізу СП за деревом досяжності.

7. Пояснити, в чому полягають умови покривності, живучості та пріоритетності в СП.
8. Дати означення інгібаторної СП та обґрунтувати роль таких сіток в аналізі процесів.
9. Навести означення та окреслити область застосування розфарбованої СП.
10. Охарактеризувати два напрями модифікації СП як об'єктно-орієнтованого розширення під задачі моделювання процесів у складних об'єктах.
11. Пояснити, в чому полягає підхід щодо структурного перетворення СП.
12. Дати означення і прокоментувати формальне подання сіткових блоків, зв'язуючої сітки, локалізованого вузла, ієрархічно розфарбованої СП з типових блоків.
13. Навести тлумачення першої модифікації СП – формування автоматичного, умовного, паралельного, циклічного та конвеєрного типових блоків.
14. Сформулювати та прокоментувати умови живучості та безпечності розфарбованої СП, що складається з типових блоків.
15. Пояснити, в чому полягає друга модифікація СП як перший етап проектування системи керування ГКІС.
16. Дати означення  $T$ -сітки, типової розфарбованої СП і сіткової моделі виробничого процесу.
17. Навести основні риси другого етапу проектування системи керування ГКІС.
18. Пояснити, в чому полягають перевірки живучості, безпеки, правильності параметризації та структуризації для  $T$ -сіткової моделі під час аналізу умов організації виробничого процесу за цією моделлю.
19. Охарактеризувати ознаки третього етапу проектування системи керування ГКІС – синтезу алгоритмів керування.
20. Навести і прокоментувати типові конфліктні ситуації у ході виробничого процесу і виокремити методи розв'язання конфліктів на переходах відображаючої СП.
21. Розкрити механізм встановлення пріоритетів заявок на обслуговування в разі виникнення конфліктів першого, другого і третього типів і розкрити зміст цих конфліктів.
22. Дати означення сіткової моделі алгоритму керування і прокоментувати його складові кроки.
23. Розкрити послідовність реалізації третього етапу проектування системи керування ГКІС – синтезу моделей коректних алгоритмів керування.
24. Розкрити зміст основних підзадач автоматизованого синтезу алгоритмів керування процесами у ГКІС.



## ІНТЕЛЕКТУАЛІЗАЦІЯ МОДЕЛЮВАННЯ СКЛАДНИХ ОБ'ЄКТІВ НА ОСНОВІ РОЗШИРЕНЬ АПАРАТУ СП

*Не презирай совета ничьего,  
но прежде рассмотри его.*

Іван Крилов

*Можна пожалкувати про німе кіно,  
де прийнято було бачити жінку,  
яка відкриває рота,  
проте її не було чути.*

Чарлі Чаплін

### 8.1. Логічні сітки Петрі

**Формування логічного виведення на основі модифікованих сіток Петрі.** Для реалізації процедури логічного виведення як у межах пропозиційної логіки, так і в більш загальному випадку логіки предикатів запропоновано модифікації СП, які називаються *логічними СП* (ЛСП) [83; 125; 140].

Враховуючи характер параметрів і характеристик ОМ – виробничих процесів, скористаємося апаратом ЛСП для інтерпретування умов у межах пропозиційної логіки, розглядаючи дві модифікації ЛСП – прості та нечіткі.

**Означення 8.1.** *Простою ЛСП (ПЛСП)* є сітка Петрі  $N$ , в основі якої лежить відповідність переходів  $T$ -сітки продукційним правилам, вхідних позицій переходів – *логічним умовам* (посиланням), а вихідних позицій – *висновкам* (виведенням), тобто:

$$N_{\text{л.пр}} = (P, T, F, H, \mu_0, C, S, \beta, \gamma), \quad (8.1)$$

де  $P, T$  – скінченні множини позицій і переходів відповідно (причому  $|P| = n; |T| = m$ ;  $F: P \times T \rightarrow \{\emptyset, 1\}$ ,  $H: T \times P \rightarrow \{\emptyset, 1\}$  – вхідна та вихідна функції відповідно (функції інцидентності);  $\mu_0: P \rightarrow \{\emptyset, 1\}$  – початкове маркування;  $C, S$  – скінченні множини пропорційних змінних і продукційних правил відповідно (причому  $|C| = n; |S| = m$ . Антецедент кожного з правил подається кон'юнкцією тверджень  $c_j, c_j \in C$ ; консеквент – також твердження, що належить  $C$ );  $\beta: P \rightarrow C$ ;  $\gamma: T \rightarrow S$  – функції позначень (відповідності позицій пропорційним змінним і переходів – продукційним правилам).

Домовимось позначати позицію  $p$  та змінну  $c$ , що з нею зв'язана, однаковими індексами (тобто  $\forall p_i \in P: \beta(p_i) = c_i$ ). Це саме стосуватиметься також переходів і правил.

Наявність маркера в позиції  $p_i$  інтерпретується як свідчення істинності висловлювання  $c_i$ . Ненульове маркування всіх вхідних позицій переходу  $t_j$  свідчить про істинність антецедента правила  $s_j$ . Про перехід, що має ненульове маркування всіх вхідних позицій, кажуть, що він є *збудженим*, якщо принаймні одна з його вихідних позицій має нульове маркування.

Збуджений перехід може спрацювати. При цьому він приписує значення 1 маркуванню усіх вихідних позицій (тобто значенням консеквента є значення «Істина»), тоді як вхідні позиції переходу зберігають своє маркування ( $\forall p_i \in t_j$ ) ( $\mu(p_i) = 1$ ).

Позиції переходу, що асоціюються з початковими умовами (тобто ті, істинні значення яких (маркування) встановлюються перед виконанням виведення), називаються *позиціями-джерелами*; множина всіх таких позицій позначається  $P_0$ . Шлях у ПЛСП — це така послідовність  $p_a t_j p_k t_l \dots t_x p_b$ , що  $p_a \in t_j$ ,  $t_j \in p_k$ , ...,  $t_x \in p_b$ , де  $\forall i, j : (i \neq j) \rightarrow (p_i \neq p_j)$ ,  $(i \neq j) \rightarrow (t_i = t_j)$ , за винятком, можливо,  $p_a = p_b$  (в цьому разі шлях називається *циклом*).

*Довжина шляху* — це кількість позицій переходу за винятком однієї. *Імплікантним* називається шлях  $L$ , в якому позиція або перехід не зустрічаються більше одного разу. Позначимо через  $W(p_i)$  довжину максимального імплікантного шляху, що з'єднує будь-яку з позицій-джерел із позицією  $p_i$ . При цьому вважаємо, що кожна позиція є або джерелом, або існує шлях, що веде до неї від будь-якої позиції-джерела.

**Означення 8.2.** *Нечіткою ЛСП (НЛСП)* є ПЛСП, для якої вводяться додатково вагова  $h$  та граничного рівня спрацьовування переходу  $d$  функції, а саме:

$$N_{\text{л.н}} = (N_{\text{л.сп}}, h, d), \quad (8.2)$$

де  $h : P \times T \rightarrow \{\emptyset, 1\}$ ; ( $\forall p_i, t_j : (F(p_i, t_j) = 0) \rightarrow (h(p_i, t_j) = 0)$ );  $d : T \rightarrow \{\emptyset, 1\}$ . Крім того,  $\mu_0 : P \rightarrow \{\emptyset, 1\}$ . Маркування  $\mu(p_i)$  задає в НЛСП нечітке значення істинності  $c_i$  (іншими словами, можна вважати, що маркування в НЛСП задає ступінь належності фактичного стану НЛСП до нечіткої множини станів, яка відповідає змінній  $c_i$ ).

**Загальний алгоритм виведення на основі ЛСП.** У загальному випадку організація виведення на основі ЛСП аналогічна традиційній схемі функціонування СП і полягає в послідовному виконанні всіх можливих спрацьовувань переходів до стану, коли збуджених переходів немає.

**Твердження 8.1.** *Кількість етапів спрацьовувань переходів, необхідних для досягнення скінченного маркування будь-якої з позицій  $p_i$  в ЛСП, не перевищує  $W(p_i)$ .*

**Висновок 8.1.** Кількість етапів спрацьовувань переходів при повному виведенні на ЛСП не перевищує  $M^*$ , де  $M^* = \max(W(p_i))$ .

Наступне твердження констатує незалежність результату від стратегії виконання ЛСП.

**Твердження 8.2.** *Нехай  $N_{\text{л}}$  — ЛСП;  $\mu^G$  — скінченне маркування, утворене застосуванням загального алгоритму виведення до  $N_{\text{л}}$ ;  $\mu^A$  — маркування, що утворилося після застосування до  $N_{\text{л}}$  будь-якої іншої процедури виконання дій в ЛСП. Якщо при  $\mu^A$  збуджених переходів немає, то  $\mu^A = \mu^G$ .*

**Висновок 8.2.** Якщо ЛСП  $N'$  і  $N''$  однакові, а їх поточні маркування не мають збуджених переходів, то ці маркування будуть однаковими за умови, що вихідні маркування збігаються (інакше кажучи, різні процедури виконання ЛСП, що вводять до стану без збуджених переходів, якщо вони застосовуються до однакових сіток, дають однакові скінченні маркування).

Саме твердження 8.1–8.2 і відповідні висновки лежать в основі побудови логічних виведень за допомогою відповідних модифікацій ЛСП. При цьому загальний алгоритм виведення є доцільним за наявності циклів у ЛСП, тоді як для ациклічних ЛСП з метою підвищення ефективності виведення застосовуються запропоновані у праці [140] схеми «швидких» алгоритмів матричного і блочного виведень в ЛСП.

**Матричне виведення.** За наявності циклів у ЛСП доцільно застосовувати розглянутий вище загальний алгоритм виведення. Схеми виведення, що розглядаються нижче, стосуються циклічних ЛСП і забезпечують більш ефективне (швидке) виведення, оскільки дають змогу обійтись без повторних перевірок кожного з переходів, гарантуючи виведення завдяки однократному послідовному встановленню маркування позицій.

Розглянемо два типи ЛСП: структуру  $N'$ , для якої  $\forall p_i : |p_i| \leq 1$ ; структуру  $N''$ , для якої  $\exists p_i : |p_i| > 1$ . Зауважимо, що коли в  $N'$  є цикли, що містять позиції тільки з нульовим початковим маркуванням, всі переходи таких циклів будуть «мертвими»; при цьому загальний алгоритм виведення також завершиться невдало.

1. **Структура  $N'$ .** Оскільки маркування позиції визначається кон'юнкцією входів єдиного переходу, є можливість виведення без явного врахування переходів, тобто за прямими зв'язками між позиціями.

Сформуємо допоміжну матрицю  $A = I \cdot O$ , де  $I \cdot O$  – логічний добуток матриць:

$$A_{ij} = (I_{i1} \wedge O_{1j}) \vee (I_{i2} \wedge O_{2j}) \vee \dots \vee (I_{im} \wedge O_{mj}). \quad (8.3)$$

Згідно з (8.3) одиниці  $j$ -го стовпця матриці  $c$  розміщуються в рядках, пов'язаних з передумовами для змінної  $c_j$ .

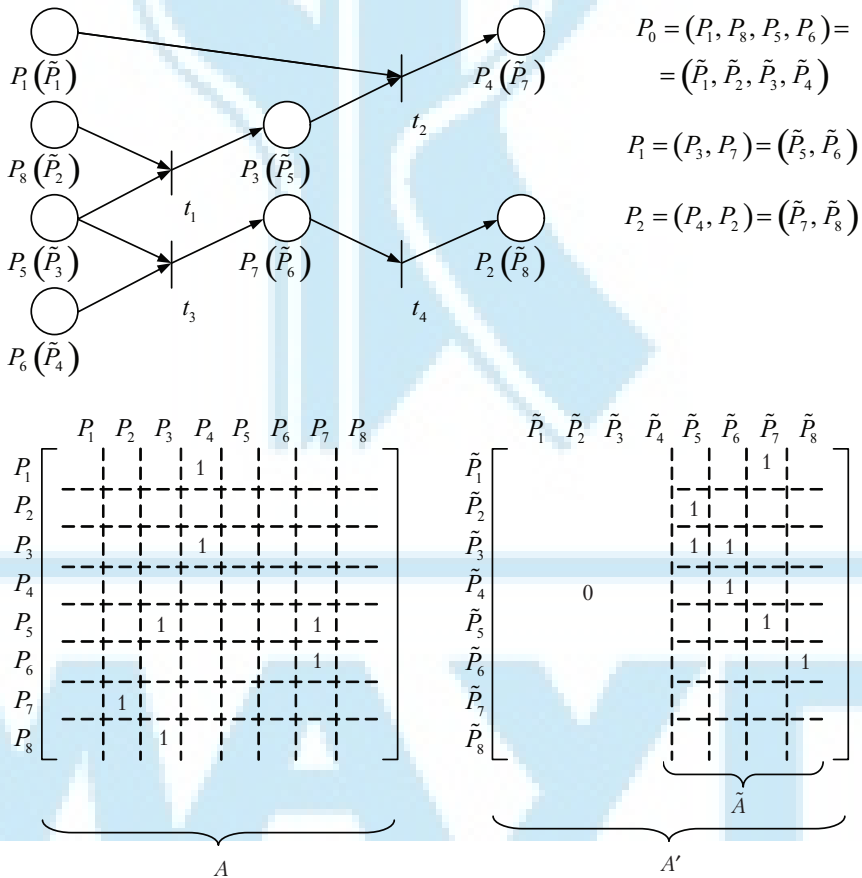


Рис. 8.1. Формування груп позицій та матриці виведення

Сформуємо групи позицій  $P_1 = (p_i | W(p_i) = 1)$ , ...,  $P_{M^*} = (p_i | W(p_i) = M^*)$ , вважаючи, що  $\forall p_i \in P_0 : |p_i| = \emptyset$ , де позиція  $P_0$  містить позиції-джерела. Зауважимо, що  $\forall i, j : (i \neq j) \rightarrow (p_i \cap p_j = \emptyset)$ ,  $\bigcup p_i = P$ .

Пронумеруємо позиції за збільшенням номера їх групи, а в межах груп – у довільному порядку (при новій нумерації писатимемо  $\tilde{p}_i$  замість  $p_i$ ):

$$\forall \tilde{p}_i \in P_k, \forall \tilde{p}_j \in P_m : (k < m) \rightarrow (i < j).$$

Сформуємо також новий вектор маркування  $\tilde{\mu}$ .

Перетворимо матрицю  $A$  в  $A'$  відповідно до зміненої нумерації позицій. Вилучивши перші  $|P_0|$  стовпців з  $A'$ , дістанемо матрицю  $\tilde{A}$ , яку називатимемо *матрицею виведення* (описане ілюструє рис. 8.1).

Процедура виведення із застосуванням матриці  $\tilde{A}$  схожа на множення вектора на одиницю, але є дві відмінності. *По-перше*, тепер кожний добуток вектора  $\tilde{\mu}$  і поточного стовпця матриці  $\tilde{A}$  не дає наступного елемента окремого результуючого вектора, а тільки заміняє відповідний елемент вектора  $\tilde{\mu}$ , який потім уже в модифікованому вигляді множиться на наступний стовпець матриці  $\tilde{A}$  і т. д. Ілюстрацію процедури формування груп позицій та матриці виведення наведено на рис. 8.2.

*По-друге*, відмінною є форма розрахунку нового значення елемента.

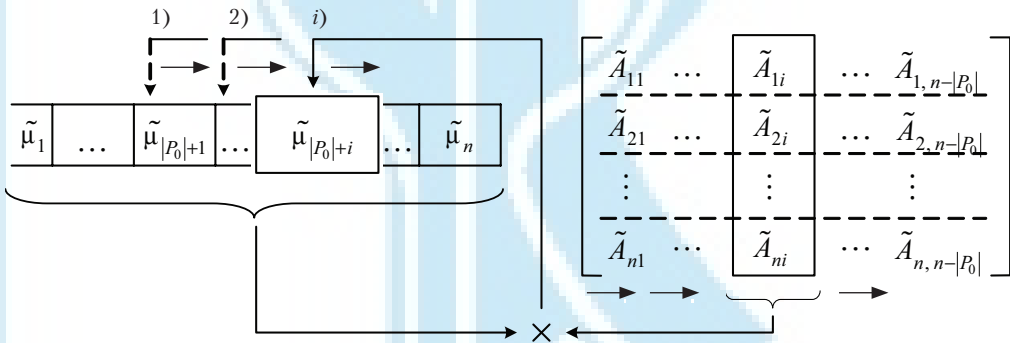


Рис. 8.2. До розгляду процедури матричного виведення

Ввівши позначення  $Q_i = (p_i | \tilde{A}_{i, j-|P_0|} = 1)$ , стосовно простої ЛСП матимемо:

$$\tilde{\mu}_j^* = \begin{cases} 1, & \text{якщо } \forall p_j \in Q : \tilde{\mu}(\tilde{p}_j) = 1; \\ \emptyset, & \text{у протилежному разі} \end{cases}$$

Для забезпечення роботи в нечіткій ЛСП пов'яжемо функцію граничного рівня кожного переходу з його вихідною позицією, ввівши функцію

$$d'(\tilde{p}_j) = d(t_h), \text{ де } \{\tilde{p}_j\} = t_k.$$

Аналогічно введемо вагову функцію  $h'$ , тобто

$$h'(\tilde{p}_i, \tilde{p}_j) = h(\tilde{p}_i, t_k).$$

Тепер позначимо

$$\gamma = \min_{\tilde{p}_i \in Q_j} (h'(\tilde{p}_i, \tilde{p}_j)) \cdot \tilde{\mu}(\tilde{p}_i), \tag{8.4}$$

і тоді для НЛСП матимемо:

$$\tilde{\mu}_j^* = \begin{cases} \gamma, & \text{якщо } \gamma \geq d(\tilde{p}_j); \\ \emptyset, & \text{у протилежному разі} \end{cases}$$



**Алгоритм 8.1** (формування виведення в ЛСП):

1. Установити початкове маркування позицій-джерел  $j := |P_0| + 1$ .
2. Поки ( $\gamma \leq n$ ), виконувати:

розрахунок  $\tilde{\mu}_j^*$ ;

модифікування вектора  $\tilde{\mu} : \tilde{\mu}_j := \tilde{\mu}_j^*, j = j + 1$ .

3. Кінець.

Прийнятність такого підходу впливає з *твердження* 8.1. Справді, маркування позицій  $p_i$  не може змінитися, якщо всі позиції, що належать множині  $\Omega_i = \{p_j | p_j \cap p_i \neq \emptyset\}$ , вже мають скінченне маркування. Оскільки в ациклічній ЛСП ( $\forall k > 0$ ) ( $\forall p_i \in P_k$ ) ( $\Omega_i \subseteq P_1 \cup P_2 \cup \dots \cup P_{k-1}$ ), то послідовність  $P_1 \rightarrow P_2 \rightarrow \dots \rightarrow P_{M^*}$  (а саме вона реалізується в наведеному підході) прийнятна для забезпечення виведення в ЛСП.

2. **Структура  $N''$** . Наявність у позицій альтернативних (диз'юнктивних) вхідних дуг від різних переходів потребує явного виділення кожного з них. Аналогічно розглянутій вище структурі  $N'$  сформуємо в ЛСП групи позицій  $P_1, P_2, \dots, P_{M^*}$ , а також групи переходів  $T_1, T_2, \dots, T_{M^*}$ :

$$T_i = \left\{ t_j \mid (t_j \cap P_{i-1} \neq \emptyset) (\forall i \leq M) \left[ (l \geq i) \rightarrow (P_l \cap t_j = \emptyset) \right] \right\}.$$

Пронумеруємо переходи за збільшенням номера їх груп, а в межах груп – у довільному порядку. Відповідно до змін номерів позицій і переходів перетворимо матриці  $I, O, h$  і вектори  $\bar{\mu}, \bar{n}, \bar{d}$  в  $\tilde{I}, \tilde{O}, \tilde{h}, \tilde{\mu}, \tilde{n}, \tilde{d}$ :

$$T_1 = \{t_1, t_2, t_4\} = \{\tilde{t}_1, \tilde{t}_2, \tilde{t}_3\}; T_2 = \{t_3\} = \{\tilde{t}_4\}$$

(див. рис. 8.1).

Кожній групі  $P_i$  позицій припишемо два числа:  $\xi(P_i) = \min_{p_k \in P_i} k$ ;  $\chi(P_i) = \max_{p_k \in P_i} k$ . Аналогічні числа припишемо групам переходів. Зауважимо, що  $\xi(P_{M^*}) = n$ ,  $\chi(T_{M^*}) = m$ . Тоді алгоритм виведення в ЛСП полягатиме у виконанні серій коригувань векторів  $\tilde{\mu}$  і  $\tilde{n}$  по чергово для груп позицій (переходів) у кожній із серій у порядку збільшення номерів груп.

$$I = \begin{bmatrix} 1100 \\ 0010 \\ 0000 \\ 1000 \\ 0000 \\ 0101 \\ 0010 \\ 0001 \end{bmatrix}; \tilde{I} = \begin{bmatrix} 1100 \\ 1000 \\ 0110 \\ 0010 \\ 0001 \\ 0000 \\ 0001 \\ 0000 \end{bmatrix}; O = \begin{bmatrix} 01000000 \\ 00100000 \\ 00001000 \\ 00000010 \end{bmatrix}; \tilde{O} = \begin{bmatrix} 00001000 \\ 00000100 \\ 00000010 \\ 00000001 \end{bmatrix}.$$

**Алгоритм 8.2** (формування виведення в ЛСП):

1. Установити початкове маркування позицій-джерел  $i := 1$ .
2. Поки ( $i \leq M^*$ ), виконувати  $j := \xi(T_i)$ .
3. Поки ( $j \leq \chi(T_i)$ ), виконувати:

розрахунок  $\tilde{n}_j^*$ .

4. Поки ( $j \leq \chi(P_i)$ ), виконувати:

розрахунок  $\tilde{\mu}_j^*$ ;

модифікування вектора  $\tilde{\mu} : \tilde{\mu}_j := \tilde{\mu}_j^*, j = j + 1$ .

Кінець  $i := i + 1$ .

5. Кінець.

У наведеному алгоритмі  $\tilde{n}_j^*$  обчислюється аналогічно розрахунку вектора  $\tilde{\mu}_j^*$  для ПЛСП типу  $N'$  або розраховується як параметр  $\gamma$  (див. (8.4)) для НЛСП типу  $N'$ , причому  $\tilde{I}, \tilde{h}, \tilde{d}$  використовуються відповідно замість  $\tilde{A}, h', d'$ , тоді як вектор  $\tilde{\mu}_j^*$  обчислюється за формулою

$$\tilde{\mu}_i^* = \max_{\tilde{t}_s \in Q_i'} \{ \tilde{n}(\tilde{t}_s) \}, \text{ де } Q_i' = \{ \tilde{t}_k \mid (\tilde{t}_k \in \tilde{P}_i) (\tilde{n}(\tilde{t}_k) \geq \tilde{d}(\tilde{t}_k)) \}.$$

Зокрема, для ПЛСП можна застосувати формулу  $\mu_n^*(\tilde{P}_j) = \sigma(\tilde{n}^H \tilde{Q}_j)$ , де

$$\sigma(x) = \begin{cases} 1 & \text{при } x \geq \emptyset; \\ \emptyset & \text{при } x < \emptyset. \end{cases}$$

Наведений алгоритм у його початковому варіанті передбачає встановлення груп елементів по черзі для векторів  $\tilde{n}^H$  і  $\tilde{\mu}^H$  при розподіленому використанні матриць  $\tilde{I}$  та  $\tilde{O}$ .

Можна запропонувати таку організацію виведення на основі мішаної матриці інцидентності  $E$ . В ході виведення послідовно використовуються її стовпці з першого до  $(n + m - |P_0|)$ -го. Алгоритм виведення аналогічний алгоритму 8.2; необхідні зміни випливають зі структури матриці  $E$ :

$$\left. \begin{array}{cccccccccccc} t_i, P_i & \tilde{t}_1 & \tilde{t}_2 & \dots & \tilde{t}_{1n_+}^{T_1} & \tilde{P}_{n_-}^{P_1} & \dots & \tilde{P}_{n_+}^{P_1} & \tilde{t}_{n_-}^{T_2} & \dots & \tilde{t}_{n_+}^{T_2} & \dots & \dots & \tilde{P}_n \\ i=1 & \left[ \begin{array}{cccccccccccc} \tilde{I}_{11} & \tilde{I}_{12} & \dots & \tilde{I}_{1n_+}^{T_1} & \tilde{Q}_{1n_-}^{P_1} & \dots & \tilde{Q}_{1n_+}^{P_1} & \tilde{I}_{1n_-}^{T_2} & \dots & \tilde{I}_{1n_+}^{T_2} & \dots & \dots & \tilde{O}_{1n} \\ \dots & \tilde{I}_{21} & \tilde{I}_{22} & \dots & \tilde{I}_{2n_+}^{T_1} & \tilde{Q}_{2n_-}^{P_1} & \dots & \tilde{Q}_{2n_+}^{P_1} & \tilde{I}_{2n_-}^{T_2} & \dots & \tilde{I}_{2n_+}^{T_2} & \dots & \dots & \tilde{O}_{2n} \\ \dots & \tilde{I}_{31} & \tilde{I}_{32} & \dots & \tilde{I}_{3n_+}^{T_1} & \tilde{Q}_{3n_-}^{P_1} & \dots & \tilde{Q}_{3n_+}^{P_1} & \tilde{I}_{3n_-}^{T_2} & \dots & \tilde{I}_{3n_+}^{T_2} & \dots & \dots & \tilde{O}_{3n} \\ \dots & & & \dots & & & \dots & & & & & & & & \end{array} \right] \\ \dots & & & & & & & & & & & & & & \end{array} \right\} \max(m, n)$$

$\underbrace{\hspace{15em}}_{m+n-|P_0|}$

У цій матриці, якщо  $m \neq n$ , то стовпці, що складаються з елементів  $\tilde{I}_{ij}$  (при  $n < m$ ) або  $\tilde{O}_{ij}$  (при  $n > m$ ), доповнюються знизу нулями. Аналогічно нулями доповнюються вектори — відповідно  $\tilde{n}^H$  або  $\tilde{\mu}^H$ .

В окремих випадках (наприклад, при малих вимірностях сітки) можна застосувати спільне подання позицій і переходів на основі допоміжного вектора  $\tilde{u}$  та мішаної матриці  $D$ :

$$\tilde{u} = (u_1, \dots, u_{n+m-|P_0|}) = (\mu^n(\tilde{P}_1), \dots, \mu^n(\tilde{P}_{n_+}^{P_0}), n^n(\tilde{t}_1), \dots, \dots, \mu^n(\tilde{P}_n));$$

$$D = \left[ \begin{array}{cccccccccccc} \tilde{t}_1 & \dots & \tilde{t}_{n_+}^{T_1} & \tilde{P}_{n_-}^{P_1} & \dots & \tilde{P}_{n_+}^{P_1} & \tilde{t}_{n_-}^{T_2} & \dots & \tilde{t}_{n_+}^{T_2} & \dots & \dots & \tilde{P}_n \\ \dots & I_{ij} & \dots & 0 & \dots & \dots & \tilde{I}_{kl} & \dots & \dots & \dots & \dots & 0 \\ 0 & & & \dots & \tilde{O}_{ps} & \dots & 0 & \dots & \dots & \dots & \dots & \tilde{O}_{wv} \\ 0 & & & 0 & & & & & & & & 0 \\ 0 & & & 0 & & & 0 & & 0 & \dots & \tilde{O}_{wz} & \dots \\ 0 & & & 0 & & & 0 & & 0 & & 0 & \dots \end{array} \right] \left. \vphantom{\begin{array}{c} \\ \\ \\ \\ \end{array}} \right\} n+m$$

$\underbrace{\hspace{15em}}_{m+n-|P_0|}$

У цьому разі виведення полягає в послідовному коригуванні вектора  $\tilde{u}$ ; вектор  $u_i^*$  визначається як результат поелементних дій над вектором  $\tilde{u}$  та відповідним стовпцем матриці  $D$  з урахуванням того, чому відповідає вектор  $u_i^*$  – переходу чи позиції.

Для ефективного розрахунку НЛСП в матрицях  $E, D$  замість елементів  $\tilde{I}_{ij}$  можна відразу зазначити значення елементів  $\tilde{h}_{ij}$ . Ці матриці доцільно також записувати в розрідженому вигляді.

**Блочне виведення.** В основі виведення лежать поділ ЛСП на окремі неперетинні фрагменти (блоки) та призначення правил їх взаємодії. В межах блоку виведення здійснюється відповідно до розглянутих вище алгоритмів.

Блочне виведення є корисним не тільки для сіток великої вимірності, а й для забезпечення наочності та зручності (шляхом відокремлення окремих змістових фрагментів) виділення фрагментів сіток різних типів у загальній мішаній сітці з подальшою організацією матричного виведення. Воно дає можливість при зміні істинності посилань запобігати обов'язковим переглядам усіх переходів сітки, обмежившись виконанням окремих фрагментів при локалізації змін маркувань тільки в них. Перевагою підходу є можливість компонувати з окремих фрагментів різні процедури; при цьому в конкретній ситуації активізуються тільки блоки, що забезпечують необхідну процедуру.

Можливі два варіанти використання блочної структури ЛСП: при *повному* (щоразу виконуються всі блоки сітки) та при *частковому* (виконується частина блоків, утворюючи окрему процедуру) *виведеннях*. Часткове виведення відповідає повному виведенню з підсітки, яку утворено вибраними блоками. Розглянемо особливості повного виведення.

**Формування блоків.** Блок  $b_i$  в сітці  $N$  є її підсіткою, утвореною множиною позицій  $p_i \subset P$  та множиною переходів  $t_i \subset T$ . Якщо перехід інцидентний до позицій кількох блоків, то умовно вважатимемо його таким, що належить кожному з блоків (це є адекватним дублюванню переходу).

Нехай блоки сформовано так, що

$$(\forall b_i \in N)(\forall t_j \in b_i)(t_j \subset b_j) \left[ \left( t_j \subset b_i \right) \vee \left( t_j \cap b_i = \emptyset \right) \right].$$

Виконання цієї умови зумовлює незалежність виведення в кожному блоці.

**Означення 8.3.** *Вхідною для певного блоку* називатимемо позицію  $\cdot p_i$  блоку  $b_k$ , якщо вона є *джерелом*, тобто  $(\exists t_j)(t_j \in \cdot p_i)(\cdot t_j \cap b_k = \emptyset)$ .

**Означення 8.4.** *Вихідною для певного блоку* називатимемо позицію  $p_i \cdot$  блоку  $b_k$ , якщо вона є *витоком*, тобто  $(\exists t_l)(t_l \in p_i \cdot)(t_l \notin b_k)$ .

**Означення 8.5.** *Вхідним для певного блоку* називатимемо перехід  $\cdot t_r$  блоку  $b_k$ , якщо для цього переходу справедливою є умова  $(t_r \in b_k \cdot)(\cdot t_r \cap b_k = \emptyset)$ .

Множину вхідних позицій блоку  $b_i$  позначатимемо  $T_{b_i}$ . Будемо також вважати, що в деякій групі блоків всі вхідні позиції є джерелами (такі блоки називаються ще *витоками*, їх множину позначатимемо  $B_{дж}$ ), а в решті — жодна з позицій не є джерелом. Блоки, що не є витоками, мають нульове початкове маркування всіх позицій. Приклад формування блочної структури ЛСП зображено на рис. 8.3, а.

Підсумкова блочна структура ЛСП може містити і не містити циклів, причому тут нема однозначної відповідності з початковою СП. Так, цикл, що є у початковій СП, може виявитися охопленим одним блоком і, таким чином, не бути репрезентованим у блочній структурі, і навпаки, виділення блоків у сітці без циклів може спричинити цикл між блоками (рис. 8.3, б).

Кожному блоку  $b_i$  поставимо у відповідність множини його вхідних і вихідних блоків – відповідно  $\cdot b_i$  та  $b_i^*$ , тобто:

$$\cdot b_i = \{b_j \mid (\exists t_l \in b_i) (\cdot t_l \cap b_j \neq \emptyset) (b_j \neq b_i)\},$$

$$b_i^* = \{b_j \mid (\exists t_l \in b_j) (\cdot t_l \cap b_i \neq \emptyset) (b_j \neq b_i)\}.$$

До того ж у деяких блоках може бути  $\cdot b_k = \emptyset$  (у витоках) або  $b_k^* = \emptyset$ .

**Організація виведення.** Вважатимемо, що блок буде виконаний, якщо здійснюється виведення з усієї відповідної підсітки (тобто забезпечено всі можливі спрацьовування переходів), причому при виведенні принаймні одне спрацьовування є можливим.

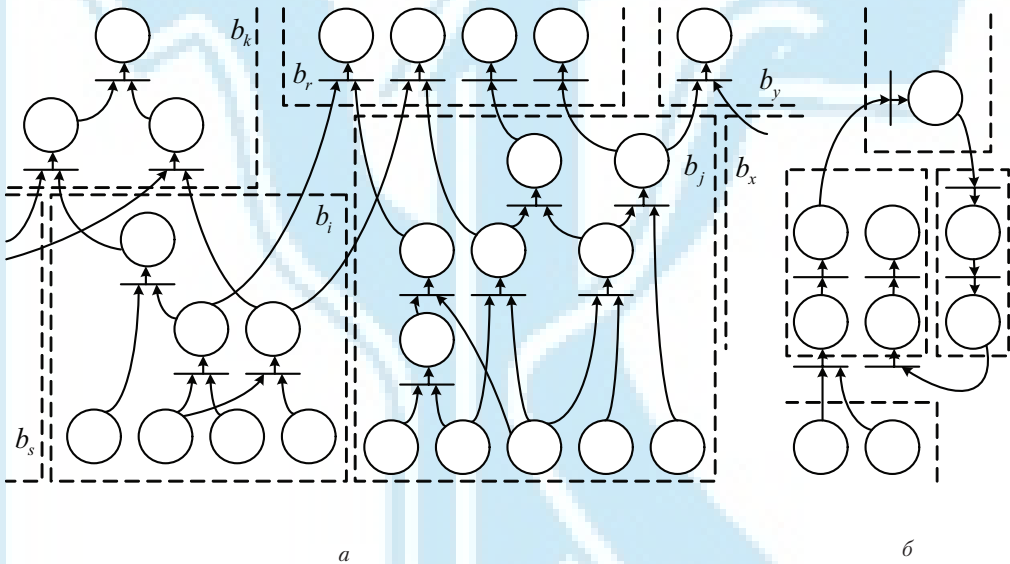


Рис. 8.3. Приклади формування блочної структури ЛСП (а) та виділення блоків у сітці з циклом між ними (б)

У загальному випадку виведення з блочної структури подання ЛСП полягає у виборі по черзі блоків і спробі виконання кожного з них. Процедура повторюється доти, поки жоден з блоків не можна буде виконати.

Порядок вибору блоків може бути довільним. Однак для чергового виконання блоку  $b_i$  необхідним є спрацьовування принаймні одного з його вхідних переходів, оскільки саме вони можуть змінити маркування, встановлене перед першим виконанням або після попереднього виконання блоку  $b_i$ . У свою чергу, умови для нових спрацьовувань вхідних переходів створюються лише при виконанні блоків з  $\cdot b_i$ . Отже, доцільно переглядати і, якщо можна, виконувати блоки в міру реалізації вхідних блоків  $\cdot b_i$ .

Для контролю черговості переглядів і виконання окремих блоків передбачимо такі елементи:  $\text{time}$  – умовний лічильник поточного часу;  $L_i$  – момент останнього перегляду блоку  $b_i$ ;  $F_i$  – момент останнього виконання цього блоку;  $V'$  – допоміжна множина (список). Тоді можна запропонувати алгоритм організації блочного виведення, що розкривається нижче.

**Алгоритм 8.3** (організація блочного виведення):

1. Установити маркування вхідних позицій витокових блоків, а також

$$(\forall b_i)(F_i := \emptyset), (L_i := \emptyset), B' := \emptyset, \text{time} := \emptyset.$$

2. Виконати всі можливі блоки  $b_i \in B'$ ; встановити  $\text{time} := \text{time} + 1$ , потім для всіх виконаних блоків задати  $F_i := \text{time}$ ,  $L_i := \text{time}$  і занести це в  $B'$ .
3. Якщо  $B' = \emptyset$ , то кінець.
4. Вибрати блок  $b_i \in B'$ ; сформуванати множину  $g_i = \{b_j \mid (b_j \in b_i)(L_j < F_j)\}$ .
5. Якщо  $g_i = \emptyset$ , то вивести  $b_i$  з  $B'$  і перейти до п. 3.
6. Вибрати блок  $b_p \in g_i$  і визначити можливість його виконання (тобто перевірити умови спрацювання вхідних переходів); установити  $\text{time} := \text{time} + 1$ , а потім  $L_p := \text{time}$ . Якщо п. 6 виконати неможливо, то перейти до п. 3.
7. Виконати блок  $b_p$ ; установити  $F_p := \text{time}$ ; занести  $b_p$  у  $B'$  (за умови, що він там ще не зазначений); перейти до п. 4.

Ефективність наведеного алгоритму можна підвищити, якщо модифікувати його другий та сьомий кроки, записавши таке:

- 2<sup>1</sup>. Виконати всі можливі блоки  $b_i \in B'_{\text{дж}}$ ; встановити  $\text{time} := \text{time} + 1$ , після чого для всіх виконаних блоків задати  $F_i := \text{time}$ ,  $L_i := \text{time}$ . Занести виконані блоки, в яких змінилося маркування вихідних позицій, до списку  $B'$ .
- 7<sup>1</sup>. Виконати блок  $b_p$ ; встановити  $F_p := \text{time}$ ; якщо відбулася зміна маркування вихідних позицій блоку, то занести це в  $B'$  (за умови, що він там ще не зазначений); перейти до п. 4.

У загальному випадку кожна вхідна позиція блоку  $b_i$  може бути інцидентною до кількох його вхідних переходів. Отже, важливо, щоб існували різні набори вхідних переходів, у кожному з яких для кожної вхідної позиції є інцидентний їй перехід.

Такому набору відповідає деяке сполучення, що складається з блоків  $b_i \in \cdot b_p$ , які пов'язані з переходами цього набору. Таке сполучення називатимемо *альтернативним вхідним сполученням* блоку  $b_i$  і позначатимемо  $e_i^j$ , де  $j=1, n_{b_i}$ , а  $n_{b_i}$  — кількість

альтернативних вхідних сполучень блоку  $b_i$ . Спостерігається:  $\bigcup_j \left( \bigcup_{b_i \in e_i^j} b_i \right) = \cdot b_i$ .

**Твердження 8.3.** На однакових ЛСП за умови, що початкові маркування їх однакові, виконання алгоритму 8.3 та загального алгоритму виведення завершується скінченням маркуваннями.

Враховуючи *твердження 8.2*, достатньо довести, що після виконання *алгоритму 8.3* збуджених переходів у ЛСП не буде. Припустимо, що перехід  $t_i$  є внутрішнім переходом блоку  $b_j$  (тобто  $\cdot t_i \subset b_j$ ). Якщо блок  $b_j$  вже був виконаний, то збуджений перехід  $t_i$  не може залишитися без спрацювання при виконанні загального чи матричного алгоритму виведення, що суперечить початковій передумові.

Але якщо  $b_j$  все-таки не спрацював, то збудження переходу  $t_i$  могло відбуватися лише за умови ненульового маркування деякої початкової позиції в блоці  $b_j$ , а це суперечить змісту виконання *алгоритму 8.3*, оскільки його робота з блоком починається саме з розгляду початкових позицій (див. п. 1 *алгоритму 8.3*).

Припустимо, що перехід  $t_i$  є вхідним переходом блоку  $b_j$ . Збудження переходу  $t_i$  могло відбуватися лише за умови зміни маркування вихідних (кінцевих) позицій деякого блоку  $b_l \in \cdot b_j$ . Збуджений стан переходу  $t_i$  суперечить тому факту, що *алгоритм 8.3* завершено (оскільки згідно з цим алгоритмом успішне виконання блоку  $b_l \in \cdot b_j$  привело до перевірки та виконання блоку  $b_j$  — див. п. 7 цього алгоритму).

Після перевірки та виконання окремих блоків ЛСП може спрямовуватися зовнішньою процедурою, побудованою на основі *алгоритму* 8.3, або реалізуватися допоміжною керуючою сіткою, що моделює цей алгоритм. Останній підхід забезпечує додаткові можливості для аналізу та управління процесом виведення.

Щоб підвищити ефективність виведення й одночасно зменшити можливу кількість безрезультатних спроб виконання блоків ЛСП, доцільно переглянути блок (включити його в множину  $g_i$  п. 4 *алгоритму* 8.3) лише за умови, що хоча б одне з його вхідних сполучень повністю складається з блоків, які зазнали спроб виконання. Проте необхідно стежити за тим, щоби блоки, які за *алгоритмом* 8.3 зазнали цих спроб, не були тепер пропущені.

**Означення 8.6.** Блок називається умовно виконаним, якщо він зазнав спроб виведення, але його вхідні переходи не спрацювали.

**Означення 8.7.** Блок  $b_i$  називається активним, якщо  $b_i \in B_{\text{дж}}$  і до позицій  $p_i \in X_{b_i}$  виконано встановлення початкового маркування (можливо, й нульового) або якщо  $(\exists e_i^j)(\forall b_l \in e_i^j)$  (« $b_l$  – виконаний»  $\vee$  « $b_l$  – умовно виконаний»). Блоки називаються пасивними, якщо вони не є активними.

Альтернативні вхідні сполучення блоку  $b_i$ , що складаються з виконаних і/або умовно виконаних блоків, позначатимемо  $\tilde{e}_i^j$ , а всю їх множину –  $\tilde{E}_i$ .

Тепер вимагатимемо, щоби спроби виконання стосувалися тільки активних блоків.

**Означення 8.8.** Блок  $b_i$  називається умовно живучим, якщо в блочній структурі ЛСП можлива така послідовність умовних і/або фактичних виконань блоків, яка приводить до активізації блоку  $b_i$ .

**Означення 8.9.** Блочна структура ЛСП є умовно живучою, якщо всі її блоки – умовно живучі.

Умовна живучість блочної структури ЛСП означає, що навіть при вказаних обмеженнях щодо вибору блоків для їх виконання гарантується можливість такої спроби для кожного блоку.

**Означення 8.10.** Шлях у блочній структурі ЛСП – це така послідовність  $B_L = b^{\text{п}} b_1 b_i \dots b_j b^{\text{к}}$ , що  $(b^{\text{п}} \in \cdot b_p, b_i \in \cdot b_p, \dots, b_j \in \cdot b^{\text{к}})$ . При цьому вважається, що жоден з блоків не входить у  $B_L$  більш як один раз, за винятком, можливо, випадку  $b^{\text{п}} = b^{\text{к}}$ , коли шлях є цикловим. Такий шлях позначимо як  $V_{\text{ц}}$ .

Тепер введемо поняття *сукупності перетинних циклів* (СПЦ):

$$(\text{СПЦ}) = V_{\text{ц}} \mid (\text{СПЦ}) \cap V_L,$$

де для  $V_L$  маємо:

$$b^{\text{п}} \in (\text{СПЦ}), b^{\text{к}} \in (\text{СПЦ}) \text{ (можливо, } b^{\text{п}} = b^{\text{к}}).$$

**Твердження 8.4.** Для умовної живучості блочної структури ЛСП необхідно й достатньо, щоби кожна (СПЦ) містила блок з альтернативним вхідним сполученням, що складається з блоків, які до (СПЦ) не належать:

$$\forall (\text{СПЦ})_i (\exists b_j \in (\text{СПЦ})_i) (\exists e_j^k) (e_j^k \cap (\text{СПЦ})_i) = \emptyset.$$

Для виведення в блочній структурі ЛСП, що задовольняє умову *твердження* 8.4, доцільно модифікувати *алгоритм* 8.3, змінивши правило формування множини  $g_i$  п. 4:

4. Вибрати блок  $b_i \in V'$ ; сформувати множину

$$g_i = \{b_i \mid (\exists \tilde{e}_i^r \in \tilde{E}_i)(b_i \in \tilde{e}_i^r)\}.$$

Тепер розглянемо особливості виведення залежно від варіанта блочної структури ЛСП.

**Варіант 1. Відсутні (СПЦ).** Така блочна структура є умовно живучою, і застосовувати модифікований алгоритм 8.3 доцільно. При цьому для управління послідовністю переглядів і виконання блоків, що належать до блочної структури ЛСП (рис. 8.4, а), зручно використати відповідну допоміжну СП (рис. 8.4, б), сформовану так, щоб кожному блоку  $b_i$  в ній відповідав перехід  $t_i$ , а множині  $\bullet b_i$  — множина вхідних позицій  $\bullet p_i$  переходу  $t_i$ .

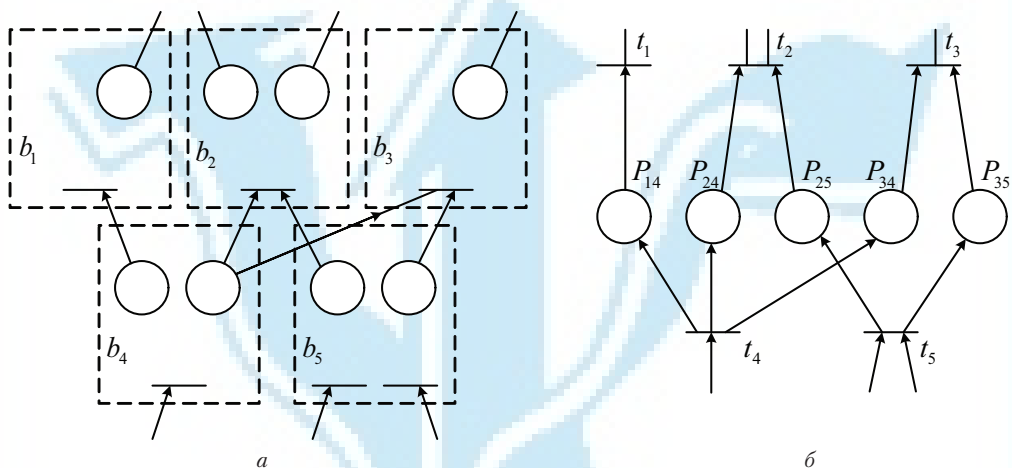


Рис. 8.4. Фрагмент блочної структури ЛСП (а) та відповідна допоміжна СП (б)

**Варіант 2. Маємо (СПЦ)  $i (\forall b_i) (n_{b_i} \leq 1)$ .** У цьому разі блочна структура є умовно живучою. Для виведення слід застосувати алгоритм 8.3.

**Варіант 3. Маємо (СПЦ)  $i (\forall b_i) (n_{b_i} > 1)$ .** Якщо блочна структура є умовно живучою, то треба застосувати модифікацію алгоритму 8.3, у протилежному разі — алгоритм 8.3.

## 8.2. Орієнтоване на інтелектуалізоване моделювання розширення апарату СП

**Модифіковані навантаження СП.** Тепер з'ясуємо можливості СП-моделювання при інтерпретації ієрархічних розподілених процесів у складних об'єктах і відображенні динаміки їх проходження. Зокрема, при описі ГКІС за допомогою СП для розв'язання задач оперативного керування істотним обмеженням може бути наявність у кожній партії деталей власного технологічного маршруту (ТМ):

$$M_i = (m_{i_1}, m_{i_2}, \dots, m_{i_k}, \dots, m_{i_j}); \quad (8.5)$$

$$\tau_i = (\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}, \dots, \tau_{i_j}), \quad (8.6)$$

де  $m_{i_k}$ ,  $\tau$  — тип устаткування та час обробки на  $k$ -му етапі ТМ відповідно.

Складність формалізування підвищується, якщо при цьому задано номери інструментальних налагоджень  $H_i$ , тобто

$$H_i = (H_{i_1}, H_{i_2}, \dots, H_{i_k}, H_{i_j}), \quad (8.7)$$

а також якщо кількість носіїв з деталями  $S_i$  є різною на окремих етапах (це може бути тоді, коли частину деталей було оброблено перед черговим складанням плану):

$$S_i = (s_{i_1}, s_{i_2}, \dots, s_{i_k}, \dots, s_{i_l}). \quad (8.8)$$

Часто в робототехнічній системі (РТС) треба призначати відносні пріоритети  $PR_i$  щодо різних партій (наприклад, при складанні за методом послідовного нарощування):

$$PR_i = (pr_{i_1}, pr_{i_2}, \dots, pr_{i_k}, \dots, pr_{i_l}) \quad (8.9)$$

Додаткова вимога до моделі може полягати в урахуванні тривалості переналагоджувань

$$\tau_{j'j''}^{\text{нал}} = \|\tau_{j'j''}^{\text{нал}}\|, \quad (8.10)$$

де  $\tau_{j'j''}^{\text{нал}}$  — час переходу від інструментального налагодження  $j'$  до іншого  $j''$ .

В цьому випадку звичайні СП і розглянуті вище розширення не відображають усієї множинності зазначених властивостей ГКІС. Тому аналогічно мові навантажених графів ( $P$ -технології проектування послідовних програм) для подання моделей розподілених структур процесів, ресурсів, алгоритмів і програм впроваджено мову, побудовану на основі модифікованих навантажених СП [101; 125; 139].

**Означення 8.11.** *Навантаженою  $N_n$  сіткою Петрі (НСП) є звичайна сітка Петрі  $N$ , доповнена функціями  $E$  навантажувального відображення і  $\tau$  визначення часу спрацьовування переходів, а саме:*

$$N_n = (N, Q, E, \tau), \quad (8.11)$$

де  $Q$  — множина описувачів маркерів;  $E$  — навантажувальне відображення, в якому  $E_1 : A_1 \rightarrow \eta_1$ ,  $E_2 : A_2 \rightarrow \eta_2$ ,  $E = E_1 \cup E_2$ ,  $\eta_1 = \{y\}$  — множина керуючих функцій;  $\eta_2 = \{\eta_{ij}\}$ ;  $\tau : T \times q_\tau \rightarrow R$  (причому  $R$  — множина невід'ємних раціональних чисел;  $q_\tau$  — атрибут мітки  $m$ , який керує часом).

За умови, коли в НСП спостерігається  $Q = E = 0$ , термін НСП звужується до поняття *часової СП* (ЧСП), формальне подання якої може здійснюватися за двома принципами:

- **перший принцип** полягає в тому, що кожному переходу  $t$  ставиться у відповідність пара величин  $a$  та  $l$  таких, що  $0 \leq a \leq l$  [124]. Якщо перехід  $t$  активізований на момент часу  $\tau$ , то він не може бути реалізований раніше, аніж у момент часу  $(\tau + a)$ , але повинен бути реалізований до або в момент часу  $(\tau + l)$ , якщо тільки умови його активізації не зміняться в цей проміжок через реалізацію інших переходів. Це так звані *тайм-аутні СП*, і їх використовують для опису і перевірки протоколів обчислювальних мереж з механізмами тайм-аутів, а також для опису систем із властивостями відновлення;
- **другий принцип** застосовується для побудови часових СП-моделей (ЧСП) ГКІС, коли кожному переходу СП ставиться у відповідність певний інтервал часу  $\tau_j$ , що дорівнює тривалості його реалізації, тобто довгочасності зіставленої з ним операції [113].

**Означення 8.12.** *Часовою  $N_\tau$  сіткою Петрі є звичайна сітка Петрі  $N$ , доповнена функцією  $\tau : T \rightarrow R$  часів спрацьовування переходів, за якою для кожного переходу зіставляється постійний час спрацьовування  $\tau_j \in \theta$  ( $\theta$  — множина часів спрацьовування переходів), тобто*

$$N_\tau = N, \tau. \quad (8.12)$$



Множини  $P$ ,  $T$  і їх відображення  $P \times T$ ;  $T \times P$  однозначно визначаються структурою описуваної ГКІС, а множина  $\theta$  і функція  $\tau$  – часовими показниками роботи компонентів ГКІС на виконавчому рівні керування останньою.

Якщо перехід збуджується, то мітки, що викликали його пуск, покидають вхідні позиції  $PRE(t_i)$ . Породження міток у вихідних позиціях  $POST(t_i)$  відбувається через час  $\tau_i$ .

Обмеженнями другого принципу формального подання ГКІС у вигляді ЧСП є: апіорна незмінність часів  $\tau_j \in \theta$  виконання основних і допоміжних виробничих операцій; використання найближчих і усереднюючих статистичних оцінок тривалості роботи технологічного устаткування (в результаті знижується ступінь адекватності ЧСП-моделі ОМ щодо оптимальності за швидкодією керування реальною ГКІС, а не моделлю); можливість виникнення випадків неіснування розв'язків задачі організації оперативного керування ГКІС за необхідності синхронізації роботи устаткування в умовах мінімізації часу на виконання виробничого завдання:

$$\tau_{\Pi}(c_i) \leq [\tau_{\Pi}(c_i)] | c_i \in C, \quad (8.13)$$

де  $\tau_{\Pi}$  – тривалість виконання завдання;  $[\tau_{\Pi}(c_i)]$  – максимально допустима довгочасність його виконання;  $c_i$  – варіант плану-графіка;  $C$  – множина допустимих розкладів завантаження устаткування.

У цьому разі при заданих значеннях  $P$ ,  $T$ ,  $\theta$ ,  $P \times T$ ,  $T \times P$  задача організації оперативного управління ГКІС, що задовольняє умову (8.13), може не мати розв'язку. Один із шляхів одержання коректного розв'язку полягає у зміні складових набору  $\theta = \{\tau_1, \tau_2, \dots, \tau_p, \dots, \tau_n\}$  і проведенні повторного динамічного навантаження моделі. Цього можна досягти зміною або методів керування (зокрема ПР, технологічним устаткуванням тощо), або параметрів пристроїв на виконавчому рівні.

Для вирішення зазначених проблем використання ЧСП-моделі ГКІС необхідно враховувати зв'язок між рівнями управління (рис. 8.5), що дасть змогу: шляхом аналізу виконавчого рівня здійснити коректне навантаження НСП-моделі ГКІС і тим самим підвищити ступінь її адекватності об'єкту моделювання; в разі необхідності провести повторне динамічне навантаження НСП-моделі після зміни часових показників процесів, що відбуваються на виконавчому рівні управління.

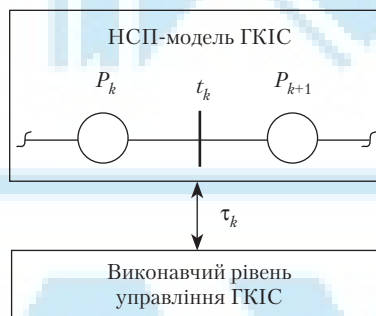


Рис. 8.5. Зв'язок між рівнями управління ГКІС

Модифікована таким чином НСП-модель дістала назву *ієрархічної НСП-моделі* (ІНСП-модель) ГКІС [140; 142].

**Означення 8.13.** Динамічне навантаження СП-моделі – це таке відображення двох скінченних непорожніх множин переходів  $T = \{t_1, t_2, \dots, t_n\}$  і часів спрацьовувань переходів  $\theta = \{\tau_1, \tau_2, \dots, \tau_n\}$  сітки, що  $T \times \theta \equiv \theta \times T$ .

**Означення 8.14.** *Навантажена СП-модель ГКІС* — це модель, яку одержано за допомогою процедури динамічного навантаження СП-моделі, що відповідає ГКІС.

**Означення 8.15.** *Ієрархічно-навантажена СП-модель (ІНСП-модель) ГКІС* — це модель, побудована на основі аналізу процесів, які відбуваються на рівнях керування ГКІС, та взаємодії цих процесів.

При цьому можливості роботи ІНСП-моделі в реальному часі забезпечуються автоматизацією динамічного навантаження СП-моделей виконавчого рівня ГКІС, яке може бути реалізоване шляхом використання модуля планування експериментів (МПЕ), що передбачає побудову замкненого контуру керування експериментами в межах власних засобів зовнішнього програмного забезпечення (див. твердження 4.1 і 4.2).

### 8.3. Організація динамічного навантаження при плануванні і проведенні експериментів на ІНСП-моделі

**СП-моделі в автоматизації проведення експериментів з ГКІС.** Побудова систем автоматизованого управління експериментами, яку розглянуто в підрозд. 4.2–4.4, передбачає створення замкненого програмно-реалізованого контуру управління СП-моделлю в межах засобів зовнішнього програмного забезпечення [125; 140]. Спрямовані серії експериментів відповідно до заданої мети та з урахуванням обмежень змін параметрів моделі ГКІС організовують спеціально введені до складу зовнішнього програмного забезпечення модулі, які в загальному випадку задають набори початкових даних, ініціюють прогони моделі, обробляють результати і приймають рішення про дальший перебіг експериментів згідно з алгоритмом моделювання, що реалізується.

Функціональну компоненту системи автоматизації управління експериментами відповідно до конкретної стратегії виконує пошуковий модуль ПМ, який є інтелектуальною надбудовою СП-моделі ГКІС. Модуль ПМ і СП-модель, що становлять замкнений контур, утворюють автоматизовану СП-модель, результатом функціонування якої є алгоритм оперативного управління ГКІС. Цей алгоритм синтезується ПМ шляхом багаторазового прогону СП-моделі ГКІС.

У термінах СП задачу синтезу алгоритму управління ГКІС можна описати так [139]. Поставимо у відповідність ГКІС деяку СП  $N$ , стан якої характеризується її маркуванням. Управління ГКІС можна відобразити функцією її стану  $Y = f(X)$ , де  $X = \{\mu_i\} \mid i = \overline{1, m}$ ;  $Y = \{y_j\} \mid j = \overline{1, n}$ ;  $i, j$  — множини станів об'єкта керування ОК (в цьому разі ГКІС), що визначаються маркуванням СП, а також операторів управління ГКІС відповідно.

Тоді в інтерпретації СП ця задача зводиться до необхідності управління переходами сітки залежно від маркування її позицій. Відтак алгоритм управління РТС як ОК можна задати у формі модифікованої СП, тобто

$$N_a = (N, \mu, \varphi, Y, f) \quad (8.14)$$

за умов

$$\forall p \in P, \forall \mu \in R(N): \mu(p) \leq 1; \quad (8.15)$$

$$\forall \mu_i, \mu_j \in R(N) \exists \sigma_l: \mu_i \xrightarrow{\sigma_l} \mu_j; \quad (8.16)$$

$$\forall t \in T \exists \mu_k, \mu_l \in R(N): \mu_k \xrightarrow{\sigma_t} \mu_l, \quad (8.17)$$

де  $N$  – СП, що визначає структуру алгоритму управління;  $\mu$  – множина станів ОК;  $\sigma_t$  – послідовність запуску переходів;  $\varphi : T \rightarrow 2^\mu$  – функція позначення переходів, яка кожному з них ставить в однозначну відповідність оператор або сукупність операторів алгоритму управління;  $2^\mu$  – множина всіх підмножин  $\mu$ ;  $Y$  – множина операторів управління;  $f : \mu \rightarrow Y$  – функція, яка кожному маркуванню ставить в однозначну відповідність підмножину станів ОК.

Наведені умови визначають СП  $N_a$  як правильну. Зокрема, умова (8.15) означає, що сітка  $N_a$  є безпечною; умови (8.16) і (8.17) визначають цю сітку як живучу. При цьому задача синтезу алгоритму управління ГКІС полягає у відшуванні компонентів набору (8.14), що задовольняють умови (8.15)–(8.17).

Тепер сформулюємо означення деяких понять, які далі використовуватимуться.

**Означення 8.16.** *Пошуковий модуль ПМ* – це функціональна компонента автоматизованої системи СП-моделювання, що організовує певну серію експериментів з СП-моделлю відповідно до заданої стратегії за певним методом пошуку, для якого СП-модель виступає як засіб розрахунку заданих показників.

**Означення 8.17.** *Автоматизована СП-модель* – це замкнений контур, який складається з СП-моделі, ПМ, а також МЛВ.

**Самонавантажувальні СП-моделі.** При використанні автоматизованих ІНСП-моделей ГКІС можна запропонувати таку послідовність синтезу алгоритмів оперативного управління останніми:

1. Проведення серії експериментів на виконавчому рівні управління ГКІС з метою одержання коректних часових показників роботи ПР та іншого технологічного устаткування.
2. Динамічне навантаження СП-моделі ГКІС одержаними часовими показниками.
3. Синтез алгоритму оперативного управління ГКІС за допомогою ПМ.
4. У разі необхідності (наприклад, якщо синтезований алгоритм не приводить до задоволення умови (8.13)) – проведення повторної серії експериментів на виконавчому рівні з метою інтенсифікації процесів, які там відбуваються, та одержання нових часових показників динамічного навантаження СП-моделі ГКІС.

Схему планування експериментів на виконавчому рівні управління ГКІС показано на рис. 8.6.



Рис. 8.6. Схема планування експериментів на виконавчому рівні управління ГКІС

Згідно з нею дослідник формує стратегію організації експериментів відповідно до мети, що визначається вхідним словом  $W_1$ . Останнє задається автоматизованою СП-моделлю ГКІС і може мати вигляд задач: одержання коректних часових показників роботи ПР та іншого технологічного устаткування; зменшення використовуваних часових показників. При цьому вихідне слово  $W_2$  відповідає множині  $\theta = \{\tau_1, \tau_2, \dots, \tau_n\}$  і застосовується для динамічного навантаження автоматизованої СП-моделі ГКІС.

Тоді зображена на рис. 8.6 схема планування експериментів як складова ІНСП-моделі ГКІС повністю відобразить властиві їй функції в загальній схемі організації динамічного навантаження ІНСП-моделей ГКІС (рис. 8.7).

Наявність людини-дослідника в *модулі планування експериментів* (МПЕ) суттєво обмежує можливості ІНСП-моделей ГКІС щодо професійних здібностей дослідника, його досвіду, можливості роботи ІНСП-моделі в реальному часі, достовірності оцінок результатів проведених експериментів тощо.

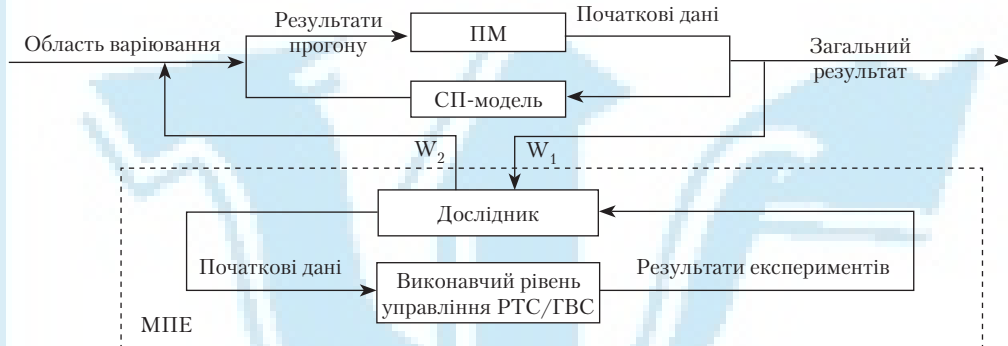


Рис. 8.7. Загальна схема організації динамічного навантаження ІНСП-моделей ГКІС

Вирішення цієї проблеми забезпечується автоматизацією динамічного навантаження СП-моделей виконавчого рівня ГКІС, коли аналогічно верхньому рівню автоматизації СП-моделі ГКІС створення МПЕ передбачає побудову замкнутого контуру управління експериментами в межах власних засобів зовнішнього програмного забезпечення виконавчого рівня. При цьому функції дослідника на виконавчому рівні виконує спеціально введений до складу зовнішнього програмного забезпечення пошуковий модуль ПМ2, що є інтелектуальною надбудовою МПЕ. У загальному випадку до складу задач цього модуля входять формування множини  $\theta = \{\tau_1, \tau_2, \dots, \tau_n\}$  і динамічне навантаження СП-моделі верхнього рівня управління ГКІС обґрунтованими значеннями складових  $\theta$ .

СП-модель ГКІС, що містить МПЕ з інтелектуальною надбудовою, називатимемо *самонавантажувальною*.

**Означення 8.18.** Самонавантажувальна СП-модель (СНСП-модель) – це ієрархічна автоматизована СП-модель, що взаємодіє з МПЕ на виконавчому рівні управління через програмно-реалізовані інтелектуальні надбудови – ПМ.

Метою функціонування СНСП-моделі є розв’язання задачі синтезу алгоритму оперативного управління ГКІС з одночасним впливом на динаміку процесів, які відбуваються на виконавчому рівні управління.

З урахуванням викладеного можна сформулювати такі вимоги до структури та функціональних модулів СНСП-моделей ГКІС:

1. Структура СНСП-моделі ГКІС повинна поєднувати ІНСП-модель і МПЕ на виконавчому рівні управління.

2. ІНСП-модель РТС/ГВС має виконувати такі функції: динамічне навантаження СП-моделі на основі аналізу вихідного (з МПЕ) слова  $W_2$ ; синтез алгоритму оперативного управління ГКІС; формування вхідного (до МПЕ) слова  $W_1$ . Перелічені функції повинна реалізувати інтелектуальна надбудова ІНСП-моделі – пошуковий модуль ПМ1.

3. МПЕ на виконавчому рівні управління ГКІС повинен забезпечувати формування стратегії планування експериментів на основі аналізу вхідного слова  $W_1$ , проведення серії експериментів за цією стратегією та формування вихідного слова  $W_2$ . Ці функції має виконувати інтелектуальна надбудова МПЕ – пошуковий модуль ПМ2.

4. НСП-модель і МПЕ повинні неперервно взаємодіяти через ПМ за певним алгоритмом обміну й обробки інформації, що формується на кожному з рівнів у вигляді вхідних і вихідних слів.

Зазначені вище вимоги задовольняє структура СНСП-моделі ГКІС (рис. 8.8). Стосовно одного з основних її компонентів – ПР логічну структуру розв'язання задач побудови СНСП-моделей можна розбити на три етапи.

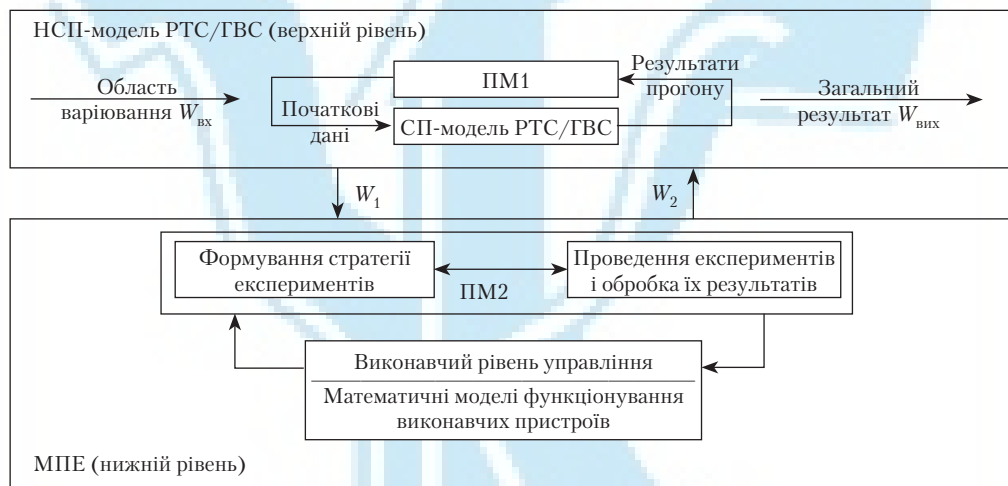


Рис. 8.8. Схема взаємодії між елементами СНСП-моделі ГКІС

**Перший етап.** Враховуються два аспекти: змістові основи побудови виконавчого рівня управління ПР; розроблення алгоритмів керування ПР в реальному часі на виконавчому рівні.

Подання структури виконавчого рівня у вигляді ієрархічної сукупності кількох підрівнів суттєво полегшить формування стратегії проведення експериментів з ПР на другому етапі.

Алгоритми реального часу для керування ПР будуть потрібні, якщо вхідне слово  $W_1$  матиме вигляд задачі зменшення часових показників роботи ПР, тобто інтенсифікації процесів, які відбуваються на виконавчому рівні управління ГКІС. Тут необхідно зауважити, що інтенсифікація може бути забезпечена двома шляхами: вибором технічних засобів з найбільш прийнятними фізичними показниками (наприклад, інерційністю, точністю, швидкодією тощо); поліпшенням якості керування технічними засобами, що використовуються (наприклад, завдяки оптимальному плануванню рухів маніпуляційної системи ПР).

У подальшому приділимо увагу другому шляху інтенсифікації процесів на виконавчому рівні управління ГКІС. Зокрема, якщо йдеться про інтенсифікацію процесів у ГКІС за рахунок поліпшення якості керування ПР, можна скористатись рядом розроблених спеціально для цієї мети алгоритмів керування ПР на виконавчому рівні, а саме [16]:

- алгоритмом параболічної інтерполяції траєкторії руху робочого органа, що дає змогу зобразити вузлові ділянки кусково-лінійних траєкторій у вигляді парабол. Це сприяє запобіганню зупинень робочих органів у вузлових точках, а отже, зменшує загальний час відпрацювання руху ПР;
- алгоритмом планування рухів робочого органа ПР, що ґрунтується на побудові оптимальної за швидкістю фазової траєкторії, яка відображає залежність швидкості руху робочого органа від його положення. Завдяки використанню так званих характеристичних точок перемикання (в них фазова траєкторія переходить до кривої максимальної швидкості без порушення обмежень) при інтегруванні рівнянь динаміки не потрібна побудова кривої максимальної швидкості, що при плануванні траєкторій руху багатоланкових структур ПР забезпечує виграш у часі при виконанні обчислень у десятки разів;
- алгоритмом числового розв'язування задач кінематики ПР, що ґрунтується на використанні прогнозатора та коригувача узагальнених координат у сукупності з лінійними інтерполяційними методами обчислення функції.

Використання в першому та третьому алгоритмах тільки лінійних залежностей підвищує швидкістю системи управління під час прийняття рішень не менше ніж у два рази і забезпечує можливість роботи ПР в реальному часі.

**Другий етап.** Цей етап пов'язаний з проведенням експериментів на виконавчому рівні керування ПР для одержання коректних (інтенсифікованих) часових показників. Експерименти виконуються за стратегією, сформованою на основі аналізу вхідного слова  $W_1$  і бібліотеки розроблених на першому етапі алгоритмів.

**Третій етап.** Розв'язуються задачі побудови алгоритмів взаємодії СНСП-моделі ГКІС і МПЕ. Ці алгоритми повинні реалізовувати обмін інформацією, що містить значення вхідного  $W_1$  і вихідного  $W_2$  слів, для динамічного навантаження СП-моделі ГКІС.

Докладніше про реалізацію розглянутих етапів планування експериментів СНСП-моделі ГКІС на виконавчому рівні, а також про реалізацію умов взаємодії рівнів СНСП-моделі йдеться в наступному підрозд. 8.4.

#### 8.4. Алгоритмічне забезпечення модуля планування експериментів

**Планування експерименту на нижньому рівні СНСП-моделей ГКІС (детермінований підхід).** Планування експерименту на виконавчому рівні СНСП-моделі ГКІС здійснюється пошуковим модулем ПМ2 на основі аналізу вхідного слова  $W_1$ , що відображає множину  $T'$  переходів СП-моделі з часовими показниками, які не забезпечують заданого рівня якості функціонування системи (тобто, зокрема, не задовольняють умову (8.13)). Наприклад, якщо  $W_1 = T' = \{t_3, t_4, t_7, t_{15}, t_{18}, t_{19}\}$ , де  $T' \subset T$ , то це означає, що відповідні часові показники  $\theta' = \{\tau_3, \tau_4, \tau_7, \tau_{15}, \tau_{18}, \tau_{19}\}$ , де  $\theta' \subset \theta$ , необхідно інтенсифікувати шляхом впливу на виконавчий пристрій ГКІС. Інтенсифікацію можна забезпечити проведенням експериментів з виконавчими пристроями ГКІС.

Для планування та проведення експериментів використовуватимемо ієрархічне уявлення про структуру задач виконавчого рівня керування ПР, що по суті визначає етапи розв'язання задач реалізації функціонування окремого робота.

**Етап А.** Побудова геометричної траєкторії  $z = (x, y)$  руху робочого органа в декартовій системі координат  $xuz$ , пов'язаній з базовою системою координат ПР, тобто в його робочому просторі. Коли ПР має позиційне керування, ця задача зводиться до визначення координат точок позиціонування  $K_1(x_1, y_1, z_1), K_2(x_2, y_2, z_2), \dots$

**Еман В.** Визначення функції залежності швидкості руху робочого органа  $V = f(x, y, z)$  від його положення на геометричній траєкторії, тобто планування фазової траєкторії руху робочих органів ПР. Якщо ПР має позиційне керування, то задача зводиться до визначення швидкостей  $V_1$  переміщення від точки  $K_0$ , яка характеризує початкове положення робочого органа ПР, до точки  $K_1$ , потім швидкості  $V_2$  переміщення від  $K_1$  до  $K_2$  і т. д.

**Еман С.** Розрахунок векторів  $Q_i = (q_{1i}, q_{2i}, \dots, q_{ni})$  узагальнених координат ПР для переміщення робочого органа вздовж траєкторії  $z = f(x, y)$  або між точками  $K_0, K_1, K_2$  і т. д. ( $n$  – кількість ступенів рухомості;  $i = 1, 2, \dots$ ).

**Еман D.** Визначення векторів керуючих дій і  $U_i = (u_{1i}, u_{2i}, \dots, u_{mi})$  на приводи ПР для відпрацювання векторів  $Q_i$  узагальнених координат.

Що стосується етапів А–С, то в деяких випадках недоцільно проводити експерименти для визначення найкоротших геометричних траєкторій (етап А) і/або оптимальних за швидкістю фазових траєкторій (етап В). Зокрема, при роботизації зварювання складальних конструкцій геометрична траєкторія визначається формою зварюваного шва, а фазова траєкторія (швидкість руху робочого органа із зварювальним пальником) – технологією зварювання. Це означає, що для розроблення алгоритмів планування експериментів необхідно мати інформацію про можливість застосування того чи іншого методу оптимального планування рухів робочих органів ПР. Цю інформацію повинен містити кожний перехід вхідного слова  $W_1$ .

Визначимо переходи СП-моделі ГКІС як бінарні функції кількох змінних:  $t_i = f(r_{1i}, r_{2i}, r_{3i})$ , де  $r_{1i}, r_{2i}, r_{3i}$  – змінні, що визначають характеристику дій ПР, які заміщуються переходом, і можливість інтенсифікації цієї дії (у двійковому коді);  $i$  – номер переходу СП. При цьому

$$r_{1i} = \begin{cases} 1, & \text{якщо ПР має контурне керування;} \\ \emptyset, & \text{якщо ПР має позиційне керування;} \end{cases}$$

$$r_{2i} = \begin{cases} 1, & \text{якщо передбачається можливість геометричної траєкторії;} \\ \emptyset & \text{у протилежному разі;} \end{cases}$$

$$r_{3i} = \begin{cases} 1, & \text{якщо передбачається можливість модифікації фазової траєкторії;} \\ \emptyset & \text{у протилежному разі.} \end{cases}$$

Кожний перехід  $t_i$  може набувати таких значень:

$$t_i = \begin{cases} 1, & \text{(встановлення на модифікацію геометричної траєкторії);} \\ \emptyset & \text{у протилежному разі;} \\ 1, & \text{(встановлення на модифікацію фазової траєкторії);} \\ \emptyset & \text{у протилежному разі,} \end{cases}$$

що визначаються функцією  $f$ .

Цю функцію можна задати у вигляді таблиці.

	$t_i$	$r_{1i}$	$r_{2i}$	$r_3$
0	0	1	0	0
0	0	0	0	0
0	1	1	0	1
1	1	1	1	1
1	0	1	1	0
–	–	0	0	1
–	–	0	1	0
–	–	0	1	1

Таким чином, при побудові СП-моделі ГКІС кожному переходу  $t_i$  ставиться у відповідність пара двійкових кодів, яка визначається функцією  $f$ , причому остання вказує можливі шляхи модифікацій динаміки  $\tau_i$  переходу  $t_i$ . Таку СП можна назвати *динамічною*.

**Означення 8.19.** *Динамічна СП* – це двійка вигляду  $N_D = (N_t, \Phi = f(r))$ , де  $N_t$  – звичайна ЧСП;  $f$  – функція, що визначає можливості модифікації динамічного навантаження СП (задається таблично):  $r = \{(r_{11}, r_{21}, r_{31}, \dots, r_{m1}), (r_{12}, r_{22}, \dots, r_{m2}), \dots, (r_{1k}, r_{2k}, \dots, r_{mk})\}$  – множина груп двійкових змінних, які визначають властивості подій (переходів) сітки («одиниця» – властивість спостерігається; «нуль» – у протилежному разі);  $\Phi = \{f(r_{11}, r_{21}, r_{31}, \dots, r_{m1}), f(r_{12}, r_{22}, \dots, r_{m2}), \dots, f(r_{1k}, r_{2k}, \dots, r_{mk})\}$  – множина значень функції  $f$ , які визначають шляхи модифікації динамічного навантаження СП;  $m$  – кількість властивостей подій;  $k$  – кількість переходів, причому кожному переходу  $t_i$  відповідає одна група  $(r_{1i}, r_{2i}, \dots, r_{mi}) \in r', r' \subset r$ .

Використовуючи запропоновану модифікацію СП, можна визначити можливості планування експериментів з ПР шляхом аналізу значень переходів сітки. Так, якщо  $r_{1i} = 1, r_{2i} = 1, r_{3i} = 1$ , то  $t_i = f(r_{1i}, r_{2i}, r_{3i}) = (1; 1)$ . Це означає, що припускається можливість проведення експериментів з ПР, які відповідають переходу  $t_p$ , на етапі А (перша «одиниця»), на етапі В (друга «одиниця») та на етапі С (на цьому етапі немає обмежень щодо можливості проведення експериментів при будь-яких значеннях  $t_p$ , оскільки швидкість розв'язання зворотної задачі кінематики доцільно інтенсифікувати за будь-яких обставин).

Отже, виконавши аналіз функції  $f(r_{1i}, r_{2i}, r_{3i})$  для кожного з переходів  $t_i \in T'$  вхідного слова  $W_1$ , можна прийняти рішення про можливість застосування алгоритмів оптимального планування рухів для модифікації роботи відповідного ПР і часу  $\tau_i$  спрацьовування переходу.

Схему алгоритму прийняття рішень про проведення експериментів для модифікації часу  $\tau_i$  спрацьовування переходу  $t_i$  зображено на рис. 8.9.

Можна навести також схему збільшеного алгоритму функціонування пошукового модуля ПМ2 в структурі МПЕ.

**Крок 1. Аналіз вхідного слова  $W_1$ .** Відповідно до переходу  $t_i$  вхідного слова  $W_1$ , з СП-моделі ГКІС викликаються відповідний час  $\tau_i$  (якщо він визначений) і група змінних  $(r_{1i}, r_{2i}, r_{3i})$ , що характеризують властивості переходу.

**Крок 2. Прийняття рішень.** За алгоритмом прийняття рішень (див. рис. 8.9) визначаються обмеження можливості застосування алгоритмів модифікації роботи ПР. Виділяються можливі шляхи проведення експериментів.

**Крок 3. Проведення експериментів.** Якщо використовується система програмування OFF–LINE, то виділені шляхи проведення експериментів передаються до цієї системи. Експерименти та формування модифікованого часу  $\tau_i$  виконуються в автоматичному режимі. Якщо ж застосовується програмування ON–LINE, то зазначені шляхи передаються людині-досліднику, який і здійснює експерименти з реальним ПР для формування  $\tau_i$ .

**Крок 4. Формування вихідного слова  $W_2$ .** Попередні кроки повторюються для кожного переходу  $t_i$  із слова  $W_1$ . За результатами експериментів формується вихідне слово  $W_2$ , яке містить модифіковані часи спрацьовування переходів.

**Моделювання проведення експериментів з ПР (імовірнісний підхід).** Для підвищення ефективності проведення експериментів (див. крок 3) з ПР, які мають контурне керування, перед проведенням експериментів доцільно визначити опти-



мальні умови функціонування ПР. Це можна зробити, використовуючи попереднє моделювання (організацію) серії експериментів при переміщенні робочого органа ПР вздовж неперервної траєкторії [109].

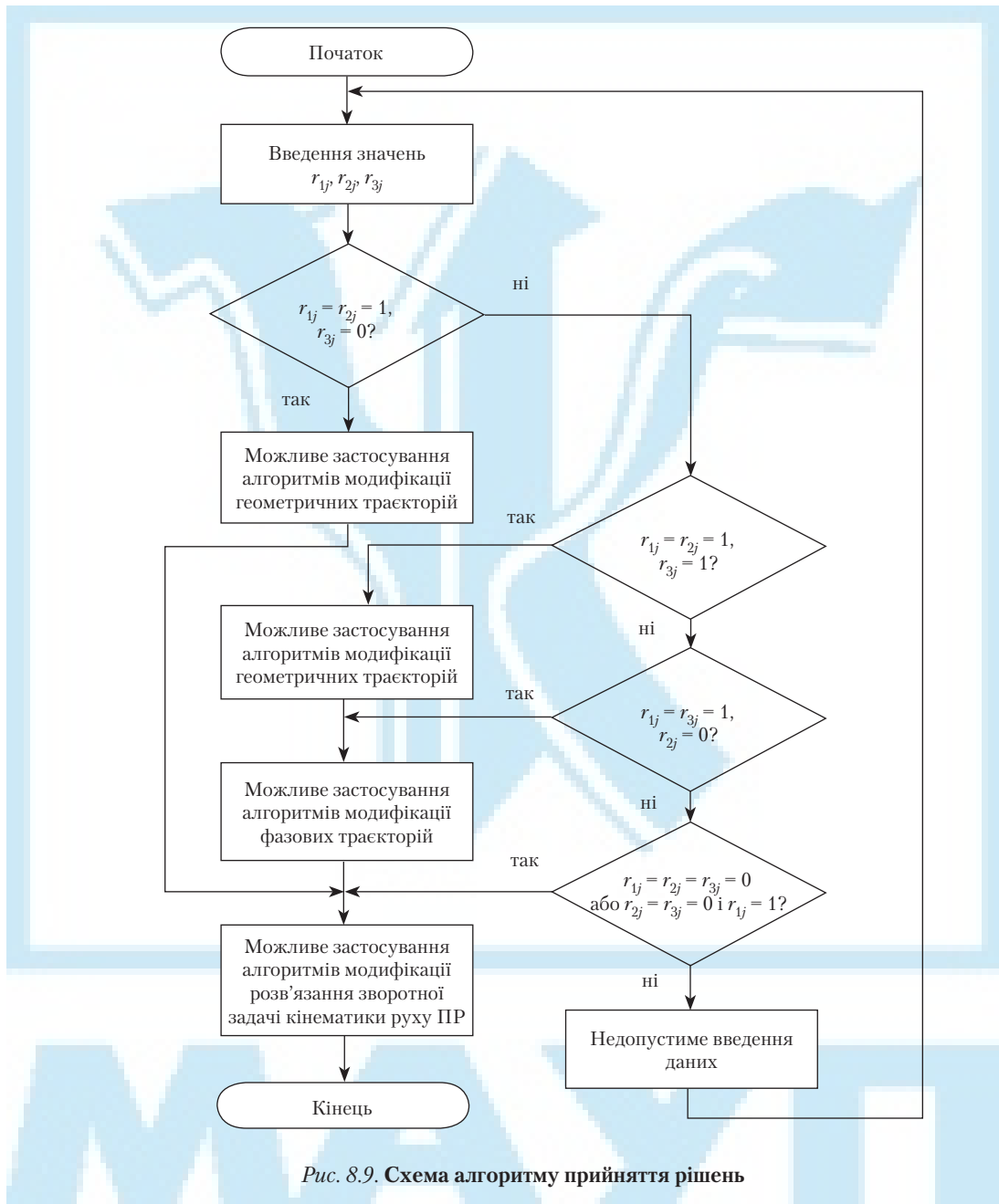


Рис. 8.9. Схема алгоритму прийняття рішень

Отже, при проведенні експериментів з ПР успішною буде тільки певна частка робочих циклів (спроб). Цю частку можна назвати ймовірністю успіху експерименту (ЙУЕ).

**Означення 8.20.** Ймовірністю успіху експерименту назвемо числову характеристику можливості появи прийнятної якості ОМ (зокрема, відтворення траєкторії руху робочого органа ПР, яка в реальних умовах його функціонування забезпечує виготовлення якісного виробу).

На основі ЙУЕ і тривалості циклу (який непрямо впливає на частку успіхів) виразимо міру продуктивності моделі ПР через міру точності апроксимації (інтерполяції) траєкторії його робочого органа і точності керування ПР.

**Означення 8.21.** Продуктивністю моделі ПР є величина, яка чисельно дорівнює кількості успішних робочих спроб, здійснених за одиницю часу виконання експерименту.

**Теорема 8.1.** Продуктивність моделі ПР, за допомогою якої досліджується рух його робочого органа вздовж неперервної траєкторії, прямо пропорційна ЙУЕ.

*Доведення теореми 8.1.* Введемо такі модельні параметри експерименту:  $k_i(t)$  – фактичне положення  $i$ -го зчленування в момент часу  $t$ ;  $k'_i(t)$  – обчислене генератором траєкторії положення  $i$ -го зчленування на той самий момент часу  $t$ ;  $\epsilon_i(t)$  – випадкова похибка  $i$ -го зчленування на момент часу  $t$ .

Припустимо, що всі похибки, крім випадкових, можна компенсувати. Тоді правильною є така рівність:

$$k_i(t) = k'_i(t) + \epsilon_i(t). \quad (8.18)$$

Випадкову похибку  $\epsilon_i(t)$  можна зменшити при використанні більш точного сервокерування ПР. Нехай ця похибка підпорядковується закону нормального розподілу Гауса з нульовим середнім і стандартним  $\sigma_{ip}$  відхиленнями, де  $p = 1, 2, \dots$  – індекс, який відповідає рівню замкненої системи керування ПР. Тоді

$$\sigma_{ip} = \sigma_i \beta_p, \quad (8.19)$$

де  $0 < \beta_p < 1$ ;  $\sigma_i$  – стандартне відхилення похибки для нижчого рівня керування ПР із зворотним зв'язком;  $\beta_p$  – коефіцієнт зв'язку цього рівня з використовуваним.

Фактичні положення та орієнтацію  $A(t)$  робочого органа ПР можна знайти з (8.19), розв'язуючи пряму задачу кінематики робота.

Ймовірність успіху експерименту з неперервною траєкторією робочого органа ПР можна визначити як імовірність того, що ця траєкторія не виходить за межі «трубки» похибок потрібної траєкторії. Введемо додаткові позначення:  $K = (x, y, z)$  – фактичне положення робочого органа ПР в заданій точці траєкторії;  $K^* = (x^*, y^*, z^*)$  – потрібне його положення в цій точці;  $h, l, g$  – фактичні ортогональні одиничні вектори орієнтації робочого органа в певній точці траєкторії;  $h^*, l^*, g^*$  – потрібні вектори орієнтації в цій точці;  $d$  – максимально припустима відстань між фактичним і потрібним положеннями робочого органа ПР;  $B = [\gamma_h, \gamma_l, \gamma_g]$  – матриця максимально припустимих похибок орієнтації робочого органа ПР.

З урахуванням уведених позначень маємо

$$\begin{aligned} (x - x^*)^2 + (y - y^*)^2 + (z - z^*)^2 &\leq d^2; \\ \arccos \frac{hh}{|h||h^*|} &\leq \gamma_h; \quad \arccos \frac{ll}{|l||l^*|} \leq \gamma_l; \quad \arccos \frac{gg}{|g||g^*|} \leq \gamma_g. \end{aligned}$$

Нехай  $C = (d, \gamma_h, \gamma_l, \gamma_g)$  – вектор мір допусків ефективності, а  $A^*(t)$  означає потрібні положення та орієнтацію робочого органа ПР в момент часу  $t$ . В цьому разі успіх функціонування робочого органа ПР в кожній спробі залежить від  $c, A(t), A^*(t)$ .

Задамо двійкову функцію успіху в спробі  $j$  у вигляді

$$F_j(a, b, \rho, c) = \begin{cases} 1, & \text{якщо спроба } j \text{ є успішною;} \\ 0, & \text{якщо спроба } j \text{ є невдалою,} \end{cases}$$

де  $a$  – кількість вузлових точок;  $b$  – тип методу апроксимації (інтерполяції) траєкторії.

При заданих значеннях  $a, b, \rho, c$  можна визначити функцію  $F_j$  для однієї спроби. Тоді функцію ЙУЕ, що складається з достатньої кількості  $G$  спроб, можна визначити так:

$$I(a, b, \rho, c) = \frac{1}{G} \sum_{j=1}^G F_j(a, b, \rho, c). \quad (8.20)$$

Отже, продуктивність моделі ПР визначається виразом

$$H(a, b, \rho, c) = \frac{1}{t_c} I(a, b, \rho, c), \quad (8.21)$$

де  $t_c$  – тривалість однієї спроби, що й треба було довести.

Нехай  $L_p$  – довжина траєкторії;  $V_{ab\rho}$  – максимальна швидкість переміщення робочого органа ПР. Тоді

$$H(a, b, \rho, c) = \frac{V_{ab\rho}}{L_p} I(a, b, \rho, c).$$

Оскільки потрібно максимізувати продуктивність, оптимізаційну модель ПР можна подати у вигляді

$$\max H(a, b, \rho, c) / V_{ab\rho} \leq V^*, \quad (8.22)$$

де  $V^*$  – обмеження швидкості функціонування ПР.

З урахуванням ЙУЕ  $I(a, b, \rho, c)$  дістаємо задачу стохастичної оптимізації, розв'язок якої відшукується за допомогою стохастичного методу пошуку.

Викладений підхід можна проілюструвати такою схемою алгоритму:

**Крок 1.** Для заданих кількості вузлових точок  $a$  методу інтерполяції  $b$  та рівня керування зі зворотним зв'язком  $\rho$  пошуковим модулем ПМ2 генеруються випадкові похибки положення зчленувань згідно з рівністю (8.18).

**Крок 2.** За допомогою прямого кінематичного перетворення визначаються фактичні положення та орієнтація робочого органа ПР для кожної апроксимованої (інтерпольованої) точки траєкторії.

**Крок 3.** Порівнюються фактичне й потрібне положення та орієнтація робочого органа ПР для всіх точок траєкторії відносно заданих меж допуску  $c$ .

**Крок 4.** За результатами порівнянь приймається рішення про успіх або невдачу виконаної робочої спроби.

**Крок 5.** Повторюючи моделювання для значного часового проміжку, обчислюється ЙУЕ за формулою (8.20).

**Крок 6.** Виходячи зі співвідношення (8.22) будується оптимізаційна модель.

**Крок 7.** За допомогою стохастичного методу відшуковуються оптимальні значення керуючих параметрів  $a, b$  та  $\rho$ , які забезпечують максимальну продуктивність моделі ПР і дають змогу в подальшому при проведенні експерименту зменшити часові атрибути СП-моделі ГКІС.

## 8.5. Детермінований та ймовірнісний підходи до моделювання ієрархічних структур ГКІС

### Алгоритми взаємодії рівнів СНСП-моделей ГКІС

**Детермінований підхід.** Отже, відповідно до встановлених вимог (див підрозд. 8.3) структуру СНСП-моделі ГКІС визначає ієрархічна сукупність НСП-моделі верхнього рівня керування ГКІС (організаційно-технічне, оперативне керування) та МПЕ на виконавчому рівні керування ГКІС, причому ці рівні взаємодіють один з одним через пошукові модулі  $ПМ1$  і  $ПМ2$  (див. рис. 8.7 і 8.8). До задач вищого рівня входять організація прогону СП-моделі для формування алгоритму оперативного керування ГКІС і призначення переходів СП з часовими показниками, які не задовольняють заданої якості керування; задачею нижчого рівня є організація серії експериментів на ньому з цими компонентами ГКІС (АлгМПЕ), що відповідають виявленим за допомогою пошукового модуля  $ПМ1$  переходам СП-моделі і потребують уточнення (*задача аналізу*) чи формування (*задача синтезу*) їх часових характеристик. При цьому узагальнений алгоритм СНСП-моделі (АлгСНСП) є композицією АлгНСП та АлгМПЕ [109] – на вхід АлгСНСП надходить слово  $W_{вх}$ , що відповідає множині  $D_{вх}$  – області варіювання параметрів (на практиці вона може задаватись непрямо – за допомогою обмежень). Вихідне слово  $W_{вх}$  визначає основні результати моделювання ГКІС.

Наведене вище дає змогу скористатись твердженням 4.1 і для випадку визначення умов функціонування СНСП-моделей ГКІС.

Така схема взаємодії останніх визначає *детермінований підхід* до моделювання ГКІС. Взаємодія верхнього та нижчого рівнів СНСП-моделі ГКІС відбувається в такій послідовності:

1. Пошуковий модуль  $ПМ1$  здійснює апріорне часове навантаження СП-моделі ГКІС, синтезує алгоритм оперативного управління з урахуванням заданого критерію функціонування ГКІС і визначає переходи СП-моделі з часовими атрибутами, що не забезпечують заданої алгоритмом управління ефективності експлуатації ГКІС.

2. Модуль  $ПМ2$  організовує серію експериментів для зменшення часу спрацювання визначених переходів СП-моделі завдяки підвищенню якості керування відповідним устаткуванням ГКІС на виконавчому рівні.

3. Модуль  $ПМ2$  здійснює динамічне навантаження СП-моделі інтенсифікованими часовими атрибутами, модифікує алгоритм оперативного керування ПР з урахуванням нового навантаження СП-моделі і викликає запуск останньої для перевірки ефективності функціонування ГКІС загалом.

Наведеній послідовності відповідає алгоритм, схему якого показано на рис. 8.10 (пронумеровані блоки виконують функції, про які йдеться нижче).

#### **Алгоритм 8.4 (послідовність взаємодії рівнів СНСП-моделі ГКІС).**

**Крок 1. Формування СП-моделі ГКІС.** У відповідність до ГКІС ставиться СП  $N = (P, T, I, O, \mu_0)$ , елементи якої визначено в підрозд. 8.1. Шляхом виділення для кожної дії (переходу) передумов (умов, які передують дії) та постумов (подій, що є результатами виконання дії) задаються функції  $I, O$  відображення множин  $P \times T, T \times P$ , за допомогою яких об'єднуються позиції та переходи СП.

Задаються також множина  $r$  груп двійкових змінних, які визначають властивості переходів СП, функція  $f$  модифікування динамічного навантаження сітки і визначається множина значень  $\Phi$  функції  $f(r)$ .

Вводиться скінченне маркування  $\mu_1$ , яке відповідає кінцевому стану СП-моделі (тобто стану, в який необхідно перевести ГКІС) і обмеженню роботи ГКІС (наприклад, умова (8.13)).

*Крок 2. Визначення часів спрацьовування переходів СП-моделі.*

Може виконуватись двома шляхами: проведенням експериментів на виконавчому рівні управління ГКІС за допомогою АлгПМ2; використанням наближених і/або статистичних методів.

*Крок 3. Формування множини  $\theta = \{\tau_1, \tau_2, \dots, \tau_m\}$ .* За результатами кроку 2 формується множина часів спрацьовування переходів СП.

*Крок 4. Динамічне навантаження СП-моделі.* Кожному переходу СП-моделі ставиться у відповідність час функціонування заміщуваного елемента (ПР або іншого технологічного устаткування) шляхом відображення добутку  $\theta \times T$ . Таким чином формується навантажена динамічна СП-модель.

*Крок 5. Прогін НСП-моделі.* Шляхом прогону НСП-моделі ГКІС визначається множина послідовностей  $\sigma_t = \{\sigma_1, \sigma_2, \dots\}$  пусків переходів для переведення СП з початкового стану, що характеризується маркуванням  $\mu_0$ , у кінцевий стан з маркуванням  $\mu_1$ . З цієї множини вибирається послідовність  $\sigma_r$ , що відповідає мінімальному часу переведення СП з початкового стану в кінцевий. Одержаній послідовності відповідає алгоритм оперативного управління, який забезпечує переведення ГКІС у заданий стан за мінімальний час.

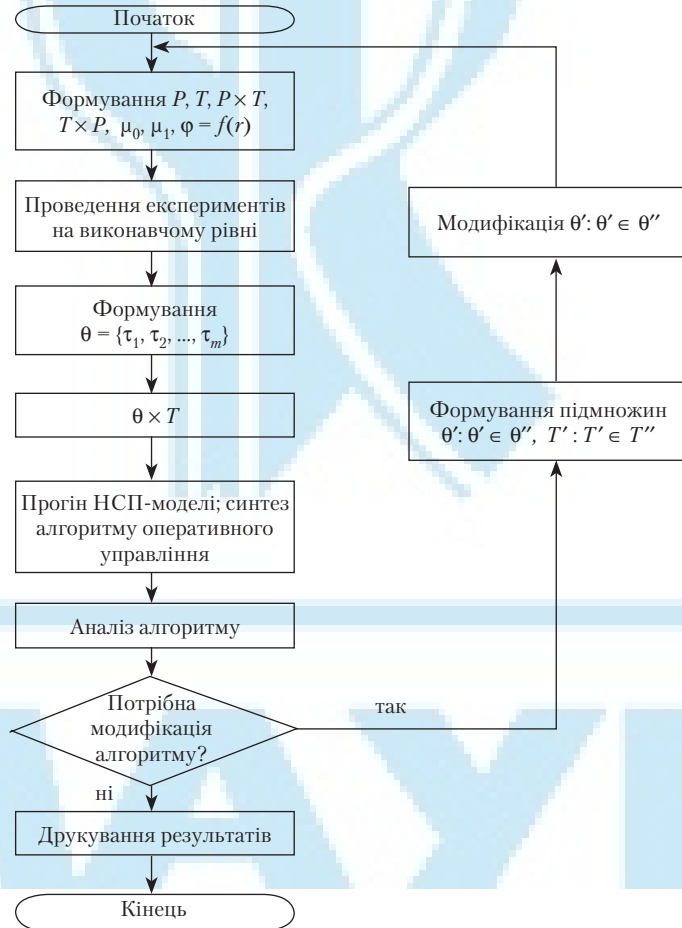


Рис. 8.10. Схема алгоритму функціонування НСП-моделі ГКІС

*Крок 6. Аналіз алгоритму оперативного управління.* Якщо обмеженням роботи ГКІС була умова (8.13) і синтезований алгоритм задовольняє її, то в модифікований алгоритм немає потреби. За інших обставин алгоритм необхідно модифікувати.

*Крок 7. Умовний перехід.* Він відповідає результату аналізу алгоритму (крок 6): якщо алгоритм не потребує модифікації, то він виводиться на друк; якщо модифікація необхідна, то управління передається кроку 8.

*Крок 8. Виділення підмножин.* Виділяються підмножини переходів  $T' : T' \subset T$  СП з підмножинами  $\theta' : \theta' \subset \theta$  часових показників, які знижують швидкодню алгоритму.

*Крок 9. Модифікування  $\theta' : \theta' \subset \theta$ .* Виконуються експерименти на виконавчому рівні управління ГКІС з метою інтенсифікування роботи відповідного (підмножини переходів  $T' : T' \subset T$ ) технологічного устаткування та одержання нових часових показників спрацьовування переходів.

*Крок 10. Друкування результатів.* Результатом є послідовність пуску переходів СП, що відповідає алгоритму оперативного управління ГКІС.

**Ймовірнісний підхід.** Ймовірнісне моделювання дає змогу підвищити «рівень інтелекту» моделі порівняно з детермінованим моделюванням і є ефективним при побудові складних інтегрованих систем керування ОМ з елементами штучного інтелекту [109; 140; 196]. Ймовірнісне моделювання інтелектуальних складних ГКІС ґрунтується на побудові ієрархічної тривірневої ймовірнісної моделі з функціями ймовірності та ентропії як аналітичними мірами. При цьому структура моделі повністю відповідає класичній ієрархії задач керування ГКІС на стратегічному, тактичному і виконавчому рівнях. Середовище робочого простору ГКІС ймовірнісно моделюється в термінах множини неповторюваних і повторюваних примітивних подій, які становлять область завдань системи і показують її можливості. Далі найкращий план переноситься на тактичний рівень, де розробляється детальний сценарій його виконання, відображений у різних функціях координуючих засобів. Система починає функціонувати зі стану максимальної невизначеності, вираженого спочатку функціями рівномірного розподілу ймовірностей, які послідовно коригуються за допомогою навчаючих алгоритмів кожного робочого циклу, завдяки чому невизначеності системи асимптотично зменшуються. До того ж стратегічний рівень забезпечений деякою кількістю означень, які встановлюють його робочий простір, і множиною операційних процедур (правил), що керують його функціями. Як і при використанні детермінованого підходу, ймовірнісній СНСП-моделі ГКІС відповідає узагальнений АлгСНСП-моделі, на вхід якого надходить вхідне слово  $W_{вх}$  (вхідні команди).

**Означення 8.22.** Множина систематизованих за допомогою ПМ компільованих вхідних команд — це команди користувача, які передаються інформаційним каналом зв'язку на верхній рівень СНСП-моделі ГКІС у вигляді множини команд  $U = \{u_1, u_2, \dots, u_z\}$  з відповідними ймовірностями  $I(u_j / u_n | j = \overline{1, z}, n = \overline{1, z}, z = \text{const})$ , які вважаються істинними входами верхнього рівня моделі.

Запис вигляду  $I(A/B)$  означає умовну ймовірність виникнення подій  $A$  за умови здійснення подій  $B$  ([див. (3.22), підрозд. 3.5), що визначається аксіомами:

- а) ймовірність поєднання умов  $A \cap B$  становить  $I(A/B) = I(B) \cap I(A/B)$ ;
- б) ймовірність  $I(A/B)$  є невизначеною, якщо  $I(B) = 0$ .

**Означення 8.23.** Область задавання СНСП-моделі ГКІС — це множина незалежних взаємовиключних диз'юнктивних підмножин переходів СП, які відповідають неповторюваним і повторюваним примітивним подіям у ГКІС, тобто:

$$T = \{T_{nr}, T_r\} = \{t_1, t_2, \dots, t_{k-1}, \dots, t_{k-1-1}, \dots, t_k\},$$

де  $k = \text{const}$ ;  $(k - l)$ ,  $l$  — відповідно кількості переходів, що відповідають неповторюваним (тобто таким, що зустрічаються в плані тільки один раз) і повторюваним (які можуть з'являтися в плані більше як один раз) примітивним подіям.

Поставимо у відповідність кожному переходу  $t_i$  двійкову випадкову змінну  $x_i$ :  $x_i = (1, 0)$ .

**Означення 8.24.** *Перехід  $t_i$  є збудженим (подія — активна), якщо  $x_i = 1$ . В цьому разі кожна позиція (передумова події в ГКІС)  $p_i^{\text{BX}}$ , що є вхідною для переходу  $t_i$ , має необхідне для спрацьовування переходу маркування  $\mu_i^{\text{BX}} \geq 1$ .*

**Означення 8.25.** *Ймовірність збудження переходу  $t_i$  при цій команді — це ймовірність того, що  $p((x_i = 1)/u_i)$ .*

**Означення 8.26.** *Перехід  $t_i$  є незбудженим (подія — неактивна), якщо  $x_i = \theta$ .*

**Означення 8.27.** *Ймовірність того, що перехід  $t_i$  не буде збуджений при цій команді  $u_i$  — це ймовірність того, що  $p((x_i = \theta)/u_i)$ .*

**Означення 8.28.** *Дія  $a_{jm}$  — це послідовність переходів СП, що відповідає групі примітивних подій, взаємозв'язок яких утворює складне завдання для модельної ГКІС.*

Згідно з цим означенням:

- кожен  $m$ -ту дію  $a_{jm}$ , що відповідає команді  $u_j$ , можна подати у вигляді матриці-рядка двійкових випадкових  $\mathbf{X}_{jm} = [x_1, x_2, \dots, x_k]_m$ , де  $m = 1, (2^k - 1)$ , причому кожній компільованій команді  $u_j$  відповідає максимальна кількість  $(2^k - 1)$  дій;
- ця матриця вказує, які переходи  $t_i$  є збудженими, а які не збуджені в  $m$ -й дії, що відповідає  $j$ -й команді;
- кожна дія  $a_{jm}(x_{jm})$  має ймовірність  $I(x_{jm} / u_j)$ .

**Означення 8.29.** *Множина матриць  $\mathbf{M}_{jmr}$  перестановок з відповідними ймовірностями  $I(\mathbf{M}_{jmr} / u_i)$  — це множина, яка визначається і використовується для одержання впорядкованих матриць-рядків дій  $\mathbf{A}_{jmr}$  шляхом упорядкування переходів СП, що відповідають активним примітивним подіям усередині кожної матриці-рядка  $\mathbf{X}_{jm}$  ( $\mathbf{r}$  означає  $\mathbf{r}$ -те впорядкування, що відповідає рядку  $\mathbf{X}_{jm}$ ; максимальна кількість упорядкувань дорівнює кількості перестановок переходів, які відповідають активним примітивним подіям у  $\mathbf{X}_{jm}$ , тобто  $d!$ , де  $d$  — кількість активних примітивних подій). Множину сумісних упорядкованих дій  $\mathbf{A}_{jmr}$  з ймовірностями  $I(\mathbf{A}_{jmr} / u_i)$  подаватимемо у вигляді матриць-рядків двійкових випадкових змінних.*

**Означення 8.30.** *Множина розширених матриць  $\mathbf{M}_{jmr}(y_s)$  перестановки з відповідними ймовірностями  $I(\mathbf{M}_{jmr}(y_s) / \mathbf{A}_{jmr})$  — це множина, яка визначається і використовується для одержання сумісних розширених матриць-рядків дій  $\mathbf{A}_{jmr}(y_s)$  з кожної  $\mathbf{r}$ -упорядкованої дії  $a_{jm}$  (тобто з  $\mathbf{A}_{jmr}$ ).*

**Означення 8.31.** *Множина сумісних розширених упорядкованих рядків дій  $\mathbf{A}_{jmr}(y_s)$  — це множина, яка визначається шляхом розміщення переходів СП, що відображують повторювані примітивні події на відповідних позиціях елементів кожної дії  $\mathbf{A}_{jmr}(y_s)$  означає  $\mathbf{r}$ -й розширений рядок, одержаний з дії  $\mathbf{A}_{jmr}$ , який має ймовірність  $I(\mathbf{A}_{jmr}(y_s) / \mathbf{A}_{jmr})$ .*

**Означення 8.32.** *Модельне мислення — це поєднання компільованої вхідної команди  $u_j \in U$  і множини придатних матриць-рядків дій.*

**Означення 8.33.** *Модельне планування — це визначення повних і сумісних упорядкованих рядків дій для виконання виробничого завдання модельованої ГКІС і відхилення несумісних, а також неповних упорядкованих встановлень з метою зменшення вимірності моделі.*

**Означення 8.34.** Зворотний зв'язок між рівнями моделі — це обчислення функцій вартості, що відповідають розв'язку задач кожного рівня для виконання виробничого завдання, і використання перспективних імовірностей [141].

Припустимо, що кожна з  $M$  уведених команд  $u'_n (n=1, M)$  користувача має априорну ймовірність  $p(u'_n)$  і передається на верхній рівень СНСП-моделі РТС каналом зв'язку у вигляді сталої напруги  $v_n$ , забрудненої адитивним шумом. Сигнал на виході кожного каналу квантується з постійними часовими інтервалами, внаслідок чого утворюється  $\theta$  сигналів  $\omega_1, \omega_2, \dots, \omega_\theta$ , які відображують кожну команду  $u'_n$  у заданій точці  $\theta$  вимірного простору спостережень.

Шум у кожній точці квантування вважатимемо неперервною випадковою величиною  $\eta_f$ , яка підпорядковується закону нормального розподілу Гауса з нульовим середнім відхиленням і дисперсією  $\gamma^2$ :

$$p_{\eta_f}(x) = \frac{1}{\gamma\sqrt{2\pi}} e^{-\frac{x^2}{2\gamma^2}}. \quad (8.23)$$

Спостереження сигналу від кожної команди  $u'_n$  матиме вигляд

$$\omega_f = v_n + \eta_f, \quad (8.24)$$

де  $f = 1, 2, \dots, \theta$  для будь-якого значення  $n = 1, 2, \dots, M$ .

**Теорема 8.2.** Умовний ризик розв'язання  $RR(u_j/\omega)$ , що команда  $u'_n$  класифікується як  $u_j$ , при заданому спостереженні  $\omega$ , дорівнює  $\sum_{n \neq 1} p(u'_n/\omega)$ .

*Доведення теореми 8.2.* З виразу (8.24) знаходимо  $\eta_f = \omega_f - v_n$ . Після підставлення  $x = \eta_f$  у (8.23) дістанемо

$$p(\omega_f / u'_n) = p(\omega_f - v_n) = \frac{1}{\gamma\sqrt{2\pi}} e^{-\frac{(\omega_f - v_n)^2}{2\gamma^2}}. \quad (8.25)$$

Враховуючи те, що спостереження  $\omega_f$  є незалежною подією, ймовірність  $p(\omega)$  суміщення всіх спостережень можна визначити як добуток імовірностей окремих спостережень:

$$p(\omega / u'_n) = \prod_{f=1}^{\theta} p(\omega_f - v_n) = \frac{1}{\gamma^{\theta}\sqrt{2\pi}^{\theta}} e^{-\frac{(\omega_f - v_n)^2}{2\gamma^2}}, \quad \forall n = \overline{1, M}. \quad (8.26)$$

Згідно з правилом Байєса маємо

$$p(u'_n / \omega) = p(\omega / u'_n) p(u'_n) / p(\omega), \quad (8.27)$$

де

$$p(\omega) = \sum_{n=1}^M p(\omega / u'_n) p(u'_n). \quad (8.28)$$

Отже, умовний ризик розв'язання класифікації визначається виразом

$$RR(u'_n / \omega) = \sum_{n=1}^M \lambda(u_j / u'_n) p(u'_n / \omega), \quad \forall n = \overline{1, M}, \quad (8.29)$$

де  $\lambda(u_j / u'_n)$  — функція втрат, яка вводиться для класифікації команд з найменшою інтенсивністю похибки і має такий вигляд:

$$\lambda(u_j / u'_n) = \begin{cases} 0, & \text{якщо } j = n \\ 1, & \text{якщо } j \neq n \end{cases}; \quad n = \overline{1, M}; \quad j = \overline{1, M}.$$

Оскільки функція втрат є симетричною, умовний ризик (8.29) можна записати так:

$$RR(u'_n / \omega) = \sum_{n \neq 1} p(u'_n / \omega) = 1 - p(u'_n / \omega), \quad (8.30)$$

що й треба було довести.



Отже, команду  $u'$  (як одну з  $M$  класифікованих команд  $u$ ) можна класифікувати, визначивши множину згортки дискримінантних функцій ризику

$$g_i(\omega) = RR(u_i / \omega), \quad (8.31)$$

і прийнявши рішення, що копійованою є команда  $u_j$ , коли

$$g_i(\omega) > p(u'_n / \omega), \quad \forall j \neq i.$$

Цьому окремому випадку класифікації команд з мінімальною інтенсивністю похибки і симетричною функцією втрат відповідає така дискримінантна функція:

$$g_i(\omega) = p(u'_n / \omega). \quad (8.32)$$

З (8.32) випливає, що максимальній дискримінантній функції відповідає максимальна ймовірність появи команди  $u'_j$  у  $\theta$ -вимірному просторі спостережень  $\omega$ . Тому для мінімізації похибки класифікації команд сформулюємо правило прийняття рішення: «Прийняти рішення про появу команди  $u_j$ , якщо

$$p(u'_j / \omega) > p(u'_i / \omega), \quad \forall j \neq i.$$

Отже, на вхід верхнього рівня СНСП-моделі РТС надходить істинна команда  $u_j$  з максимальною ймовірністю, яка є гарантом мінімуму ризику неправильної класифікації команд і визначається виразом

$$p(u_j) = \max_j p(u'_j / \omega), \quad j = \overline{1, M}. \quad (8.33)$$

Використовуючи означення 8.20–8.34 і вирази (8.29), (8.30) та (8.33), можна побудувати алгоритм функціонування ймовірнісної СНСП-моделі складальної РТС.

**Алгоритм функціонування моделей.** Алгоритм функціонування ймовірнісної СНСП-моделі ГКІС можна подати у вигляді шести кроків, які описують один її робочий цикл (у загальному випадку – не перший).

**Алгоритм 8.5** (функціонування ймовірнісної СНСП-моделі ГКІС).

*Крок 1. Класифікація вхідної команди на верхньому рівні моделі.* Цю класифікацію здійснює пошуковий модуль ПМ. Для заданої вхідної команди  $u'_n$  користувача її ймовірність  $p(u'_n)$  визначається виразом  $p(u'_n) = V_n / V_T = V_n / \sum_{n=1}^M V_n$ , де  $V_n$  – відносна кількість появ команди  $u'_n$ , а  $V_T$  – загальна кількість усіх команд користувача. Початкова ймовірність появи кожної команди  $u'_n$  становить  $p(u'_n) = 1 / M$ .

Згідно з виразом (8.33) визначається ймовірність появи копійованої класифікованої вхідної відносно стратегічного рівня моделі команди  $u_j$ .

Через функцію ентропії розподілу ймовірностей для одновимірної випадкової величини обчислюється невизначеність, пов'язана з командами користувача та копійованими командами:

$$H(u'_n) = -p(u'_n) \log p(u'_n); \quad (8.34)$$

$$H(u_j) = -p(u_j) \log p(u_j). \quad (8.35)$$

Визначаються повні ентропії, що відповідають усім командам користувача й усім копійованим класифікованим командам:

$$H(U') = \sum_{n=1}^M H(u'_n); \quad (8.36)$$

$$H(U/U') = \sum_{n=1}^M H(u_n). \quad (8.37)$$

Оскільки команди користувача (і копільовані команди) в загальному випадку є незалежними випадковими подіями, то передавання інформації між двома різними командами  $u'_n$  і  $u'_{n'}$  користувача (або копільованими  $u_j$  і  $u_{j'}$ ) не буде:

$$T(u'_n : u'_{n'}) = 0, \quad n \neq n'; \quad (8.38)$$

$$T(u_j : u_{j'}) = \emptyset. \quad (8.39)$$

На підставі (8.33) і (8.34) визначаються умовні ентропії подій:

$$H(u'_n : u'_{n'}) = H(u'_n) \text{ при } n \neq n'; \quad (8.40)$$

$$H(u_j : u_{j'}) = H(u_j) \text{ при } j \neq j'. \quad (8.41)$$

*Крок 2. Модельне мислення.* Здійснюється пошуковим модулем ПМ1 на основі таких міркувань. Після ідентифікації вхідної копільованої команди  $u_j$  їй відповідає максимальна множина  $(2^N - 1)$  придатних дій – матриць-рядків збуджених і/або незбуджених переходів СП, що відображують примітивні події в РТС. Виходячи з цього кожна дія  $a_{jm}$  подається матрицею-рядком двійкових випадкових змінних  $X_{jm} = [x_1, x_2, \dots, x_N]_m$ , яка відповідає  $m$ -й дії та  $j$ -й копільованій команді.

Кожній матриці-рядку відповідає ймовірність

$$p(a_{jm} / u_j) = p(X_{jm} / u_j) = p_{jm}, \quad \forall j = \overline{1, M}; \quad \forall m = \overline{1, (2^N - 1)},$$

яка спочатку визначається як

$$p_{jm} = 1 / (2^N - 1), \quad (8.42)$$

а потім коригується за допомогою навчаючого рівняння (див. крок 6).

Ентропія, що відповідає дії  $a_{jm}$  (матриці-рядку двійкових випадкових змінних  $X_{jm}$ , які відображують збудження/незбудження переходів-подій СП-моделі РТС) за умови появи події у вигляді класифікованої копільованої вхідної команди  $u_j$ , визначається виразом

$$H(X_{jm} / u_j) - p_{jm} \log_2 p_{jm}. \quad (8.43)$$

Повна ентропія множини придатних дій (матриць-рядків переходів СП), що відповідають одній класифікованій копільованій команді  $u_j$ , задається вектором імовірностей  $S_j = (X_{j_1}, X_{j_2}, \dots, X_{j_{(2^N-1)}})$  як

$$H(S_j / u_j) = \sum_{m=1}^{2^N-1} H(X_{jm} / u_j), \quad j = \overline{1, M}, \quad (8.44)$$

а ентропія всіх дій (матриць-рядків) на всі класифіковані копільовані команди обчислюється за формулою

$$H_\Sigma = \sum_{j=1}^M H(S_j / u_j), \quad (8.45)$$

яка визначає сумарну невизначеність у процесі модельного мислення.

З (8.39) і (8.43) випливає, що

$$T(X_{jk} : X_{jl} / u_j) = \emptyset, \quad k \neq l; \quad (8.46)$$

$$H(X_{jk} / (X_{jl}, u_{jl})) = H(X_{jk} / u_j), \quad (8.47)$$

де  $X_{jk}, X_{jl}$  – дві різні матриці-рядки двійкових випадкових змінних, які відображують дії  $a_{jk}$  та  $a_{jl}$ .

*Крок 3. Модельне планування.* Здійснюється пошуковим модулем ПМ1 у чотири етапи.

*Еман 1.* За допомогою матриць перестановок, що переставляють змінні  $x_i$ , дістаємо впорядковані дії (матриці-рядки) шляхом переупорядкування змінних  $x_i$  вектора  $X_{jmr}$ , які відповідають збудженим переходам СП-моделі. Матрицю перестановки записуємо як матрицю вимірністю  $M \times N$  з елементами 1 і  $\emptyset$  так, щоб у кожному рядку та кожному її стовпці була тільки одна одиниця. Впорядковані (сумісні) матриці-рядки двійкових змінних  $A_{jmr}$  формуються так:

$$A_{jmr} = X_{jmr} M_{jmr}; \quad m=1, (2^N - 1); \quad r = \overline{1, d!}, \quad (8.48)$$

де  $M_{jmr}$  –  $r$ -та матриця перестановки, що відповідає вектору  $X_{jmr}$ ;  $d$  – кількість збуджених переходів – активних примітивних подій у модельованій РТС.

Позначивши через  $p(M_{jmr}/u_j)$  ймовірність того, що матриця  $r$ -го спільного подання використовується для одержання  $A_{jmr}$  з  $X_{jmr}$  при заданій команді  $u_j$ , визначимо ймовірність одержання  $r$ -ї впорядкованої матриці-рядка  $A_{jmr}$  за умови появи команди  $u_j$ .

$$p(A_{jmr} / u_j) = p(M_{jmr} / u_j) p(X_{jmr} / u_j), \quad (8.49)$$

причому  $\sum_m \sum_r p(M_{jmr} / u_j) = 1$ .

*Еман 2.* Послідовно внесемо переходи СП, що відповідають повторюваним примітивним подіям, у сумісні впорядковані дії (рядки) і за можливості спростимо (наприклад, так:  $t_j, t_j \equiv t_j$ ) цю послідовність.

*Еман 3.* Кожній розширеній сумісній впорядкованій матриці-рядку двійкових випадкових змінних, яка формується з деякого набору матриць-рядків дій  $A_{jmr}$ , відповідає ймовірність  $p(M_{jmr}(y_s)/A_{jmr})$ , де  $M_{jmr}(y_s)$  –  $r$ -та розширена матриця перестановки, одержана з  $r$ -ї матриці  $M_{jmr}$ . Розширена матриця перестановки формується внесенням рядка  $[\emptyset \emptyset \dots 1 \dots \emptyset]$  у відповідну матрицю переустановлення, де 1 вказує на внесений перехід примітивної події СП-моделі.

Ці ймовірності коригуються за допомогою зворотного зв'язку (див. крок 6) після виконання кроку 3 за умови, що

$$\sum_m \sum_r \sum_s p(M_{jmr}(y_s) / A_{jmr}) = 1. \quad (8.50)$$

Отже, ймовірність  $p(A_{jmr}(y_s)/u_j)$  розширеної матриці-рядка  $A_{jmr}(y_s)$  задається співвідношенням

$$p(A_{jmr}(y_s) / u_j) = p(M_{jmr}(y_s) / A_{jmr}) p(A_{jmr} / u_j), \quad (8.51)$$

а ентропія, що відповідає кожній матриці-рядку  $A_{jmr}$ , визначаються так:

$$\begin{aligned} H(A_{jmr} / u_j) &= -p(A_{jmr} / u_j) \log_2 p(A_{jmr} / u_j) = \\ &= p(X_{jmr} / u_j) H(M_{jmr} / u_j) + p(M_{jmr} / u_j) H(X_{jmr} / u_j) \end{aligned} \quad (8.52)$$

і є сумою початкової невизначеності дії (матриці-рядка) та невизначеності, яка відповідає  $r$ -му переустановленню змінних у матриці.

Аналогічно визначається ентропія, що відповідає  $s$ -й розширеній сумісній впорядкованій матриці-рядку двійкових випадкових змінних:

$$\begin{aligned} H(A_{jmr}(y_s) / u_j) &= p(A_{jmr} / u_j) H(M_{jmr}(y_s) / A_{jmr}) + \\ &+ p(M_{jmr}(y_s) / A_{jmr}) H(A_{jmr} / u_j). \end{aligned} \quad (8.53)$$

Вираз (8.53) є зваженою сумою невизначеностей, які відповідають початковій впорядкованій дії (матриці-рядку), та невизначеності, що відповідає  $s$ -му розширенню матриці.

*Етап 4.* Вибираємо повні та сумісні плани, використовуючи обмеження, які є аналогічними виразам (8.46) (8.47):

$$T(A_{jmr} : A_{jml} / u_j) = 0; \quad (8.54)$$

$$H(A_{jmr} / (A_{jmr}, u_j)) = H(A_{jmr} / u_j), \quad (8.55)$$

де  $r$  та  $l$  мають той самий зміст, що й у (8.46), (8.47).

*Крок 4. Модельне прийняття рішень.* Реалізується воно пошуковим модулем ПМ1. Припустимо, що існує  $h$  повних і сумісних виробничих планів. Тоді процес прийняття рішень полягатиме у виборі такого плану, який відповідав би розширеній матриці-рядку з максимальною умовною ймовірністю, тобто

$$\max_h P(A_{jmr}(y_s) / u_j). \quad (8.56)$$

*Крок 5. Реалізація рішень і динамічне навантаження моделі.*

Пошуковий модуль ПМ1 передає вибраний найкращий можливий план виконання необхідної роботи на нижчий рівень у вигляді послідовності виділених переходів, які відповідають діям плану. На нижчому рівні пошуковий модуль ПМ2 визначає всі деталі, пов'язані з виконанням плану, в наведеній далі послідовності.

Кожному переходу СП (дії в РТС) приписується деяка вартість, яка підсумується при повторних пусках переходу. Одержана сума є накопиченою вартістю пусків кожного переходу СП. Визначається повна накопичена вартість нижчого рівня, що відповідає виконанню виробничого завдання, у вигляді суми накопичених вартостей пусків переходів СП ( $J_B = J_1 + J_2 + \dots$ ), виражених через відповідні ентропії. Ця вартість передається на верхній рівень (динамічне навантаження) і використовується для коригування інформації про ймовірність, яка зберігається у довгочасній пам'яті (модифікація динамічного навантаження).

Нижче наведено етапи функціонування пошукового модуля ПМ2.

*Етап 5.1.* Переглянути план виконання виробничого завдання, переданий пошуковим модулем ПМ1 з верхнього рівня, і сформулювати задачу управління, що підлягає виконанню на виконавчому рівні. Вона містить активізацію виконавчих пристроїв і виконання певних завдань, коли збуджується відповідний перехід СП.

Формулювання задачі управління виконавчими пристроями можна подати у вигляді ряду етапів;

- а) зібрати інформацію про об'єкти та інструментальні засоби, що є в наявності в середовищі робочого простору РТС;
- б) ідентифікувати об'єкти та засоби РТС, необхідні для виконання виробничого завдання;
- в) зібрати інформацію про початкове (індекс П) й кінцеве (індекс К) положення необхідних об'єктів і засобів; утворити відповідні декартові вектори:

$$r_{\text{П}} = (r_x, r_y, r_z, \alpha, \beta, \gamma); \quad r_{\text{К}} = (r'_x, r'_y, r'_z, \alpha', \beta', \gamma');$$

- г) перетворити цю інформацію в простір векторів узагальнених координат (за необхідності):

$$r_{\text{П}} \rightarrow q_{\text{П}} = (q_1, q_2, q_3, q_4, q_5, q_6); \quad r_{\text{К}} \rightarrow q_{\text{К}} = (q'_1, q'_2, q'_3, q'_4, q'_5, q'_6)$$

- д) вибрати динамічні моделі ПР з множини досяжних моделей;
- е) вибрати конкретний алгоритм керування приводами з множини доступних алгоритмів (як критерій може бути вибраний вид траєкторії — гладка, зі зломом тощо), типи різних рухів (наприклад, з використанням сили тяжіння або протидії цій силі);

ж) вибрати показник якості функціонування ПР у складі РТС з урахуванням обмежень задачі (наприклад, показник якості за швидкодією).

*Етап 5.2.* Розрахувати виконавчий рівень моделі, тобто обчислити накопичену вартість  $J_B$ , що відповідає виконанню виробничого плану, виходячи з індивідуальних накопичених вартостей кожного переходу СП.

*Етап 5.3.* Здійснити динамічне навантаження СНСП-моделі РТС шляхом передавання інформації, одержаної на нижчому її рівні, на верхній рівень після моделювання виконання виробничого завдання.

*Етап 5.4.* Скоригувати інформацію, що зберігається в довгочасній пам'яті нижчого рівня.

*Крок 6. Модифікація динамічного навантаження моделі.* Цей крок передбачає пошук і корекцію інформації про ймовірності, що надходять з довгочасної пам'яті верхнього рівня СНСП-моделі. Ймовірності коригує пошуковий модуль ПМ1 за допомогою навчаючого ітераційного алгоритму, ймовірностей апроксимування, використовуючи для цього одержане на нижчому рівні значення накопиченої вартості:

$$p((\psi + 1) / u_j) = p(\psi / u_j) + v_{\psi+1}[\varepsilon - p(\psi / u_j)], \quad (8.57)$$

де  $\varepsilon = \begin{cases} 1, & \text{якщо } J = J_B; \\ 0 & \text{у протилежному разі;} \end{cases}$   $\psi$  – номер ітерації;  $v_{\psi+1}$  – послідовність, яка задовольняє умову збіжності алгоритму;  $p(\psi / u_j)$ ,  $p(\psi + 1) / u_j$  – умовні ймовірності компільованої команди  $u_j$ , що відповідають  $\psi$ -й та  $(\psi + 1)$ -й ітераціям;  $J$  – фактична вартість виконання сформованого плану.

Скоригована інформація заноситься в довгочасну пам'ять верхнього рівня СНСП-моделі РТС.

Аналіз описаного алгоритму показує, що СНСП-моделлю РТС накопичується інформація і досвід, що забезпечує асимптотичне зменшення її невизначеності за принципом зростаючої точності при поступовому зниженні необхідного на перших етапах функціонування рівня інтелекту.

## 8.6. Обґрунтування нових модифікацій СП для динамічного моделювання ГКІС

**Гнучкі сітки Петрі.** Як було зазначено вище, ГКІС має ряд особливостей функціонування, а саме: паралельність та асинхронність процесів, їх розгалуження і синхронізацію, наявність черг, конкуренцію технологічного устаткування за ресурси, одночасний рух декількох видів матеріальних потоків тощо. Ці особливості мають складний опис у термінах звичайних СП і їх стандартних модифікацій, тому було розроблено ряд модифікацій, наведених у розділах 7 та 8 і призначених для моделювання ГКІС [74; 125; 140], зокрема:

1. Ієрархічні розфарбовані СП, побудовані на основі типових блоків. Суть полягає у виділенні типових сіткових конструкцій (блоків), що описують послідовне, рівнобіжне, альтернативне і безумовне виконання ієрархією дискретних процесів. З огляду на те, що типові блоки коректні за визначенням, аналіз коректності СП зводиться до аналізу ієрархічної системи типових блоків.

2. Ієрархічні розфарбовані СП побудовані на основі сіткових конструкцій, які вводяться для опису і розв'язування типових конфліктів у процесах, що моделюються. Суть полягає в забезпеченні на основі типових блоків і сіткових конструкцій

можливості автоматизованого синтезу структур коректних алгоритмів функціонування і керування.

3. Нечіткі логічні СП. Тут позиції відповідають умовам правил (посилань), а самі переходи — продукціям (правилам). Простим сіткам Петрі притаманна звичайна логіка висловлювання (маркування набувають значення 0 або 1), а в нечітких сітках Петрі маркування набувають значення від 0 до 1 майже на неперервному інтервалі з кроком, наприклад, 0,1; 0,01 і т. д. Нечітка логіка, яка інтерпретується сітками, дає змогу маркуванню визначити ступінь істинності висловлювання у відповідній позиції сітки.

Однак, згадані вище модифікації спрямовані на можливість синтезу коректних і безпечних сіткових моделей без підвищення моделюючої потужності стандартних модифікацій СП, що дає змогу скоротити витрати на аналіз сіток і при цьому розрахувати більшість характеристик функціонування ОМ. Проте при моделюванні процесів у ГКІС часто необхідно враховувати їхні часові характеристики. Це можна зробити (див. *означення* 8.11–8.15), навантаживши кожен перехід СП часовим атрибутом [63; 124; 159].

**Означення 8.35.** Часові СП визначаються як  $N_\tau = (N, \tau)$ , де  $N = (P, T, F, H)$  — сітка Петрі, а  $\tau: T \rightarrow \mathbf{R}$  — функція часів спрацьовування переходів, яка навантажує кожен перехід постійним часом його реалізації.

Правило збудження переходу у часовій СП збігається з правилом, що діє для звичайних СП, формулюється як  $\forall p_i \in P : \mu(p_i) \geq H(p_i, t_j)$  і полягає в тому, що кожна з позицій попередників збудженого переходу  $t_j$  в момент початку спрацьовування переходу втрачає кількість фішок, визначену значенням вихідної функції  $H(p_i, t_j)$ , а кожна з позицій послідовників  $p_i \in \text{POST}(t_j)$  одержує кількість фішок, визначену значенням вхідної функції  $F(t_j, p_i)$  через проміжок часу  $\tau_i$  після початку спрацьовування переходу.

З урахуванням часового фактора навантаження було запропоновано нову модифікацію СП, яка дістала назву *гнучкої сітки Петрі* (ГСП) [41] з такими моделюючими спроможностями:

- для адекватного моделювання роботи технологічного устаткування враховуються часові характеристики виконуваних ним операцій, тому переходи мають часовий атрибут, а СП — засоби навантажування переходів часовими атрибутами;
- для можливості реалізації перевірки на нуль введено інгібіторні дуги, що полегшує практичне застосування апарату СП через спрощення структури СП-моделі реального об'єкта;
- гнучкість забезпечується можливістю зміни параметрів СП у процесі її функціонування.

Таким чином, ГСП матимуть властивості одночасно інгібіторних і часових СП, що дасть змогу промоделювати будь-яку систему, яка працює в реальному часі.

На графі ГСП (рис. 8.11, *a*) позиції  $p_1, p_2, \dots$ , як і для звичайної СП, позначаються колами, а переходи  $t_1, t_2, \dots$  — планками; вхідні та вихідні функції задаються спрямованими дугами (наприклад,  $p_2, t_1; t_2, p_2; \dots$ ); інгібіторна функція подається дугою, яка закінчується кружечком (на рис. 8.11, *a* це  $p_5, t_1$ ); функція часів спрацьовування задається числовими значеннями останніх для кожного переходу (на рис. 8.11, *b* наведено відповідні для переходів  $t_1, t_2, t_3, t_4$  їх часові еквіваленти  $\tau = (10, 30, 4, 20)$ ). Останнє означає, що на відміну від звичайних СП, де спрацьо-

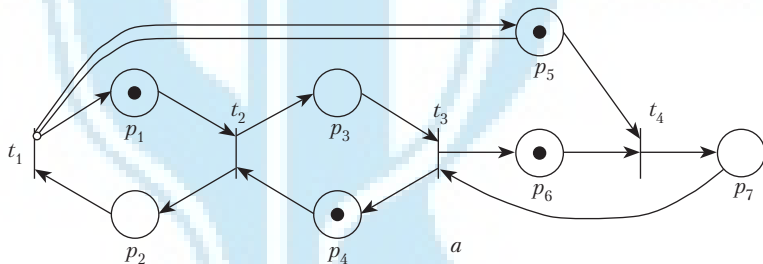
ування переходів відбувається миттєво, в разі ГСП для спрацьовування переходу необхідний час, визначений функцією  $\mathbf{R}$ , що треба враховувати при маркуванні сітки.

**Формальний опис ГСП.** Дамо означення ГСП:

**Означення 8.36.** Гнучка сітка Петрі (ГСП) визначається як:

- $N_{\text{ГП}} = (N, \bar{H}, \tau, \gamma)$ , де  $N = (P, T, F, H)$  – сітка Петрі;
- або  $N_{\text{ГП}} = (P, T, F, H, \bar{H}, \tau, \gamma)$ .

В обох випадках  $P$  – скінченна множина позицій, причому  $P = \{p_i\}$ ,  $i = \overline{1, n}$ ;  $T$  – скінченна множина переходів, причому  $T = \{t_j\}$ ,  $j = \overline{1, m}$ ; множини  $P$  та  $T$  не перетинаються, тобто  $P \cap T = \emptyset$ ;  $F: T \times P \rightarrow V_{\text{вх}}$  – вхідна функція;  $H: P \times T \rightarrow V_{\text{вих}}$  – вихідна і  $\bar{H}: P \times T \rightarrow \{0, 1\}$  – вихідна функції, які визначають інгібіторні дуги;  $\tau: T \rightarrow \mathbf{R}$  – функція часів спрацьовування, яка надає кожному переходові постійний час його реалізації;  $\gamma: \Omega \rightarrow \tau$ , де  $\Omega$  – множина параметрів об'єкта моделювання,  $\gamma$  – зовнішня процедура, яка визначає функцію  $\tau$  на основі параметрів об'єкта моделювання (рис. 8.11, б).



$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}; T = \{t_1, t_2, t_3, t_4\};$$

$$F = \begin{pmatrix} 1000100 \\ 0110000 \\ 0001010 \\ 0000001 \end{pmatrix}; H = \begin{pmatrix} 0100 \\ 1000 \\ 0010 \\ 0100 \\ 0001 \\ 0001 \\ 0010 \end{pmatrix}; \bar{H} = \begin{pmatrix} 0000 \\ 0000 \\ 0000 \\ 0000 \\ 1000 \\ 0000 \\ 0000 \end{pmatrix}$$

$$\mu = \{(1, 0, 0, 1, 1, 1, 0), (\emptyset, \emptyset, \emptyset, \emptyset)\}; \tau = (10, 30, 4, 20); \gamma = \text{ПМ1}$$

б

Рис. 8.11. Гнучка сітка Петрі  $F = (P, T, I, H, \bar{H}, \tau, \gamma)$  з маркуванням  $\mu$ :

а – граф, б – формальне визначення

**Означення 8.37.** Під гнучкістю в ГСП розуміють здатність до перевизначення функції часів спрацьовування переходів  $\tau$  у процесі функціонування моделі.

**Означення 8.38.** Маркування ГСП  $\mu = (\mu_p, \mu_t)$ , де  $\mu_p: P \rightarrow N^n$  – це функція, що визначає кількість фішок у позиціях;  $\mu_t: T \rightarrow R \cup \{\emptyset\}$  – функція, що визначає стан переходів і поточний час у переходах, що спрацьовують, а функція  $(\mu_t(\mu_j))$  набуває значення:

$$\mu_t(t_j) = \begin{cases} \emptyset, & \text{якщо перехід } t_j \text{ вважається пасивним;} \\ t_j, & \text{якщо перехід } t_j \text{ вважається активним} \end{cases}; t_j \in [0, \tau(t_j)].$$

**Означення 8.39.** *Перехід є активним, якщо він перебуває в режимі спрацьовування на поточний час, у протилежному разі він — пасивний.*

Умови спрацьовування переходу в ГСП враховують інгібіторні і часові властивості сітки.

**Означення 8.40.** *Перехід  $t_j$  ГСП є збудженим, коли виконується умова*

$$A = A_1 \wedge A_2 \wedge A_3,$$

де  $A_1 = (\forall p_i \in P) : (\mu_p(p_i) \geq H(p_i, t_j))$ ;  $A_2 = (\forall p_i \in P) : (\mu_p(p_i) \times \bar{H}(p_i, t_j) = 0)$ ;  $A_3 = (\mu_t(t_j) = 0)$ .

Іншими словами, у кожній вхідній позиції переходу повинна бути кількість фішок, яка є більшою або дорівнює кратності дуги, що веде з цієї позиції в перехід; перехід не повинен бути забороненим інгібіторними дугами; перехід повинен бути пасивним. Умова  $A$  є необхідною і достатньою, тобто перехід починає спрацьовувати в момент збудження.

Правила спрацьовування переходу  $t_j$  формулюються в такий спосіб: а) спрацьовування переходу відбувається в проміжок часу  $[t_0, t_0 + \tau(t_j)]$ , де  $t_0$  — момент часу початку спрацьовування; б) в момент часу  $t_0$  кожна з позицій попередників  $p_i \in PRE(t_j)$  втрачає кількість фішок, визначену значенням вихідної функції  $H(p_i, t_j)$ , а перехід стає активним  $(\forall p_i \in P) : \mu_p(p_i) = \mu_p(p_i) - H(p_i, t_j)$ ;  $\mu_t(t_j) = \emptyset$ ; в) у момент часу  $t_0 + \tau(t_j)$  кожна з позицій послідовників  $p_i \in POST(t_j)$  одержує кількість фішок, визначену значенням вхідної функції  $F(p_i, t_j)$ , а перехід стає пасивним  $(\forall p_i \in P) : \mu_p(p_i) = \mu_p(p_i) + F(p_i, t_j)$ ;  $\mu_t(t_j) = \emptyset$ ; г) на час спрацьовування переходу сітка не блокується, тобто в сітці одночасно може спрацьовувати декілька переходів.

**Алгоритм побудови дерева досяжності ГСП.** Через те що функціонування моделі відбувається в реальному часі, алгоритм побудови дерева досяжності ГСП значно відрізняється від аналогічного алгоритму для звичайних СП (алгоритм 8.1).

Вважатимемо, що всі можливі зміни стану ГСП у певний момент часу відбуваються миттєво. Дерево досяжності будується для моментів часу, у яких відбувається зміна маркування позицій і переходів, тобто фактично для моментів закінчення спрацьовування переходів.

**Означення 8.41.** *Момент часу  $\bar{t}_k$ ,  $k \in K$  називається моментом  $k$ -ї зміни стану ГСП.*

**Означення 8.42.** *Локальне дерево досяжності — це дерево досяжності, побудоване для певного моменту зміни стану.*

Між моментами часу  $\bar{t}_k$  і  $\bar{t}_{k+1}$  стан системи змінюється тільки за рахунок зміни значень функції  $\mu_p$ , що не становить інтересу для дослідження і не вважається зміною маркування.

У дереві досяжності можуть існувати вершини таких типів: 1) кореневі — вершини, що відповідають початковому маркуванню СП; 2) внутрішні — вершини, що мають вхідну і хоча б одну вихідну дугу; 3) термінальні — вершини, що відповідають тупиковому маркуванню; 4) дублюючі — вершини, для яких відповідні їм



маркування дорівнюють маркуванням, що відповідають іншим вершинам дерева досяжності; 5) локально-кореневі — вершини, з яких будується дерево досяжності для моменту часу  $\bar{t}_k$ ; 6) локально-термінальні — термінальні вершини для моменту часу  $\bar{t}_k$ , причому локально-термінальні вершини для  $\bar{t}_k \in$  локально-кореновими для  $\bar{t}_{k+1}$ .

При побудові дерева досяжності використовується ще поняття граничної вершини — вершини, для яких ще не побудовані вихідні дуги.

Алгоритм побудови дерева досяжності ГСП працює в такий спосіб:

**Алгоритм 8.6** (побудови дерева досяжності ГСП).

*Крок 1.* Прийняти за поточний момент зміни стану момент  $\bar{t}_0$ . Оголосити початкове маркування ГСП  $\mu$  кореневою вершиною дерева досяжності.

*Крок 2.* Прийняти як локально-кореневу вершину — кореневу:  $\mu_{\text{лк}} = \mu$ .

*Крок 3.* Побудувати локальне дерево досяжності (алгоритм 8.1).

*Крок 4.* Якщо локальне дерево досяжності не має локально-термінальних вершин, перейти до кроку 9.

*Крок 5.* Обчислити значення наступного моменту зміни стану  $\bar{t}_{k+1}$  і прийняти його за поточний.

*Крок 6.* Прийняти як локально-кореневі вершини моменту часу  $\bar{t}_{k+1}$  локально-термінальні вершини локального дерева досяжності для часу  $\bar{t}_k$ .

*Крок 8.* Перейти до кроку 3.

*Крок 9.* Закінчити роботу алгоритму.

Для побудови локального дерева досяжності зробимо структурно-тотожну заміну часових переходів  $t_j$  елементарними структурами, що складаються з двох переходів  $t_j^{\text{II}}$  і  $t_j^{\text{K}}$ , які не мають часу спрацьовування і фіктивної позиції  $p_j^t$ . Переходи  $t_j^{\text{II}}$  і  $t_j^{\text{K}}$  позначають початок і кінець спрацьовування переходу, а позиція  $p_j^t$  містить фішку протягом часу  $\tau(t_j)$  (рис. 8.12).

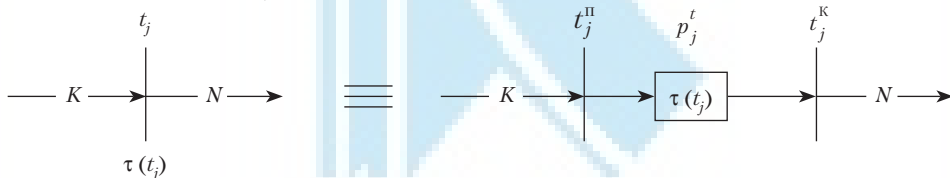


Рис. 8.12. Тотожне перетворення часового переходу

Правило порушення переходу  $t_j^{\text{II}}$  збігається з правилом порушення часового переходу  $t_j$ , а перехід  $t_j^{\text{K}}$  збуджується, коли фішка пробуде в позиції  $p_j^t$  протягом часу  $\tau(t_j)$ , тобто якщо  $\mu_t(t_j) = \tau(t_j)$ .

Тепер, якщо визначити всі локально-кореневі вершини локального дерева досяжності (рис. 8.13) як граничні і мати на увазі, що значення компонентів  $\mu_t$  не може дорівнювати  $\omega$ , для побудови локального дерева досяжності можна застосувати звичайний алгоритм побудови дерева досяжності СП. Обчислення значення наступного моменту зміни стану для стану  $\bar{t}_k$  здійснюється за формулою

$$\bar{t}_{k+1} = \bar{t}_k + \min_{j: \mu_t(t_j) \neq \omega} (\tau(t_j) - \mu_t(t_j)). \quad (8.58)$$

**Аналіз властивостей ГСП за допомогою дерева досяжності.** Для дослідження властивостей ГСП за допомогою дерева досяжності необхідно визначити коло задач, яке можна розв'язати з використанням останнього.

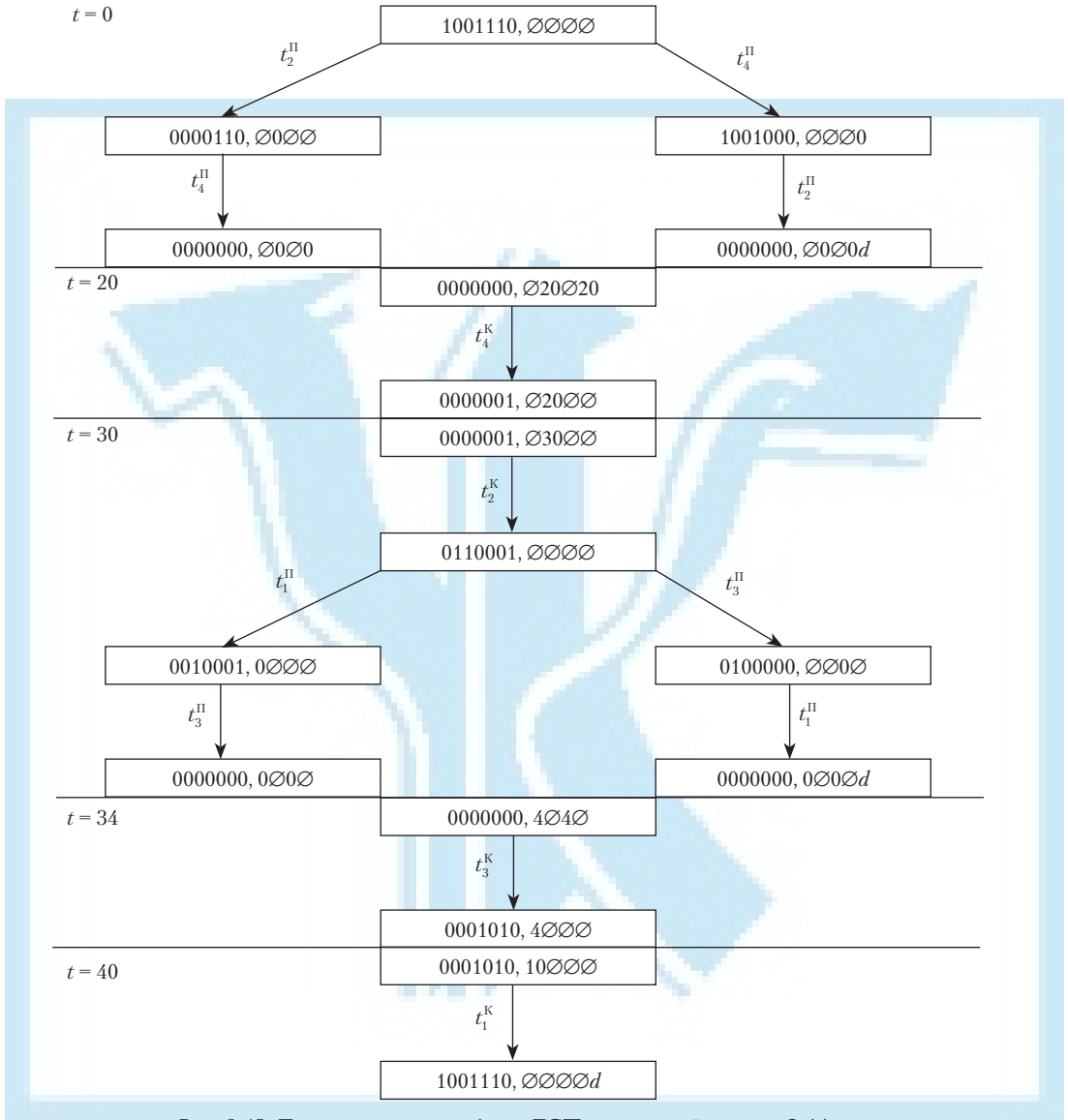


Рис. 8.13. Дерево досяжності для ГСП, наведеної на рис. 8.11, а

**Теорема 8.3.** Дерево досяжності скінченної ГСП у загальному випадку є нескінченним.

*Доведення теореми 8.3.* Кількість вершин дерева досяжності обмежено кількістю можливих маркувань. З означення 8.38 маркування ГСП маємо  $\mu = (\mu_p, \mu_t)$ , де  $\mu_t : T \rightarrow R \cup \{\emptyset\}$  – функція, що визначає стан переходів і поточний час у переходах, що спрацьовують. Таким чином, функція  $\mu_t(t_j)$  може набувати нескінченну кількість значень, отже, може існувати нескінченна кількість маркувань, і дерево досяжності може містити нескінченну кількість вершин. Що й потрібно було довести.

З теореми 8.3 випливає важлива властивість ГСП, що має часовий характер. За допомогою ГСП із нескінченним деревом досяжності неможливо змоделювати

реальну виробничу систему, тому що виробничий цикл скінченний, а нескінченне дерево досяжності передбачає наявність нескінченних послідовностей спрацьовувань переходів. Тому робота такої сітки повинна бути синхронізована за часом, що дасть їй змогу досягти початкового маркування за скінченну кількість спрацьовувань переходів. Синхронізація може бути досягнута двома шляхами: 1) за рахунок введення затримок синхронізації коригуванням часів спрацьовування переходів (цю методику розглянуто в праці [124]); 2) за рахунок структурних обмежень на функціонування часової СП.

Другий шлях ускладнює структуру сітки (тому що потребує застосування інгібіторних дуг), але забезпечує її працездатність при будь-яких значеннях часових параметрів переходів.

**Означення 8.43.** Синхронізована ГСП — це така ГСП, дерево досяжності якої скінченне при будь-яких значеннях часових параметрів переходів.

Як очевидний висновок з *теорему 8.3* і *означення 8.43* справедливим є таке твердження.

**Твердження 8.5.** Будь-яка нечасова СП є синхронізованою.

Порівнюючи ГСП з аналогічною їй інгібіторною СП (тобто фактично з аналогічною їй сіткою, з якої виключено часові атрибути переходів), можна довести таку теорему (причому необхідно мати на увазі, що алгоритм побудови дерева досяжності інгібіторної СП аналогічний відповідному алгоритму для звичайної СП, за винятком модифікованого правила порушення переходів).

**Теорема 8.4.** Множина послідовностей спрацьовування переходів, які дають змогу досягти початкового маркування гнучкої (синхронізованої) ГСП  $N_{\text{ГН}} = (N, \bar{H}, \tau, \gamma)$ , є підмножиною множини відповідних послідовностей спрацьовування переходів інгібіторної СП  $N_{\text{ІН}} = (N, \bar{H})$ .

*Доведення теорему 8.4.* Подамо правило спрацьовування переходу ГСП, яке функціонально тотожне наведеним тут, у такий спосіб: при порушенні переходу  $t_j$  фішки вилучаються з позицій попередників і поміщаються в позиції-послідовники миттєво, але доступними для збудження переходів фішки в позиціях-послідовниках стають через час  $\tau(t_j)$ . При такому трактуванні роботи синхронізованої ГСП  $F$  зміна її маркування в результаті спрацьовування переходу аналогічна зміні маркування при спрацьовуванні відповідного переходу інгібіторної СП  $N_{\text{ІН}}$ . Часові параметри, що відрізняють  $N$  від  $N_{\text{ІН}}$ , впливають тільки на пріоритети спрацьовування переходів і можуть привести до заборони деяких послідовностей. Отже, виключивши заборонені послідовності з множини послідовностей  $N_{\text{ІН}}$ , можна одержати множину послідовностей  $N$ , що й потрібно було довести.

З *теорему 8.4* випливають умови достатності для властивостей обмеженості і живучості.

**Умова 8.1.** Синхронізована ГСП  $N_{\text{ГН}} = (N, \bar{H}, \tau, \gamma)$  — обмежена, якщо обмежена інгібіторна СП  $N_{\text{ІН}} = (N, \bar{H})$ .

**Умова 8.2.** Синхронізована ГСП  $N_{\text{ГН}} = (N, \bar{H}, \tau, \gamma)$  — живуча, якщо живуча інгібіторна СП  $N_{\text{ІН}} = (N, \bar{H})$ .

Також з *теорему 8.4* впливає умова перевірки ГСП на синхронізованість.

**Умова 8.3.** ГСП  $N_{\text{ГН}} = (N, \bar{H}, \tau, \gamma)$  є синхронізованою, якщо кількість дуг  $t_j^k$  її дерева досяжності менша або дорівнює кількості дуг  $t_j$  дерева досяжності інгібіторної СП  $N_{\text{ІН}} = (N, \bar{H})$ .

Для синхронізованої ГСП із дерева досяжності легко визначити час циклу її роботи для будь-якої послідовності спрацьовування переходів, що дорівнює моменту

зміни стану, в якій у дереві досяжності з'явилася дублююча вершина з маркуванням, рівним початковому. Відшукавши час циклу для всіх послідовностей, можна вибрати найменший, тим самим визначивши найменший час виконання виробничого циклу в модельованій системі.

Таким чином, застосовуючи для моделювання виробничих систем ГСП, можна визначити, чи є система, в якій існують рівнобіжні процеси, синхронізованою, і визначити час циклу роботи такої системи, що неможливо зробити, моделюючи систему звичайними СП.

**Формування ефективного керування на основі дерева досяжності.** На рівні подання ГКІС у вигляді ГСП керування полягає в ініціюванні певної послідовності спрацьовування часових переходів (виконання елементарних операцій устаткування ГКІС). Синтез такого керування можливий на основі аналізу дерева досяжності ГСП ГКІС. Побудова дерева досяжності рідко виправдує себе через великий розмір, однак при пропонованому підході існують розуміння на підтримку саме цього методу аналізу властивостей СП: застосування інгібіторних дуг істотно ускладнює, якщо не унеможливує застосування інваріант для аналізу ГСП; ієрархічна структура моделі обмежує розмірність окремих реалізацій ГСП (оптимальним для застосування методу є розмір сітки з кількістю позицій не більше десяти); при інтенсифікації досліджується тільки та область ієрархічної структури моделі, що безпосередньо веде до критичної операції.

Нижче пропонуються два шляхи проведення аналізу дерева досяжності: а) шляхом його безпосереднього дослідження; б) шляхом перетворення його в діаграму Ганта.

**Безпосередній аналіз дерева досяжності.** Безпосереднім аналізом дерева досяжності можна розв'язати задачу синтезу мінімального за часом керування ГВМ. Для цього необхідно відшукати послідовність спрацьовування переходів ГСП, яка переводила б сітку в початкове маркування за найменший проміжок часу. Передбачається, що ГСП, яка описує функціонування ГКІС, обмежена і живуча. Задачу можна розв'язати в такий спосіб. З усіх вершин дерева досяжності, рівних кореневій вершині, береться така, момент часу зміни стану якої найменший. Далі визначається найкоротший шлях (тобто послідовність спрацьовування переходів) від кореневої вершини до знайденої.

Зазначимо, що для синтезу мінімального за часом керування не потрібно було визначати всі можливі послідовності спрацьовувань переходів. Це важливий момент, оскільки дерево досяжності будується в порядку збільшення часу, і, отже, для розв'язання наведеної задачі немає необхідності будувати його цілком, а тільки до моменту появи першої вершини, рівної кореневій, що може значно зменшити ступінь використання обчислювальних ресурсів.

Також варто зазначити, що в деяких окремих випадках можна синтезувати керування для систем, ГСП яких не має властивості обмеженості і живучості. Для цього необхідно, щоб дерево досяжності такої ГСП містило хоча б одну послідовність спрацьовування переходів, що приводить систему з початкового стану в початковий.

**Перетворення дерева досяжності ГСП у діаграму Ганта.** Діаграма Ганта часто використовується в робототехніці для визначення різних параметрів функціонування систем, зокрема часів простоювання устаткування [124], і являє собою графік сім'ї функцій

$$f_i(t) = \begin{cases} i, & \text{якщо устаткування } i \text{ працює;} \\ \emptyset, & \text{якщо устаткування } i \text{ простоює;} \end{cases} \quad i = 1, \dots, n,$$

де  $n$  — кількість елементів ГВМ.

Процедура перетворення дерева досяжності ГСП у діаграму Ганта проста алгоритмічно, але потребує значних обчислювальних ресурсів, тому що необхідно побудувати повне дерево досяжності. Полягає ця процедура в обчисленні всіх можливих послідовностей спрацьовування переходів і побудові для кожної послідовності діаграми у вигляді сім'ї функцій  $f_i(t)$ , що відповідають стану переходів  $t_i$  ГСП.

У результаті одержимо множину діаграм, що відповідають різним реалізаціям керування ГВМ. Порівняльний аналіз діаграм дасть змогу вибрати керування, яке найбільше відповідає висунутим вимогам. Крім того, за діаграмою Ганта легко обчислити критичні операції: якщо при зменшенні або збільшенні часу виконання операції на деяке значення  $\Delta t$  загальний час виконання ТП зменшується або збільшується на час, кратний  $\Delta t$ , то така операція є критичною.

Обидва розглянуті підходи до синтезу керування ГВМ на рівні його подання у вигляді ГСП легко піддаються алгоритмізації і їх можна використати для моделювання і дослідження властивостей ГВМ.

### Контрольні завдання

1. Обґрунтувати доцільність використання модифікованої логічної СП для реалізації процедур логічного виведення у межах пропозиційної логіки.
2. Дати означення простої ЛСП.
3. Пояснити, як визначається довжина шляху виведення і що являє собою імплікантний шлях.
4. Навести означення нечіткої ЛСП і прокоментувати загальний алгоритм виведення на такій сітці.
5. Обґрунтувати доцільність та особливості використання матричного виведення для циклічних ЛСП.
6. Розкрити методологію блочного виведення в ЛСП.
7. Розтлумачити процедури формування блоків та організації блочного виведення з поясненням алгоритму такого виведення.
8. Дати означення умовно виконаного, активного та живучого блоків, а також умовно живучої блочної структури ЛСП.
9. Розкрити поняття сукупності перетинних циклів і зв'язати його з умовою живучості блочної структури ЛСП.
10. Означити орієнтоване на інтелектуалізоване моделювання розширення апарату СП.
11. Дати означення навантаженої СП та динамічно навантаженої СП-моделі.
12. Прокоментувати структуру автоматизованої СП-моделі та розкрити узагальнену роль інтелектуалізованої компоненти — пошукового модуля в ній.
13. Розкрити суть самонавантажувальної СП-моделі та охарактеризувати функції модуля планування експериментів у такій моделі.
14. Прокоментувати логічну послідовність етапів побудови СНСП-моделі.
15. Розкрити процедуру планування експерименту на нижньому рівні СНСП-моделі ГКІС і прокоментувати алгоритм функціонування пошукового модуля у складі МПЕ при цьому.

16. Розкрити термін «ймовірність успіху експерименту» та вказати граничні (прийнятні) умови його проведення.
17. Розкрити сутність детермінованого підходу до моделювання ієрархічних структур ГКІС.
18. Прокоментувати алгоритм функціонування імовірнісної СНСП-моделі ГКІС і розкрити поняття «модельне мислення» та «модельне прийняття рішень».
19. Обґрунтувати доцільність нових модифікацій СП для динамічного моделювання процесів ГКІС.
20. Розкрити зміст і дати формальний опис гнучкої СП, а також навести означення часової СП.
21. Прокоментувати розгорнутий алгоритм побудови дерева досяжності гнучкої СП і розкрити відмінності від аналогічної процедури для звичайної СП.
22. Пояснити процедуру аналізу властивостей ГСП за допомогою дерева досяжності та сформулювати властивості синхронізованої ГСП.
23. Прокоментувати процедуру перетворення дерева досяжності у діаграму Ганта як процес ефективного керування ГКІС.



## ШТУЧНИЙ ІНТЕЛЕКТ У СИСТЕМАХ ПРОЕКТУВАННЯ ТА ПЛАНУВАННЯ

---

---

*Всі ми помиляємося.  
Проте, одні більше, інші — весь час.*

Закон життя

*Давай поради не ті, які найприємніші,  
а ті, які найкращі.*

Солон з Афін

*Одні протилежності  
породжують інші*

Марк Тулій Цицерон

### 9.1. Методи та засоби відображення розв'язків оптимізаційних задач

**Метод повного перебирання варіантів.** Для цього методу типовою є загальноінтелектуальна процедура: якщо інтелектуальна система потрапляє в нову ситуацію, вона намагається звести планування своїх цілеспрямованих дій до оптимізаційної задачі — визначення множини  $M$  припустимих розв'язків і цільової функції  $f(x)$ , і якщо це вдається, повним перебиранням отримується оптимальний розв'язок.

**Означення 9.1.** *Метод повного перебирання* — це універсальний метод гарантованого розв'язання оптимізаційних задач за умови скінченності множини  $M$  припустимих розв'язків (ситуація, характерна для дискретного програмування) і існування ефективного алгоритму породження будь-якого елемента з  $M$  та обчислення на цьому елементі цільової функції  $f(x)$ .

**Метод евристичного пошуку.** Оскільки для складних практичних ситуацій кількість варіантів перебирання надто велика (що потребує значних часових і машинних ресурсів), а також враховуючи ту обставину, що далеко не завжди слід шукати оптимальний розв'язок, часто обмежуються субоптимальним (наближеним до оптимального) або просто припустимим для поставленої мети розв'язком. У такому випадку необхідно виробити деяке правило (обмеження), яким перебивалося б повне перебирання варіантів пошуку при отриманні результату, який для більшості вхідних даних був би задовільним. Саме в цьому й полягає суть *евристичного підходу*.

**Означення 9.2.** *Евристичним пошуком* називають процедуру систематизованого перебирання на основі послідовного аналізу можливих варіантів та оцінки їх наслідків [21] з такою схемою алгоритмічних кроків:

- вибрати певну дію з області можливих;
- здійснити дію; це приведе до зміни ситуації;

- оцінити нову ситуацію;
- якщо досягнуто успіху — кінець; якщо ні — почати з першого кроку.

Одним з варіантів реалізації методу евристичного пошуку є стратегія *послідовних поліпшень*, яка полягає в тому, що за наявності певного наближення  $x^l$  до розв'язання застосовується деяка процедура для переходу до іншого наближення  $x^{l+1}$ , і якщо воно буде кращим за попереднє, продовжується генерація і випробування нового наближення і т. д. (нагадує процедуру градієнтних підходів). Основні методи дослідження операцій ґрунтуються саме на цій стратегії. Проте слід зазначити, що евристичний пошук супроводжується експоненціальною складністю його реалізації. Для пояснення цієї тези розглянемо такий приклад [21].

**Приклад 9.1.** Задача про виконуванисть булевого виразу формулюється так: для будь-якого булевого виразу від  $n$  змінних знайти хоча б один набір значень змінних  $(x_1, \dots, x_n)$  за якого цей вираз набуває значення 1.

Можна запропонувати розв'язання цієї задачі у такий спосіб: спочатку надається одне з двох можливих значень (0 або 1) змінній  $x_1$ , потім  $x_2$  і т. д. Після розставлення значень усіх змінних визначається значення булевого виразу, і якщо воно дорівнює 1, то задача розв'язана, а якщо ні — треба повернутися назад і змінити значення деяких змінних.

Можна інтерпретувати цю задачу як задачу пошуку на певному *дереві перебирання*, кожна з вершин якого відповідає набору встановлених значень  $x_1, \dots, x_r$ . Оскільки для будь-якої змінної можна встановити значення 0 або 1, отримуємо вибір з двох дій, кожній з яких відповідає одна з двох дуг, які йдуть від цієї вершини. Інакше кажучи, кожна з вершин  $x_1, \dots, x_r$  має два породження:  $x_1, \dots, x_r$  із значенням 0 і  $x_1, \dots, x_r$  із значенням 1.

Для наочності спростимо задачу до виразу від  $n = 3$ , для якого дерево можливостей (рис. 9.1) має вершини, помічені наборами змінних, і дуги — розв'язки, які отримуємо на черговому кроці. Вершини, що відповідають розв'язкам задачі для виразу  $(x_1 x_2) = x_3$ , а також відповідні дуги, які визначають шляхи до них, пофарбовані більш жирною лінією на дереві перебирання.

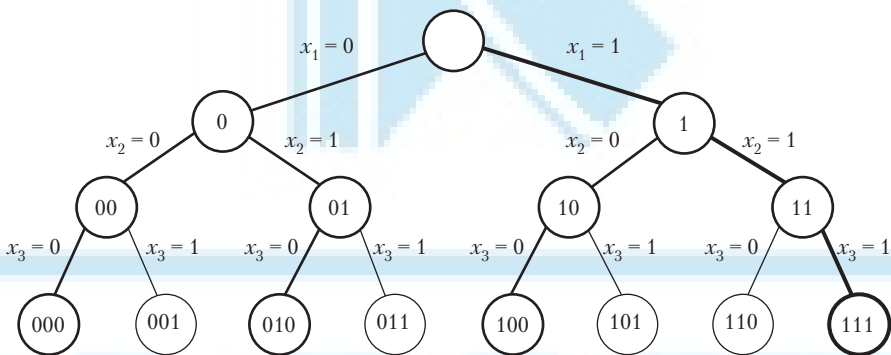


Рис. 9.1. Дерево перебирання для задачі про виконуванисть булевого виразу

Як впливає з рисунка, розв'язання задачі зводиться до перебирання коріння дерева з метою виявлення їх відповідності наборам, що перетворюють заданий булевий вираз на 1. Переглянувши набори витоків дерева, легко визначити, що з восьми розв'язків цієї задачі на перебирання загальну умову  $(x_1 x_2) = x_3$  задовольняють набори 000, 010, 100, 111, а перетворенню булевого виразу на 1 відповідає тільки набір 111, який і є оптимальним шуканим розв'язком (шлях до відповідної вершини і сама вершина рельєфно наведені на дереві).



Як бачимо з наведеного прикладу, при  $n = 3$  коріння дерева має  $2^3 = 8$  витоків. У загальному ж випадку кількість можливих варіантів дорівнює  $2^n$ , тобто із збільшенням  $n$  кількості булевих змінних перебірний алгоритм евристично має експоненціальний характер, і ця обставина змушує шукати способи його реалізації, які базуються на спрощених стратегіях розв'язання таких задач.

**Стратегії організації пошуку вершин дерева перебирання.** Найпоширенішими стратегіями організації пошуку необхідної вершини на дереві перебирання і шляху до неї є *пошук у ширину* і *пошук у глибину*.

**Означення 9.3.** *Стратегією пошуку в ширину* називається організація такої процедури, яка передбачає проведення аналізу на кожному кроці породжень усіх вершин, що були проаналізовані на попередньому кроці, тобто *паралельну* перевірку всіх можливих альтернатив.

Ця стратегія для наведеного прикладу 9.1 для булевого виразу з  $n = 3$  ілюструється рис. 9.2.

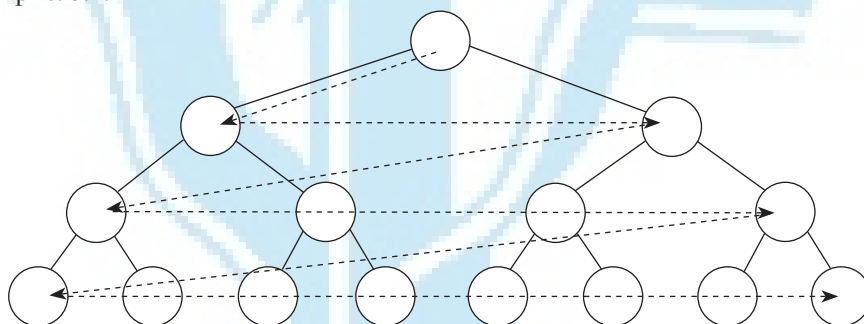


Рис. 9.2. Стратегія з процедурою пошуку в ширину

**Означення 9.4.** *Стратегією пошуку у глибину* називається організація такої процедури, якою передбачається першочерговий аналіз породжень тих вершин, що були проаналізовані останніми, тобто *послідовну* перевірку всіх альтернатив одна за одною, причому аналіз деякої альтернативи завершується тільки тоді, коли точно встановлюється, що вона приводить до успіху. У разі невдачі відбувається *повернення на крок* і розглядаються інші альтернативи.

Ілюстрацією стратегії пошуку у глибину для умов того ж прикладу є аналогічне дерево перебирання з наведеною пунктирними стрілками послідовністю процедурного крокування (рис. 9.3).

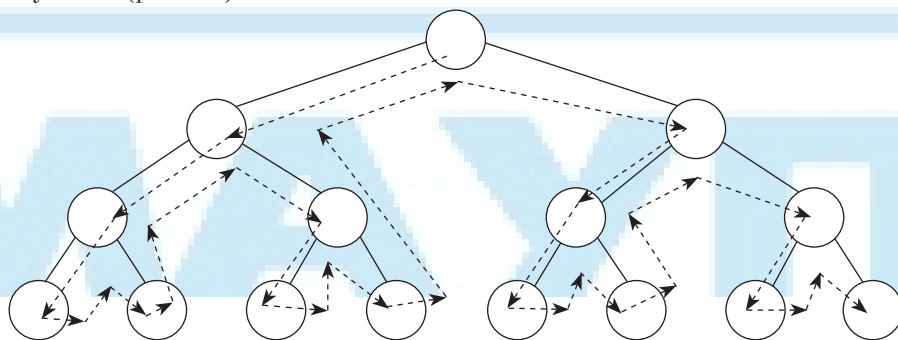


Рис. 9.3. Стратегія з процедурою пошуку у глибину

Стратегія пошуку у глибину набула більшого поширення на практиці і стала класичною загальноінтелектуалізованою процедурою в сучасних методиках планування цілеспрямованих дій.

**Простори станів і задач.** Методики реалізації цілеспрямованих дій при організації пошуків розв'язків поділяються на два класи: планування в просторі станів і планування в просторі задач [42].

При плануванні в просторі станів заданим вважається набір станів (ситуацій) і відомі дії, які може здійснювати система і які визначають перехід від одного стану до іншого в процесі розв'язання задачі, тобто граф її станів.

**Означення 9.5.** *Граф станів задачі* — це орієнтований граф, вершини якого відповідають можливим станам об'єкта дослідження, а дуги — методам переходу від стану до стану.

Дуги можуть бути поіменованими мітками з інтерпретацією вартості або довжини відповідного переходу, і тоді загальний шлях від початкового стану до цільового визначає витрати на його реалізацію; при цьому типовою є вимога оптимізації цього розв'язку за певним (так званим енергетичним) критерієм.

Фактично задача зводиться до пошуку в графі, основними машинними способами завдання якого є матриця інцидентності, матриця суміжності та списки суміжності.

Планування у просторі задач передбачає декомпозицію початкової задачі на підзадачі доти, поки не буде досягнуто зведення до елементарних задач, для яких є готові алгоритми розв'язання. Якщо ж для деякої задачі готового алгоритму не існує, а її подальша декомпозиція неможлива, таку задачу вважають нерозв'язною.

Для планування у просторі задач традиційно використовується формалізм І–АБО-графів.

**Означення 9.6.** *І–АБО-графом* називається орієнтований граф, вершини якого відповідають задачам, а дуги — відношенням між задачами, причому між дугами вводяться відношення «І» та «АБО».

**Приклад 9.2.** Нехай задано задачу  $D$ , яка зв'язана дугами з підзадачами  $E$  та  $F$ . При плануванні розв'язків це означає, що задачу  $D$  можна звести шляхом декомпозиції до підзадач  $E$  та  $F$ , що позначатиметься відповідними дугами  $DE$  і  $DF$ . При цьому, якщо ці дуги зв'язані між собою оператором «І», то це означає, що для  $D$  треба розв'язати і  $E$ , і  $F$ . Якщо ж для визначення  $D$  достатньо розв'язку тільки будь-якої однієї з цих підзадач, то відповідні дуги зв'язуються оператором «АБО».

Будь-який І–АБО-граф можна звести до певної *нормальної форми*, в якій з будь-якої вершини виходять або тільки «І»-дуги, або тільки «АБО»-дуги.

**Означення 9.7.** *«І»-вершиною* називається вершина, з якої виходять лише «І»-дуги; *«АБО»-вершиною* є вершина з виключно «АБО»-дугами.

**Означення 9.8.** *Початковою вершиною І–АБО-графа* є вершина, з якої розпочинаються цілеспрямовані дії щодо розв'язання задачі.

**Означення 9.9.** *Вершина є розв'язною*, якщо задача, що відповідає цій вершині, має розв'язок.

З урахуванням означення 9.9 метою пошуку, що реалізується на І–АБО-графі, є встановлення факту розв'язуваності початкової вершини.

Оскільки процедурно розв'язання задачі на І–АБО-графах нагадує механізм виведення продукційної системи, можна стверджувати, що фактично *такі графи є фрагментами послідовностей виведення з продукційних систем*.



**Метод динамічного програмування.** Цей метод ґрунтується на схемі послідовного перебирання варіантів з використанням процедур, які на основі побічних оцінок відхиляють всі ті допустимі розв'язки, серед яких не може бути оптимального, тобто відбувається стиснення множини конкурентоспроможних варіантів до одного або декількох обмежених, які й порівнюються.

**Означення 9.10.** *Динамічне програмування* — це такий алгоритмічний метод, який доцільно використовувати, коли розв'язання задачі можна звести до певної послідовності розв'язків.

При такому підході на кожному кроці перевіряються всі можливі послідовності розв'язків і за рахунок вилучення тих послідовностей, які не можуть бути оптимальними, задача значно спрощується. В цьому й полягає *принцип оптимальності*.

Тоді під динамічним програмуванням ще розуміють такий спосіб побудови алгоритмів, який можна застосувати тоді, коли розв'язання загальної задачі можна розбити на розв'язання її підзадач і отримувати розв'язок наступної задачі як суперпозицію попередніх розв'язків.

**Твердження 9.1.** *Оптимальна послідовність розв'язків полягає в тому, що, незважаючи на початковий стан і розв'язок, серед розв'язків, які залишилися, завжди міститься оптимальна послідовність розв'язків стосовно стану, що утворився після прийняття першого розв'язку.*

У динамічному програмуванні може утворюватися кілька послідовностей розв'язків. Приклади використання методик реалізації цілеспрямованих дій у плануванні розглядаються далі (див. підрозд. 9.3).

## 9.2. Експертні системи в задачах проектування і планування

**Етапи проектування.** Впровадження елементів ШІ в задачі проектування та планування є необхідним засобом підвищення ефективності конструкторської і технологічної підготовки виробництва на підготовчій стадії життєвого циклу виробу.

Системи з ШІ у задачах САПР можна охарактеризувати зокрема за етапами проектування (на яких вони використовуються), за рівнем «творчого потенціалу», за ступенем участі саме у створенні нових об'єктів тощо.

Специфічні задачі, що відображують потенціальні галузі ефективного застосування технології з ШІ, виникають на всіх етапах *процесу проектування* (табл. 9.1).

Рівень «творчого потенціалу» інтелектуальної САПР визначається її здатністю набувати нових (у тому числі й принципово нових) розв'язків у процесі створення проекту. Фактичні рівні компетенції інтелектуальної складової САПР можуть лежати в широкому діапазоні: від можливості розв'язувати задачі винахідницького характеру (активна творча діяльність у межах дослідного конструювання) до простого вибору найбільш придатної альтернативи серед готових з використанням евристичних правил віддавання переваги. Значною мірою така градація визначається наявністю глибинних знань.

## Задачі інтелектуальних систем на різних етапах проектування

Етап проектування	Функція ЕС
Технічна пропозиція	Уточнення та формалізація поставленої задачі. Узгодження постановки задачі зі стандартами, нормами, правилами. Зведення технічного завдання до вигляду, потрібного для подальшої обробки та подання в системі проектування. Пошук готових рішень та аналогів, оцінювання можливості їх застосування стосовно цієї задачі й доопрацювання аналогів. Аналіз переважного орієнтування на аналог або нову розробку; оцінювання компромісного співвідношення
Ескізний проект	Вибір концептуальних схем і базових принципів, які закладаються в об'єкт проектування. Вибір структурних схем і компоновальних рішень
Технічний проект	Визначення та призначення параметрів об'єкта проектування, а також його окремих складових
Робочий проект	Вибір алгоритмів розрахунків. Інтерпретування результатів числового та імітаційного моделювання. Загальне управління (організація) спрямованим процесом розрахунку та оптимізації параметрів об'єкта проектування (тобто координування роботи модулів числових розрахунків, моделювання, статистичної обробки) відповідно до заданих критеріїв. Вибір стандартизованих й уніфікованих вузлів з БД, які мають характеристики, що задовольняють поставлені вимоги
Випробування дослідного зразка	Формування плану експериментального аналізу обробки й узагальнення результатів експериментів. Підготовка рекомендацій для коригування проекту

За ступенем участі в проектуванні можна виділити дві групи функцій ЕС: основні та супровідні. Перші пов'язані зі створенням нового об'єкта (серед них зокрема виділяють структурний синтез, розрахунок характеристик тощо); другі — з аналізом патентної чистоти, редагуванням тощо.

На попередніх етапах проектування основна роль відводиться власне ЕС, спряженій з великим банком типових розв'язків і моделей. На дальших етапах, де значне місце займають числові розрахунки, основні задачі розв'язуються системою, що вписується в концепцію гібридної інтелектуальної системи (ГІС), про яку детальніше йдеться в роз. 6. Особливості, які виникають при цьому в контексті задач САПР, можна розглянути на прикладі взаємодії ЕС і *методу скінченних елементів* (МСЕ).

**Експертні системи і МСЕ.** Велике поширення й висока ефективність МСЕ у поєднанні зі специфічними труднощами його застосування стимулюють спільне використання ЕС і МСЕ у межах інтелектуальних САПР.

Функціями ЕС, пов'язаними з підтримкою МСЕ, можуть бути: вибір типових елементів для окремих конструкцій та їх складових; аналіз структури матриці жорсткості; вибір алгоритмів розрахунків; інтерпретування результатів розрахунків; вибір кроку дискретизації; формалізація характеристик навантажень; урахування властивостей матеріалу тощо. Прикладами використовуваних при цьому правил є такі:

ЯКЩО  
ТО  
об'єктом аналізу є вал,  
застосувати вісесиметричні елементи

ЯКЩО  
ТО  
матриця суміжності має рядкову структуру,  
використати фронтальний метод (розрахунок).

Для розвинених САПР, які покладають на ЕС, наприклад, задачу вибору елементів із значної кількості типових або навіть самостійне формування їх конфігурації, а також для систем, які працюють з об'єктами дуже складних конструкцій, особливу роль відіграють глибинні знання, що відображають основні фізичні закономірності і є універсальними або спеціалізованими математичними моделями, пов'язаними з МСЕ. Зокрема, такі знання можуть відображати співвідношення сил у стані рівноваги (рис. 9.5, а), залежність сили пружності від переміщення (рис. 9.5, б), типи напружень та умови їх виникнення і різні критичні співвідношення (рис. 9.5, в) тощо.

Для подання знань в інтелектуальних компонентах систем, орієнтованих на МСЕ, найбільш доцільними (ефективними) є комбіновані моделі: фрейми в поєднанні з правилами (крім числових процедур саме МСЕ).

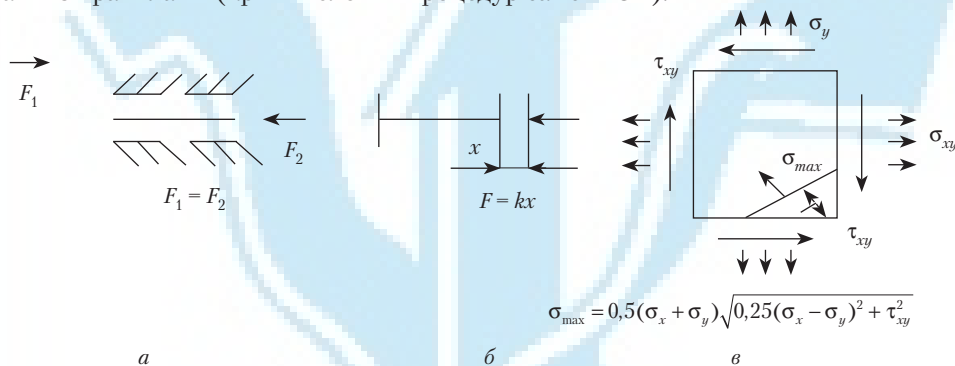


Рис. 9.5. Співвідношення, закладені в глибинних знаннях: умови виникнення критичних співвідношень ( $\sigma_{\max}$  – максимум головного скорочення)

Продукції спрямовують здійснення загальної стратегії методу, ініціюють на окремих етапах звернення до конкретних фреймів, а також допомагають у виборі елементів тощо. Фрейми відображають основну суть, пов'язану з реалізацією методу: скінченні елементи та характеристики їх застосування, основні алгоритми розрахунку й галузі їх використання, форми конструкцій і варіанти їх дискретизації тощо.

Умовний приклад організації розв'язання задачі при реалізації МСЕ з підтримкою ЕС показано на рис. 9.6.

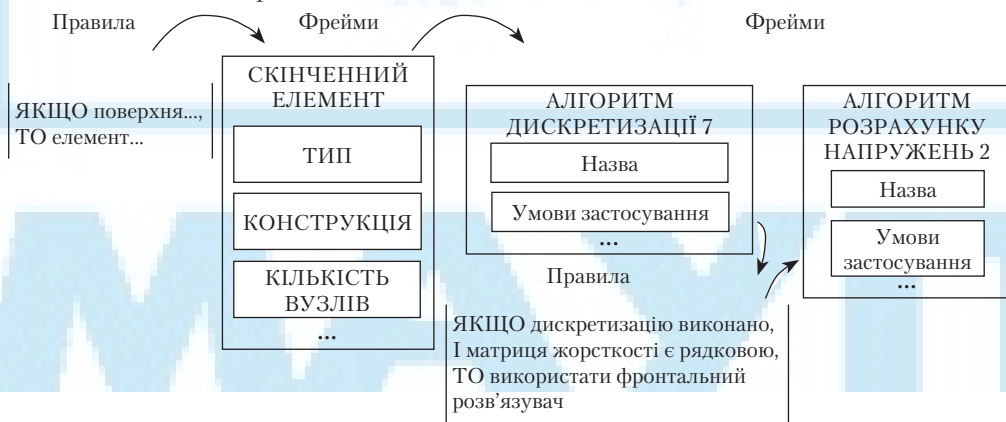


Рис. 9.6. Умовний приклад послдовності активізування фрагментів БЗ при реалізації МСЕ

**Етапи планування.** В роботі інтелектуальної системи планування технологічного процесу можна виділити два основних етапи:

- перетворення та інтерпретування початкових даних, які надходять у зовнішньому форматі;
- аналіз опису і синтез плану.

*Початкові дані* до планування технологічного процесу надходять у вигляді файлів, сформованих САПР, або задаються користувачем безпосередньо. Ці дані відображають геометричну модель об'єкта.

Зміст **першого етапу** роботи інтелектуальної складової системи зводиться до аналізу початкового опису технологічного процесу, поданого в термінах геометричних (графічних) *примітивів*, а також до інтерпретування цього опису на основі використовуваних понять, що стосуються технологічного процесу. Інакше кажучи, на основі геометричного опису відбуваються пошук, виділення та розпізнавання основних конструктивних елементів об'єкта. Наприклад, короткий відрізок  $AB$ , що з'єднує два інших, розташованих під прямим кутом один до одного на геометричній моделі тіла обертання (рис. 9.7, *a*), інтерпретується як фаска.

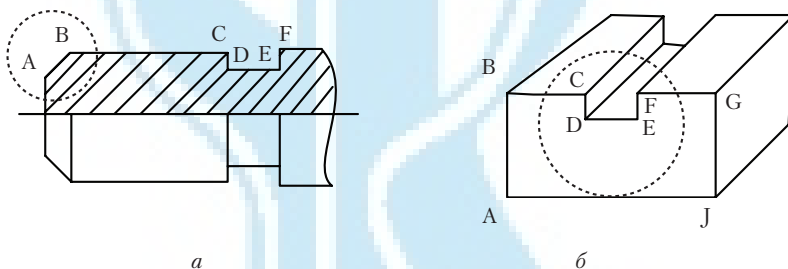


Рис. 9.7. Виділення конструктивних елементів фаски (*a*) та паза (*б*) за геометричними моделями деталей

Ефективність процесу пошуку та ідентифікація конструктивних елементів значною мірою залежать від стратегії обходу геометричної моделі. Один з варіантів — попереднє ланцюгове впорядкування примітивів уздовж зовнішньої контурної лінії (як правило, для тіл обертання), коли залежно від з'єднання суміжних елементів (відрізків, дуг тощо) визначається тип конструктивних елементів. Наприклад, фрагмент  $CDEF$  (рис. 9.7, *б*) інтерпретується як паз.

У більш складних випадках може застосовуватися впорядкування примітивів на основі черговості перетину окремих елементів моделі з деякими умовними площинами, як правило, перпендикулярними до осей координат і переміщуваними уздовж них (рис. 9.8). Найбільш розвинені підходи є комбінацією різних стратегій.

Після виділення конструктивних елементів об'єкта й аналізу їх взаємного розташування настає **другий етап** роботи ЕС — аналіз опису та синтез плану, коли у загальному випадку виділяються такі фази:

- вибір технологічних операцій, пов'язаних з конкретними конструктивними елементами об'єкта і, можливо, часткове впорядкування виконуваних операцій;
- локальна оптимізація черговості виконання операцій;
- визначення загальної черговості виконання операцій.

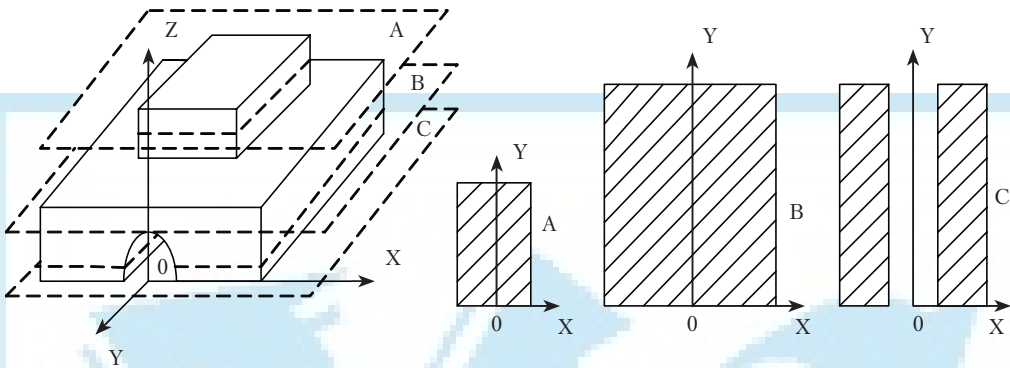


Рис. 9.8. Впорядкування примітивів на основі черговості перетину окремих елементів моделі з умовними площинами

*Перша фаза* визначається характером конструктивних елементів об'єкта і здійснюється на основі інформації про зв'язки «дія (операція) – результат». Найпростішим прикладом тут є правило «ЯКЩО елемент – (наскрізний отвір), ТО операція – (свердління)». Часткове впорядкування операцій здійснюється на основі фіксованих апріорних відомостей про припустимі варіанти черговості. Наприклад, «ЯКЩО конструктивний елемент об'єкта утворюється внаслідок операцій фрезерування та шліфування, ТО фрезерування передує шліфуванню».

*Друга фаза* спрямована на реалізацію окремих (заданих) конструктивних елементів об'єкта та їх груп за допомогою мінімальної сукупності дій (або найбільш простих дій). Зокрема, локальна оптимізація здійснюється поєднанням кількох однотипних дій, які необхідні для одержання різних конструктивних елементів. Наприклад, деталь на рис. 9.9, *а* можна одержати після виконання послідовності операцій згідно з рис. 9.9, *б*. Проте послідовність операцій відповідно до рис. 9.9, *в* технологічно більш ефективна. Остання може бути забезпечена застосуванням правил типу

ЯКЩО	об'єкт – тіло обертання
І	є відрізки різного діаметра,
ТО	спочатку виконувати токарну обробку найбільших відрізків, що перекриваються поточним значенням діаметра, в порядку зменшення діаметрів,

тобто те, що відповідає рис. 9.9, *в*.

*Третя фаза* складається з окремих операцій і/або їх локальних упорядкованих наборів, одержаних у другій фазі (локальної оптимізації). При цьому враховуються загальні відомості про технологічний процес, розташування устаткування тощо. Як правило, визначення загальної черговості виконання операцій має ітераційний характер з подальшим її поліпшенням. Тут для попередньої фази може використовуватися впорядкування виконуваних операцій, одержане на першому етапі (тобто при інтерпретуванні геометричної моделі об'єкта).



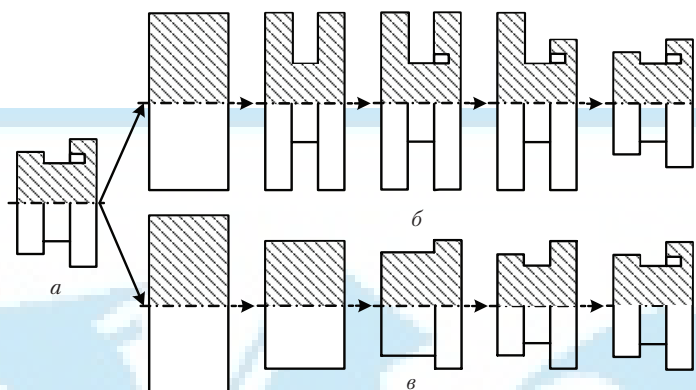


Рис. 9.9. Деталь (а) та варіанти черговості операцій (б, в) при її виготовленні

### 9.3. Методи пошуку розв’язків на прикладах задач планування

**Пошук у просторі станів.** Розглянемо можливе подання задачі стосовно формування плану-графіка завантаження устаткування. Початковий його стан  $I_0$  характеризується множиною завдань, які складаються з частково впорядкованих наборів операцій (рис. 9.10). Дія кожного оператора  $O_i$  полягає в перенесенні найближчої операції одного з наборів у чергу до одного з відповідних верстатів. У кінцевому стані устаткування всі операції переміщуються в черзі. Процесу розв’язання такої задачі відповідає граф послідовності подій (рис. 9.11).

Якщо мета полягає лише у виконанні операцій, а часові критерії відсутні (це, мабуть, нереально для практики), то кожна термінальна вершина графа служить розв’язком. При накладенні обмежень типу директивних строків без використання будь-якої цільової функції розв’язання досягається «пошуком у глибину» або «пошуком у ширину» з перевіркою виконання обмежень у кожній новій вершині та із завершенням пошуку в першій термінальній вершині, що задовольняє обмеження.

Очевидно, така схема розв’язання розглядуваної задачі є вкрай неефективною й може призвести до повного перебирання. Мало того, за наявності критерію оптимальності саме ця схема й призводить до повного перебирання, оскільки заздалегідь невідомо, яка з допустимих термінальних вершин виявиться кращою (адже множина цих вершин задається не в явному вигляді, а умовою). Отже, такий підхід у чистому вигляді не підходить для складних реальних застосувань.

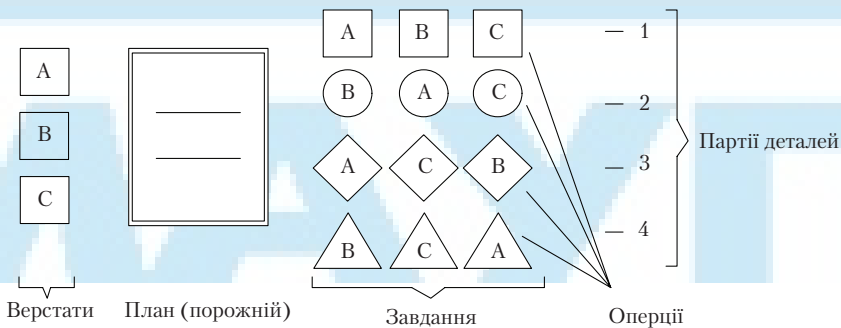


Рис. 9.10. Частково впорядковані набори операцій

Скорочення кількості варіантів оптимальних (або раціональних) розв'язків досягається використанням спеціальних оцінних функцій, евристичних правил, додаткових обмежуючих узгоджень. Так, в основі багатьох практичних методів лежить схема, що імітує послідовність подій у реальній системі (див. рис. 9.11). Ця схема реалізує односпрямований рух уздовж тільки одного з шляхів графа  $G$ , які об'єднують початкову та термінальну вершини.

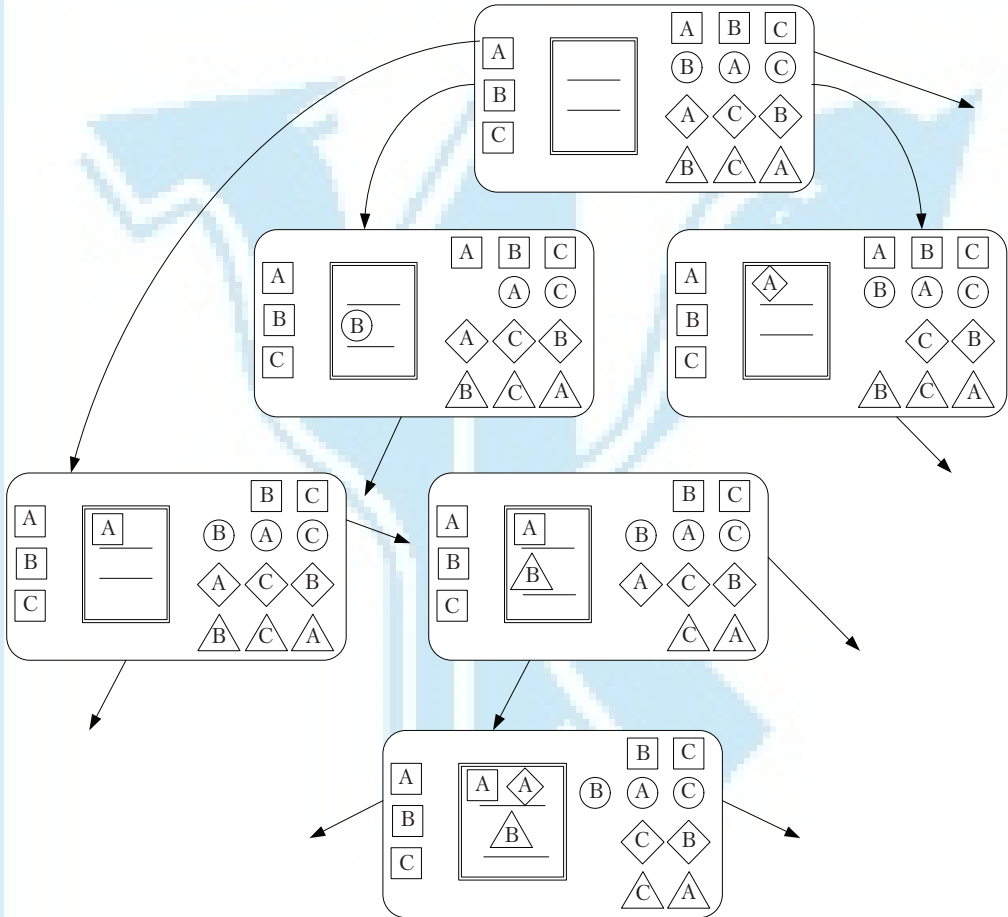


Рис. 9.11. Послідовність подій у реальній системі

Справді, в цьому разі хоча й існують альтернативні варіанти в кожній з вершин, однак після прийняття рішення (вибору партії серед претендентів) повернення не здійснюються, а всі потенційно можливі наступні стани подій локалізуються розташованим нижче підграфом. При цьому оптимальний розв'язок не гарантується, оскільки він може розміщуватися поза цим підграфом. Як компенсація цього недоліку — простота реалізації, скорочення простору пошуку при прийнятних показниках одержуваного субоптимального розв'язку.

Практично описана схема розв'язання наведеної задачі доцільна в алгоритмічному вигляді, а не засобами ЕС, хоча в спрощеному вигляді вона може моделюватися набором таких правил:

ЯКЩО	операції не виконувались,	} (9.1)
ТО	виконувати початкове завантаження;	
ЯКЩО	є невиконані операції,	
ТО	вибрати непозначений верстат з мінімальним часом очікування;	
ЯКЩО	верстат вибрано,	
ТО	здійснити пошук претендентів;	
ЯКЩО	претенденти відсутні,	
ТО	замінити верстат;	
ЯКЩО	претендентів знайдено,	
ТО	вибрати кращий з них;	
ЯКЩО	кращий з претендентів вибрано,	}
ТО	здійснити завантаження верстата й звільнити верстати від міток	

Очевидно основним кроком, який визначає ефективність пошуку в загальній схемі розв'язання задачі й у стратегіях розглянутого вище типу, є вибір чергового оператора (відповідно вибір кращого претендента в правилах (9.1)). Зміст цього кроку відрізняє одну стратегію пошуку від іншої. Зміст інших кроків (можливо, за винятком початкового завантаження) схожий. Тому нижче розглянемо приклади можливих правил саме щодо цього кроку (основного), тобто щодо вибору кращого претендента.

Тоді за наявності тільки вимоги виконання завдання (без критерію оптимальності) можливі правила матимуть вигляд:

ЯКЩО	здійснюється вибір кращого претендента
І	використовуються обмеження за ...,
І	...,
ТО	вибрати партію з ...,

а за наявності критерію оптимальності — вигляд

ЯКЩО	здійснюється вибір кращого претендента
І	використовуються обмеження за ...,
І	критерій оптимальності ...,
І	...,
ТО	вибрати партію з ...

У подальшому для зручності опису наведених нижче прикладів загальні елементи в лівій частині правил (тобто умови «вибір кращого претендента» та «використовуються обмеження за...») упускатимемо. Спочатку розглянемо випадок наявності тільки вимоги виконання завдань. Як правило, в цьому разі фігурують директивні строки їх завершення і можуть використовуватися, наприклад, такі евристичні правила для визначення напрямку перебирання кращого претендента:

ЯКЩО	значення директивних строків наближаються,
ТО	вибрати завдання з мінімальною тривалістю операцій, що залишилися;

ЯКЩО	директивні строки відрізняються,
ТО	вибрати завдання з мінімальною різницею $a_i - d_i$ ,

де  $a_i = \sum_j t_{ij}^*$ ,  $t_{ij}^*$  — тривалість обслуговування партії деталей  $i$  на  $j^*$ -й операції її технологічного маршруту (ТМ);  $j^*$  — номери решти операцій  $d_i$  — директивний строк обслуговування партії деталей  $i$ .

Як оцінка може бути вибрана або функція  $\xi_1 = \sum_i \sigma(z_i)$ , де  $z_i = t_i^* + \sum_{j^*} t_{ij^*} - d_i$ , а  $t_i^*$  — момент завершення останньої виконаної (на поточний момент) операції над  $i$ -ю партією деталей, причому

$$\begin{cases} \sigma(x) = x \text{ при } x > 0; \\ \sigma(x) = 0 \text{ при } x \leq 0, \end{cases}$$

або функція

$$\xi_2 = \lambda_1 \sum_i \sigma(z_i) + \lambda_2 \frac{\sum_i \sum_{j^*} t_{ij^*}}{\sum_i \left( \sum_{j^*} t_{ij^*} + \sigma(-z_i) \right)},$$

де  $\lambda_1, \lambda_2$  — вагові коефіцієнти, або функція

$$\xi_3 = \lambda_1 \sum_i \sigma(z_i) + \max_i \left\{ \frac{\sum_{j^*} t_{ij^*}}{\left( \sum_{j^*} t_{ij^*} + \sigma(-z_i) \right)} \right\},$$

і т. д. (припускається, що для кожної з цих функцій  $\xi_k \rightarrow \min$ ).

За наявності критерію оптимальності залежно від його конкретного вигляду можуть застосовуватися такі евристичні правила, а також їх комбінації:

- ЯКЩО критерієм оптимальності є мінімум тривалості циклу,
- ТО вибрати претендента з мінімальною тривалістю на цьому кроці вибору поточної операції;
- ЯКЩО критерієм оптимальності є мінімум тривалості циклу,
- ТО вибрати претендента з мінімальною кількістю операцій, що залишилися;
- ЯКЩО критерієм оптимальності є мінімум тривалості циклу,
- ТО вибрати претендента з мінімальною різницею  $\lambda_i$ ;
- ЯКЩО критерієм оптимальності є мінімум тривалості простоїв устаткування,
- ТО вибрати претендента, для якого наступна операція виконується на верстаті, що простоює.

У наведених прикладах з критерієм оптимальності як оцінки можна вводити, наприклад, такі функції

$$\xi_4 = \sum_i \sum_{j^*} t_{ij^*}; \quad \xi_5 = \max_i \left\{ \sum_{j^*} t_{ij^*} \right\}; \quad \xi_4, \xi_5 \rightarrow \min.$$

**Пошук в ієрархії просторів.** В основі цього підходу лежить ієрархічне подання компонентів проблемної галузі, наприклад, планування. Ієрархія може утворюватись у межах: власне *простору станів* (планів); *множини цілей*; *множини операторів і стратегій* щодо формування плану. Це досягається як виділенням безпосередніх складових планування, так і формуванням рівнів абстракції, коли підпростір верхнього рівня є узагальненим, абстрагованим від деяких елементів подання підпростору нижнього рівня (рис. 9.1).

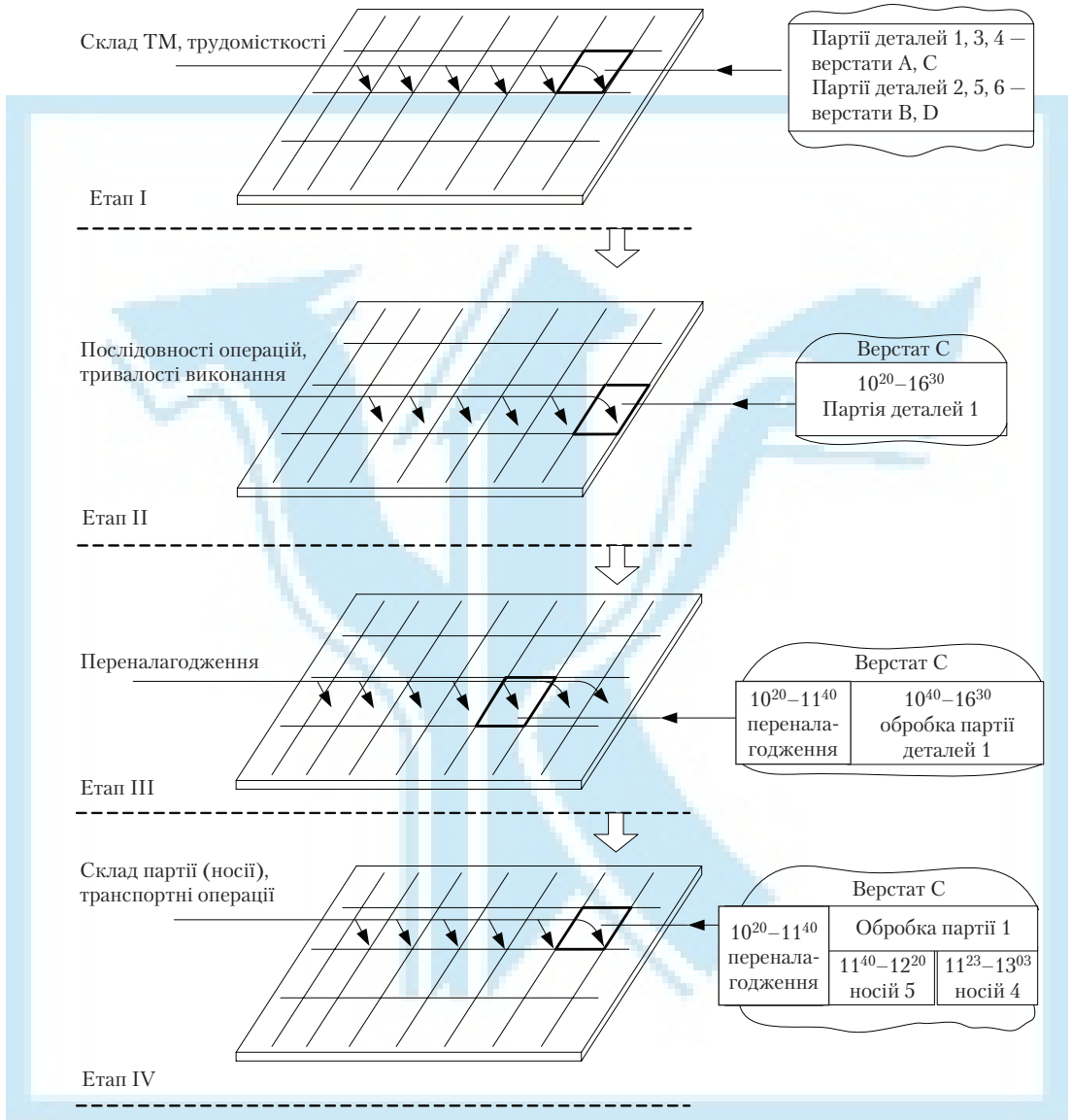


Рис. 9.12. Приклад низхідного уточнення виробничого плану

Основним об'єктом зазначеного ієрархічного подання є план. У цьому разі підставою для виділення рівнів є різний ступінь урахування в ньому основних компонентів і параметрів виробничого процесу, а саме:

- повне подання фрагмента плану (наприклад, транспортні операції явно наводяться в плані, зазначаються їх часові інтервали і технічні засоби для виконання);
- непряме подання фрагмента (наприклад, транспортні операції та їх виконавці в плані не зазначаються, але тривалості виконання операцій враховуються в тривалостях обробки відповідних партій деталей);

- ігнорування фрагмента (наприклад, коли посилання на пересування вантажів відсутні, а тривалості їх виконання не враховано).

До основних фрагментів планування, що впливають на виділення рівнів, належать: *переналагодження устаткування* (його наявність, тривалості використання тощо); *структури завдань* (партія – набір носіїв – сукупність деталей тощо); *транспортні операції* (розглядалися вище); *склад ресурсів* (однотипні групи устаткування із сумарною продуктивністю, окремі одиниці устаткування тощо).

Схема розв'язання задачі будується на *принципі низхідного уточнення* (top-down refinement): план, одержуваний на більш високому рівні, послідовно деталізується на більш низьких рівнях.

У загальному випадку прийняття рішень на різних рівнях планування може бути рознесено в часі. До того ж окремі рівні можуть виконуватися на різних стадіях виробничого процесу. Так, формування плану в межах етапів I–III (див. рис. 9.12) здійснюється до запуску партій деталей на обробку, проте рішення, що деталізують план до рівня руху носіїв (етап IV на рис. 9.12), приймаються в процесі диспетчеризування при виконанні завдання (тобто в режимі реального часу).

На різних рівнях планування можуть використовуватися й різні моделі. Так, ефективним є підхід, коли на верхніх рівнях застосовується фреймовий граф (рис. 9.12) як пошук у просторі станів (для початкового групування партії деталей за верстатами, призначення початкових налагоджень тощо), тоді як детальний розклад формується на нижчих рівнях з використанням імітаційної моделі (ІМ).

**Принцип найменших завершень.** Ідея цього принципу полягає у відкладанні розв'язування окремих задач до одержання додаткової інформації, яка може бути одержана в процесі розв'язування інших задач. У міру надходження такої інформації, яка сприяє прийняттю кращого рішення, розв'язуються відкладені задачі, тобто здійснюється чергування окремих розв'язків задач, яке супроводжується спрямованим інформаційним обміном – одержаними окремими розв'язками, даними тощо.

Як правило, принцип найменших завершень застосовується при ієрархічному поданні простору пошуку, коли уточнення розв'язків, одержуваних на більш високих рівнях абстракції, відбувається після розв'язання окремих, більш деталізованих задач. Однак суть підходу залишається справедливою і щодо підзадач одного рівня. Більше того, змістову узагальненість за цим принципом можуть мати стратегії, що характеризуються суттєвим часовим розподілом етапів й умовами їх виконання.

У галузі планування можна навести приклад, коли попередньо формується розклад, який закріплює партії деталей за окремими групами устаткування, а конкретне чергування носіїв (у межах партії деталей) і пункти призначення транспортного засобу визначаються вже в процесі диспетчеризування, тобто в режимі реального часу.

**Приклад 9.4.** Для ілюстрації принципу найменших завершень розглянемо такий спрощений приклад. Припустимо, що при виборі конкуруючих претендентів на обслуговування (партії деталей 1 і 2 на рис. 9.13, а) ресурсом А враховується, зокрема, як скоро до нього надійдуть наступні претенденти. Проте потенційні претенденти (партії деталей 3 і 4) на момент прийняття рішення можуть бути ще не закріплені за тими ресурсами, які відповідають першочерговим етапам їх ТМ, тобто за верстатом 8. До того ж вони можуть конкурувати як між собою, так і з іншими партіями деталей.

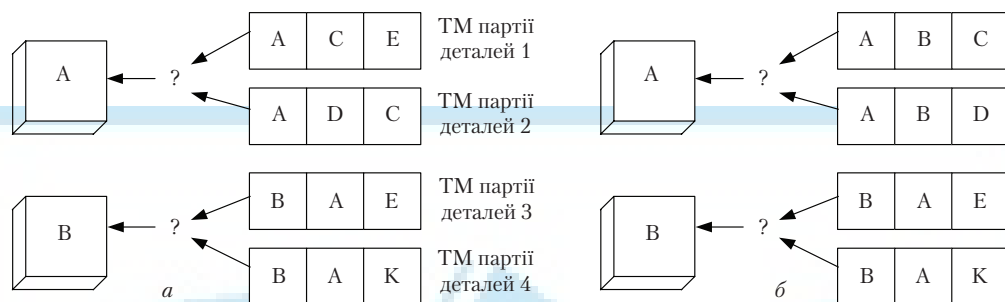


Рис. 9.13. Урахування потенційних претендентів (розв'язання конфліктів при їх виборі)  
а – рівні обліку; б – тупикова ситуація

У цьому разі призначення ресурсу (верстата) А може бути відкладене, а здійснюватиметься закріплення відповідних партій деталей за ресурсом (верстатом) В з подальшим поверненням до уточнення їх закріплення за ресурсом А. Проте оскільки вибір серед партій деталей 3 і 4 здійснюється за тим самим принципом, це може призвести до досить довгих ланцюжків очікування залежних підзадач, а в гіршому разі – до появи тупикових ситуацій (рис. 9.13, б). Тут вибір серед партій деталей 1 і 2 відкладено до закріплення партії деталей 3 або 4 за ресурсом В, що, у свою чергу, залежить від вибору партій деталей, що закріплюються за ресурсом А. Практично можуть виникати довгі та складніші цикли залежності.

Щоб запобігти надмірному збільшенню кількості відкладених задач, можна використовувати обмеження ланцюжків очікування із застосуванням на кінцевих підзадачах інших критеріїв або мішаних стратегій вибору. Для розв'язання тупикових ситуацій, зокрема, використовуються рівноймовірнісний випадковий вибір, додаткові евристичні, різні для кожної з підзадач критерії, а також вибір претендентів за заздалегідь призначеним пріоритетом. Зазначені підходи можуть комбінуватись або може застосовуватися випадковий вибір з імовірностями, які пропорційні відносним пріоритетам претендентів за схемою методу Монте-Карло та ін.

**Приклад 9.5.** Використання принципу найменших завершень у разі ієрархічного пошуку розглянемо на такому прикладі. Нехай верхній рівень (рис. 9.14) складання плану полягає у формуванні груп різних претендентів, які закріплюються за різними (чи різнотиповими) наборами ресурсів на певному часовому інтервалі, а на нижньому рівні розв'язуються конфлікти між конкуруючими претендентами (якщо такі залишаються після вибору на верхньому рівні з урахуванням конкретних значень часу), тобто власне формування розкладу запусків партій деталей.

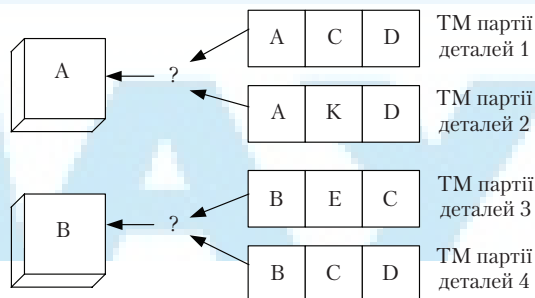


Рис. 9.14. Урахування потенційної конкуренції претендентів

При цьому в процесі формування груп партій деталей (верхній рівень) критерієм є зменшення їх конкуренції на подальших етапах обробки (тобто стратегія тут впливає з евристичного припущення про бажаність запуску в одночасну обробку тих партій деталей, які не створюють потенційних перешкод одна одній у подальшому). Так, згідно з рис. 9.14, на ресурси (верстати) А та В надходять відповідно партії деталей 2, 3.

Очевидно, початкове призначення груп претендентів для всіх ситуацій, що відрізняються ступенем обробки партій деталей, практично неможливе й недоцільне внаслідок дуже великої кількості його можливих варіантів. Звуження ж набору ситуацій, які аналізуються, на кожному наступному часовому інтервалі можливе лише після виділення конкретних значень моментів часу для інтервалу, що минув (тобто в результаті формування його розкладу). При цьому може статися, що зайнятість деякого ресурсу охоплює й наступний часовий інтервал і, таким чином, виключає цей ресурс із списку доступних.

Зауважимо, що поняття часового інтервалу є досить умовним: йдеться не про попередній розподіл планового періоду на фіксовані проміжки, а скоріше говорять про плаваючий (поточний) проміжок часу, до того ж не жорсткої тривалості, а такий, що займає деякий прийнятний допуск і підлягає аналізу через звільнення в ньому кількості (відносно великої) ресурсів. У такому разі формування плану полягає в чергуванні етапів формування груп претендентів з етапами деталізації розкладу в межах груп.

За наявності претендентів з близькими параметрами група може формуватися із включенням до неї претендентів, які конкурують на цей ресурс, і тоді конфлікт розв'язуватиметься на нижньому рівні. В разі відсутності невизначеності, а також у разі повної узгодженості наступних конкуруючих етапів значення часового інтервалу може призначатись у межах етапів формування груп.

**Використання глибинних знань.** Застосування цих знань при розв'язанні задач планування передбачає врахування основних змістових і причинних взаємозв'язків між компонентами (поняттями), що лежать в основі поняття «план», а також урахування основних принципів його формування та впливу можливих ситуацій на значення базових критеріїв.

**Приклад 9.6.** Розглянемо такий спрощений приклад. Маємо один верстат і кілька деталей-претендентів на їх обробку. Вибраним критерієм оптимальності є мінімум сумарного часу очікування. Використовуємо правило, яке називається *SPT-впорядкуванням* (Shortest Processing Time Sequencing):

ЯКЩО критерієм є мінімум часу очікування,  
ТО впорядкувати заявки за низхідною тривалістю виконання операцій.

Якщо ж деякі деталі потребують обов'язкового переналадження устаткування, то це правило набуває вигляду:

ЯКЩО критерієм є мінімум часу очікування,  
ТО впорядкувати заявки за низхідною сумою тривалостей переналадження устаткування й обробки деталей.

Іноді ставиться вимога враховувати тривалість виконання транспортних операцій. Для цього необхідно заздалегідь зазначити в консеквенті всі можливі складові або деталізувати антецеденти та згрупувати правила за умовами їх виконання:

ЯКЩО критерієм є ...  
I переналадження устаткування не потрібне,  
I час транспортування деталей несуттєвий,



ТО	впорядкувати заявки за низхідною тривалістю виконання операцій.
ЯКЩО	критерієм є ...
І	переналагодження устаткування потрібне,
І	...,
ТО	впорядкувати заявки за низхідною сумою ...

і т. д.

Очевидно, в реальній ситуації розглянутий підхід потребуватиме суттєвого розширення БЗ, однак і це не дасть змоги передбачити всі можливі ситуації. Водночас, як це легко передбачити, праві частини наведених вище правил мають загальний зміст: вони лише відображають той факт, що дія може завершитися тільки після закінчення всіх передумов.

Стосовно ситуації планування можна записати таке правило:

ЯКЩО	допоміжна дія $D_i$ безпосередньо пов'язана з операцією $w_i$	}	(9.2)
І	$D_i$ відбувається неодноразово з $w_i$ ,		
ТО	включити тривалість дії $D_i$ в сумарну трудомісткість операції $w_i$		

Очевидно також, що в БЗ будуть необхідними такі твердження (факти):

переналагодження – допоміжна дія;	}	(9.3)
переналагодження передують обробці		

Тоді правило віддавання переваги щодо впорядкування заявок матиме вигляд:

ЯКЩО	критерієм є мінімум часу очікування,	}	(9.4)
ТО	впорядкувати заявки за низхідною сумарною трудомісткістю		

Це дасть змогу запобігати надмірній деталізації й стосовно інших критеріїв, при розрахунку яких застосовується та сама величина. Так, за наявності директивних строків справедливим буде правило:

ЯКЩО	критерієм є мінімум середнього запізнювання,	}	(9.5)
ТО	впорядкувати заявки за низхідною сумарною трудомісткістю		

Як бачимо, правила (9.2) і (9.3) не пов'язані безпосередньо з будь-яким критерієм або видом упорядкування, а виражають більш загальні закономірності і відношення, які ґрунтуються на уявленні про тривалість, причинно-наслідкові зв'язки, особливості організації виробничого процесу тощо. Водночас саме використання цих правил дає змогу підвищити ступінь «універсальності» рішень стосовно окремих критеріїв і, що суттєво, підготувати деякий базис для можливих майбутніх рішень, у тому числі й за іншими критеріями.

Знання, що розглядаються як глибинні при розв'язанні задач планування, містять:

- знання, *інваріантні щодо конкретного методу формування плану;*
- знання, *пов'язані з конкретною стратегією.*

У свою чергу, **перша група** знань складається:

а) із знань про загальні взаємозв'язки сутностей, що враховуються при формуванні плану. Прикладами таких знань є правило (9.2), а також правило:

ЯКЩО	заявки впорядковано,
ТО	сумарний час очікування дорівнює сумі добутків трудомісткостей заявок на кількість подальших заявок;

б) із знань, які виражають деякі закономірності загальноматематичного характеру. Наприклад:

ЯКЩО мінімуму сумарної величини досягнуто,

ТО досягнутим є й мінімум її середнього значення,

з чого випливають відповідні твердження щодо загальних і середніх простоїв, очікувань тощо. Ще один приклад таких знань: мінімум суми попарних добутків елементів двох послідовностей досягається при їх протилежному впорядкуванні, звідки з урахуванням правила (9.4) за змістом впливає правило (9.5);

в) із знань, що виражають загальні закономірності та сутності виробничого процесу. Прикладом цих знань є правило (9.5).

**Друга група** знань складається зі знань, які мають вузькоспеціалізований характер і пов'язані з принципами, що лежать в основі конкретних методів розв'язань.

Застосування глибинних знань може бути доцільним у системах планування, що використовуються в умовах складної та динамічної структури зв'язків компонентів виробничої системи, при широкій та швидкозмінюваній номенклатурі продукції, наявності численних додаткових обмежень (у тому числі директивних строків), тобто якщо неможливо визначити досить малий набір ефективних детермінованих стратегій для всього комплексу задач, які виникають.

Водночас використання глибинних знань є зайвим в умовах порівняно стабільного та передбачуваного потоку заявок або в умовах диспетчеризування з прийняттям рішень у реальному масштабі часу (див. розд. 10).

#### 9.4. Експертні системи в плануванні та моделюванні процесів підготовчої стадії життєвого циклу виробів

**Інженерно-конструкторська діяльність як об'єкт моделювання та керування.** Об'єктивні умови розвитку сучасних виробництв висунули задачу підвищення рівня автоматизації на всіх стадіях життєвого циклу виробу — підготовчій, виробничій та експлуатаційній [131]. Зокрема, збільшення номенклатури виробів за умови їх випуску малими серіями підвищує вимоги щодо скорочення підготовчої стадії. Отже, збільшується актуальність запровадження нових методів і засобів прискорення процесу розроблення виробів, які б ґрунтувалися на комплексному підході до планування та управління процесами на всіх стадіях життєвого циклу виробів.

Власне наявність «наскрізного» життєвого циклу виробу й спонукає використовувати нові організаційні інженерно-виробничі структури, які б об'єднували фази наукового пророблення, технологічної та конструкторської підготовки виробництва безпосередньо із стадією виробництва. Йдеться про створення *інженерно-виробничих комплексів* (ІВК), які дають змогу спростити планування роботи підрозділів на підготовчій стадії, значно полегшити коригування планів, застосувати єдиний математичний апарат і методи формалізованого опису процесів ІВК як *об'єкта планування та керування*.

**Скінченно-автоматна інтерпретація керування дискретними процесами у ГКІС.** Окрім функціональної та організаційної аналогій (які розглядаються у цьому підрозділі далі) в процедурах оперативного керування процесами підготовки виробництва, освоєння та виготовлення нових виробів можна відзначити також багато спільного в інтерпретуванні структури та функціонуванні засобів технічного оснащення ГВС і комплексу технічних засобів системи керування. При цьому *подання принципів функціонування інформаційних потоків у підрозділах ІВК, що під-*

готовляють виробництво нових виробів, може базуватися на єдиній методологічній основі з описом принципів функціонування матеріальних потоків у підрозділах, які освоюють та виготовляють ці вироби. Спробуємо це довести.

Почнемо з того, що як інформаційні, енергетичні, так і заготовки, продукція, оснащення інструменту тощо являють собою *матеріальні потоки*. *Методи доступу* до внутрішніх накопичувачів (нагромаджувачів) інструментів, заготовок та оснащення, які реалізуються у ГВС, можна формально подати методами доступу до довгочасної (на вінчестерах в ЕОМ та ЦС у ГВС) та оперативної («робоча/операційна пам'ять» на ЕОМ та міжопераційні накопичувачі заготовок/деталей/оснащення або магазини інструментів на обробних центрах у ГВС) пам'яті. *Інтегрування* окремих гнучких виробничих модулів та обробних центрів (як і будь-яких інших обробляючих ресурсів, до яких належать також промислові роботи, засоби упорядкування середовища, окремі технологічні машини тощо) як *обробляючих ресурсів* у ГВС *організовано за принципом мультипроцесора*, коли транспортна мережа, що зв'язує окремі обробляючі ресурси, легко асоціюється з транспортним каналом «загальна шина». Виробничі підрозділи в ІВК за допомогою такого каналу інтегруються за принципом «загальна пам'ять», якою виступають АСк та ЦС заготовок, готової продукції, інструменту та оснащення. *Роль інтерфейсу введення/виведення* для устаткування ГВС відіграють засоби завантаження/розвантаження заготовок/виробів/оснащення/інструменту, в тому числі ПР, автооператори. Перелічені компоненти ГКІС з'єднані між собою *шинами: адрес і даних* (потоки запитів, відповідей, повідомлень); *даних* (потік заготовок, виробів, оснащення, інструментів тощо); *керування* (потік диспетчерських команд). Окремі підсистеми та пристрої ГКІС можуть обмінюватися інформацією та отримувати диспетчерські вказівки за відповідними шинами, що необхідно для координації їх спільної роботи.

Наведене дає змогу сформулювати поняття про *технологічний процесор ГКІС* та скласти на його основі уявлення про *технологічне мультипроцесорне середовище ГКІС*, відповідно до якого процеси у ГКІС стають подібними до обчислювальних процесів у мультипроцесорних структурах ЕОМ [125; 141]. При цьому програмно-керовані засоби технічного оснащення ГКІС подаються у вигляді скінченних автоматів  $A_{o,k}$  — об'єктів керування з мікропроцесорним керуванням [23], об'єднаним датчиками стану устаткування та зовнішнього середовища в єдину інформаційну підсистему. Таке припущення є справедливим через те, що устаткування може виконувати скінченну кількість операцій, реагуючи на скінченну множину інструкцій. Локальні пристрої, які керують роботою устаткування, також будуються як скінченні автомати  $A_{z,k}$  — засоби керування. Ці засоби подають на входи  $A_{o,k}$  символи  $Y_j$ , що відповідають інструкціям, і зчитують з виходів символи  $X_i$ , які несуть інформацію про стан устаткування та виріб, що обробляється. При такій інтерпретації властивостей підхід до розгляду керування ГКІС з позицій теорії скінченних автоматів полягає в описі всіх функціональних модулів як скінченних автоматів і побудові для керування ними відповідної системи, яка теж є скінченним автоматом (або їх сукупністю). При цьому враховується, що процес переходу устаткування (обробляючого ресурсу — процесора у більш загальному трактуванні для всіх підрозділів ІВК) з одного стану в інший при надходженні керуючої дії (інструкції або директиви) не контролюється і фіксується лише результат переходу після досягнення бажаного ефекту.

Такий підхід дає змогу отримати формалізований опис виробничої системи (в тому числі подати функціонування підрозділів ІВК та окремих його виконавців) у

вигляді логічної сітки, яка складається із скінченних автоматів, та інтерпретувати керування ІВК на логічному рівні керування мережею автоматів, об'єднаних входами та виходами [139].

І якщо йдеться про виробничу стадію, то тут запропоновано цілий ряд формалізованих методів опису (теорію масового обслуговування, імітаційне моделювання, сіткові моделі, графові методи тощо). Водночас підготовчу стадію, зокрема її фазу конструкторської підготовки виробництва (КПВ), в цьому плані вивчено мало.

**Конструкторська підготовка виробництва як об'єкт скінченно-автоматної інтерпретації функціонування.** Робота конструкторських підрозділів спрямована на розроблення комплексу конструкторської документації (КД) в усьому обсязі  $V$  продукції за плановий період  $T_{\text{пл}}$  при обмежених ресурсах  $F$  (наявності певної кількості виконавців необхідної кваліфікації).

У табл. 9.2 наведено порівняльну характеристику параметрів «виконавців» або «обробляючих ресурсів», які забезпечують випуск продукції на підготовчій стадії (конструктор) і на стадії власне виробництва виробів (технологічне устаткування). Тобто віртуально конструкторські і технологічні підрозділи ІВК можна розглядати як багатопроцесорне (власне, як і в обчислювальних мережах) середовище, де обробляючими ресурсами є автоматизовані робочі місця (АРМ) з користувачами – розробниками технологічної та конструкторської документації. Отже, в обох випадках ресурс можна подати у вигляді технологічної позиції, що характеризується рядом параметрів, тобто

$$F_j = \{ \Delta T_{ij}, T_{F_{ij}}, C, J, H_j, \Phi \}, \quad (9.6)$$

де  $\Delta T_{ij}$  – трудомісткість  $i$ -го виду робіт, що виконуються виконавцем  $j$ -ї кваліфікації (тривалість  $i$ -ї операції на  $j$ -му устаткуванні);  $T_{F_{ij}} = T_{\text{н}j} - \sum_{i=1}^I \Delta T_{ij}$  – резерв часу, причому  $T_{\text{н}j}$  – нормативний фонд часу для виконавця  $j$ -ї кваліфікації (або  $j$ -го типу устаткування);  $C$  – спеціалізація виконавця (або тип устаткування);  $H_j$  – досвід роботи виконавця (або показник надійності устаткування);  $\Phi$  – потік робіт (завдань або фонд робочого часу).

Таблиця 9.2

Порівняльна характеристика параметрів технологічного устаткування («обробляючих ресурсів») та конструктора

Технологічне устаткування	Конструктор
<i>Часові параметри</i>	
Директивний строк випуску $T_D$	Строк розробки комплексу КД $T_D$
Тривалість операції $\Delta T$	Трудомісткість $\Delta T$
Резерв часу $T_A$	Резерв часу $T_A$
<i>Технологічні параметри</i>	
Тип операції $J$	Вид роботи $J$
Тип необхідного устаткування $C$	Спеціалізація виконавця $C$
Клас складності $G$	Кваліфікація $G$
Надійність $H$	Досвід роботи $H$

Як і на виробничій стадії, у фазі КПВ, крім основної, задається деяка множина цілей  $Z = \{z_i\}$ , які реально відображають ситуацію на цьому етапі (наприклад, рівномірність завантаження конструкторського бюро (КБ), рівномірність завантажен-

ня виконавців усередині КБ, видача вчасно розробленого комплексу КД, прискорене розроблення КД на дефіцитні вироби тощо).

У фазі КПП на вхід ІВК надходять заявки на розробку робочої КД на виріб. На кожному з етапів КПП конструктори виконують визначений планом перелік проектувальних робіт, який закінчується випуском обов'язкових документів. Увесь перелік цих документів у фазі КПП становить *комплект робочої КД і є виходом цієї фази підготовчої стадії виробництва*.

Інформаційна обробка комплексу для планового замовлення  $P_l$  за  $l$ -м виробом визначається так:

$$\bar{P}_l = [D_l, Q_l, K_{il}^0], \quad (9.7)$$

де  $l \in L$  — множина виробів, які проектуються на плановий період;  $D_l$  — децимальний номер виробу, що проектується;  $Q_l = (i_{1l}, i_{2l}, \dots, i_{kl})$ ,  $i_{kl} \in I$  — проектна технологічна послідовність виконання робіт за  $l$ -м виробом;  $K_{il}$  — номер роботи в  $Q_l$ , з якої починається процес проектування виробу  $l$ .

Інформацію про ступінь готовності КД, яка має важливе значення при переплануванні та коригуванні планових завдань усередині конструкторських підрозділів, а також при розподілі завдань між ними, можна записати у вигляді вектора готовності

$$\bar{X}_{ik} = [D_l, A_{lk}, K_{il}, Q_i], \quad (9.8)$$

де  $A_{lk}$  — табельний номер виконавця, зайнятого  $k$ -ю проектною технологічною операцією за виробом  $l$ ;  $K_{il}$  — номер останньої завершальної проектною технологічної операції в  $Q_i$ .

При розробленні нового виробу враховується, що частина його деталей вже проектувалася для інших виробів і тому доцільно скористатися готовими кресленнями і технологічними картами. Для визначення наявності готової документації на інформаційний вхід ІВК на цьому етапі надходить інформація

$$U_{ln} = (D_l, M_l, \gamma_{ln}), \quad (9.9)$$

де  $M_l$  — тип складової виробу (складальна одиниця, деталь, кріпильні вироби тощо);  $\gamma_{ln}$  — параметр пошуку в БД проектною технологічною інформацією на  $M_l$ -складову;  $n \in E$  — множина пошукових критеріїв у БД.

Множина векторів  $\bar{P}_l$  (планові проектні замовлення) становить виробничу програму ІВК у фазі КПП на період  $T_{pl}$ .

*Керування здійснюється за подійними моментами  $\tau_s(S)$ , коли відбувається стрибкоподібна зміна ступеня готовності КД. За станом  $S(\tau_s) \rightarrow S(\tau_s+1)$  і контролюється цей ступінь.*

Залежно від алгоритму керування, вірогідності та якості надходжуваної інформації про стан ОК визначаються ефективність і якість керуючих дій.

Враховуючи зображальні здатності, а також можливості відображення характерних для процесів створення КД у фазі КПП особливостей (дискретність, альтернативність, паралельність і конкурентність), як базовий формальний апарат найкраще використовувати *ієрархічні пофарбовані СП* (ІПСП), в яких застосовуються дані про логічний взаємозв'язок подій; при цьому множина «умова — подія» визначає сукупність причинно-наслідкових зв'язків між подіями.

**Формалізація задачі організації керування ІВК.** Побудова моделі функціонування ІВК на СП аналогічна побудові математичної моделі будь-якого ОК і визначається такими діями:

1. *Процеси*, що відбуваються в ІВК (так само як і керування), *однозначно описуються* множиною «умова – подія» та *причинно-наслідковими відношеннями*, що нею встановлюються.
2. *Визначаються події*, послідовність появи яких встановлюється станами ОК. Стани задаються множиною умов у вигляді предикатів. Кількісним еквівалентом умов є їх ємність, яка може мати тільки натуральні значення.
3. *Умови настання подій можуть бути такими*, що виконуються і не виконуються; це й зумовлює можливість настання подій. Умови, з фактом яких пов'язується можливість реалізації події, називаються *передумовами*.
4. *Постумови* – це умови, які настають після виконання подій і перебувають з передумовами у причинно-наслідковому зв'язку.

Для кращого усвідомлення процесу використання СП при описі функціонування ІВК на підготовчій стадії життєвого циклу розроблюваних виробів нагадаємо деякі властивості звичайної СП – дводольного орієнтованого графа (означення 7.2) із зазначенням його елементів, які наведено в підрозд. 7.1. Формальною є така умова спрацьовування переходу  $p_i$  СП:

$$\mu(p_i) - I(p_i, t_j) \geq 0, \quad \forall p_i \in P, \quad (9.10)$$

де порівнюється кількість міток  $\mu(p_i)$  у позиції  $p_i$  з кількістю дуг графа між  $p_i$  та  $t_j$ .

Як результат спрацьовування переходу  $t_j$ , що задовольняє умову (9.10), маркування у вхідній позиції зникає і переходить у вихідну позицію:

$$\mu(p_{i+1}) = \mu(p_i) - I(p_i, t_j) + O(t_j, p_{i+1}), \quad \forall p_i \in P.$$

Отже, функціонування СП – це послідовна зміна маркувань  $\mu_i \xrightarrow{\sigma_i} \mu_j$  як наслідок спрацьовування її переходів, а стан сітки на поточний момент часу визначається її маркуванням.

**Означення 9.11.** *Позиція  $p \in P$  є обмеженою*, якщо існує таке значення  $k$ , що  $\mu(p) \leq k \quad \forall \mu \in R(N)$ . Коли  $k = 1$ , то позиція  $p$  буде безпечною. СП є безпечною, якщо безпечною є будь-яка її позиція. СП є  $k$ -обмеженою, якщо обмеженими є всі  $k$  її позицій.

**Означення 9.12.** *СП є живучою*, якщо:

$$\left. \begin{aligned} &\forall t \in T \exists \mu_i, \mu_j \in R(N), \mu_i \xrightarrow{\sigma_i} \mu_j; \\ &\forall \mu_i, \mu_j \in R(N), \mu_i \xrightarrow{\sigma_i} \mu_j \end{aligned} \right\}. \quad (9.11)$$

**Означення 9.13.** *Проблема досяжності полягає* у виявленні алгоритму, за допомогою якого для будь-якої СП  $N$  і для будь-якого її маркування  $\mu$  можна з'ясувати, чи належить  $\mu$  до множини досяжності ( $\mu \in R(N)$ ).

**Означення 9.14.** *Пофарбованою СП* є сітка, подана набором

$$N_\Phi = (P, T, \Omega, I, O, \sigma, \Phi, \mu_0), \quad (9.12)$$

де функція  $\sigma: (P \times \Omega) \times T \rightarrow (\emptyset, 1)$  визначає умови спрацьовування переходів сітки та розподілу кольорів маркерів за вхідними позиціями її переходів; функція  $\Phi: T \times (P \times Q) \rightarrow (P \times Q)$  визначає кольори маркерів вихідної позиції для переходу, що спрацював, і розподіляє ці кольори за вхідними позиціями переходів сітки;  $\Omega = \{\omega\}$  – непорожня скінченна множина кольорів маркування;  $\mu_0: P \times \Omega \rightarrow \{\emptyset, 1, 2, \dots\}$  – початкове маркування сітки.

Умовою спрацьовування переходу  $t$  СП при деякому її маркуванні  $\mu$  є така умова:

$$\left. \begin{aligned} &\forall p_{ik} \in \bullet t \exists \omega_{jk} \in \Omega : \mu(p_{ik}, \omega_{jk}) \geq I(p_{ik}, t); \\ &\lambda_t((p_{i1}, \omega_{j1}), \dots, (p_{in}, \omega_{jn})) = 1; k = \overline{1, n}; n = |\bullet t| \end{aligned} \right\}. \quad (9.13)$$

Тепер визначимо умови відповідності між ІВК як ОМ та деякою модифікацією СП. Нехай модель ІВК задано у вигляді пофарбованої СП  $N_\Phi$  (див. означення 9.14), стан якої визначається її маркуванням. Введемо функцію  $U$ , що описує послідовність спрацьовування переходів сітки. Тоді для побудови алгоритму керування ІВК необхідно сформувати модель

$$N_a = (N_D, U^*) \quad (9.14)$$

таку, що виконується умова  $\tau_k(U^*, \mu^*) = \min \tau(U, \mu)$ , причому

$$\begin{aligned} \mu_{i+1}(p, \omega) &= \mu_i(p, \omega) - \sigma_{t_j}(\omega) I(p, t_j) U_i(t_j, \omega) + \\ &+ g_{t_k}(\omega) O(t_k, p) U_i(t_k, \omega), \end{aligned}$$

де

$$g_{t_k} = \begin{cases} \emptyset, & \text{якщо } \Phi_{t_k}(\omega) = [\dots(p, \omega)\dots] \\ 1 & \text{у протилежному разі} \end{cases}$$

Тут  $\mu_i \in R(N_\Phi, \mu_0)$ ;  $U_i \in \{\emptyset, 1\}$  – множина функцій спрацьовування переходів сітки, причому значення  $U_i$  на  $i$ -му кроці алгоритму залежить від кольору збудження переходу  $t_k$ , за умови:

а) безпечності

$$\forall p \in P, \forall \omega \in \Omega, \forall \mu \in R(N_\Phi, \mu_0): \mu(p, \omega) \leq 1, \quad (9.15)$$

інтерпретується так: будь-який виконавець у фіксований момент часу виконує конкретну роботу і не може дістати нову роботу до завершення попередньої;

б) живучості

$$\begin{aligned} \forall \mu_i, \mu_j \in R(N_\Phi, \mu_0): \mu_i \xrightarrow{\sigma_{t_j}} \mu_j; \\ \forall t \in T \exists \mu_k, \mu_l \in R(N_\Phi, \mu_0): \mu_k \xrightarrow{\sigma_{t_j}} \mu_l. \end{aligned} \quad (9.16)$$

Цією умовою враховується те, що всі роботи з планових завдань будуть завжди виконані із залученням усіх виконавців;

в) збудження переходу  $t_j$

$$\forall p \in P \exists \omega \in \Omega: \mu_i(p, \omega) - I(p, \omega) \geq \emptyset; \sigma_{t_j}(\omega) = 1. \quad (9.17)$$

Умова передбачає, що для будь-якого виконавця завжди існує робота з планового завдання;

г) завершення спрацьовування переходу  $t_k$

$$\tau - \vartheta_{t_k} \omega \geq \theta_{t_k}(\omega), \quad (9.18)$$

де  $\vartheta_{t_k}(\omega)$  – початок цього спрацьовування за  $\omega$ ;  $\theta_{t_k}(\omega)$  – час затримки маркера  $\omega$  в переході  $t_k$ .

Ця умова підтверджує, що будь-яка робота з планового завдання завжди буде виконаною за час  $\theta_{t_k}(\omega)$ .

У (9.12) введено такі позначення:  $N_\Phi$  – СП, що визначає структуру алгоритму керування;  $\mu$  – стан ОК з множини станів  $R(N_\Phi, \mu_0)$ ;  $\Phi: T \rightarrow 2^\mu$  – функція позначення переходів сітки, яка кожному переходу ставить в однозначну відповідність підмножину станів ОК;  $2^\mu$  – множина всіх підмножин  $\mu$ ;  $P = \{p\}$  – непорожня скінченна множина виконавців (позицій);  $T = \{t\}$  – непорожня скінченна множина робіт КБ (переходів);  $\Omega = \{\omega\}$  – непорожня скінченна множина планових завдань (кольорів маркерів);  $I, O$  – функції інцидентності, що описують структурно-функціональний взаємозв'язок виконавців у КБ;  $\sigma$  – функція розподілу кольорів маркерів

за вхідними позиціями переходів, яка відображає послідовність виконання робіт згідно з плановим завданням;  $\mu_0$  — початкове маркування сітки, що відповідає початковому завантаженню виконавців у підрозділах КБ.

Отже, умови (9.15) і (9.16) визначають модель (9.14) як правильну, а задача синтезу алгоритму керування ІВК полягає у відшукуванні складових набору (9.14), які задовольняють умови (9.15)–(9.18).

**Визначення сіткової моделі ІВК.** Керування ІВК зводиться до необхідності управління переходами СП залежно від маркування її позицій. Отже, треба враховувати колір маркера при управлінні збудженням переходів сітки. Такий маркер називатимемо *маркером кольору спрацьовування* переходу.

Перевірку умов живучості та безпечності для довільного маркування СП завжди можна виконати за скінченну кількість кроків. Однак з експоненціальної залежності кількості можливих маркувань від кількості позицій сітки випливає, що при досить складній СП потужність множини  $R(N_\Phi)$  може бути дуже великою і це суттєво обмежує можливості безпосереднього аналізу сіток, потребуючи пошуку інших підходів до розглядуваної проблеми.

Одним з можливих її розв'язань є використання конструктивних прийомів, які дають змогу будувати заздалегідь правильні СП (в загальному випадку — сітки з будь-якими наперед заданими властивостями) із скільки завгодно великою кількістю елементів з деяких простих СП — *типових блоків*, правильність (або будь-які інші властивості) яких можна легко встановити [125; 139]. При інтерпретуванні процесів керування ІВК такими блоками вибираються [125]: блок К (відображає конвеєрність функціонування ІВК); блок П (відображає властивості паралельних процесів); блок У (умова альтернативності); блок Ц (визначає циклічність процесів в ІВК).

**Означення 9.15.** ІПСП є сітка  $N_{i\Phi} = \{N_i\}$ , що складається із скінченної множини коректно сформованих блоків К, П, У та Ц, де

$$N_i = (P, T, I, O, \Omega, \sigma, \varphi, \mu_0). \quad (9.19)$$

В алгоритмах керування ІВК переходи-дублери виконують такі функції: кожному дублеру відповідає підалгоритм, що сам є алгоритмом, опис якого задається блоком. Якщо в дубльованому блоці є свій перехід-дублер, то йому також відповідає свій підалгоритм, але ще нижчого рівня ієрархії, і так до найнижчого рівня, де перехід-дублер є оператором. Зв'язок між алгоритмами за рівнями ієрархії задається деревом відношень.

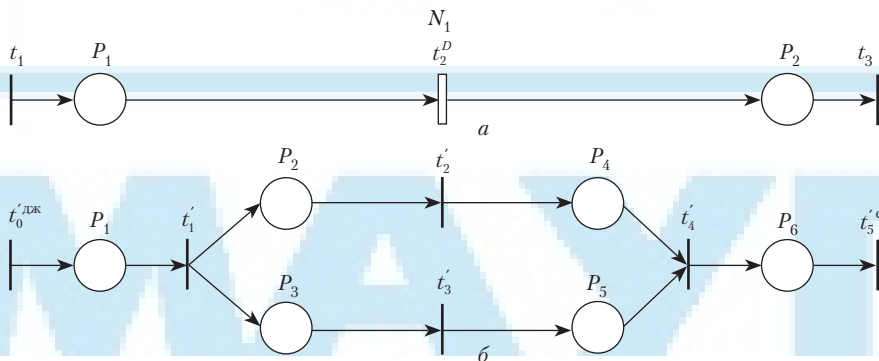


Рис. 9.15. Приклад старшої ІПСП з переходом-дублером (а) та блок-дублер сітки (б):  $t_0^{\text{дж}}$  — перехід-джерело;  $t_5^{\text{ст}}$  — перехід-стік



Як приклад розглянемо ІПСП верхнього рівня, що описує функції ІВК у фазі КПВ (рис. 9.16). Ця фаза КПВ складається з чотирьох етапів: технічної пропозиції; ескізного проекту; технічного проекту; розроблення робочої КД.

Переходи сітки відображатимуть обсяг робіт на кожному з етапів, а позиції — умови їх виконання. Збудження переходу сітки означає початок виконання робіт, спрацьовування — закінчення етапу.

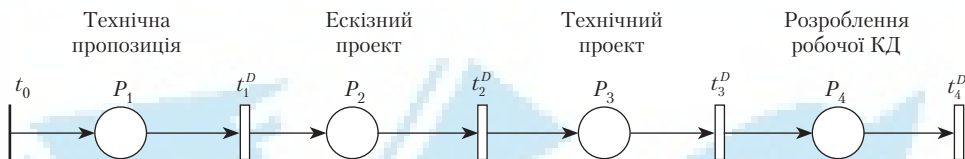


Рис. 9.16. Приклад ІПСП верхнього рівня, що описує функції ІВК у фазі КПВ

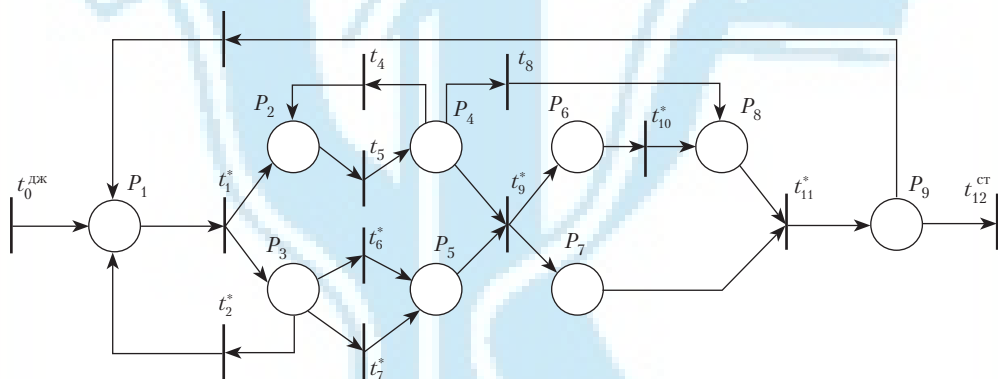


Рис. 9.17. Приклад ІПСП, що описує функціональну організацію окремого КВ в складі ІВК

Як підалгоритм, поданий кожним з переходів-дублерів  $t_i^D$ , можна розглянути роботу окремого КВ в складі ІВК (рис. 9.17). ІПСП, що описує цю роботу, складається з: позицій — множини виконавців КВ; переходів — множини робіт, які виконуються в КВ; вхідних і вихідних дуг в ІПСП, що відповідають організаційно-функціональному взаємозв'язку виконавців; початкового маркування сітки, що визначається конкретним завданням (переліком робіт) для КВ.

Умова живучості ІПСП, яка описує алгоритм керування ІВК, інтерпретується як обов'язкове виконання всіх планових завдань (збудження всіх переходів) за заданих початкових умов (початковим маркуванням сітки).

Умова безпечності в ІПСП, що описує алгоритм керування ІВК, інтерпретується як неможливість повторного ініціювання виконання будь-якої з робіт до повного її завершення (неможливість повторного спрацьовування збудженого переходу сітки).

**Означення 9.16.** Сітковою моделлю структури алгоритму організації керування ІВК є ІПСП  $N_k = \{N_{ij}\}$ , адекватна конструкції  $K = (B, C, TH, A)$ , така, що  $N_{ij}$  визначається набором (9.19) — сітковою моделлю підалгоритму, де:

$$P = \{p_i | i = \overline{1, m}, p_i = f_1(\alpha_i), \alpha_i \in A\};$$

$$T = \{t_j | j = \overline{1, n}, t_j = f_2(\beta_j), \beta_j \in B\};$$

$B = \{\beta_j | j = \overline{1, n}\}$  – множина функцій (робіт) КБ в складі ІВК;

$\Omega = \{\omega_r | r = \overline{1, l}, \omega_r = f_3(c_r), c_r \in C\}$ ;

$C = \{c_r | r = \overline{1, l}\}$  – множина робіт (замовлень) на проектування;

$f_{\text{тн}} : B \times B \rightarrow \{\emptyset, 1\}$  – взаємозв'язок пар проектних робіт;

$f_c : C \rightarrow B^*$  – відображення, що ставить у відповідність плановим замовленням підмножини планових робіт, причому  $B^*$  – множина всіх підмножин  $B$ ;

$f_B : B \rightarrow A$  – відображення, що визначає підмножину можливих виконавців робіт;

$A = \{\alpha_i | \alpha_i = f_B(\beta_j), \beta_j \in B, i = \overline{1, W}\}$  – множина виконавців КБ ( $W$  – потужність цієї множини);

$TH = \left\{ \prod_{j=1}^r f_1(c_r) | c_r \in C, r = \overline{1, L} \right\}$  – послідовність виконання проектних робіт для планового замовлення  $r$  (проектна технологія);

$\sigma = \{\sigma_t | t \in T, \sigma_t = f_2(TH)\}$ ;  $\varphi = \{\varphi_t | t \in T, \varphi_t = f_3(TH)\}$ .

Тепер дамо означення *коду функціональної відповідності* (КФВ) структурних підрозділів/елементів ІВК.

**Означення 9.17.** *КФВ структурного елементу ІВК (підрозділу)* є вираз вигляду  $\langle P^*, B^*, f \rangle$ , де  $P^* \subset P$  – підмножина функціональних елементів підрозділу;  $B^* \subseteq B$  – підмножина функцій, що виконуються підрозділом;  $f : P^* \rightarrow B^*$  – функція відповідності кваліфікації виконавця складності виконуваної проектної роботи.

Таким чином, КФВ описує набір функцій підрозділу із загального переліку робіт ІВК, а через цей набір – і функцію  $f_{\text{тн}}$ , що визначає зв'язки функціональної відповідності елементів підрозділу ІВК, які подаються фрагментом ІПСП. Зазначимо, що, задаючи КФВ підрозділу, легко побудувати сіткове подання останнього, а побудувавши матрицю інцидентності пар проектних робіт – визначити маркування цієї ІПСП.

**Модель функціонування конструкторських підрозділів інженерно-виробничого комплексу.**

**Синтез алгоритмів оперативного керування ІВК.** Процедура синтезу алгоритму оперативного керування ІВК у фазі КПВ починається з побудови ІПСП, що описують структурно-функціональну взаємодію виконавців у конструкторських підрозділах (наприклад, таку, яку зображено на рис. 9.17). При цьому використовуються інформація про склад цих підрозділів з точки зору їх «працевдатності» і перелік проектних робіт, які виконуються кожним з підрозділів. Питання про оцінювання «працевдатності» виконавця розглядається в наступному підрозділі. Тут тільки зауважимо, що із загального переліку конструкторських робіт ІВК вибираються ті, які відповідають плановим завданням для підрозділів, а також проводиться процедура спрощення ІПСП, яка відображає повний структурно-функціональний взаємозв'язок виконавців.

Початкове маркування в такій спрощеній сітці відповідає початковому (поточному) завантаженню виконавців. У разі необхідності зміни сіткового подання функціонування ІВК (зміни КФВ підрозділів, зміни планових завдань тощо) починає працювати верхня частина моделі – її інтелектуальна надбудова (рис. 9.18).

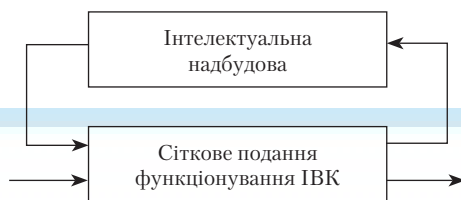


Рис. 9.18. Модель функціонування ІВК

Зміна КФВ підрозділів може спричинюватися рядом зовнішніх дій: хворобою окремого виконавця, скороченням штатів, доповненням плану новим завданням, зміною строків виконання завдань тощо. В таких випадках інтелектуальна надбудова, згідно з деревом відношень, здійснює вибір тих сіткових блоків, які відображають якісні ознаки необхідних сіткових перетворень. Далі береться найнижчий з вибраних сіткових блоків і здійснюється його модифікування: вилучення (додавання) позиції, вилучення (додавання) переходу.

Після модифікування сіткового блоку здійснюється аналіз його властивостей на коректність і виконання, і якщо він задовольняє необхідні для цього вимоги, аналіз їх виконання для блоків, що розташовані вище по дереву відношень, проводити не треба. Проте якщо зміна структури блоку змінює умови спрацьовування переходу-стоку та переходу-джерела, то необхідно повторити аналіз властивостей для суміжного блоку верхнього рівня. Це пояснюється тим, що в разі спрощення КФВ деякі роботи, які надходять для виконання в підрозділи, вже не можна буде виконати.

Такий аналіз слід проводити послідовно, рухаючись по дереву відношень вгору від блоку доти, поки зміни в нижньому сітковому блоці вже не впливатимуть на умови спрацьовування переходу-стоку та переходу-джерела. Якщо ж сітковий блок на будь-якій ітерації аналізу виявиться некоректним, то потрібно змінити його пофарбування. При переміщенні по дереву відношень застосовується операція підстановки сіткового блоку замість блоку-дублера верхньої сітки (див. означення 9.18).

**Означення 9.18.** Підстановкою сіткового блоку  $N_{bl}$  замість переходу-дублера  $t_i^D$  ІПСП  $N$  є операція, що породжує сітку  $\bar{N}(\bar{P}, \bar{T}, \bar{I}, \bar{O}, \bar{\Omega}, \bar{\sigma}, \bar{\phi}, \bar{\mu})$ .

Інакше кажучи, операція підстановки блоку замість переходу полягає у вилученні з ІПСП  $N$  переходу  $t_i^D$  і заміні його сітковим блоком  $N_{bl}$  так, щоб всі вхідні позиції переходу  $t_i^D$  стали вхідними позиціями кожного з переходів-джерел  $t_{bl}^{дж}$ , а всі вихідні позиції переходу  $t_i^D$  — вихідними позиціями кожного з переходів-стоків  $t_{bl}^{ст}$ . Маркування всіх позицій ІПСП  $N$  і сіткового блоку  $N_{bl}$  залишаються без змін. При цьому сітковий блок буде правильним, якщо його підстановка замість переходу-дублера  $t_i^D$  ІПСП  $N$  породжує правильну сітку.

Використання інтелектуальної надбудови може зумовлюватися й іншими факторами, зокрема необхідністю одержання інформації про поточний стан виконання плану, про можливість реалізації додаткових завдань у директивні строки тощо. Тоді відпадає потреба в аналізі властивостей сіткової моделі ІВК, але виконується відповідне часове навантаження переходів сіткових блоків нижчого рівня дерева входження і здійснюється динамічна трансляція всієї сіткової моделі ІВК.

Для реалізації функцій інтелектуальної надбудови можна організувати автоматизоване робоче місце адміністратора підрозділу.

**Часова інтерпретація ІПСП.** У загальному випадку перехід в ІПСП є підсіткою нижчого рівня, яка, у свою чергу, складається з більш простих підалгоритмів.

На найнижчому рівні сітка відображає, наприклад, технологічну послідовність роботи окремого КБ протягом планового періоду. При цьому кожен перехід у такій сітці відповідає виконанню окремої технологічної операції або проектної роботи.

Кожна така операція (робота) має свою тривалість і залежить від

$$t_i = f(KS, GR, G, H, C), \quad (9.20)$$

де  $KS$  — коефіцієнт складності виробу, що проектується;  $GR$  — група новизни цього виробу;  $G$  — коефіцієнт, яким враховують кваліфікацію виконавця КБ;  $H$  — досвід його роботи (в роках);  $C$  — спеціалізація виконавця, що визначає можливість його використання при виконанні  $i$ -ї роботи.

Значення функції (9.20) визначається за статистичними даними її параметрів або експертним шляхом.

Кожний перехід нижчого рівня управління сіткової моделі не є переходом-дублером і характеризується часом спрацювання  $\tau_i$ , що відповідає тривалості технологічної операції (проектної роботи). Для переходів вищого рівня ієрархії (тобто переходів-дублерів) часова функція обчислюється шляхом послідовних покрокових перетворень на кожному з етапів збільшення сітки (заміною блоків дублерами).

Такі перетворення стосовно типових блоків К, П, У та Ц, які відповідно відображують властивості конвеєрності, паралельності, умовності (альтернативності) та циклічності описуваних процесів (рис. 9.19), зводяться до такого:

1. **Блок К** (рис. 9.19, а). Нехай перехід  $t_i^*$  має час спрацювання  $\tau_1 = \tau(p_1, \omega_j)$ ,  $\omega_j \in C_{t_i^*}$ , де  $p_1$  — вхідна позиція цього переходу, визначає умови його спрацювання. Тоді час спрацювання блоку визначається виразом

$$\tau_{bl} = \tau_0 + \tau_1, \quad (9.21)$$

де  $\tau_0$  — момент надходження маркера до переходу  $t_0^{дж}$ .

Тут йдеться про мінімально необхідний час спрацювання блоку, але для подання реальної динаміки його роботи треба враховувати часову затримку маркерів у позиціях блоку.

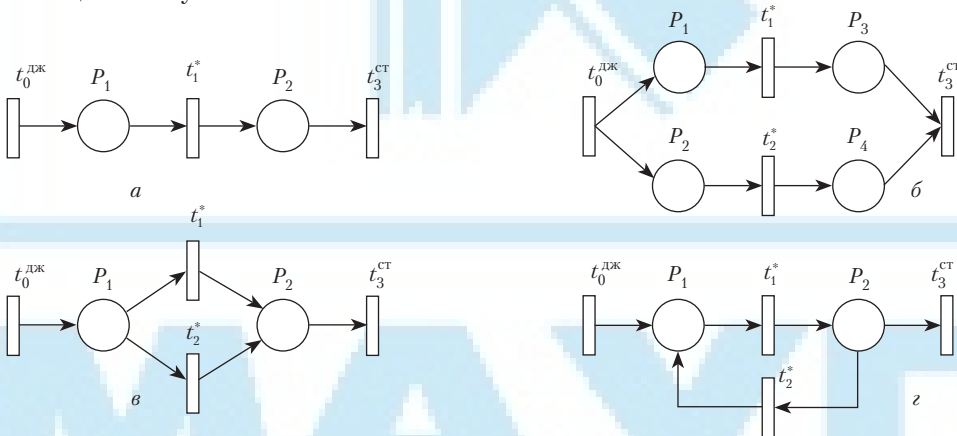


Рис. 9.19. Типові блоки, що застосовуються для побудови сіткової моделі ІВК

2. **Блок П** (рис. 9.19, б). Нехай переходи  $t_1^*$ ,  $t_2^*$  мають часи спрацювання  $\tau_1$  і  $\tau_2$  відповідно. При спрацюванні переходу  $t_0^{дж}$  складові роботи, що інтерпретується вхідним маркером, одночасно надходять до позицій  $p_1$  і  $p_2$ . Нехай  $\tau_0$  — момент

надходження маркера до переходу  $t_0^{дж}$ , тоді моменти завершення робіт (спрацьовування переходів  $t_1^*$ ,  $t_2^*$ ) дорівнюють відповідно  $\tau_0 + \tau_1$  і  $\tau_0 + \tau_2$ . Отже, загальний час спрацьовування блоку визначається як

$$t_{bl} = \max_{t_i \in N_{bl}} (\tau_i), \tau_i = \tau_i(i, \omega_j), \omega_j \in C_{t_i}^* \quad (9.22)$$

3. **Блок У** (рис. 9.19, е). Нехай часи спрацьовування переходів  $t_0^{дж}$ ,  $t_1^*$ ,  $t_2^*$  відповідають значенням  $\tau_0, \tau_1, \tau_2$ . Тоді загальний час спрацьовування блоку становитиме

$$\tau_{bl} = \tau^*(Y) + \tau_0, \quad (9.23)$$

де  $\tau^* = \tau_1$  (або  $\tau_2$ ) залежно від стратегії керування  $Y$  (практично це – вибір кваліфікації  $G$  виконавця відповідної роботи).

4. **Блок Ц** (рис. 9.19, з). Нехай переходи  $t_0^{дж}$ ,  $t_1^*$ ,  $t_2^*$  мають часи спрацьовування  $\tau_0, \tau_1, \tau_2$  відповідно. Тоді загальний час спрацьовування блоку визначається як

$$\tau_{bl} = \tau_0 + \tau_1 + \sum_{i=1}^n (\tau_i + \tau_2), \quad (9.24)$$

де  $n$  – кількість циклів, яка залежить від стратегії керування  $Y$ . Тут слід зауважити, що при надходженні вказівки від інтелектуальної надбудови про проведення нової динамічної трансляції сіткової моделі функціонування ІВК з метою заощадження часу доцільно обчислити час спрацьовування тільки тих сіткових блоків, які зазнали структурних або часових змін, а також блоків, розташованих на шляху до верхньої ІПСП по дереву відношень.

**Приклад 9.7.** Для розрахунку часової затримки в ІПСП верхнього рівня розглянемо ІПСП верхнього рівня (рис. 9.20, а), що описує підготовчу стадію ІВК у фазі КПВ. Часи спрацьовування переходів  $t_0^{дж}, t_1^*, t_2^*, t_3^D, t_4^D$  позначимо через  $\tau_0^{дж}, \tau_1^*, \tau_2^*, \tau_3^D, \tau_4^D$  відповідно. Тоді тривалість фази КПВ (часова затримка в сітці) становитиме

$$\tau_{КПВ} = \tau_0 + \tau_1^D + \tau_2^D + \tau_3^D + \tau_4^D. \quad (9.25)$$

Тепер розрахуємо часи спрацьовування  $\tau_1^D, \tau_2^D, \tau_3^D, \tau_4^D$  кожного з переходів-дублерів, припустивши, що КБ виконує ескізний проект виробу. Побудуємо КФВ підрозділу виходячи з його штатного розпису:  $p_1$  – провідний інженер-конструктор;  $p_2, p_2$  – інженери-конструктори першої категорії;  $p_3, p_3$  – інженери-конструктори другої категорії;  $p_4, p_4$  – інженери-конструктори третьої категорії;  $p_5$  – технік.

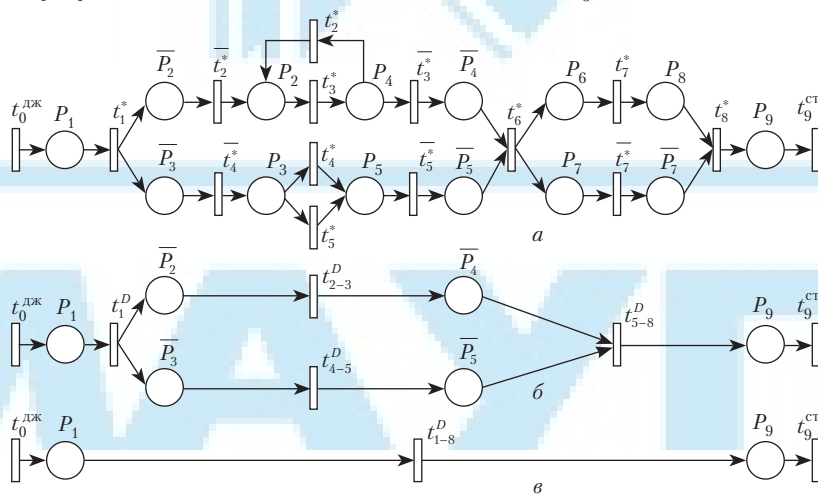


Рис. 9.20. ІПСП алгоритму керування КБ (а) та її покрокове спрощення (б, в)

Тоді КФВ цього КБ матиме вигляд  $p_1 2p_2 2p_3 2p_4 2p_5$ . При цьому кожен з виконавців може здійснювати суворо визначений посадовою інструкцією перелік проектних робіт, який встановлюється табл. 9.3 (вона стосується етапу ескізного проектування виробу).

За цим КФВ побудуємо ПСП, що описує функціональну організацію КБ в складі ІВК. Ця сітка охоплює всі функціональні зв'язки між виконавцями КБ. На плановий період для виконання певного обмеженого переліку робіт будуть задіяні не всі зв'язки і ПСП буде відповідно відкоригована, наприклад, так, як це показано на рис. 9.20, *a* (в сітку введено додаткову фіктивну позицію  $p_9$ , яка відображає готовність комплекту КД).

Колір дуг, які входять у позиції, визначає складність виконуваної роботи, переходи сітки відображають виконання окремим виконавцем певної планової роботи.

Розрахуємо час спрацьовування блоку. Для цього виділимо типові блоки на рис. 9.20, *a* і виконаємо покрокове спрощення сітки (рис. 9.20, *b*). З цією метою введемо кілька фіктивних позицій, а також переходів  $\bar{p}_2, \bar{p}_3, \bar{p}_4, \bar{p}_5, \bar{p}_7$  і  $\bar{t}_2^*, \bar{t}_3^*, \bar{t}_4^*, \bar{t}_5^*, \bar{t}_7^*$ . Вважаючи, що тривалість введених переходів дорівнює нулю, на час спрацьовування всього блоку така корекція сітки не вплине.

Переходи  $\bar{t}_2^*, \bar{t}_3^*, \bar{t}_4^*, \bar{t}_5^*$  утворюють блок Ц:

$$\tau_{2-3}^* = \bar{\tau}_2 + \bar{\tau}_3 + \tau_3 + k_{\text{ц}}^* (\tau_2 + \tau_3) = \tau_3 + k_{\text{ц}}^* (\tau_2 + \tau_3),$$

де  $k_{\text{ц}}^*$  — коефіцієнт циклу для цього блоку.

Переходи  $\bar{t}_4^*, \bar{t}_5^*, \bar{t}_7^*$  утворюють блок У:

$$\tau_{4-5}^* = \bar{\tau}_4 + \bar{\tau}_5 + \tau^*(Y) = \tau_4,$$

де значення  $\tau^*(Y)$  визначається стратегією керування  $Y$ .

Переходи  $\bar{t}_6^*, \bar{t}_7^*, \bar{t}_8^*$  утворюють блок П з часовою затримкою

$$\tau_{6-8}^* = \bar{\tau}_6 + \bar{\tau}_8 + \max(\tau_7, \bar{\tau}_7) = \tau_6 + \tau_7 + \tau_8.$$

Переходи  $\bar{t}_1^*, \bar{t}_{2-3}^*, \bar{t}_{4-5}^*, \bar{t}_{6-8}^*$  також утворюють блок П (рис. 9.20, *в*):

$$\tau_{1-8}^* = \tau_1 + \tau_{6-8}^* + \max(\tau_{2-3}^*, \tau_{4-5}^*).$$

Отже, час спрацьовування переходу-дублера  $\tau_1^D$  визначається як

$$\tau_1^D = \tau_0 + \tau_{1-8}^*, \tag{9.26}$$

де  $\tau_0$  — момент надходження маркера до переходу  $t_{1-8}$  сітки.

Таблиця 9.3

**Приблизна перелік робіт умовного конструкторського підрозділу ІВ**

Шифр	Перелік робіт	Перелік посад, що залучаються до виконання роботи*						
		6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9
2.01	Виконання варіантів можливих рішень, їх конструкторське пророблення	—	1,0	1,3	—	—	—	—
2.02	Попереднє вирішення питань пакування і транспортування	—	—	1,0	1,3	—	—	—
2.03	Розроблення й обґрунтування технічних рішень, спрямованих на забезпечення надійності	—	—	1,0	1,2	—	—	—
2.04	Оцінювання виробу на технологічність і правильність вибору засобів і методів контролю	—	—	1,0	1,2	—	—	—

\* Тут 6 — провідний інженер-конструктор; 7 — інженер-конструктор першої категорії; 8 — інженер-конструктор другої категорії; 9 — інженер-конструктор третьої категорії; 10 — конструктор першої категорії; 11 — конструктор другої категорії; 12 — конструктор третьої категорії.

1	2	3	4	5	6	7	8	9
2.05	Оцінювання виробу за показниками стандартизації та уніфікації	—	—	—	1,0	1,4	—	—
2.06	Оцінювання виробу щодо вимог ергономіки та технічної естетики	—	1,0	1,1	1,2	—	—	—
2.07	Перевірка варіантів на патентну чистоту і конкурентоспроможність	—	—	—	—	1,0	1,6	—
2.08	Перевірка відповідності варіантів вимогам техніки безпеки	—	—	—	1,0	1,4	—	—
2.09	Вибір оптимального варіанта	1,0	1,1	1,4	—	—	—	—
2.10	Виявлення на основі прийнятих рішень нових виробів і матеріалів, які повинні бути розроблені іншими підприємствами	—	—	—	1,0	1,4	—	—
2.11	Складання відомості ескізного проекту	—	—	—	—	—	1,0	—
2.12	Складання пояснювальної записки до ескізного проекту	1,0	1,1	1,3	1,6	—	—	—
2.13	Підготовка пропозицій для розроблення стандартів	—	—	—	1,0	1,4	—	—

Аналогічно розраховуються значення  $\tau_2^D, \tau_3^D, \tau_4^D$  з урахуванням конкретної функціональної структури КБ і планового переліку робіт.

Підставляючи в (9.25) знайдені значення часів спрацьовування окремих переходів сітки, дістаємо значення тривалості фази КПП.

**Експертний аналіз при диспетчеризуванні проектних завдань.** Застосування методики *експертного оцінювання* (ЕкО) в задачах керування ІВК на підготовчій стадії, зокрема у фазі КПП, спричинюється виникненням конфліктних ситуацій, що потребують для свого розв'язання опитування думок фахівців. Так, при розподілі та диспетчеризуванні робіт усередині КБ можуть виникнути дві задачі: два чи кілька виконавців претендують на виконання одного проектного завдання; вибір виконавцем проектного завдання для виконання з двох або кількох завдань. Такі задачі ЕкО можуть виникати і на виробничій стадії, де ресурсом виступає технологічне обладнання, а завданням — технологічна операція. Ці задачі слід зарахувати до типу *інтерпретування*.

Для розв'язання таких задач залучаються експерти (фахівці) — конструктори, технологи тощо. Роботою групи експертів керує фахівець у галузі організації та проведення експертизи, який інтерв'ює експертів, обробляє результати, спрямовує процедуру опитування. Оцінювання проводиться за допомогою збирання статистики (опитування експертів), а також використанням вже готової БЗ у галузі. Залежно від кількості експертів, стратегії обробки результатів визначається процедура опитування та формування оцінки.

Найбільш простим і зручним методом ЕкО названих вище задач керування ІВК у фазі КПП є *метод «Делфі»*. Основними етапами експертного аналізу конфліктних ситуацій, що виникають, є:

- *експертне оцінювання професійних якостей виконавця.* Критеріями оцінювань є кваліфікація, спеціалізація, «надійність», досвід роботи виконавця тощо, і професійні його якості визначаються вектором

$$\mathbf{Val}_{\text{вик}}(i) = \{k_{ni} \mid k_{ni} \in K_{\text{вик}}, i \in W\}, \quad (9.27)$$

де  $i$  — номер виконавця;  $n$  — номер критерію ЕкО;  $K_{\text{вик}}$  — множина коефіцієнтів виконавця за критеріями ЕкО;

- *експертне оцінювання характеристик виробу з боку технології проектування* (під терміном «виріб» розуміють сукупність проектних робіт, які виконуються у фазі КПП стосовно цього виробу). Результат ЕкО також можна подати у вигляді вектора

$$\text{Val}_{\text{вир}}(j) = \{g_{mj} | g_{mj} \in G, j \in J\}, \quad (9.28)$$

де  $j$  – децимальний номер виробу;  $g_{mj}$  – коефіцієнт виробу за відповідним критерієм ЕкО;

- експертний аналіз відповідності «виріб – виконавець» з використанням результатів перших двох етапів ЕкО (тут слід наголосити, що останні не обов'язково виконуються в реальному часі, оскільки можуть використовуватись вже накопичені статистичні дані). Зміст третього етапу полягає у формуванні бінарного відношення на множинах  $f$  – виконавець і  $V$  – продукція, яке визначає ступінь «придатності» виконавця до виконання проектної роботи стосовно певного виробу.

Розглянемо приклади таких бінарних відношень:

$$f_1 = \sum_{i,j} f_0(k_{ij}, g_{mj}) \text{Val}_m(j) \text{Val}_l(i), \quad (9.29)$$

$$f_2 = \sum_{i,j} f_0(k_{ij}, g_{mj}) (\text{Val}_m(j) + \text{Val}_l(i)), \quad (9.30)$$

де  $f: K_{\text{вик}} \times G_{\text{вир}} \rightarrow [0,1]$  – функція взаємозв'язку критеріїв оцінювання виконавців і виробів. Одним з варіантів функції  $f_0$  може бути така матриця:

$$f_0 = \begin{bmatrix} 0.1 & 0.3 & 0.4 & 0.2 \\ 0.4 & 0.2 & 0.1 & 0.3 \\ 0.3 & 0.3 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.3 & 0.3 \end{bmatrix}$$

Значення елементів цієї матриці знайдемо так. На першому етапі для одержання часових параметрів стосовно виконання робіт за базові значення характеристик виконавців і виробів було взято значення, запропоновані після проведення ЕкО і наведені в табл. 9.4 і 9.5.

Хронометраж дає змогу визначити тривалості виконання проектних операцій для різних поєднань «виконавець – виріб». Результат кожного випробування відповідає деякій сукупності факторів, поданих таким поєднанням. Одержана множина точок відклику досліджується на вплив попарних узгоджень значень характеристик виконавців і робіт за часовими показниками.

Таблиця 9.4

Результати ЕкО професійних якостей конструкторів

№ пор.	Склад КБ	Критерії ЕкО			
		Кваліфікація	Спеціалізація	Досвід роботи	Показник надійності
1	Стасюк В. А.	5	5	6	8
2	Лавринюк О. С.	3	4	3	5
3	Петренко Л. В.	4	2	4	3
4	Герашенко Н. Ю.	5	9	9	5
5	Половников Г. Ф.	4	5	2	3
6	Шаванова Е. А.	3	3	2	1
7	Миколайчук М. В.	4	5	5	4
8	Ординець Т. Д.	8	4	6	4
9	Самофаленко Р. І.	2	3	2	4
10	Пічкуренко О. П.	5	3	4	5
11	Антонюк С. Л.	1	1	1	3



Далі для призначення вагових коефіцієнтів визначається ступінь впливу кожної характеристики виконавця на різні характеристики виробу. Для цього при кожному фіксованому значенні параметра виконавця встановлюється єдине значення (у відносних одиницях) для всіх параметрів виробу і розраховуються відносні ваги для кожного з них відповідно до значень часу. Тоді відносні значення параметрів виробів «пробігають» весь заданий діапазон.

Сукупність одержаних даних утворює залежність типу «час – кваліфікація/складність». Як результат дослідження цих залежностей формуються рекомендації для варіанта, якому віддається перевага щодо розв'язання конфліктних ситуацій, які виникають під час реалізації алгоритму.

Користуючись формулою (9.30), з урахуванням значень  $f_0$ ,  $Val_{\text{вик}}$  і  $Val_{\text{вир}}$  для перших виконавця та виробу (див. перші рядки табл. 9.4, 9.5), знаходимо

$$\begin{aligned} f_{2,11} = & 0,1 \cdot (5+4) + 0,3 \cdot (5+3) + 0,4 \cdot (5+2) + 0,2 \cdot (5+5) + \\ & + 0,4 \cdot (5+4) + 0,2 \cdot (5+3) + 0,1 \cdot (5+2) + 0,3 \cdot (5+5) + \\ & + 0,3 \cdot (6+4) + 0,3 \cdot (6+3) + 0,2 \cdot (6+2) + 0,2 \cdot (6+5) + \\ & + 0,2 \cdot (8+4) + 0,2 \cdot (8+3) + 0,3 \cdot (8+2) + 0,3 \cdot (8+5). \end{aligned}$$

Повний розрахунок відношення  $f_{2,11}$  за спеціальною програмою дає результати, які фрагментарно наведено в табл. 9.6. Ці результати і є підставою для розв'язання конфліктних ситуацій.

Зупинимося докладніше на типах конфліктних ситуацій, алгоритмах їх розв'язання та правилах використання результатів ЕкО відповідності професійних якостей виконавця складності виробу, який проектується.

Таблиця 9.5

Результати ЕкО технології проектування виробу

№ пор.	Найменування виробу	Критерії ЕкО			
		Коефіцієнт складності	Група новизни	Терміновість замовлення	Відповідальність
1	РТС установлення мікросхем на друковані плати	4	3	2	5
2	Координатний стіл	1	1	4	1
3	Стенд для перевірки друкованих плат	3	3	1	5
4	Транспортний модуль	3	2	4	3
5	Блок живлення	2	1	5	2

Нехай треба визначити виконавця проектної роботи, якщо є кілька виконавців з резервом часу, достатнім для її виконання. Тоді згідно з табл. 9.3 вибираються посади виконавців, які мають ненульовий коефіцієнт  $K_{\text{вик}}$ . Кожна з виконуваних робіт має свій унікальний код  $C_{ij}$ , тобто тільки властивий для неї колір маркування. Значення індексу  $i$  відповідає номеру етапу КПВ, а індексу  $j$  – порядковому номеру операції. Якщо напроти найменування роботи/операції в графі відповідної посади стоїть ненульовий коефіцієнт  $K_{\text{вик}}$ , це означає, що цей виконавець може виконати цю роботу/проектну операцію в  $K_{\text{вик}}$  разів повільніше, ніж виконавець з  $K_{\text{вик}} = 1$ .

Таблиця 9.6

Результати ЕкО відповідності «виріб – конструктор»

№ пор.	Склад КБ	Вироби, що проектуються			
		1	2	3	4
1	Стасюк В. А.	38,0	31,0	36,0	34,0
2	Лавринюк О. С.	29,0	22,0	27,0	25,0
3	...	...	...	...	...

Отже, наявність ненульового значення  $K_{\text{вик}}$  відповідає наявності дуги до відповідної операції – переходу з позиції, що відображає виконавця з  $K_{\text{вик}}$ . Якщо після відбирання за ненульовим коефіцієнтом  $K_{\text{вик}}$  і резервом часу на роботу претендують більше одного виконавця, слід скористатися табл. 9.6. Перевага віддається виконавцю з більшим значенням коефіцієнта ЕкО.

За наявності вільного виконавця і кількох робіт для виконання згідно з табл. 9.3 вибираються роботи, що відповідають кваліфікації виконавця. Потім на основі таблиці пріоритетів визначається послідовність проектування виробу. Проектні роботи, що залишились, також утворюють чергу на виконання і чекають звільнення виконавця необхідної кваліфікації.

**Методика моделювання функцій ІВК та проектування алгоритму керування.** Основними етапами цієї методики, відповідно до алгоритму рис. 9.21, є такі:

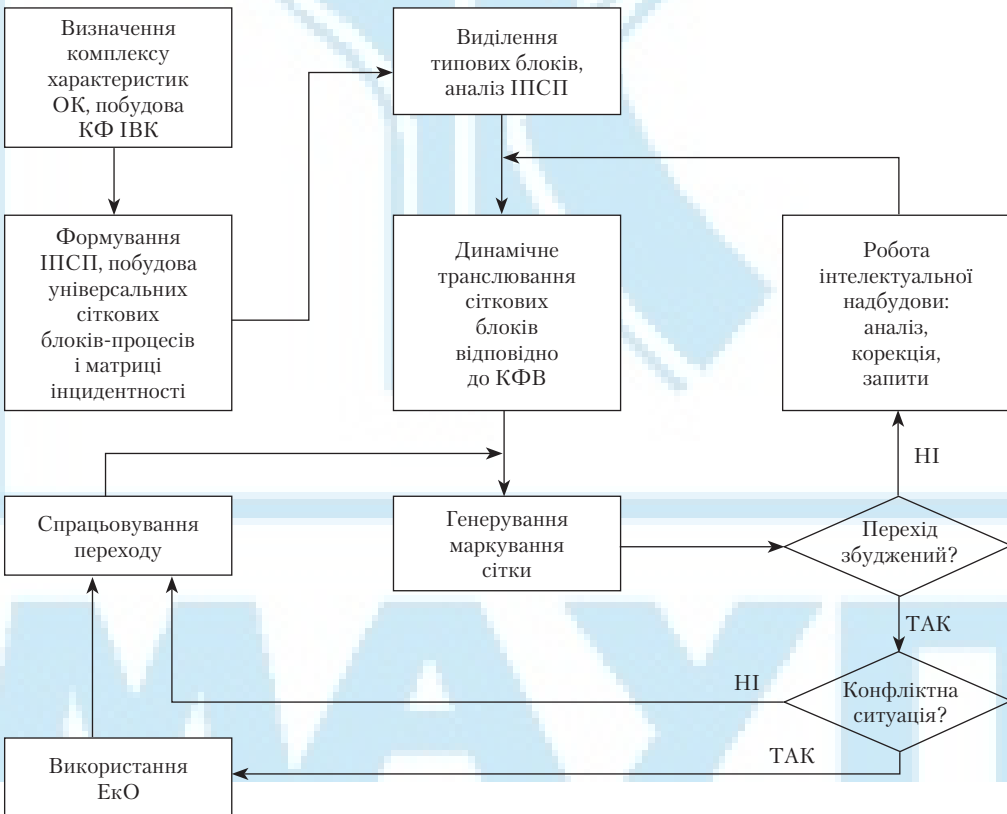


Рис. 9.21. Послідовність етапів методики синтезу функціонування ІВК

1. Визначення комплексу характеристик ОК (ІВК) у фазі КПВ: облікового складу КБ, переліку проектних замовлень, набору проектних робіт і комплектів КД, часу виконання останніх для кожного із замовлень.

2. Формування сіткової моделі функціонування ІВК у фазі КПВ. Інтерпретування характеристик ОК в термінах ІПСП та створення універсальних сіткових блоків-процесів, що описують динаміку виконання проектних замовлень на рівні конструкторських підрозділів. Позиції цих блоків описують перелік виконавців, здатних виконати проектну роботу, яка зображається переходом на виході окремої позиції.

3. Здійснення операції вибору конкретного виконавця для виконання кожної з планових робіт, формування варіантів сіткових блоків нижчого рівня шляхом варіювання цього вибору з урахуванням КФВ підрозділів.

4. Аналіз алгоритму керування за сітковою структурою моделі, що відображає повні зв'язки підрозділів: виділення типових блоків, аналіз належності старшої компоненти ІПСП системі типових блоків. Повторний аналіз алгоритму керування необхідно проводити тільки при зміні КФВ, а також тоді, коли сітковий блок неможливо подати через систему типових блоків.

5. Синтез алгоритму керування ІВК у фазі КПВ: виконання процедури покрокового спрощення ІПСП шляхом вилучення типових блоків, що містять переходи, для яких не виконується умова збудження; динамічна трансляція сіткової моделі шляхом часового навантаження переходів; генерація та занесення в БД множини послідовностей спрацьовування переходів (проектних технологічних послідовностей), які задовольняють умову функціонування ІВК – розроблення виробу у встановлені строки. В разі виникнення конфліктних ситуацій при генерації маркувань блоків у структурі моделі – проведення експертного аналізу для встановлення відповідності «виріб – конструктор» (див. вище) або ж використання готової статистики (див. табл. 9.6) для їх розв'язання згідно з алгоритмом, схему якого зображено на рис. 9.21.

6. Ініціювання роботи модуля інтелектуальної надбудови для формування нових вхідних даних до переформування сіткової моделі або регенерації послідовностей маркувань у разі виникнення зовнішнього впливу на функціонування елементів ІВК. В цьому модулі формуються також різного роду інформаційні запити, наприклад, визначення резерву часу для підрозділів ІВК.

**Приклад 9.8.** На основі наведеної методики розглянемо приклад побудови сіткової моделі та організації керування умовним підрозділом ІВК.

1. Визначаємо КФВ підрозділу

$$\text{КФВ} = p_1 p_2 p_3 4 p_4 2 p_5 11 p_6 8 p_7 12 p_8 25 p_9 13 p_{10} 10 p_{11} 14 p_{12} 2 p_{14} p_{15} \times \\ \times p_{16} 6 p_{19} p_{20} 3 p_{21} 4 p_{22} p_{23} 2 p_{24} 2 p_{25} p_{26}.$$

Кодування позицій  $p_i$  здійснюємо відповідно до табл. 9.7.

Таблиця 9.7

### Класифікація множин виконавців ІВК

Позиція	Посада	Кількість посадових одиниць
1	2	3
$p_1$	Начальник ІВК	1
$p_2$	Заступник з виробництва	1
$p_3$	Заступник з техніки	1
$p_4$	Начальник відділу	4
$p_5$	Заступник начальника відділу	2

Закінчення табл. 9.7

1	2	3
$p_6$	Провідний інженер-конструктор	11
$p_7$	Інженер-конструктор першої категорії	8
$p_8$	Інженер-конструктор другої категорії	12
$p_9$	Інженер-конструктор третьої категорії	25
$p_{10}$	Конструктор першої категорії	13
$p_{11}$	Конструктор другої категорії	10
$p_{12}$	Конструктор третьої категорії	14
$p_{13}$	Технік-конструктор першої категорії	—
$p_{14}$	Технік-конструктор другої категорії	2
$p_{15}$	Технік-конструктор третьої категорії	1
$p_{16}$	Технік першої категорії	1
$p_{17}$	Технік другої категорії	—
$p_{18}$	Технік третьої категорії	—
$p_{19}$	Кресляр	6
$p_{20}$	Інженер-програміст першої категорії	1
$p_{21}$	Інженер-програміст другої категорії	3
$p_{22}$	Інженер-програміст третьої категорії	4
$p_{23}$	Старший технолог	1
$p_{24}$	Технолог першої категорії	2
$p_{25}$	Технолог другої категорії	2
$p_{26}$	Технолог третьої категорії	1

Вибірку проектних робіт на плановий період і тривалість їх виконання наведено в табл. 9.8. Як приклад вибрано роботи щодо проектування транспортного засобу — ЕРК. Згідно зі спеціалізацією відділів ІВК це замовлення виконується групою конструкторів транспортно-нагромаджувальних засобів відділу, що займається розробленням механічного устаткування й має  $KФВ = p_6 p_7^2 p_8 p_9$ .

Матрицю інцидентності планових робіт подано у вигляді табл. 9.8, де наведено перелік робіт, які стосуються етапу ескізного проектування виробу.

Таблиця 9.8

Перелік проектних робіт на плановий період

№ пор.	Найменування роботи	Кількість нормо-годин	$p_6$	$p_7$	$p_8$	$p_9$
			4	5	6	7
1	2	3	4	5	6	7
1	Виконання варіантів можливих рішень, їх конструкторське пророблення	467	—	1,0	1,3	—
2	Перевірка варіантів на патентну чистоту	122	—	—	—	1,0
3	Вибір оптимального варіанта	55	1,0	1,1	1,4	—
4	Складання відомості ескізного проекту	36	—	—	1,0	1,1
5	Складання пояснювальної записки	101	—	—	—	1,0
6	Розроблення конструкторських рішень	180	1,0	1,2	1,4	—
7	Виконання необхідних розрахунків для підтвердження техніко-економічних показників	120	—	—	1,0	1,2
8	Аналіз конструкції на технологічність	98	1,0	1,2	1,5	—
9	Розроблення, виготовлення та випробування макета	150	—	1,0	1,3	1,5
10	Узгодження габаритних, установочних і приєднувальних розмірів із замовником	20	—	1,0	1,1	—
11	Розроблення креслень складальних одиниць і деталей	300	1,0	1,1	1,3	1,4
12	Складання відомостей технічного проекту	55	—	1,0	1,1	—
13	Складання пояснювальної записки до технічного проекту	101	1,0	1,1	1,3	1,4
14	Розроблення креслень загального вигляду	300	1,0	1,1	1,4	—

Закінчення табл. 9.8

1	2	3	4	5	6	7
15	Розроблення креслень деталей	250	—	1,0	1,1	1,4
16	Розроблення складального креслення	450	1,0	1,1	1,3	1,7
17	Складання специфікацій	55	—	1,0	1,0	1,0
18	Складання відомостей специфікацій	35	—	1,0	1,0	1,0
19	Складання відомостей розсильних документів	35	—	—	1,0	1,0
20	Складання відомостей узгодження застосування покупних виробів	35	—	—	1,0	1,0
21	Складання технічних умов	20	1,0	1,0	1,0	—
22	Складання програми та методики випробувань	52	1,0	1,0	1,2	—
23	Конструктивний контроль	150	1,0	1,1	—	—
24	Нормоконтроль	130	1,0	1,1	—	—
25	Технологічний контроль	120	1,0	1,1	—	—
26	Допоміжні роботи	—	1,0	1,0	1,0	1,0

2. На рис. 9.22 показано ІПСП, що описує динаміку процесу проектування вибраного замовлення, а на рис. 9.23 — сіткові моделі згенерованої сітки.

Макропозиції цих блоків відображають деякий список потенційних виконавців відділу для кожного переходу (тобто робіт) відповідно до їх кваліфікації та складності виконуваної роботи.

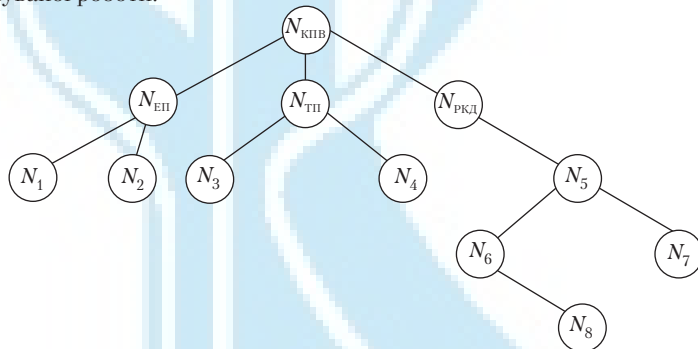


Рис. 9.22. Дерево відношень алгоритму керування ІВК: ЕП — ескізне проектування; ТП — технічний проект; РКД — розроблення робочої КД

3. Згідно з КФВ КБ для кожної роботи складаємо список виконавців з достатньою для її виконання кваліфікацією (в табл. 9.8 вони мають ненульові коефіцієнти  $K_{\text{вик}}$ ). Замість макропозицій сіткових блоків здійснюємо підстановку вільного на поточний момент виконавця з використанням результатів експертного аналізу (див. табл. 9.6). При цьому здійснюється генерація всіляких послідовностей спрацьовування переходів, тобто всіляких варіантів проектування транспортного модуля.

4. Завдяки тому, що старша компонента ІПСП на рис. 9.22 і рис. 9.23 належить до системи типових блоків, побудованих на сітці алгоритм керування є коректним і задовольняє умови безпеки, живучості, збудження та завершення.

5. Проводимо динамічне транслявання побудованої сіткової моделі, тобто навантажуюмо відповідними тривалостями виконання проектних робіт переходи сіткових блоків нижчого рівня й виконуємо часові оцінювання одержаних варіантів розподілу планових завдань усередині КБ з урахуванням директивного строку виконання замовлення.

6. Здійснюємо ініціалізацію роботи модуля інтелектуальної надбудови для розв'язання питання вибору оптимального завантаження виконавців з урахуванням забезпечення мінімального строку проектування транспортного модуля. Визначаємо резерв часу для кожного виконавця ІВК, коригуємо інформацію БД.

При зміні КФВ підрозділів або переліку планових робіт п. 5 та 6 слід повторити.

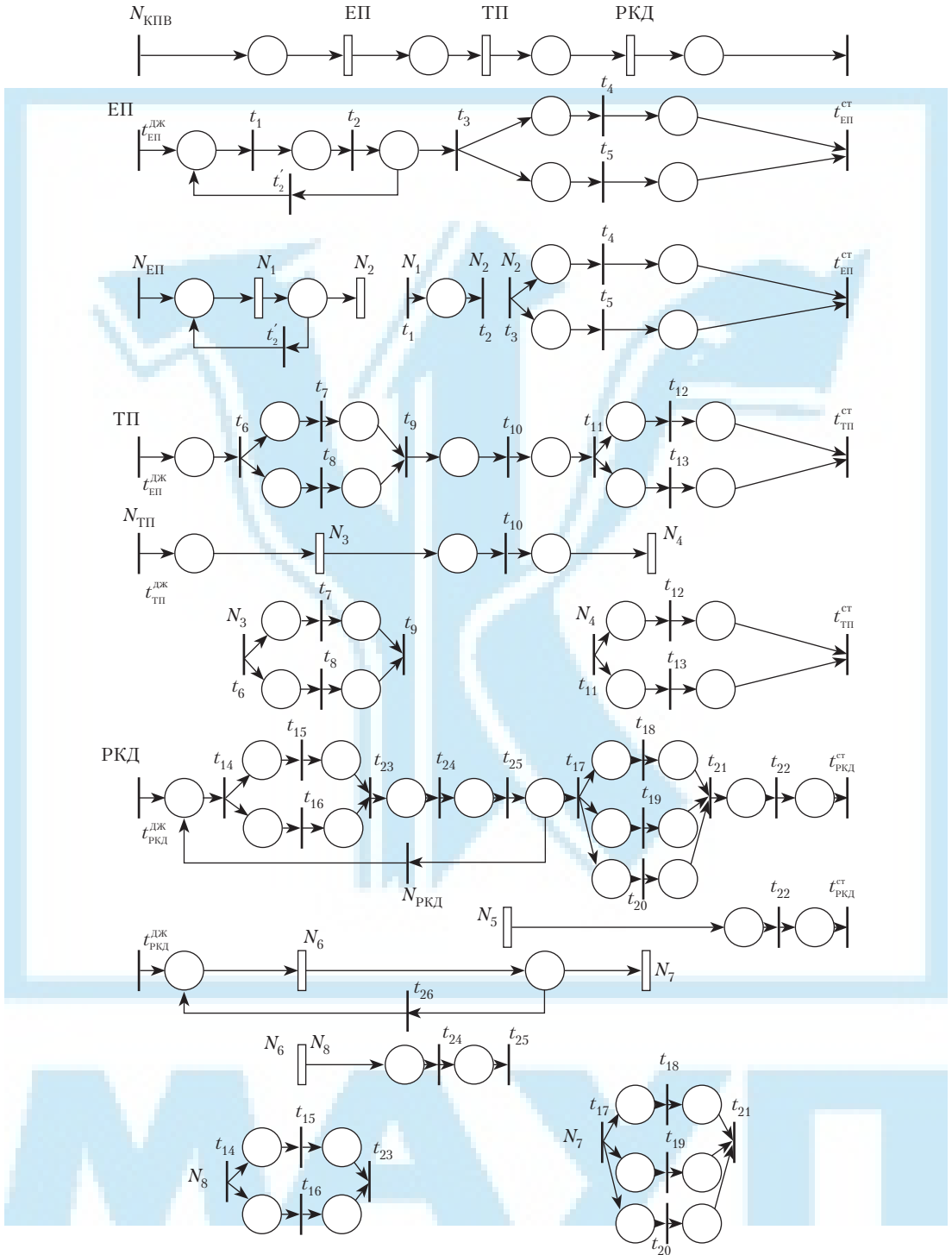


Рис. 9.23. Алгоритм керування конструкторським підрозділом ІБК

### Контрольні завдання

1. Дати означення та розкрити сутність методу повного перебирання як підходу до розв'язання оптимізаційних задач.
2. Пояснити, в чому полягає метод евристичного пошуку і дати його означення.
3. Навести і схарактеризувати основні стратегії організації пошуку вершин дерева перебирання.
4. Визначити організацію пошуку розв'язків при плануванні у просторі задач і станів.
5. Пояснити, на чому ґрунтується і що являє собою метод динамічного програмування.
6. Охарактеризувати етапи проектування і впровадження інтелектуалізованих систем.
7. Розкрити особливості взаємодії ЕС і методу скінченних автоматів у межах інтелектуалізованих САПР.
8. Прокоментувати два основних етапи планування, які формуються при створенні технологічного процесу, і розкрити роль інтелектуальної складової системи планування при цьому.
9. Навести і проаналізувати приклад задачі пошуку розв'язків у просторі станів і прокоментувати функціональність правил, на яких ґрунтуватиметься розв'язок задачі.
10. Назвати і розтлумачити основні критерії оптимальності, що використовуються при пошуку розв'язків у просторі станів.
11. Охарактеризувати метод пошуку в ієрархії просторів і проілюструвати його прикладом низхідного уточнення плану.
12. Розкрити принцип найменших завершень і проілюструвати прикладом розв'язання задачі в умовах потенційної конкуренції претендентів на обслуговування.
13. Охарактеризувати принцип формування правил, які використовуються в глибинних знаннях.
14. Пояснити термін «глибинні знання».
15. Дати означення стадій життєвого циклу виробу.
16. Охарактеризувати принципи скінченно-автоматної інтерпретації керування процесами на підготовчій стадії виробництва виробів.
17. Дати тлумачення технологічного процесора і технологічного процесорного середовища ГКІС.
18. Назвати порівняльні характеристики ОК на підготовчій (фаза КПВ) та виробничій стадіях життєвого циклу виробу.
19. Розкрити складові інформаційної обробки комплексу КД.
20. Дати означення вектора готовності та охарактеризувати його складові.
21. Прокоментувати поданий за допомогою апарату СП формальний опис задачі організації керування ІВК.
22. Дати означення ІПСП.
23. Розкрити умови відповідності між ІВК та модифікованою СП.
24. Розкрити принцип побудови ієрархічних СП-моделей ІВК у фазі КПВ на підготовчій стадії життєвого циклу виробу.

25. Дати означення сіткової моделі структури алгоритму організації керування ІВК.
26. Охарактеризувати роль КФВ структурного елемента ІВК при формуванні плану завантаження структурного підрозділу.
27. Прокоментувати часову інтерпретацію СП з урахуванням послідовності виконання операцій у проектному підрозділі ІВК.
28. Навести приклад розрахунку часової затримки в сітці верхнього рівня, що інтерпретує розподіл робіт у КБ ІВК.
29. Розкрити особливості застосування методики ЕКО в задачах планування завантаження підрозділів ІВК на підготовчій стадії виробництва.
30. Навести й охарактеризувати основні етапи методики моделювання функцій ІВК та розробки алгоритму керування останнім.





## ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ РЕАЛЬНОГО ЧАСУ

*Всі ідеї в науці народилися у драматичному конфлікті між реальністю і нашими спробами її зрозуміти.*

Альберт Ейнштейн

### 10.1. Загальні відомості

Поняття функціонування в реальному часі є досить умовним і може стосуватися лише конкретних дій і процесів. Шкала реального часу для задач керування крилатою ракетою, хімічним процесом або складальним ПР буде різною. Коли говорять про інтелектуальні системи (ІС) з точки зору їх роботи в режимі реального часу, часто мають на увазі не тільки і не так швидкість прийняття рішення, як можливість обробки динамічних подій складних ситуацій з урахуванням часового фактора. Водночас існують застосування ІС, де саме швидкодія при виробленні керуючих команд має першорядне значення.

У загальному випадку можна вважати найбільш наближеним до відображення сутності режиму реального часу для ІС таке твердження:

**Твердження 10.1.** *Функціонування ІС у режимі реального часу передбачає здатність їх «розмірковувати» про динамічні зміни в зовнішньому середовищі і приймати рішення в строки, прийнятні для адекватного реагування на ці зміни.*

Передумовами використання ІС у задачах реального часу можуть бути:

- великі навантаження операторів, що призводить до їх швидкого стомлення й зниження уваги;
- відсутність можливості побудови адекватної математичної моделі керованого процесу або її надмірна складність;
- суттєва залежність ефективності управління процесом від досвіду та кваліфікації оператора;
- часті зміни стратегії управління тощо.

*Інтелектуальні системи реального часу (ІСРЧ)* є не тільки об'єктом інтенсивних наукових досліджень і розробок, їх дедалі більше застосовують у виробничих умовах і вони є корисним комерційним продуктом.

Першим комерційним пакетом, призначеним для побудови інтелектуальних систем контролю й управління в режимі реального часу, був пакет PICON (Process Intelligent Control) фірми LMI (LISP Machine Inc.). Основою цього пакета є реалізована на Ліспі ЕС, пов'язана з розподіленою системою управління. Пакет PICON може обробляти до 20 тис. показань датчиків і сигналів тривоги. Особливістю його є те, що з розподіленої системи не передаються всі дані в режимі постійного сканування, а здійснюється звернення до них лише за необхідності при виведенні. Цим система нагадує людину-оператора, що здійснює вибіркове переглядання й аналіз характеристик, ґрунтуючись на власному досвіді. В міру необхідності активізуються

правила та процедури, розв'язуючи задачі забезпечення безпеки, діагностування несправностей, контролю.

Пакет PICON підтримує подання знань на основі правил і фреймів, передбачає пряме та зворотне виведення. Графічний інтерфейс виконує інтерактивне приймання команд від оператора, передбачає взаємодію між кількома експертами, забезпечує графічне подання змін характеристик ОК. Даним, які надходять від датчиків, у цьому пакеті можуть приписуватися часові інтервали, протягом яких вони зберігають достовірність (значущість), а також приблизні значення цих даних.

Для забезпечення ефективності обчислень використано два процесори із спільного області пам'яті: один з процесорів — Motorola 68010, який підтримує модулі системи, реалізовано на Сі; другий — LISP-процесор. Алгоритми довільної структури динамічно завантажуються в процесор Motorola 68010, надходячи з LISP-процесора. Система має механізм «фокусування уваги», що дає змогу концентруватися на окремих аспектах виробничого процесу; при цьому всі відповідні сенсорні дані та продукційні правила розглядаються частіше.

Пакет PICON знайшов застосування на підприємствах нафтохімічної промисловості, а також у розробках, які стосуються космічних досліджень.

## 10.2. Вимоги до інтелектуальних систем реального часу та шляхи підвищення їх швидкодії

Міжнародним об'єднанням із ШІ виділено ряд властивостей, якими повинні володіти інтелектуальні системи, що працюють у реальному часі. До цих властивостей зокрема належать:

- динамічне призначення пріоритетів розв'язуваних задач;
- виконання пошуку в БЗ за наявності часових обмежень;
- оптимальне співвідношення між точністю розв'язку, з одного боку, і витратами (в тому числі й часовими) на його одержання (прийняття) — з іншого;
- акцентування уваги на найважливіших подіях, які відбуваються в динамічному процесі;
- використання методів виведення з урахуванням часових відношень;
- застосування нових ефективних методів попередньої обробки інформації для скорочення обсягу та складності даних.

Метою *динамічного призначення пріоритетів* розв'язуваних задач є, по-перше, забезпечення першочергового обслуговування критичних задач, тобто таких, наслідки яких суттєво впливають на перебіг процесу (обробка аварійних ситуацій, аналіз різких змін значень параметрів); по-друге, раціональний розподіл обчислювальних ресурсів між задачами, що відрізняються обсягом, ступенем впливу на якість прийнятих рішень тощо. Механізм динамічного призначення пріоритетів передбачає наявність розвиненої системи переривань. У разі появи чергового завдання система аналізує співвідношення пріоритетів поточного завдання й того, яке щойно надійшло, і розв'язує питання про необхідність переривання поточного завдання з метою виконання нового. Значення відносних пріоритетів можуть призначатися як заздалегідь, так і в процесі роботи системи; до того ж ці значення можуть змінюватися.

*Забезпечення пошуку в БЗ за наявності часових обмежень* може досягатися групуванням окремих елементів знань залежно від типу розв'язуваних ними задач, контексту використання шляхом формування системи взаємообміну між елементами знань, запам'ятовуванням і збереженням структур проміжних зв'язків.

*Оптимальне (або близьке до нього) співвідношення між точністю розв'язання задач і витратами на її досягнення створюється за допомогою ефективних методів виведення, використанням поєднань спеціалізованих і універсальних процесорів тощо.*

*Концентрація уваги на найважливіших подіях забезпечується в найпростіших випадках оцінюванням заданих пріоритетів задач, а в складних системах — на основі аналізу контексту та поточної ситуації.*

*Методи виведення з урахуванням часових відношень ґрунтуються на використанні відповідних формалізмів подання знань, наприклад часової логіки, або передбачають обов'язкове відображення таких відношень у межах традиційних моделей (наприклад, коли в лівих і правих частинах продукцій є факти, в яких фігурують поняття «раніше», «після», «одночасно» тощо).*

*Ефективна попередня обробка даних передбачає, наприклад, компактне подання розріджених матриць, виявлення та передачу в системі тільки інформативних показників (наприклад, значення відхилень параметра, а не просто абсолютне значення останнього) та ін.*

Шляхи підвищення швидкодії ІСРЧ можна поділити на апаратні та програмно-методичні.

До *апаратних* слід зарахувати: розробку швидкодіючих процесорів загального та спеціального призначення; паралельну архітектуру обчислювальних систем; застосування інтегральних мікросхем, які реалізують операції нечіткої логіки; використання нейрокомп'ютерної технології тощо.

*Програмно-методичні* шляхи містять: створення ефективних компіляторів; розробку ефективних методів обчислення та спеціальних алгоритмів для роботи з БД в умовах жорстких часових вимог; використання спільних областей пам'яті; розподіл БЗ на спеціалізовані модулі; фіксацію типових дерев пошуку рішень до найважливіших і/або таких ситуацій (задач), які виникають часто; застосування елементів самонавчання; використання «монотонного» наближення до рішення (на противагу стратегії, що передбачає вироблення якогось рішення лише після завершення процедури пошуку).

«Монотонне» наближення полягає у послідовному поліпшенні і/або деталізації проміжних розв'язків у ході виведення (при цьому навіть у разі переривання процесу виведення чи раптової зміни (наближення) критичних термінів існуватиме варіант розв'язку, який бодай деякою мірою враховував би обставини), а також у використанні програм очищення пам'яті (так званих збирачів сміття). Завданням останніх є своєчасне виключення з пам'яті застарілих фактів, оскільки в протилежному разі весь обсяг пам'яті може заповнюватися інформацією, яка не використовується.

Ефективність організації процесу «збирання сміття» відіграє суттєву роль не тільки з точки зору раціонального використання ресурсів системи, а й може позначитися на правильності прийняття рішень. Справді, умовою активізації правил може бути не тільки наявність певних фактів, а й відсутність деяких інших. У такому разі затримка з виключенням застарілих даних може призвести до помилкової активізації правила.

Можна рекомендувати зокрема два шляхи усунення таких ситуацій: призначення задач «збирання сміття» найвищого пріоритету, але це може призводити до блокування інших задач у невідповідні моменти часу; запобігання урахуванню правилом фактів як застарілих, якщо інші правила враховують їх відсутність.

### 10.3. Побудова та особливості застосування інтелектуальних систем реального часу

У загальному вигляді склад ІСРЧ показано на рис. 10.1. Поряд з традиційними компонентами систем із ШІ (МЛВ, базами даних і знань тощо) така система має розвинену інтерфейсну підсистему, яка по суті складається з двох частин: інтерфейсу користувача (як і у звичайних інтелектуальних системах) і комплексу інтерфейсів з обладнанням, який містить вхідні та вихідні інтерфейсні модулі. Вхідні інтерфейсні модулі призначені для забезпечення передачі вхідної інформації від датчиків і приладів до системи з метою подальшого її відображення в БД. Вихідні інтерфейсні модулі передають інформацію у зворотному напрямку – від інтелектуальної системи до обладнання з метою виконання прийнятих системою рішень (команд).

Хоча всі сучасні комп'ютерні системи мають вбудовані модулі відліку часу, на рис. 10.1 такий модуль (таймер) виділено окремо, щоб наголосити на особливій важливості безпосереднього доступу до нього (з метою відображення поточного часу в БД) саме для ІСРЧ.

Можливими є різні принципи збирання та надходження інформації від зовнішніх джерел (через датчики): періодичне опитування датчиків; незалежне (асинхронне) надходження даних у міру змін параметрів; мішаний (комбінований) підхід (перший і другий принципи).

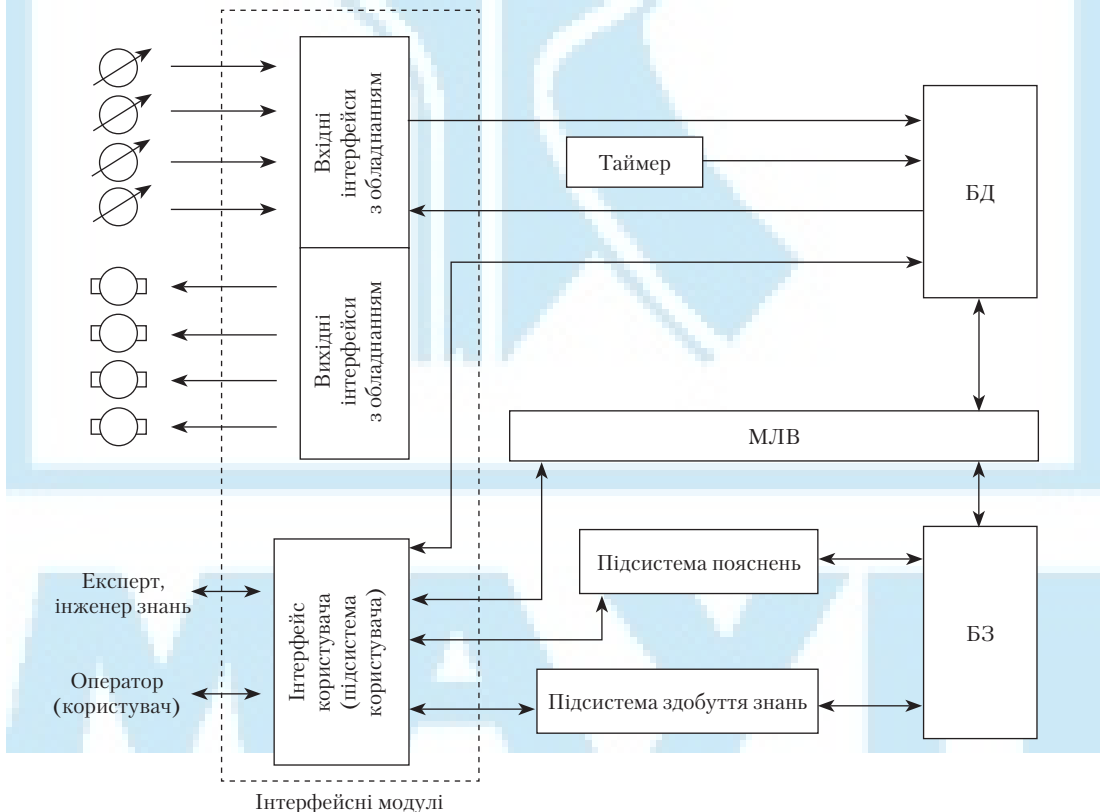


Рис. 10.1. Загальна схема ІСРЧ

*Періодичне опитування* датчиків може, у свою чергу, передбачати: послідовне сканування всіх датчиків з однаковою частотою (одного за одним у певній черговості); сканування з різними частотами для окремих датчиків (залежно від важливості відповідних параметрів, що реєструються, від швидкості зміни параметрів тощо). Періодичне опитування датчиків простіше в реалізації, але виправдане лише при відносно невеликій кількості вхідних параметрів, високій швидкості обробки (логічного виведення), невисокій (порівняно з частотою опитування) швидкості зміни параметрів, що реєструються. Недоліком є вірогідність значного процента «холостих» опитувань, імовірність пропустити «стрибки» чи різкі зміни, що відбуваються в проміжках між опитуваннями датчиків.

Підхід, який ґрунтується на *незалежному надходженні даних* у міру зміни параметрів, також має кілька варіантів реалізації: а) реєстрація кожного нового значення параметра, що відрізняється від попереднього на певний заданий крок; б) реєстрація лише значень, які виходять за передбачений діапазон; в) мішаний (комбінований) варіант (сума двох попередніх).

Варіант «а» дає змогу відображати в БД повну картину стану об'єкта, враховувати різні комбінації значень параметрів, але водночас потребує більш складних і розвинених механізмів реєстрації та доступу до БД.

Варіант «б» простіший в реалізації, потребує менше обчислювальних і апаратних ресурсів, але прийнятний лише для відносно нескладних стратегій управління (або моніторингу, діагностування). Цей варіант є неприйнятним, наприклад, у ситуаціях, коли кожний параметр окремо перебуває у своєму допустимому діапазоні, але сама комбінація (сукупність) кількох певних параметрів виявляється забороненою.

Найбільш раціональним у рамках другого підходу є комбінований варіант «в», який для різних параметрів поєднує варіанти «а» та «б». В цьому разі дані про більш важливі, швидкозмінювані параметри надходять у систему безпосередньо у разі їх зміни. Дані ж про інші параметри, враховувати які важливо і необхідно, але які не є критичними, а являють собою загальний інформаційний «фон» для прийняття рішень, скануються періодично.

Організація виведення в ІСРЧ (наприклад, у разі продукційного подання знань) може ґрунтуватися на різних схемах, а саме: 1) традиційне послідовне виконання етапів «зіставлення – вибір конфліктної множини – вирішення конфлікту – виконання продукції»; 2) асинхронне (паралельне) виконання окремих етапів.

*Перша схема* відповідає вже розглянутому в розд. 2 і 3 механізму логічного виведення. Кожний етап зіставлення може синхронізуватися із завершенням чергового циклу опитування датчиків (при періодичному скануванні), тобто спрямовуватися на обробку останнього набору вхідних даних. У загальному випадку цикли виведення й опитування (при періодичному опитуванні) не обов'язково повинні мати однакову частоту. Зауважимо лише, що робити частоту циклів виведення вищою від частоти опитувань було б надмірною вимогою; водночас, коли інтервали між окремими циклами виведення передбачаються більшими від періодів опитувань, треба бути впевненими, що реакція на зареєстровані зміни параметрів не виявиться занадто запізненою, а ймовірність втратити важливі «стрибки» параметрів, що сталися між виведеннями, є досить малою.

*Друга схема* (асинхронного виконання окремих етапів виведення) є прийнятнішою для ІСРЧ. Вона дає змогу оперативно реагувати на зміни ситуації, активізувати обчислювальні ресурси для обробки саме значущих подій тощо. Ця схема передбачає наявність кількох модулів у складі МЛВ, які можуть працювати паралельно. У загальному випадку це модулі зіставлення, вибору, виконання (рис. 10.2).

Результатом роботи модуля зіставлення є конфліктний набір правил, який передається на вхід модуля вибору. Після цього модуль зіставлення готовий знову виконати своє завдання, не чекаючи завершення роботи інших модулів. Оскільки найбільше часу при виведенні займає саме процес зіставлення, така схема забезпечує значну економію часу, дає змогу раціонально розподіляти апаратні ресурси для виведення окремих спеціалізованих функцій.

Модуль вибору передає інформацію про відібрану «продукцію» модулю виконання, після чого відразу можна виконувати черговий вибір з нового конфліктного набору правил.

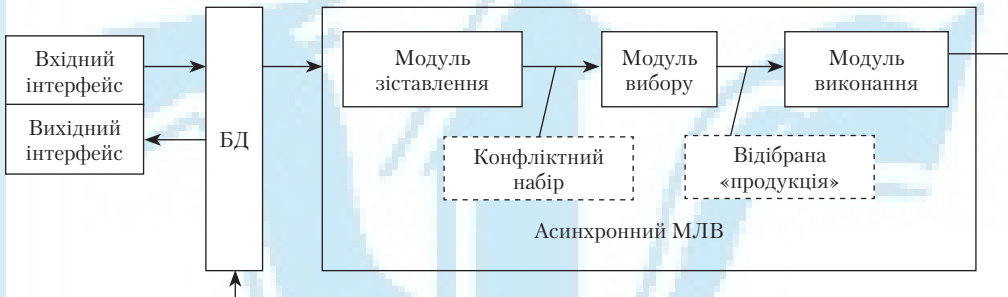


Рис. 10.2. Схема асинхронного МЛВ

У модулі виконання реалізується дія, передбачена в правій частині правила: це або занесення до БД нового факту, або виклик певної процедури.

Ураховуючи специфіку ІСРЧ, слід відзначити важливість для них функції прогнозування. Як доповнення до можливостей, що надаються звичайними моделями подання знань, може бути поєднання компонентів, які утворюють саме ЕС, з компонентами, що реалізують імітаційне моделювання процесів у предметній галузі.

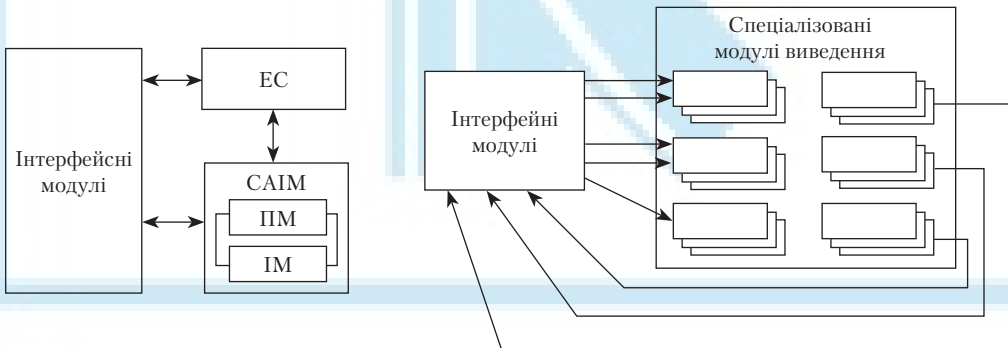


Рис. 10.3. ІСРЧ з вбудованою системою імітаційного моделювання

Рис. 10.4. Схема багатомодульної асинхронної системи виведення

Загальну схему такої системи зображено на рис. 10.3. В ній інформація про зовнішню ситуацію відображається в БД та ІМ. За рішенням ЕС відбувається автоматизоване імітаційне моделювання (див. розд. 6), результати якого аналізуються ЕС, яка й приймає остаточне рішення.

У разі потреби високої швидкодії та за умови чіткого розподілу окремих функцій (що дає змогу організувати модульну структуру БЗ) ІСРЧ може бути реалізована у вигляді сукупності спеціалізованих модулів виведення, які працюють незалежно, ма-

ючи власні (спеціалізовані, а отже, й досить обмежені) бази даних і знань. Такі модулі активізуються асинхронно, кожен за наявності своїх вхідних умов (тобто у разі зміни відповідних параметрів). Таку схему виведення спрощено показано на рис. 10.4.

Інтелектуальні системи реального часу застосовуються, як правило, в задачах керування об'єктами. Разом з оператором вони входять до складу контуру керування (рис. 10.5, а). Організація замкненого контуру керування із застосуванням інтелектуальної підтримки в системах з оператором передбачає: задавання ступеня участі ЕС у замкненому контурі керування; наявність клавіші «Виконання», набір команд замкненого керування; специфікацію передумов виконання команд; команди скасування дій.

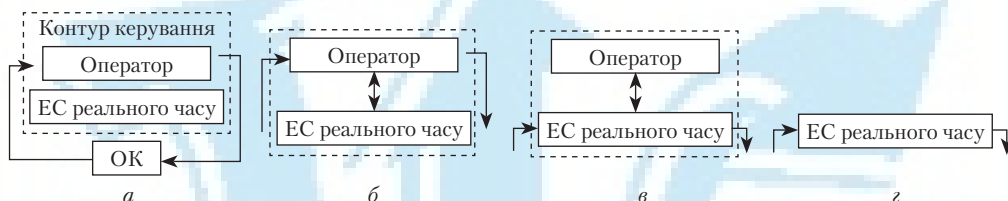


Рис. 10.5. ІСРЧ в контурі керування об'єктом:

а – загальна схема; б – ЕС виконує дорадчі функції; в – ЕС керує об'єктом за умови підтвердження команди керування; г – ЕС повністю виконує керівні функції

Ступінь участі ЕС у процесі керування визначається рангом її «повноважень» у прийнятті остаточних рішень і можливістю безпосереднього впливу на керований процес. Інтелектуальна система повинна забезпечувати гнучкий набір режимів її використання за такими рівнями участі ЕС:

- ЕС поза контуром керування. При цьому необхідним є остаточне рішення оператора, який ініціює необхідні дії (рис. 10.5, б);
- ЕС замикає контур керування тільки за рішенням оператора (тобто останній підтверджує прийняте рішення натисканням на клавішу «Виконання») або якщо оператор не втрутився протягом певного проміжку часу (рис. 10.5, в);
- ЕС постійно перебуває в замкненому контурі керування (рис. 10.5, г).

Клавіша «Виконання», що ініціює набір команд, який відповідає одержаній від ЕС рекомендації, дає змогу оператору контролювати дії ЕС в контурі керування й гарантує, що здійснюються саме операції, передбачені цією рекомендацією.

Призначення команд замкненого керування полягає в забезпеченні можливості ЕС безпосередньо впливати на керований процес (на відміну від рекомендацій, орієнтованих тільки на оператора). Так, у системі, що ґрунтується на правилах, такі команди можуть або повністю формувати консеквенти окремих правил, або включатися до них поряд із рекомендуючою частиною. Прикладом останнього варіанта може бути такий:

ЯКЩО показання датчика температури печі підвищуються більш як на 2°С за хвилину,  
ТО (рекомендація): «зменшити подавання газу на 5 % за допомогою клапанів»  
(команда): «перепускний клапан Г: = перепускний клапан 0,05 %».

Передумови виконання команд замкненого керування повинні задаватися не тільки з урахуванням стану керованого процесу, а й контексту, що зберігається в БЗ. Наприклад, важливо те, яким правилом запускається команда та з якого рівня використання ЕС. Можна, наприклад, дозволити ЕС видавати команди про роботу

об'єкта в режимах, близьких до граничних, якщо тільки це санкціонується оператором; водночас у разі автономної роботи ЕС в замкненому контурі керування логічніше не допускати прийняття таких команд.

Аналогічно, якщо ЕС замкнула на себе контур керування через відсутність реакції оператора і в такому режимі дістає інформацію про серйозні недоліки, доцільно, щоб заходи безпеки були більш радикальними, ніж за участі оператора, коли система сама може ініціювати тільки мінімально необхідний набір екстрених заходів, а про інші запитати підтвердження.

Необхідність у команді скасування впливає з можливості виникнення непередбачених ситуацій, коли оператору слід самому приймати рішення, повністю або частково виключивши ЕС з контуру керування.

#### 10.4. Інтелектуальна підтримка функціонування промислових роботів у реальному часі

**Машинний інтелект і робототехніка.** Основу будь-якого робота становить його «тіло» (рис 10.6, блок 7), тобто сукупність механічних, електромеханічних, пневматичних та інших пристроїв, що забезпечують його властивості на міцність і несучі властивості (шасі, козухи), здатність переміщуватися в просторі (колеса, гусениці, «ноги» і їх приводи) і впливати на об'єкти зовнішнього світу (ефектори — маніпулятори, штовхачі, захвати). На шасі розташовують пристрої сприйняття інформації із середовища (сенсори), пристрої управління і різні допоміжні пристрої, що розглядаються як фізичні об'єкти. Залежно від призначення робота склад його «тіла» може змінюватися.

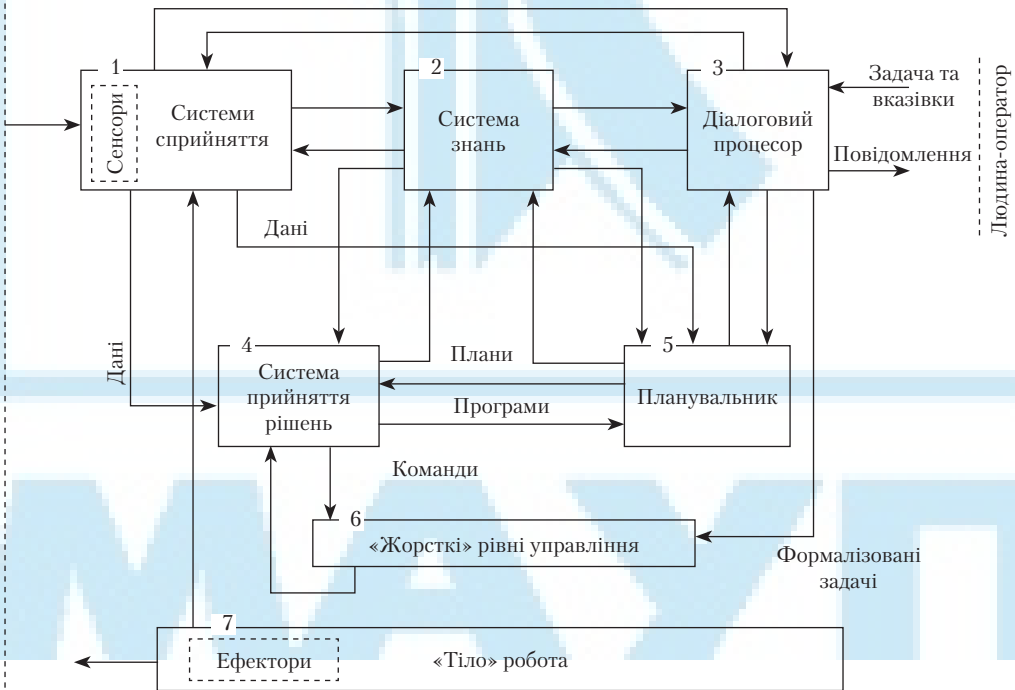


Рис. 10.6. Структурна схема робота з елементами штучного інтелекту



Неодмінним компонентом робота є система управління, яку називають «жорсткою» (блок 6). Вона містить незмінну програму або логічні пристрої, які управляють роботою приводів ефекторів, і може бути також представлена сукупністю вимикачів, упорів і т. ін. Якщо робот забезпечений більш розвинутою системою управління, «жорстка система» становить її нижній рівень.

Система сприйняття (блок 1) включає датчики, які контролюють стан «тіла» робота, і параметри зовнішньої ситуації, в якій він перебуває. Сюди можуть входити також розвинені системи технічного зору, що містять оптичні, ультразвукові, телевізійні та інші пристрої. Всі вони становлять набір сенсорів робота. Система сприйняття включає також засоби попередньої обробки інформації, що надходить від сенсорів, і засоби інтерпретації цієї інформації. Під інтерпретацією розуміють «переклад» мови віхідних сигналів сенсорів на мову опису об'єктів і станів зовнішнього середовища. Інтерпретація здійснюється за допомогою знань про зовнішній світ і «тіло» робота, що зберігається в блоці 2. Інтерпретовані повідомлення надходять до людини-оператора, що бере участь в управлінні роботом (зв'язок з блоком 3), або можуть використовуватися власною системою управління (зв'язки з блоками 4 і 5).

Пристрій зв'язку робота з людиною-оператором — діалоговий процесор — призначений для виконання ряду функцій. Одна з них полягає в сприйнятті і попередній обробці вхідних висловлювань, що надходять від оператора. За допомогою висловлювань оператор може ставити перед роботом задачі, управляти процесом їх розв'язування, коригувати дії робота тощо. Спеціальні висловлювання можуть бути адресовані системі сприйняття і керувати режимом її роботи (ввімкнення/вимкнення окремих сенсорів, задачі на пошук потрібних ознак, об'єктів). Висловлювання можуть бути мовними або мають текстову форму. В останньому випадку вони подаються, наприклад, у вигляді послідовності рукописних символів, прошивок на перфострічці або дій на спеціальну клавіатуру. Мовні висловлювання можуть реалізуватися деякою більш-менш обмеженою підмножиною природної мови.

Функція попередньої обробки полягає в здійсненні морфологічного і синтаксичного аналізу вхідних висловлювань, виконуваного на основі знань про мову, що використовується (блок 2).

Наступна функція діалогового процесора пов'язана із семантичним аналізом висловлювань. У ході цього аналізу висловлювання інтерпретується в якусь внутрішню мову робота, зрозумілу його системам управління, і цією мовою формулюється задача або повідомлення, значення яких повинно повністю збігатися із значенням сприйнятого висловлювання. Так здійснюється «розуміння» роботом людини. Повнота цього розуміння може бути проконтрольована лише за остаточними результатами дій робота.

Далі семантичний аналіз здійснюється на основі знань, які подано в блоці 2. При цьому повинні використовуватися знання не тільки про зовнішній (фізичний) світ і про самого робота, а й знання про світ задач людини, мету, способи їх вираження, характерні замовчування і т. ін. Так, наказ «принеси круглий предмет» не повинен приводити до спроб робота розглянути як круглий предмет голову оператора.

Повнота реалізації описаних функцій залежить від призначення робота і рівня мови керування, що використовується. Якщо, наприклад, вказівки формуються в термінах команд для блоку 6, діалоговий процесор може бути редукований до клавіатури управління. Ще одна функція діалогового процесора полягає в інтерпретації і видачі повідомлень і запитів, сформованих системами сприйняття, знань та управління. Інтерпретація полягає в перекладі повідомлень з внутрішньої мови робота на мову систем виведення. Як такі системи можуть використовуватися синтезатори мови, відеотермінали, графічні пристрої та інші пристрої.

Власне система управління робота, точніше її вищі рівні, представлені планувальником (блок 5) і системою прийняття рішень, або розв'язувачем (блок 4). Задача планувальника полягає в тому, щоб на основі відповідних даних здійснити автоматичне розв'язання задачі, формальний опис якої надходить з діалогового процесора. У простих випадках планувальник виступає в ролі трасувальника, що визначає оптимальну або близьку до неї траєкторію (програму) переміщення в просторі робочих органів робота або самого робота.

У складніших випадках планувальник виконує складну обробку інформації, яка подана не тільки в цифровій, а й у символній формі, і виробляє узагальнені плани досягнення поставленої мети.

Розв'язувач зазвичай виконує ті самі функції, що й планувальник. Відмінність полягає в роботі на різних рівнях деталізації задач. Рівень планувальника можна назвати стратегічним, а рівень розв'язувача — тактичним. Розв'язуючи свої задачі, планувальник використовує інформацію про загальний характер середовища, приблизне розташування перешкод і т. ін. Така інформація зберігається переважно в блоці 2 і відіграє роль карти робочого простору. А розв'язувач використовує в основному дані про поточну ситуацію, що надходять від системи сприйняття, і формує конкретні команди управління, послідовність яких повинна забезпечити виконання поточних етапів плану, синтезованого планувальником.

Якщо виконання плану призводить до невдачі, наприклад, робот натрапляє на непереборну перешкоду і чергова команда не може бути виконана, розв'язувач розпізнає це або за даними системи сприйняття, або за сигналами з «жорсткого рівня» управління і формує відповідне повідомлення для планувальника. Останній або розв'язує задачу наново, синтезуючи новий план, або через діалоговий процесор передає повідомлення оператору і чекає подальших вказівок.

Кожний з розглянутих блоків реалізує досить складний комплекс функцій, розв'язуючи при цьому свою задачу. Проблеми, пов'язані з розробкою теоретичних принципів і технічних засобів побудови кожного з блоків, і є основними проблемами робототехніки. Всі вони тісно пов'язані з ідеями і методами штучного інтелекту.

### **Особливості інтелектуалізації промислових роботів у дискретному просторі.**

Роботи заміняють людину в умовах, коли вона не в змозі через фізичне перенавантаження або небезпечні ситуації виконувати роботу, наприклад, монотонні виснажливі операції, висока продуктивність процесів на виробництві, дослідження акваторій світового океану та космічного простору, обслуговування небезпечного ядерного та хімічного устаткування тощо. При цьому значну трудомісткість становить формулювання інструкцій дій формалізованою мовою, тобто переклад інструкцій, поданих близькою до природної (скажімо, технологічною) мовою, на таку мову, яка прийнятна для обчислювальної/керуючої системи. Отже, постає проблема підвищення рівня «інтелектуалізації» сучасних гнучких комп'ютерно-інтегрованих систем, за яким ці системи зможуть «зрозуміти» безпосередньо опис поданих інструкцій.

Незважаючи на значний інтерес, що проявляється останніми десятиріччями до робототехніки, і на помітний бум у розвитку теорії штучного інтелекту, питання, пов'язані із створенням «симбіозної» інтелектуальної робототехніки, в наукових публікаціях порушувалися дуже рідко, їм приділялася явно недостатня увага.

Зокрема, нема однозначної і чіткої відповіді на питання, як врахувати невизначеність середовища «існування» робота, тобто реального виробництва, якою повинні

бути моделі його поведінки в цьому випадку, який механізм логічного виведення і т. ін., що враховують недетермінований характер дійсності при функціонуванні ІР у реальних середовищі та часі.

Розвинені, функціонально завершені системи, що є компонентами роботів різного типу, можна розділити на п'ять основних груп [5]:

- *B* – системи сприйняття зорової, слухової, тактильної та інших типів інформації про зовнішній світ;
- *M* – системи впливу на об'єкти зовнішнього світу (різного роду маніпулятори, виробничі системи і т. ін.);
- *T* – системи, що забезпечують переміщення робота (колісні, гусеничні, крокуючі, плаваючі і літаючі рухомі механізми й апарати);
- *P* – системи визначення цілей і планування дій робота, системи розв'язання задач;
- *C* – системи, що забезпечують комунікацію робота з людиною-оператором та іншими роботами мовою спілкування різних рівнів, аж до природної.

Отже, різні класи роботів іменуються набором букв, що позначають групи. Інтеграція **ступеня 1** задає п'ять класів: *B*; *M*; *T*; *P*; *C*. Інтеграція **ступеня 2** задає десять класів, наприклад *BM*, *MT* і т. д. Інтеграція **ступеня 3** задає також десять класів, наприклад *BMP*, *BTP* і т. д., а інтеграція **ступеня 5** задає лише один клас — *BMTPC*.

**Твердження 10.2.** Інтелектуальними роботами можна називати автономні роботи 3-го, 4-го і 5-го ступенів інтеграції, призначені для виконання дій у реальному світі та реальному часі.

**Означення 10.1.** Під автономністю робота розуміють функціональну незалежність, тобто здатність (отриману в результаті його програмування або «навчання») самостійно планувати і виконувати дії, спрямовані на розв'язання задач.

Традиційно вирізняють два підходи до керування складними системами та програмування робототехнічних пристроїв, які розв'язують певні задачі, — *алгоритмічний* і *декларативний*.

**Означення 10.2.** Алгоритмом називається чітка недвозначна послідовність інструкцій, яка зрозуміла виконавцеві і обов'язково приводить до гарантованого результату за скінченний час.

На відміну від алгоритму інструкції квазіалгоритму можуть бути не зовсім чіткими, отже, й результат виконання квазіалгоритмічної процедури не обов'язково є гарантованим.

*Перший підхід* (алгоритмічний) передбачає, що необхідно наперед продумати й детально розписати, **як розв'язувати** певну проблему, і полягає в наданні чітких послідовних інструкцій з цього для задачі будь-якої складності, причому інструкції повинні виконуватися системою або роботом без «усвідомлення» останніми власне змісту задачі.

*Другий підхід* (декларативний) потребує повноти знань стосовно того, **що розв'язувати**, тобто формулювання змісту завдання, проте **як** це завдання розв'язуватиметься, визначається наявністю інтелектуальної компоненти в складі системи або робота (або власне виконавцем).

Як приклади підходів наведемо дві задачі, які характерні для промислової робототехніки. *Задача першого типу* — *задача маршрутизації* (або диспетчеризації) полягає в тому, що рухомий робокар повинен обслуговувати за технологічним маршрутом обробляючі ресурси (верстати, склади, різноманітні проміжні нако-

пичувачі, контрольні позиції тощо) розподіленої у просторі розгалуженої гнучкої комп'ютерно-інтегрованої виробничої системи. Ця задача є досить складною, але піддається точній алгоритмізації, оскільки регламентується послідовністю операцій у технологічному процесі виготовлення певного виробу, а методи трасування та навігації для рухомих робочих органів дають змогу останнім відслідковувати трасу навіть за наявності певних відхилень від неї. Можна сказати, що для такого типу задач характерним є складання зовнішніх для об'єкта керування інструкцій у вигляді речень природної мови, програм, написаних алгоритмічними мовами високого рівня, тощо, тобто написання *первинних інструкцій* (інакше ще – первинного опису ситуацій), які й формують мету для об'єкта.

Інша справа, коли йдеться про постановку *задачі другого типу*, за якою промислового роботу необхідно з певного набору комплектуючих деталей зібрати складальну одиницю (вузол чи виріб у цілому), задану кресленням. Ця задача може ускладнюватися тим, що вихідне положення комплектуючих є невизначеним (наприклад, наперед не орієнтованим). У цьому випадку інтелектуальна компонента керуючої системи промислового робота за кресленнями має сама визначити послідовність складання комплектуючих (керуючись при цьому або технологічним описом, або «власним» набутим досвідом для об'єктів цього класу, тобто ще й прокласифікувати складальну одиницю і «витягнути» із своєї бази алгоритмів дій необхідний для цього випадку), розпізнати (ідентифікувати) комплектуючі деталі за класом та положенням у просторі або одну відносно іншої, скласти (або знов витягти з «пам'яті») алгоритм переорієнтації своїх робочих органів (захватних пристроїв, складальних, згинувальних головок тощо) відносно розташування на вихідній позиції чергових комплектуючих. Якщо додати, що треба контролювати надійність утримання деталей у робочих органах робота, витримувати визначені технологією режими складання (зусилля, швидкості тощо) в умовах дії випадкових факторів (наприклад, забрудненість поверхонь комплектуючих, що потребує коригування режимів складання), можливість виникнення інших (так званих нештатних) непередбачених ситуацій, то стає цілком зрозуміло, що розв'язання такого типу задач потребує наявності в системі керування роботом потужної *інтелектуальної компоненти* для безперервного аналізу поточних (первинних) ситуацій та внесення адаптуючих корекцій у його дії для досягнення початкової мети. Отже, в цьому разі можна говорити про *самокерування*.

**Означення 10.3.** *Самокерування* – це керування, за яким на основі аналізу інформації про зовнішню ситуацію (первинного опису ситуації, який у загальному випадку може ототожнюватися до вкраплення в самокерування зовнішньої компоненти) система керування об'єктом приймає рішення для досягнення визначеної мети.

В будь-якому разі *розвиненість інтелектуальних спроможностей роботів на-самперед залежить від здатності безпосередньо сприймати навколишнє середовище за допомогою потужної багатofункціональної системи спеціальних «органів чуттів» – тактильних, зорових, силомоментних у першу чергу.* Щоб розв'язати задачу планування дій ПР, необхідно скласти добре скоординовану, злагоджену схему операцій (логічних, математичних, технологічних та ін.), яка починається із заздалегідь заданих умов (передумов) і закінчується досягненням мети. Розв'язуючи такого роду інтелектуальні задачі, система ШІ (СШІ) постійно шукає шляхи обходу перешкод і досягнення заданої мети, намагається виробити певний план дій, дотримуючись

якого, можна досягти безпосередньо недоступної мети. Уміння США розв'язувати інтелектуальні задачі за допомогою використання наявних знань набувається шляхом навчання. Це уміння і пов'язані з ним навички розв'язання задач у певному сенсі набагато важливіше для США, аніж інформація, що зберігається в БЗ, хоча, звичайно, без необхідних базових знань неможливо знайти і рішення.

Враховуючи наведене на початку книги узагальнене визначення штучного інтелекту, спробуємо дати означення штучного інтелекту ГКІС.

**Означення 10.4.** Під інтелектом ГКІС розумітимемо наявність у її системі управління відповідного алгоритмічного і програмного забезпечення, що дає змогу розв'язувати задачі інтелектуального характеру за допомогою цілеспрямованого перетворення зовнішньої інформації і відповідних знань, які реалізують здатність системи до адаптації і навчання на власному досвіді. Характерними рисами інтелекту ГКІС є її здатність ефективно функціонувати в технологічній обстановці, що постійно змінюється, можливість легко перебудовуватися з розв'язання одного класу задач на інший.

В умовах функціонування в реальному часі інтелектуальна ГКІС повинна володіти здатністю аналізувати навколишнє оточення, розпізнавати польові об'єкти (деталі, інструменти), ідентифікувати перешкоди. Для розв'язання таких задач загальноприйняті методи розпізнавання образів не пристосовані, а деколи і непридатні. Тут *головна складність* полягає в тому, що промисловий ІР в ГКІС зазвичай має справу не з одним об'єктом, що належить тому чи іншому класу, а з сукупністю об'єктів з різних класів. *Інша складність* породжена багатоваріантністю об'єктів у полі зору таких роботів і ГКІС. Так, в умовах реального виробництва постійно міняється відстань до певного предмета, ракурс його сприйняття, умови та інтенсивність освітлення і т. ін. На практиці одні предмети і об'єкти часто затуляють інші, чим істотно ускладнюють їх розпізнавання.

У цьому випадку придатним слід визнати підхід до розпізнавання сцен за їх зображеннями, який базується на попередньому (в процесі навчання ІР) формуванні понять у вигляді аксіом класів з подальшим виділенням тих чи інших об'єктів сцени за допомогою логічного аналізу і розпізнавання. Цей підхід ґрунтується на *адаптивному методі логічного розпізнавання*, основна ідея якого полягає в логічному поданні (в термінах предикатів-ознак) понять, сформованих у процесі навчання ІР у вигляді проблемно-орієнтованого набору аксіом класів, і в організації процесу розпізнавання за допомогою алгоритмічних засобів пошуку логічного висновку.

При цьому в режимі навчання ІР подаються в різних ракурсах еталонні об'єкти (наприклад, деталі або інструменти) з різних класів. Далі будуються (автоматично) описи класів у вигляді формул *обчислення предикатів першого порядку*  $A_1(\omega)$ ,  $A_m(\omega)$ , які зв'язують початкові предикати-ознаки  $\xi_1, \dots, \xi_n$  і набувають істинних значень на елементах навчальної вибірки.

Потрібно наголосити, що в режимі розпізнавання перед промисловим ІР, як правило, виникають задачі трьох типів: *ідентифікація, класифікація і аналіз* (див. розд. 5). Мета ідентифікації полягає в тому, щоб ІР відшукав в умовах реального виробництва в реальному часі на зображенні сцени об'єкт цього класу. Метою класифікації є з'ясування питання: об'єкти яких класів представлені на зображенні сцени. Нарешті, мета аналізу полягає в з'ясуванні того, об'єкти яких класів і в якій кількості є на зображенні сцени, а також, що дуже важливо, у визначенні їх розташування, орієнтації і т. ін.



Коригування знань тісно пов'язане з проблемою навчання робота, що врешті виражається в розробці ефективних механізмів «бракування» знань за різними критеріями типу логічної суперечності або, навпаки, несуперечності.

В розглядуваній нами концепції функціонування системи розв'язання задач ІР ключовим моментом є роль плануючої підсистеми і підсистеми рефлекторної діяльності робота.

Синтез планів здійснюється на основі поставленої мети і накопиченого знання, причому мета може як задаватися людиною-оператором, так і вироблятися самим інтелектуальним роботом. При синтезі планів широко використовуються дедуктивні механізми планування, за яких план визначається як послідовність застосовних і придатних операторів, які вибираються з нагромадженого набору операторів. Оскільки в досить складних задачах планування на кожному кроці є альтернативні можливості вибору придатних операторів серед множини застосовних, нами буде вибрано підхід, що базується на використанні поняття нечіткого предиката для опису мети, підцілей, логічних моделей і т. ін.

Найважливішим моментом у плануванні цілеспрямованої поведінки промислового ІР є узагальнення планів, суть якого в тому, щоб робот міг узагальнювати вже побудовані плани так, аби використовувати їх надалі як додаткові укрупнені оператори, «навчаючись» розв'язувати щораз складніші задачі. Накопичуючи, таким чином, моделі узагальнених дій, ІР може ефективно застосовувати їх для побудови приблизних планів, кількість і складність яких щораз збільшує.

Зрозуміла та роль, яку в нашому укрупненому описі функціонування ІР відіграє підсистема виконання дій, відповідних побудованим планам. Саме взаємодія двох останніх підсистем визначає здатність інтелектуального робота успішно функціонувати в реальному світі.

На основі цієї інформації про зовнішнє середовище, що надходить від системи сприйняття, формується її модель, необхідна для ефективного планування цілеспрямованої поведінки ІР (в цьому випадку передбачається, що мета задається роботою ззовні). Водночас для реалізації спланованої роботом стратегії поведінки в умовах зміни стану зовнішнього середовища необхідна модель його рефлекторної діяльності, яка може бути деякою БЗ, що складається з набору продукцій вигляду «ЯКЩО..., ТО..., ІНАКШЕ...». Прийняті роботом рішення відповідно до виробленого ним плану цілеспрямованої поведінки і врахування «рефлексів», закладених у ньому, передаються в підсистему виконання дій, де й перетворюються в команди, передані на виконавчі органи, які, до речі, формують (в нашій інтерпретації) ефекторну систему ІР.

**Системи знань про зовнішнє середовище та дії робота для функціонування в реальному часі.** Розглянемо приклади застосування фреймового подання знань у дуже важливому й найпоширенішому напрямі інтелектуалізації ІР — формуванні системи знань про зовнішнє середовище робота та можливі операції в його робочому просторі в режимі реального часу проходження виробничого процесу в ГКІС.

**Система знань про зовнішнє середовище.** Ця система охоплює інформацію про його об'єкти, відношення між ними, а також про правила, використовуючи які оператор (або інтелектуальна надбудова системи керування роботом) може формувати нові об'єкти або нові відношення, перевіряючи водночас їх реалізованість з урахуванням фізичних законів. За допомогою системи технічного зору, сенсорних систем робота та системи знань про зовнішнє середовище оператор (інтелектуальна

надбудова робота) може формувати модель робочої сцени та об'єктів, з якими взаємодіє робот. Ці моделі й використовуються при задаванні цілей операції в робочому просторі робота.

Основою для створення системи знань про зовнішнє середовище [23] стала її концепція в термінах природних просторово-часових відношень [5]. Ця система складається з бібліотеки об'єктів-примітивів, сукупності просторових відношень і набору правил, які визначають дії над об'єктами. Формою подання об'єктів є фрейми (див. підрозд. 2.4), що утворюють ієрархічну структуру: *фрейм нульового рівня* — точка; *фрейм першого рівня* — лінія, пряма, відрізок; *фрейм другого рівня* — поверхня, площина; *фрейм третього рівня* — просторове тіло. Рівні пов'язані відношеннями «загальне — окреме», а фрейм, який є окремим стосовно цього, входить у його опис.

Фрейм примітива містить опис його імені, субфреймів (які входять до його опису), а також характерних ознак. Порожні слоти фрейму заповнюються користувачем. Наприклад, фрейм обмеженої площини фігури (другий рівень) має вигляд:

{*П-фігура: площа А* (її параметри); *границя Г* (параметри Г); *параметри фігури* (площа, геометричний центр, ...)}.

Опис цього фрейму містить фрейми першого (границя) і нульового (центр фігури) рівнів.

Опис фреймів-примітивів здійснюється різними способами залежно від зручності роботи з ними. Наприклад, площину можна задати за допомогою координат трьох точок, що не лежать на одній прямій, або точок перетину з осями координат, або точкою і трьома напрямними косинусами тощо. З бібліотеки примітивів (плоский прямокутник, коло, еліпс, паралелепіпед, призма, конус, куля тощо) можна утворити і складніші об'єкти, використовуючи операції об'єднання, перетину та віднімання множин, а також спеціальні програми апроксимації цих об'єктів сукупністю примітивів.

Поширеними є три способи задавання робочої сцени робота:

- *перший* полягає в повному заповненні фреймів оператором (або інтелектуальною надбудовою робота) в режимі «меню», що доцільно тоді, коли параметри об'єктів і координати їх у робочому просторі відомо заздалегідь;
- *другий спосіб* зводиться до того, що, користуючись системами технічного зору, далекомірами, іншими інформаційно-сенсорними системами, можна здійснити своєрідну «зйомку» об'єктів сцени з подальшою їх апроксимацією за допомогою бібліотечних графічних примітивів і обчисленням необхідних параметрів;
- *третій спосіб* пов'язаний з апроксимацією об'єктів сцени або її утворенням завдяки безпосередньому маніпулюванню примітивами в просторі сцени під час моделювання середовища.

Проте можливий ще один спосіб маніпулювання примітивами в просторі сцени — *застосування природних просторових відношень*. При цьому для об'єктів простору вводяться відношення, що визначаються за допомогою нечітких термінів (див. підрозд. 3.4). Наприклад: «упритул», «дуже близько», «на відстані», «далеко», «якомога далі» — відношення відстані; «ззаду», «ззаду та праворуч», «спереду та праворуч», «спереду» тощо — відношення напрямку; «на», «всередині», «за», «перед» — відношення взаємного розташування.

Правила виведення дають змогу поповнювати просторові відношення, завдяки чому формування робочої сцени робота набуває вигляду природних просторових



відношень («поставити об'єкт  $A$  на площину  $B$  поруч від об'єкта  $C$ », див. підрозд. 2.1). При цьому мається на увазі, що сцена правильно й повно сприймається інформаційними засобами робота. Існують пакети програм, що реалізують цей підхід з урахуванням конкретної галузі їх застосування.

**Система знань про можливі операції робота в його робочому просторі.** Після визначення робочого середовища можна переходити до складання опису послідовності операцій робота. Здебільшого складні маніпуляційні операції можна звести до послідовності елементарних (типових) операцій, що, як правило, відповідає практиці. Тому система знань про операції робота має охоплювати опис типових операцій і правил їх виконання, а також правил виведення при формуванні складних операцій з елементарних.

Опис елементарної операції складається з опису робочого простору (сцени), об'єкта роботи, мети операції, а також відомостей про умови її виконання. Наприклад, операцію встановлення деякого об'єкта на горизонтальну площину можна задати таким фреймом:

<Встановити об'єкт  $A$  на площину  $B$ : сцена  $C$ , об'єкт  $A$ ; мета: грань  $A'$  об'єкта  $A$  на площині  $B$ ; умови: поруч з об'єктом  $C$ ; встановити без удару>.

Фрейм містить субфрейм сцени (площину  $B$  включно) об'єкта  $A$  в початковому положенні; мета та умови формуються з використанням відношень, введених при описі сцени. Мету можна визначити й ширше, наприклад, як збіг грані  $A'$  об'єкта  $A$  з лінією на площині  $B$  чи його характерної точки  $x_A$  з відповідною точкою  $x_B$  площини  $B$ . При цьому умови можна доповнити якісними ознаками виконання операції (наприклад, у вигляді нерівності  $\|x_A - x_B\| \leq \epsilon$ , де  $\epsilon$  — заданий параметр) або використати нечіткі оцінювання різниці  $\|x_A - x_B\|$  (точно або приблизно, не дуже точно, грубо тощо).

Виділяють чотири рівні типових операцій:

- *нульовий* — операції статичного типу, мета яких полягає в забезпеченні заданого положення та орієнтації об'єкта на робочій сцені;
- *перший* — такі самі операції з урахуванням часових обмежень на їх виконання;
- *другий* — операції кінематичного типу, якщо в умови, що визначають мету операції, входять швидкість та прискорення;
- *третій* — операції динамічного типу, якщо умови включають сили та моменти, що зумовлені інерцією руху чи взаємодією з навколишнім середовищем.

Отже, додаючи кінематичні обмеження до наведеного вище прикладу операції встановлення об'єкта на площині, дістаємо опис **операції другого рівня**, що містить як субфрейм і опис **операції нульового рівня**:

<Установити об'єкт  $A$  на площину  $B$  з кінематичними обмеженнями: встановити об'єкт  $A$  на площину  $B$ ; час виконання операції не більший від  $T$ , швидкість перенесення об'єкта не більша від  $V$ , прискорення його розгону/гальмування не більше від  $W$ >.

Наведемо ще один приклад, який стосується операції руху інструмента по заданому контуру, що є **елементарною операцією другого рівня**, оскільки потребує задавання кінематичних параметрів руху вздовж лінії, а саме:

<Переміщення інструмента  $A$  по контуру  $L$ : сцена  $C$ ; інструмент  $A$ ; контур  $L$ ; мета: переміщення характерної точки  $A'$  інструмента  $A$  по контуру  $L$  у заданому напрямку; умови: швидкість руху не більша від  $V^*$  ..., не менша від  $V_*$ ; кутлова швидкість не більша від  $\omega^*$  (збереження орієнтації); похибка відстежування не більша від  $\epsilon$ >.

Оцінювання максимальних швидкостей  $V^*$ ,  $w^*$ , як і похибки  $\epsilon$  у цьому описі, можуть бути замінені нечіткими поняттями, якщо це не пов'язано з особливостями технологічного процесу (наприклад, дугове зварювання накладає жорсткі умови на швидкість руху інструмента, оскільки від цього руху залежить якість зварюваного шва).

Операція встановлення об'єкта на поверхню за умови «сила удару в момент встановлення об'єкта має бути не більшою від  $F$ » є **операцією третього рівня**. Прикладом контурної операції такого типу є обробка поверхні, коли інструменту треба рухатись по заданому контуру, забезпечуючи водночас задану силу різання.

Після складання опису елементарних операцій можна перейти і до опису складних як послідовності перших. Наприклад, операція «шліфування поверхні» містить такі елементарні операції: встановлення інструмента на маніпулятор (перший рівень); перенесення робочого інструмента в зону оброблюваної поверхні (перший рівень); обробка поверхні (третій рівень); повернення інструмента до магазину інструментів (другий рівень, оскільки виконувана операція потребує обмеження швидкості руху маніпулятора при підході до магазину інструментів).

Після того як операції описано, а їх послідовність задано, система перевіряє умови їх можливого виконання. Ці умови поділяються на дві групи:

- такі, що визначаються відношеннями між об'єктами на сцені (належність об'єкта до переміщуваних, а не до рельєфу сцени; можливість захоплення та утримання об'єкта; доступність його поверхонь для захоплення; відсутність інших об'єктів на захоплюваному об'єкті; незайнятість місця достатньої площини для встановлення об'єкта; гарантія того, що нове положення об'єкта на площині буде стійким тощо);
- такі, що зумовлені можливостями робота (об'єкт у початковому та вихідному положеннях перебуває в робочій зоні маніпулятора; за необхідності може бути забезпечена кінцева орієнтація об'єкта; конструкція захватного пристрою дає змогу захопити і надійно утримувати об'єкт у процесі роботи; робочий простір сцени є вільним у робочій зоні маніпулятора хоча б на той час, коли виконуватиметься операція з об'єктом, тощо).

Ці умови забезпечують виконання операцій *нульового рівня* і можуть ускладнюватися при збільшенні рівня операції. Наприклад, для операції *першого рівня* перевіряється можливість виконання її в заданий строк з урахуванням продуктивності робота; для операцій *другого рівня* з'ясовується, чи може робот розвинути в заданій ділянці робочої зони необхідні швидкості та прискорення; для операції *третього рівня* встановлюється, чи достатньою є миттєва потужність робота і чи можуть бути розвинені необхідні сили та моменти. Аналіз таких умов необхідно проводити заздалегідь і він має бути закладений у систему як БЗ про можливість робота.

Надалі, якщо умови реалізації операції виконуються, завдання надходить до блока алгоритмів керування та планування рухів робота, де можуть виникати свої труднощі, наприклад, при плануванні його рухів в обмеженому просторі. При цьому система знань про операції робота доповнюється даними про можливі перешкоди в його робочій зоні.

## 10.5. Складні технічні системи з агрегатними станами як об'єкти керування в реальному часі

**Загальна характеристика нештатних ситуацій та виникнення нових агрегатних станів ГКІС.** Постійне ускладнення технологічного устаткування та зв'язків між елементами складних технічних систем, зокрема тих, які працюють у паливно-енергетичних, аерокосмічних та інших галузях промисловості, спричинює підвищення ймовірності виникнення непередбачених, зокрема й надзвичайних, ситуацій, що позначається на навколишньому середовищі, призводить до збільшення матеріальних та енергетичних витрат.

У зв'язку з цим великого значення набувають методи та моделі формального подання можливих станів складних об'єктів керування (ОК), у тому числі й у процесі прогнозування, та формування дій, спрямованих на відтворення функціонування ОК у разі виникнення непередбачених обставин. В існуючих виробництвах такі функції, як правило, виконуються вручну, що призводить до втрати продуктивності системи та зниження інших характеристик функціонування виробництв.

Складні технічні системи, зокрема ГКІС, належать саме до таких ОК, які характеризуються високою ймовірністю виникнення нештатних ситуацій.

**Означення 10.5.** *Нештатною ситуацією* є така непередбачувана, зокрема надзвичайна ситуація, що виникає як внаслідок можливих відмов окремих компонентів системи, так і відхилень параметрів технологічних режимів під впливом різних факторів, а також ситуація, при якій невідомі принципи функціонування системи і невизначений алгоритм її керування.

Непередбачувані, у тому числі й надзвичайні ситуації у процесі функціонування ГКІС змінюють стан її компонентів та системи в цілому, призводять до *нештатного режиму* (НР) роботи, в результаті чого ця система переходить до нового *агрегатного стану* (АС), який не обов'язково адекватно відображається моделлю чи алгоритмом керування.

**Означення 10.6.** Під *агрегатним станом* ГКІС розуміють відображення її у дискретно-подійну систему, яке визначається множиною  $P$  подій  $p \in P$ , множинами  $X_{\text{int}_{\text{вх}}}$  та  $Y_{\text{int}_{\text{вих}}}$  атрибутів  $x \in X_{\text{int}_{\text{вх}}}$ ,  $y \in Y_{\text{int}_{\text{вих}}}$  вхідного і вихідного інтерфейсів відповідно, множиною  $G$  керуючих елементів  $g \in G$ , множиною  $Z$  внутрішніх станів  $z \in Z$ , а також випадковими операторами переходів  $I$  і виходів  $J$ , що реалізують функції  $z(p)$  і  $y(p)$  [12].

Процеси, що відбуваються в таких системах, являють собою сім'ю паралельно виконуваних послідовностей операцій, пов'язаних між собою причинно-наслідковими зв'язками, які визначаються технологічним регламентом і поточним станом ОК.

При моделюванні процесів у ГКІС широко використовуються розширення апарату сіток Петрі — логічні, нечіткі динамічні, імовірнісні, часові самонавантажені, ієрархічні та ін., які дають змогу у задачах аналізу/синтезу задавати (враховувати й прогнозувати) часовий взаємозв'язок між елементарними подіями (див. підрозд. 7 і 8).

Моделі, які будуються для дослідження таких систем і базуються на використанні відомих поширень апарату СП, можуть мати *одновекторний вимір*, коли послідовність чергування подій у «плані» являє собою планарну композицію процесу і свідчить про *просторово-подійну розподіленість* (ППР) ГКІС, або *двовекторний вимір*, тобто коли має місце ще й уточнення (деталізація) будь-якої з компонент

ГКІС шляхом заміни її групою підмоделей нижчого рівня чи, навпаки, абстракція, що свідчить про *просторово-ієрархічну розподіленість* (ПР) ГКІС.

Такі моделі відображають властивості об'єкта досліджень в *одному АС*.

Для моделювання і розв'язання задач синтезу/аналізу коректних алгоритмів керування процесами ГКІС з урахуванням непередбачуваних ситуацій (у тому числі й надзвичайних), в результаті яких система переходить до нових АС, доцільно розширити область подання об'єкта моделювання ще принаймні одним виміром — *простором АС*, що свідчить про *просторово-агрегатну розподіленість* (ПАР) ГКІС [79]. Тоді дослідження об'єктів з нештатними станами потребуватиме створення:

- моделей для кожного такого стану (будується простір моделей агрегатних станів);
- ідентифікаторів умов (інформаційних масивів), що спричиняють виникнення нештатних ситуацій;
- формалізмів логічного виведення щодо ідентифікації нештатних ситуацій в умовах імовірнісної інтерпретації нечіткої визначеності апріорної інформації про належність до певного стану;
- алгоритму синтезу оптимальної траєкторії в гіперпросторі станів об'єкта (тобто траєкторії переходу об'єкта з одного проміжного агрегатного стану до іншого за критеріями швидкодії або найменшої втрати основного показника функціонування об'єкта в зв'язку з переходом на керування у нештатному режимі).

**Класифікація нештатних режимів роботи ГКІС.** Першим кроком до створення моделей агрегатних станів ГКІС та ідентифікації умов їх виникнення є визначення суттєвих ознак, за якими відбуватиметься розпізнавання таких станів. Один з можливих варіантів класифікації нештатних режимів наведено на рис. 10.8. Не заглиблюючись у зміст самої класифікації, коротко зупинимося тільки на зв'язку між режимами функціонування, факторами, що їх викликають, а також наслідками для показників ГКІС у разі їх переходу в незаплановані агрегатні стани в умовах виникнення нештатних ситуацій.

Так, *глобальні НР*, спричинені надзвичайними обставинами, призводять до повної зупинки складових ГКІС на деякий час (наприклад, у разі відсутності живлення в мережі якоїсь технологічної дільниці, виходу з ладу устаткування в системах без дублювання тощо), тоді як *локалізовані НР* супроводжуються зниженням продуктивності внаслідок відмови деяких компонентів системи або програмного та інформаційного забезпечення, а також, наприклад, через виникнення непередбачуваних перешкод для пересування транспортних засобів.

Режими, що виникають безпосередньо в самих компонентах (устаткуванні) ГКІС, віднесемо до *внутрішніх НР* (це, наприклад, відмова управління в промисловому роботі (ПР) або у засобах чутливості орієнтуючих пристроїв тощо). Водночас збуджуючі фактори, що виникають у навколишньому стосовно певного компонента системи середовищі, є зовнішніми до нього. Тому з позиції саме цього інтегрованого до загального процесу функціонування компонента, режими, які виникають як результат нештатної ситуації в середовищі, є *зовнішніми НР*. Для ПР, скажімо, це сервісні умови його функціонування — наявність, клас, орієнтація і кондиційність виробу на робочій позиції.



Рис. 10.8. Класифікація нештатних режимів

Нарешті, саме недосконалість алгоритмів керування або обробки інформації, як і нехтування ними якихось характеристик можливих станів компонентів ГКІС, можуть спричинити програмні та інформаційні НР функціонування систем.

**Формалізоване подання гіперпростору АС системи з урахуванням штатних і нештатних режимів її функціонування.** Множина АС, у яких може перебувати ГКІС і яка у загальному випадку нескінченна, характеризує режими її функціонування.

**Означення 10.7.** Штатним режимом (ШР) роботи системи є такий режим, при якому відомо принципи функціонування системи і визначено алгоритм її керування.

**Означення 10.8.** Нештатним режимом (НР) роботи системи є такий режим, при якому невідомі принципи функціонування системи і не визначено алгоритм її керування.

Загалом НР роботи потребують впровадження методів прогнозування або планування серій експериментів для дослідження. У загальному випадку гіперпростір АС системи можна подати так:

$$\Pi_{AC} = (S_{AC1} \oplus S_{AC2} \oplus S_{AC3}) = (S_{ШС} \oplus S_{НШС}), \quad (10.1)$$

де  $\Pi_{AC}$  – гіперпростір АС системи;  $S_{AC1}$ ,  $S_{AC2}$ ,  $S_{AC3}$  – виміри можливих АС системи;  $S_{ШС}$ ,  $S_{НШС}$  – виміри штатних і нештатних АС системи відповідно.

У свою чергу, виміри  $S_{ШС}$  штатних АС системи визначаються як:

$$S_{ШС} = S_{ШС1} \times S_{ШС2} \times \dots \times S_{ШСi}, \quad (10.2)$$

а для нештатних  $S_{НШС}$  справедливим є співвідношення:

$$S_{НШС} = S_{НШС1} \times S_{НШС2} \times \dots \times S_{НШСi}. \quad (10.3)$$

**Означення 10.9.** Штатними АС системи є такі стани, для яких у системі визначено алгоритми керування.

**Означення 10.10.** Нештатними АС системи є такі стани, для яких у системі не визначено алгоритми керування.

Сукупність нештатних АС характеризують нештатні режими роботи системи (наприклад, на рис. 10.9 «траєкторія нештатного режиму»).

Якщо в процесі експлуатації або випробування систем на деякі нештатні АС буде розроблено алгоритм керування системою, то вони переходять у розряд штатних АС.

**Відображення процесу обробки надзвичайної ситуації в гіперпросторі АС системи.** Процеси, що відбуваються в ГКІС, можна подати як *рух* (або пересування) між АС системи. Сукупність АС, між якими відбувається рух системи, є *траєкторією* цього руху.

Тоді обробка цієї надзвичайної ситуації полягатиме в пошуку оптимальної траєкторії виведення системи з АС надзвичайної ситуації ( $S_{НС}$ ) через підмножину проміжних  $[S_{ПЦ}]$  до однієї з штатних ситуацій (у бажаному випадку – цільового  $S_{ПЦi}$ ) АС (рис. 10.9).

**Означення 10.11.** *Проміжними АС системи* є такі стани, в яких система ще може працювати з певними обмеженнями відповідно до штатного алгоритму функціонування, але які не є правильними станами і які займають перехідне положення на шляху від правильних до нештатних станів, і навпаки.

**Означення 10.12.** *Правильними* є стани ОК, що передбачені алгоритмами штатних режимів функціонування (тобто такі, в яких відсутні НС).

Послідовність  $S$  станів, між якими відбувається рух системи, можна подати у вигляді:

$$S = ([S_{ПРЕФ}], [S_{ОБР}]), \quad (10.4)$$

де  $[S_{ПРЕФ}] = [S_1, S_2, \dots]$  – префікс (рух системи між АС);  $[S_{ОБР}] = [S_{НСi}, \dots, [S_{ПР}], \dots, S_{ПЦj}]$  – послідовність обробки надзвичайної ситуації.

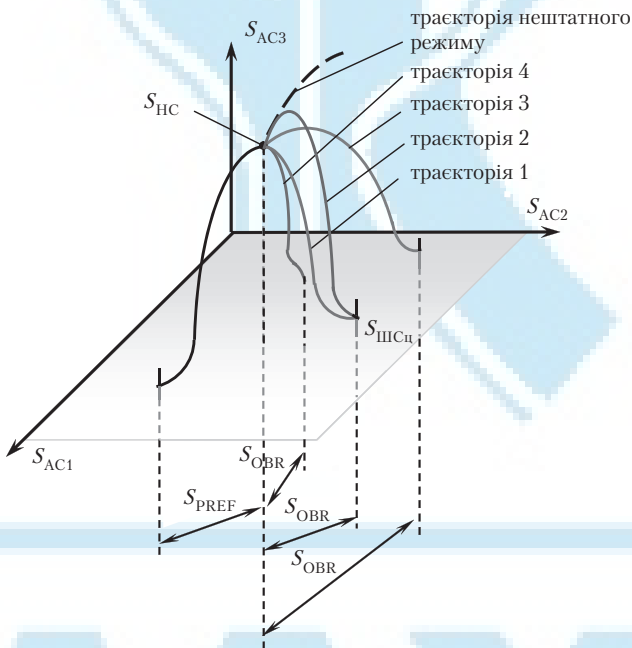


Рис. 10.9. Графічне подання процесу обробки надзвичайної ситуації в гіперпросторі агрегатних станів

Відповідно до рис. 10.9 «траєкторія 1» обробки надзвичайної ситуації належить до *штатного цільового без аномалій* режиму роботи системи, тобто такого, в якому досягається реалізація заданої цільової функції згідно з визначеним алгоритмом; «траєкторія 2» – до *штатного цільового з аномаліями* режиму роботи системи, тобто в якому досягається реалізація заданої цільової функції з відхиленнями від

визначеного алгоритму; «траєкторія 3» — до штатного, що не забезпечує виконання цільової функції, режиму роботи системи, тобто який характеризується неспроможністю реалізації заданої цільової функції; «траєкторія 4» — до штатного аварійного режиму роботи системи, тобто який характеризується наявністю відмов окремих її компонентів або системи в цілому, що не підлягають відновлюванню.

Таким чином, загальна схема обробки надзвичайних (у загальному випадку непередбачуваних) ситуацій полягає в пошуку базової траєкторії виходу у множині типових, попередньо сформованих «екземплярів».

**Імовірнісний характер визначення НР роботи.** Нехай  $S_A = \{S_1, S_2, \dots, S_p, \dots, S_n\}$  — множина класів станів системи, передбачених алгоритмами функціонування. Кожний клас  $S_i$  відповідає нечіткій множині станів з функцією належності, яка оцінює їх близькість до деякого канонічного (або характерного для цього класу) стану. Нехай також  $S^*$  — множина (у загальному випадку нескінченна) всіх станів системи. Тоді множина  $S_{НС}$  нештатних станів системи визначається як  $S_{НС} = S^* \setminus S_A$ .

В термінах апарату СП це визначення матиме вигляд: нехай  $\mu = (1, 2, \dots, n)$  — певне маркування СП;  $\mu_A = \{\mu_1, \mu_2, \dots, \mu_p, \dots, \mu_n\}$  — множина маркувань, що передбачена алгоритмом функціонування ГКІС;  $\mu^*$  — множина всіх можливих маркувань СП, що описує ГКІС. Тоді множина маркувань, яка відповідає множині  $S_{НС}$ , визначається як  $\mu_{НС} = \mu^* \setminus \mu_A$ .

Оскільки належність поточного стану ГКІС до канонічного з множини певного класу станів є імовірнісною функцією в умовах нечіткої логіки (див. розд. 3), коротко наведемо базові визначення та позначення теорії нечітких множин у термінах станів системи:  $\tilde{S} = \{s_i \mid \rho_{\tilde{S}}(s_i) \mid i = \overline{1, n}\}$  — нечітка підмножина звичайної множини  $S$  станів  $s_i \in S$  для  $i = \overline{1, n}$ ;  $\rho_{\tilde{S}}(s_i) \in [0, 1]$  — функція належності.

Для нечітких підмножин справедливими є операції: включення  $\tilde{S}_1 \subset \tilde{S}_2$ ; рівності  $\tilde{S}_1 = \tilde{S}_2$ ; доповнення  $\tilde{S}_1$ ; перетину  $\tilde{S}_1 \cap \tilde{S}_2$ ; об'єднання  $\tilde{S}_1 \cup \tilde{S}_2$ ; диз'юнктивної суми  $\tilde{S}_1 \oplus \tilde{S}_2$ ; алгебраїчної суми  $\tilde{S}_1 + \tilde{S}_2$ ;

Все наведене є справедливим для операцій над нечіткими підмножинами однієї і тієї самої звичайної множини. Проте, як зазначалося вище, постає проблема ідентифікації нечіткої визначеності поточного стану з імовірною належністю його до одного із загальної множини класів станів ГКІС. Тому в подальшій формалізації станів ГКІС в термінах нечітких множин необхідні додаткові формалізми.

Нехай  $S, A, B, C$  — звичайні множини, причому  $S = A \cup B \cup \dots \cup V$ ;  $A \subset S, B \subset S, \dots, V \subset S$ , та  $A_i \cap A_j = \emptyset$  для  $\forall i \neq j$ . Нечітка підмножина на  $S$  може бути подана в одному з таких виглядів:  $\tilde{S} = \{\tilde{A}, \tilde{B}, \dots, \tilde{V}\}$ ;  $\tilde{S} = \{s \mid \rho_{\tilde{S}}(s), \forall s \in S\}$ ;  $\tilde{S} = \{\tilde{A}, \{\rho_{\tilde{B}}(s), \forall s \in B\}, \tilde{C}, \dots, \tilde{V}\}$ . Якщо ж  $A \subset S$ , то вважатимемо, що  $A$  є фрагментом множини  $S$  (аналогічно для нечітких множин  $\tilde{A} \subset \tilde{B}$ ) і скористаємося відповідними позначками:  $S_A, \tilde{S}_A$ . Коли згадані вище операції над нечіткими множинами  $\tilde{A}$  і  $\tilde{B}$  (причому  $\tilde{A} \subset \tilde{B}, \tilde{B} \subset \tilde{C}, \tilde{A} \neq \tilde{B}$ ) зводяться до операцій над їх загальним фрагментом  $D$ , тоді позначимо це як:  $\tilde{A} \subset_D \tilde{B}, \tilde{A} =_D \tilde{B}, \tilde{A} \cup_D \tilde{B}, \tilde{A} \cap_D \tilde{B}, \tilde{A} -_D \tilde{B}, \tilde{A} \oplus_D \tilde{B}$ .

Тепер розглянемо звичайні множини  $A$  і  $B$ , на перетині яких визначено звичайне відношення  $Q$ . Нечітка підмножина  $\tilde{B}$ , яка індукована відображенням  $Q$  з нечіткої множини  $\tilde{A}$ , має вигляд:

$$\forall b \in \tilde{B}: \rho_{\tilde{B}}(b) = \begin{cases} \max[\rho_{\tilde{A}}(a)], & \text{якщо } \tilde{Q}^{-1}\{b\} \neq \emptyset; \\ 0, & \text{якщо } \tilde{Q}^{-1}\{b\} = \emptyset, \end{cases} \quad (10.5)$$

де  $a \in Q^{-1}(b)$ .

Відповідно до (10.5), нечітку підмножину  $\tilde{B}$ , індуковану відображенням з нечіткої множини  $\tilde{A}$ , позначимо як  $\tilde{B} = \tilde{Q}[\tilde{A}]$ .

Нехай тепер  $\tilde{A}$  – нечітка підмножина на  $A$ ,  $\tilde{Q}$  – відображення на  $A \times B$  та  $\tilde{B}_1$  – нечітка підмножина на  $B$ . Очевидно, що справедливими є усі операції над парою нечітких підмножин  $\tilde{Q}[\tilde{A}]$  та  $\tilde{B}_1$ .

Введемо поняття звичайної множини  $\alpha$ -рівня, найближчої до нечіткої. Нехай  $\tilde{S}$  – нечітка множина,  $S_\alpha$  – звичайна множина  $\alpha$ -рівня, найближчого до нечіткої  $\tilde{S}$ :

$$\forall s \in \tilde{S} : \rho_{\tilde{S}}(s) = \begin{cases} 0, & \text{якщо } \rho_{\tilde{S}}(s) < \alpha; \\ 1, & \text{якщо } \rho_{\tilde{S}}(s) \geq \alpha, \end{cases} \quad (10.6)$$

де  $\alpha \in [0, 1]$ .

За аналогією визначимо звичайне відношення  $\tilde{A}$ , найближче до нечіткої підмножини  $\tilde{A}$  відповідного  $\alpha$ -рівня:

$$\forall (s, a) \in \tilde{S} \times \tilde{A} : \rho_{\tilde{A}}(s, a) = \begin{cases} 0, & \text{якщо } \rho_{\tilde{A}}(s, a) < \alpha; \\ 1, & \text{якщо } \rho_{\tilde{A}}(s, a) \geq \alpha. \end{cases} \quad (10.7)$$

Нечітку множину, що індукована відношенням  $\tilde{A}$ , позначимо  $\tilde{A} = \tilde{A}[\tilde{S}]$ .

Отримані формалізми (10.5), (10.6) і (10.7) дають уявлення про ієрархічне «вкладення» за допомогою відповідних функцій  $\rho_{\tilde{S}}(s_i | i = 1, n) \in [0, 1]$  належності поточних станів ГКІС до відповідного з множини класу канонічних станів (тобто ідентифікувати поточний агрегатний стан ГКІС). Тепер дамо визначення агрегатної системи.

**Формальне подання АС системи.** Розвиненість можливостей подання особливостей ОК дає змогу створювати розширення СП і такі надлишкові моделі процесів управління об'єктами, в яких закладаються можливості прогнозування щодо утворення «шляхів» (ланцюгів) керуючих дій, не передбачених нормальною роботою устаткування та функціонуванням системи в цілому.

Проте значна розмірність моделі в цьому разі, наявність численних альтернативних варіантів управління та неповнота апріорної інформації надзвичайно ускладнюють проведення аналізу ситуацій, що виникають, і прийняття рішення стосовно корекції керуючих дій за таких умов.

Вихід із становища дають два підходи, на основі яких моделюються будь-які (ті, що виникають, і ті, які могли б виникнути за якихось обставин) ситуації в складній технічній системі, зокрема в ГКІС або окремій її досліджуваній компоненті.

**Подання АС системи на основі теорії агрегативних систем.** Зауважимо спочатку, що множина станів  $S$ , в яких може перебувати ГКІС з урахуванням НР їх функціонування, подібна до відповідної множини окремих ГКІС з одним станом з множини  $S$ . Отже, при **першому підході** складна технічна система у кожному з АС розглядається як окремий елемент – *агрегат* з відповідними певному стану властивостями (уніфікована схема, яка має динамічний характер, описує обмін сигналами із зовнішнім середовищем і враховує вплив випадкових факторів) *агрегативної* системи, або *А-системи* (клас складних систем, що відображає уявлення про взаємодію реальних об'єктів у рамках механізму обміну сигналами) [12; 79].

**Означення 10.13.** *Агрегатом* називається об'єкт керування (зокрема ГКІС або окрема її досліджувана компонента в одному з АС), який визначається сукупністю множин та операторів:

$$A = \{T, X, G, Y, S, B, H, D\}, \quad (10.8)$$



а саме множинами: моментів часу  $T = \{t_\beta | \beta = \overline{1, \rho}\}$  (фіксованою підмножиною множини дійсних чисел); вхідних сигналів  $X = \{x_j | j = \overline{1, m}\}$ ; керуючих сигналів  $G = \{g_\phi | \phi = \overline{1, k}\}$ ; вихідних сигналів  $Y = \{y_\chi | \chi = \overline{1, e}\}$ ; внутрішніх станів  $S = \{s_i | i = \overline{1, n}\}$ , причому стани, вхідні, вихідні та керуючі сигнали розглядаються як функції часу і їх значення в момент часу  $t$  визначаються як  $s(t)$ ,  $x(t)$ ,  $y(t)$ ,  $g(t)$ , відповідно, та простором параметрів  $B = \{b_r | r = \overline{1, l}\}$  агрегату (значення  $b$  фіксоване в рамках кожної конкретної задачі), а також операторами  $H$  переходів та  $D$  виходів, що реалізують функції  $s(t)$  та  $y(t)$ . Схему агрегату наведено на рис. 10.10.

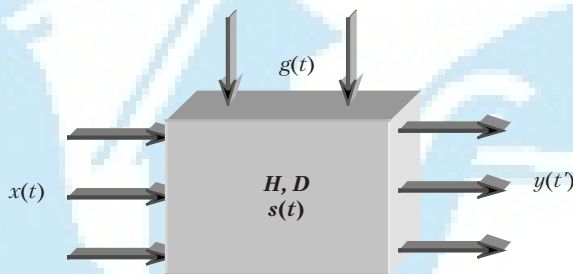


Рис. 10.10. Схеми агрегату

При цьому справедливі такі відображення:  $s(t): T \rightarrow S$  є рухом системи;  $x(t): T \rightarrow X$  – вхідним процесом;  $y(t): T \rightarrow Y$  – вихідним процесом; сукупність станів  $S$ , що відповідають усім  $t$  у цьому русі, і являє собою траєкторію цього руху.

Оператор  $D$  виходів подається сукупністю операторів  $D_1$  і  $D_2$ , які виробляють чергові моменти видавання непорожніх вихідних сигналів і їх зміст відповідно. Оскільки підмножина  $S_{g,b}^Y$  визначає момент  $t$  видавання вихідного сигналу  $y = D_2\{t, s(t), g(t), b\}$ , а оператор  $D_2$  у загальному випадку є випадковим, то даним  $t, s(t), g(t), b$  ставиться у відповідність декілька значень  $y$  з відповідним розподілом ймовірностей.

Функцією оператора  $D_1$  є визначення чергового моменту досягнення траєкторією  $s(t)$  підмножини  $S_{g,b}^Y$ , який є моментом видавання вихідного сигналу. Отже, вихідний сигнал  $y$  залежить від останнього керуючого сигналу  $g(t)$  безпосередньо через оператор  $D_2$  і множини  $S_{g,b}^Y$ . У свою чергу, від  $g(t)$  залежить стан агрегату, який є аргументом  $D_2$ . В цьому і полягає суть можливого способу керування агрегатом за допомогою керуючих сигналів, що надходять із зовнішнього середовища, в результаті чого агрегат переходить зі стану  $s(t)$  через оператор  $H$  переходів до іншого стану (наприклад, до стану  $s(t+0)$  за «малий» проміжок часу).

В деяких випадках зручно розглядати «розширений» стан  $s_p(t)$  у вигляді  $s_p(t) = (s(t), g(t)) \in S \times G$ . Тоді з урахуванням подання оператора  $H$  переходів у вигляді сукупності випадкових операторів  $U, U_1, U_2$  дістанемо такі відношення:

$$s_p(t'_n + 0) = U_1\{t'_n, s(t'_n), x'_n, b\}, g(t'_n) \text{ — для оператора } U_1;$$

$$s_p(t''_n + 0) = U_2\{t''_n, s(t''_n), b\}, g_n(t''_n) \text{ — для оператора } U_2;$$

$$s_p(t) = U\{t, t_n, s(t_n + 0), b\}, g(t_n) \text{ — для оператора } U,$$

де  $t'_n, t''_n$  – моменти надходження до агрегату вхідного  $x'_n$  і керуючого  $g_n(t''_n)$  сигналів відповідно;  $g_n(t''_n)$  – останній керуючий сигнал, що надійшов у момент  $t < t'_n$ ;

$t_n$  — момент одночасного надходження до агрегату і вхідного  $x_n$ , і вихідного керуючого  $g_n$  сигналів.

Отже, *означення 10.13* агрегату необхідно доповнити операторами переходів  $U, U_1, U_2$  до нового стану і підмножинами  $S^Y$ .

Таким чином, уведена уніфікована схема-агрегат формується зі стохастичної системи загального вигляду конкретизацією операторів переходів і виходів і тим самим позбавлена основного недоліку останньої (коли надмірні узагальнення неминуче призводять до збіднення змісту теорії — для стохастичних систем загального вигляду характерні тривіальні загальні властивості).

Однак при цьому підході зустрічаються труднощі, пов'язані, зокрема, з труднощістю та громіздкістю опису теоретико-множинних відношень агрегатів у просторово-часовій області з урахуванням численних причинно-наслідкових зв'язків між подіями, складністю синхронізації вхідних, керуючих і вихідних сигналів, які визначають взаємодію агрегатів у часі. Отже, йдеться про труднощі, які виникають при відображенні складної технічної системи в одному з АС, тоді як ідеться про подання всіх можливих її АС. При цьому ОК повинен розглядатися як сукупність агрегатів, або конструкції агрегатів.

Розглянемо клас складних систем, що уявляють собою деякі конструкції з агрегатів.

**Означення 10.14.** Системою є сукупність  $(A_i | i = \overline{1, N})$  скінченної кількості агрегатів разом із зв'язками між ними.

Крім опису елементів А-системи, необхідно мати ще й опис взаємодій між її елементами. Проте поведінка системи не може розглядатися окремо від зовнішнього середовища.

**Означення 10.15.** Зовнішнє середовище системи — це сукупність таких об'єктів, що не є елементами цієї системи, але взаємодія з якими враховується при моделюванні цієї системи.

**Припущення 10.1.** Взаємодія між системою та зовнішнім середовищем, а також між агрегатами усередині системи здійснюється тільки за допомогою обміну сигналами через канали зв'язку; взаємний вплив, що відбувається поза межами механізму обміну сигналами, не враховується.

**Припущення 10.2.** Для опису сигналу достатньо деякого скінченного набору характеристик, а саме  $X = X_1 \times X_2 \times \dots \times X_m$ ;  $G = G_1 \times G_2 \times \dots \times G_k$ ;  $Y = Y_1 \times Y_2 \times \dots \times Y_e$ , де  $X, G, Y$  — простори вхідних, керуючих і вихідних сигналів, причому  $X_j = \{x_j | j = \overline{1, m}\}$ ;  $G_\phi = \{g_\phi | \phi = \overline{1, k}\}$ ;  $Y_\chi = \{y_\chi | \chi = \overline{1, e}\}$  — елементарні виміри просторів  $X, G, Y$  відповідно.

**Припущення 10.3.** Зовнішнє середовище при моделюванні системи розглядається як фіктивний агрегат  $A_\phi$  з просторами вхідних  $X^{(\phi)} = \prod_x \{X_\lambda^{(\phi)} | \lambda = \overline{1, m}\}$  та вихідних  $Y^{(\phi)} = \prod_y \{Y_\sigma^{(\phi)} | \sigma = \overline{1, e}\}$  сигналів, де  $\prod_x, \prod_y$  — глобальні простори вхідних і вихідних сигналів фіктивного агрегату, взяті як прямі добутки просторів  $X_\lambda^{(\phi)}, Y_\sigma^{(\phi)}$  відповідно.

Сигнали, що видаються агрегатами системи в зовнішнє середовище, отримуються фіктивним агрегатом  $A_\phi$  як вхідні сигнали; сигнали, що отримуються агрегатами системи із зовнішнього середовища, є вихідними сигналами  $A_\phi$ .

**Означення 10.16.** Система — це впорядкована сукупність агрегатів  $A_1, A_2, \dots, A_N$ , множин фіктивних каналів  $[X_j^{(\phi)}]_\chi^{m+k}$  та  $[Y_\chi^{(\phi)}]_\chi^r$ , що характеризують зовнішнє сере-

довище, і оператора  $R$ , що визначає модель сполучення агрегатів у системі через елементарні канали.

**Означення 10.17.** Агрегативною (або  $A$ -системою) є така система, в якій оператор  $R$  реалізує відображення  $\cup [X_j^{(\xi)}]_x^{m+k} \rightarrow \cup [Y_\chi^{(i)}]_x^r$  і виконуються умови:

- I. Для будь-яких  $X_j^{(\phi)} \in [X_j^{(\phi)}]_x^m$  та  $Y_\chi^{(\phi)} \in [Y_\chi^{(\phi)}]_x^r$  в цій системі  $Y_\chi^{(\phi)} \neq R(X_j^{(\phi)})$ .
- II. Якщо  $Y_\chi^{(e)} = R(X_j^{(i)})$ , тоді:
  - 1)  $Y_\chi^{(e)} \subset X_j^{(i)}$ , де  $X_j^{(i)}$  та  $Y_\chi^{(e)}$  – відповідні множини елементарних сигналів;
  - 2) для будь-якого моменту  $t'$  видавання непорожнього елементарного сигналу  $y_\chi^{(e)}(t') \in Y_\chi^{(e)}$  справедливе: а)  $t' \in (T^{(e)} \cap T^{(i)})$ ; б)  $y_\chi^{(e)}(t') = x_j^{(i)}(t')$ , де  $x_j^{(i)}(t') \in X_j^{(i)}$ .

Для адекватності  $A$ -системи реальній ГКІС необхідно і засоби передачі сигналів подати у вигляді самостійних агрегатів, зв'язки між якими задовольняють наведені вище умови.

Загалом КІС може складатися з довільної кількості компонентів з агрегатними станами, і тоді власне систему можна зобразити наведеним на рис. 10.11 прикладом.

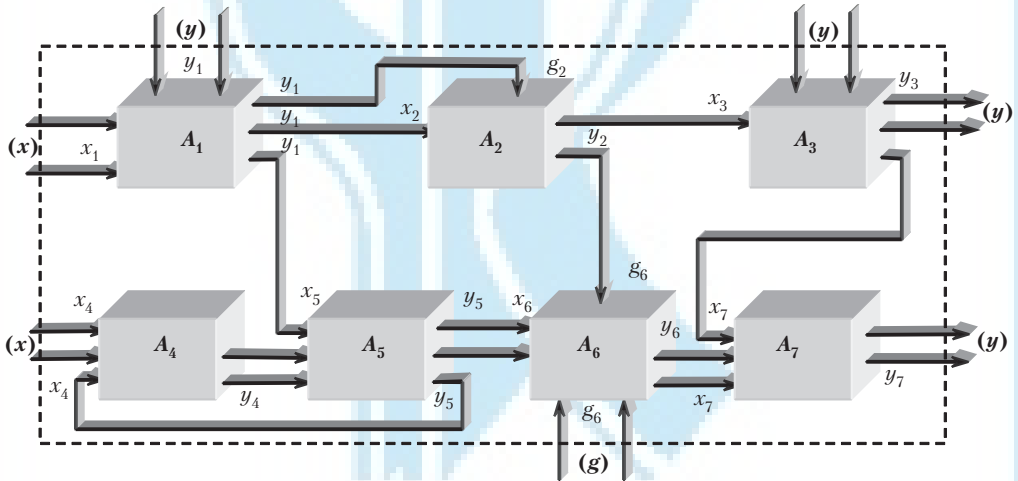


Рис. 10.11. Схема агрегатної системи

**Подання процесу управління системою у просторі АС на основі надлишкового абстрактного графа.** Вихід із становища дає **другий підхід**, запропонований у [130] і спрямований на поєднання переваг СП та агрегативних систем. Цей підхід базується на новій модифікації СП у вигляді **надлишкового абстрактного графа** (НАГ), на основі якого не тільки моделюються будь-які (ті, що виникають, і ті, які могли б виникнути за якихось обставин) ситуації в системі, а й формується логічне виведення щодо дій, спрямованих на усунення або коригування останніх, і вибирається ефективний шлях (алгоритм) управління системою в нештатному режимі її роботи у просторі можливих АС, заданих у просторово-часовій області [125].

**Означення 10.18.** Надлишковий абстрактний граф – дводольний орієнтований граф  $\Phi(\bar{P}, \bar{T}, \bar{S})$  з нескінченною кількістю дуг  $\bar{S}_i$  і вершин  $\bar{P}, \bar{T}$ , визначений у просторово-часовій області координатами вершин  $\bar{P}$ , заданих сім'єю неперетинних

підмножин  $P^{(k)}$ ,  $h \in H$ , кожна з яких відповідає певній точці просторової осі, причому

$$\hat{P} = \cup p^{(k)}, \hat{T} = \bigcup_{i \in I} t^{(i)}, \hat{P} \cap T = \emptyset, \bigcap_{h \in H} p^{(h)} = \emptyset, \bigcap_{i \in I} t^{(i)} = \emptyset.$$

Дуги графа визначають відношення між підмножинами  $P^{(h)} = \{p_{\delta}^{(h)}\}$ ,  $\delta \in \tau$ ,  $t^{(i)} = \{t_{\delta}^{(i)}\}$  у формі функції інцидентності

$$\hat{F} : (H \times I) \cup (I \times H) \rightarrow \{\emptyset, 1\}, \quad (10.9)$$

де одиниця є ознакою наявності нескінченної множини дуг, які з'єднують відповідні елементи підмножин  $p^{(k)}$ ,  $t^{(i)}$ .

Нехай  $p_{\delta}^{(h)}$  визначає умову типу  $h$ , виконану на момент часу  $\delta$  на відрізку часової осі  $\tau$  та подану в просторово-часовій області точкою з координатами  $(h, \tau)$ , а  $t_{\delta}^{(i)}$  — операція управління типу  $i$ , виконання якої розпочато з моменту часу  $\delta$ .

Тоді будь-який конкретний процес управління технічною системою в нештатному режимі її роботи може бути поданий у просторово-часовій області дводольним підграфом  $\Phi_{ек}(P_{ек}, T_{ек}, S_{ек})$  графа  $\Phi(\hat{P}, \hat{T}, \hat{S})$ , де  $P_{ек} \subset \hat{P}$ ,  $T_{ек} \subset \hat{T}$ ,  $S_{ек} \subset \hat{S}$ , причому в будь-який момент часу може існувати не більш як одна вершина підграфа з кожної підмножини  $p^{(k)}$ ,  $t^{(i)}$ . Відношення між НАГ і підграфом  $\Phi_{ек}$  у викладеній інтерпретації є аналогічним щодо викладених вище понять випадкового процесу та окремої його реалізації.

Наведені поняття та інтерпретація НАГ дають змогу сформулювати основні означення, які є основою формального апарату опису нештатних (екстремальних) режимів роботи складних організаційно-технічних систем, зокрема ГКІС.

**Твердження 10.3.** Поточний стан процесу управління в екстремальному режимі на момент часу  $\tau_j \in \tau$  визначається множиною умов  $p_{\tau_j} \in \hat{P}$ , виконаних до моменту часу  $\tau_j$ , та множиною керуючих операцій  $t_{\tau_j} \in \hat{T}$ , виконуваних на момент часу  $\tau_j$ .

Наведене твердження дає змогу сформулювати поняття початкової ситуації, яка спричинює екстремальний режим, у вигляді відповідної підмножини умов  $p^{(поч)} \subset \hat{P}$  та поняття цільової ситуації  $S_{ШСЦ}$  (див. рис. 10.9), яка завершує екстремальний режим роботи системи і визначає мету управління в цьому режимі у вигляді підмножини умов  $p^{(м)} \subset \hat{P}$ .

Поняття екстремального режиму в запропонованих термінах задається множиною

$$\{p_{ек}^{(поч)}, \{p_{\tau_j}\}, \tau_j \in \tau_{ек}, p_{ек}^{(м)}\}, \quad (10.10)$$

де індекс «ек» означає відповідність (еквівалентність) цього режиму підграфу  $\Phi_{ек}(P_{ек}, T_{ек}, S_{ек})$ .

Слід додати, що тим самим початковій  $p_{ек}^{(поч)}$  та цільовій  $p_{ек}^{(м)}$  ситуаціям може відповідати ряд альтернативних процесів управління (10.10) і підграфів  $\Phi_{ек}$  залежно від конкретних умов, збуджуючих дій з боку середовища та стану ОК в часовому інтервалі  $\tau_{ек}$  його роботи в екстремальному режимі.

У [130] запроваджено поняття ініціюючої умови та ініціюючої підмножини умов, що спричинюють необхідність використання підграфа  $\Phi_{ек}$ , а отже, й визначення алгоритму його побудови для конкретного процесу управління.

**Означення 10.19.** Ініціюючою підмножиною операції називається підмножина умов  $P_y^{(i)} \in \hat{P}$ , виконання яких є необхідним і достатнім для здійснення цієї операції  $t_{\delta}^{(i)} \in t^{(i)}$ .

Відповідно до функції інцидентності (10.9), для кожної з операцій  $t_{\delta}^{(i)} \in \widehat{T}$  можна визначити сім'ю (в загальному випадку перетинних) ініціюючих підмножин умов  $\{p_y^{(i)}\}$ ,  $y \in \theta_y$ . Об'єднання цих підмножин утворює множину  $\text{вх } p^{(i)} = \bigcup_{y \in \theta_y} p_y^{(i)}$  вхідних умов операції  $t_{\delta}^{(i)}$ .

**Твердження 10.4.** Умова  $p_{\theta}^{(h)}$  є ініціюючою щодо операції  $t_{\delta}^{(i)} \in t^{(i)}$ , якщо виконується умова

$$(p_{\theta}^{(h)} \in \text{вх } p^{(i)}) \cap (\delta = \theta). \quad (10.11)$$

Отже, як випливає з (10.11), ініціююча умова завжди є останньою задовольняючою умовою в ініціюючій підмножині.

Як результат виконання керуючої операції  $t_{\delta}^{(i)} \in t^{(i)}$  задовольняються підмножини  $\text{вих } p^{(i)}$  вихідних умов цієї операції, що відображають реакцію на операцію  $t_{\delta}^{(i)}$  і, у свою чергу, можуть, відповідно до функції інцидентності (10.9), бути вхідними умовами щодо деякої іншої операції.

**Алгоритм управління організаційно-технічною системою в нештатному режимі її роботи.** Алгоритм генерування підграфа  $\Phi_{\text{ек}}$  складається з таких етапів:

1. Пошук ініціюючої умови  $p_{\theta}^{(h)}$ .
2. При  $\tau = \theta$  запуск операції  $t_{\theta}^{(i)}$ , що визначається відношенням  $p_{\theta}^{(h)} \in \text{вх } p^{(i)}$  з функцією інцидентності (10.9).
3. При  $\tau = \theta + \delta_i$  формування в просторово-часовій області вершин підграфа  $\Phi_{\text{ек}}$ , які відповідають  $\text{вих } p^{(i)}$ , де  $\delta_i$  – час виконання операції  $t_{\theta}^{(i)}$ .
4. Перевірка виконання рівності  $p_{\tau} = p^{(i)}$  та перехід до першого етапу, якщо рівність не виконується.

Наведений алгоритм схожий з алгоритмом імітаційного моделювання, оскільки має виражений подійний характер. Проте є й ряд принципових відмінностей, а саме:

- генерування графа  $\Phi_{\text{ек}}$  здійснюється в *реальному часі* роботи керованої технічної системи;
- операції реалізуються у вигляді керуючих дій *безпосередньо на технологічні об'єкти*;
- задоволення умов  $p_{\tau}$  визначається не умовним станом ОК, а його *реальним станом*, визначеним за допомогою засобів збирання інформації у вимірвальних каналах;
- конкретна реалізація підграфа  $\Phi_{\text{ек}}$  залежить від збурення з боку навколишнього середовища, про яке *повної апріорної інформації немає*.

Інакше кажучи, наведений процес управління, з одного боку, імітується графом, який визначає умови запуску керуючих операцій (дій), а з іншого – реалізує принцип зворотного зв'язку завдяки визначенню реакцій на керуючі операції у вигляді множини вихідних умов  $\text{вих } p^{(i)}$ .

Для ілюстрації наведеної моделі управління об'єктами розглянемо рис. 10.12, а, на якому зображено фрагмент НАГ у вигляді графа  $\Phi(\widehat{P}, \widehat{T}, \widehat{S})$ , де  $\widehat{P} = \{p^{(h+k)}\}$ ,  $k = \overline{0, 4}$ ;  $\widehat{T} = \{t^{(i+n)}\}$ ,  $n = \overline{0, 1}$ .

На рис. 10.12, б у просторово-часовій області показано відповідний фрагмент імітаційно-керуючої моделі у формі графа  $\Phi_{\text{ік.м}}(P_{\text{ік.м}}, T_{\text{ік.м}}, S_{\text{ік.м}})$ , де  $P_{\text{ік.м}} = \{p_{\tau_1}^{(h+4)}, p_{\tau_2}^{(h+3)}, p_{\tau_3}^{(h+1)}, p_{\tau_4}^{(h+2)}, p_{\tau_6}^{(h)}, p_{\tau_5}^{-(h+3)}\}$ ;  $T_{\text{ік.м}} = \{t_{\tau_2}^{(i+1)}, t_{\tau_4}^{(i)}\}$ .

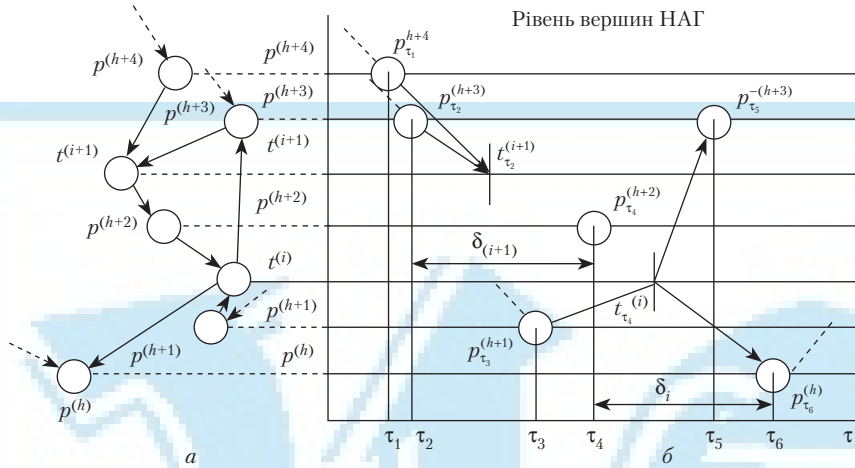


Рис. 10.12. Фрагмент НАГ процесу управління об'єктом (а) та його просторово-часова інтерпретація імітаційно-керуючою моделлю (б)

Як бачимо, у просторово-часовій області вершини НАГ мають вигляд горизонтальних прямих; ініціюючі підмножини  $p_{y=1}^{(i+1)}, p_{y=1}^{(i)}$  операцій  $t_{\tau_2}^{(i+1)} \in t^{(i+1)}$  і  $t_{\tau_4}^{(i)} \in t^{(i)}$  в цьому прикладі збігаються з множинами вхідних умов  $^{вх}p^{(i+1)}, ^{вх}p^{(i)}$  цих операцій і мають вигляд:

$$p_{y=1}^{(i+1)} = ^{вх}p^{(i+1)} = \{p_{\tau_1}^{(h+4)}, p_{\tau_2}^{(h+3)}\};$$

$$p_{y=1}^{(i)} = ^{вх}p^{(i)} = \{p_{\tau_3}^{(h+1)}, p_{\tau_4}^{(h+2)}\}.$$

Ініціюючими умовами для операцій  $t_{\tau_2}^{(i+1)}$  та  $t_{\tau_4}^{(i)}$  є відповідно  $t_{\tau_2}^{(h+3)}$  та  $t_{\tau_1}^{(h+4)}$ . Отже, аналізуючи наведене вище, можна дійти висновку:

**Твердження 10.5.** Граф екстремальних ситуацій є орієнтованим графом-деревом, якщо всі його вершини відповідають тільки початковим ситуаціям, цільовим ситуаціям або підмножинам станів і ситуацій, які включають не більш як один проміжний стан процесу управління об'єктом в екстремальному режимі, поданому у формі (10.10).

При цьому справедливим є таке твердження:

**Твердження 10.6.** Множини вершин орієнтованого графа екстремальних режимів роботи ОК, які відповідають підмножинам з цільовими  $p^{(M)}$  та проміжними (поточними)  $p_{\tau}$  станами, є порожніми.

Таке твердження дає змогу виключити з розгляду альтернативні варіанти управління, що не реалізуються, і, таким чином, набагато скоротити розмірність графа  $\Phi_{ек}$ , що генерується НАГ-моделлю.

Описаний вище підхід дає змогу сформулювати початкову задачу оптимізації управління складним організаційно-технічним об'єктом в екстремальних (нештатних) режимах його роботи у вигляді пошуку на графі можливих процесів управління  $\Phi(\bar{P}, \bar{T}, \bar{S})$  такого підграфа  $\Phi_{ек}(P_{ек}, T_{ек}, S_{ек})$ , який відповідає оптимальному значенню заданого критерію  $\Psi$ :

$$\Phi_{ек}(P_{ек}, T_{ек}, S_{ек}) = \arg \left( \underset{\Phi_{ек,ре} F}{\text{opt}} \Psi, (\Phi_{ек,р}(P_{ек,р}, T_{ек,р}, S_{ек,р})) \right), \quad (10.12)$$

де індекс «ек. р» означає екстремальний, що реалізується;  $F$  – множина всіх можливих послідовностей управління.

Як критерії ефективності управління в нештатному режимі роботи системи можуть, зокрема, застосовуватися мінімум часу її роботи в цьому режимі або мінімум матеріальних збитків і витрат енергії. Труднощі реалізації підграфа (10.12) полягають у тому, що граф можливих процесів управління  $\Phi(\hat{P}, \hat{T}, \hat{S})$  і множину  $F$  усіх можливих послідовностей керуючих операцій  $T_{\text{ек}} \subset \hat{T}$  для фактичних умов роботи системи в часовому інтервалі  $\tau$  визначити заздалегідь неможливо, а значна кількість керуючих операцій  $\hat{T}$  та неповна апріорна інформація про граф  $\Phi(\hat{P}, \hat{T}, \hat{S})$  виключають використання методів багатоетапного стохастичного програмування.

**Принципи розв'язання задачі оптимізації управління в нештатних режимах.** У [130] сформульовано ряд принципів щодо підходу до розв'язання цієї проблеми із застосуванням викладеного вище алгоритму генерування графа  $\Phi_{\text{ек}}$ .

**Принцип зворотного зв'язку** передбачає оцінку поточного стану  $p_\tau$  як результат реалізації кожної з керуючих операцій  $t_\delta^{(i)} \in \hat{T}$ . Цей принцип дає змогу діставати в реальному часі необхідну інформацію з метою визначення компонентів графа  $\Phi(\hat{P}, \hat{T}, \hat{S})$  або множини  $F$ .

**Принцип максимального розпаралелювання** визначає необхідність у будь-який поточний момент часу  $\delta = \tau^*$  виконання всіх керуючих операцій  $t_\delta^{(i)}, i \in I_\delta$ , для яких задоволено ініціюючі підмножини умов  $p_y^{(i)}, i \in I_\delta$ . Реалізація цього принципу сприяє мінімізації часу виходу організаційно-технічної системи з екстремального (нештатного) режиму її роботи.

**Принцип оптимізації керуючих операцій**  $\{t_\delta^{(i)}\}, \delta = \tau, i \in I$  передбачає визначення оптимальних керуючих дій при реалізації кожної операції з урахуванням поточної інформації про збудження та стани процесу управління  $\{p_\tau, t_\tau\}, \tau \in \tau^*$  з метою зниження втрат та скорочення шляху досягнення цільової ситуації  $p^{(M)}$  на НАГ. Реалізація цього принципу пов'язана з розв'язанням послідовності оптимізаційних задач за критеріями, узгодженими із загальними критеріями управління в задачі (10.12).

**Принцип інтелектуального моделювання** (імітаційного моделювання на основі БЗ) передбачає генерування імітаційно-керуючої моделі у формі графа  $\Phi_{\text{ек}}$  за описаним вище алгоритмом з використанням елементів БЗ у формі правил продукції, що відповідають фрагментам НАГ:

$$(p_1^i \cup p_2^i \cup \dots \cup p_y^i \cup \dots \cup p_{y_i}^i) \rightarrow \{t_\delta^{(i)}, p^{(i)}\}, i \in I. \quad (10.13)$$

Кожне з правил (10.13) подає деякий елемент функції інцидентності (10.9) і, таким чином, відповідний підграф графа  $\Phi(\hat{P}, \hat{T}, \hat{S})$ . Отже, *принцип інтелектуального моделювання дає змогу звести процес генерування моделі  $\Phi_{\text{ек}}$  до процесу логічного виведення в ЕС на основі БЗ.*

## 10.6 Підходи до розв'язання невизначеності в умовах функціонування складних ОК в реальному часі

Існує дві альтернативи розв'язання невизначеності при функціонуванні ОК в реальному часі: *перша* полягає в створенні умов, за яких невизначені ситуації переводяться у визначені умови функціонування ОК; *друга* базується на застосуванні принципів нейро-фаззи-технологій, в результаті чого відбувається зведення нечіткостей до чітких виведень, які обговорювалися в розд. 4.

Нижче розглянемо два приклади, які базуються на використанні першої і другої альтернатив.

**Приклад 10.1. Переведення невизначених ситуацій у визначені умови функціонування ГКІС.** Найбільшого поширення при багатосерійному і масовому характері виробництва набуло селективне складання, яке забезпечує потрібні характеристики спряження.

Для найпоширенішої (узагальненої для будь-яких форм з'єднаних елементів) пари, якою є з'єднання валу із втулкою, у виробничих умовах технологія виготовлення отворів (*A*) передбачає наявність максимального діапазону розсіяння похибок (що описуються нормальною кривою розподілу)  $6\sigma_A$  (рис. 10.13, *a*), яке формується на *первинних* (грубих) операціях (наприклад, свердління, розточування тощо). У подальшому виправляється лише якість поверхні отвору на *викінчувальних* (наприклад, хонінгування, шліфування) операціях.

Технологія виготовлення валів (*B*) передбачає розсіяння похибок (що також описуються нормальною кривою розподілу)  $6\sigma_B$  (рис. 10.13, *a*), яке формується на *фінішних довідних* операціях (наприклад, безцентрового шліфування), тобто в дуже вузькому діапазоні  $6\sigma_B$ , причому  $\sigma_B \ll \sigma_A$ .

Таким чином, *управління підбором складальних комплектів* здійснюється за рахунків валів.

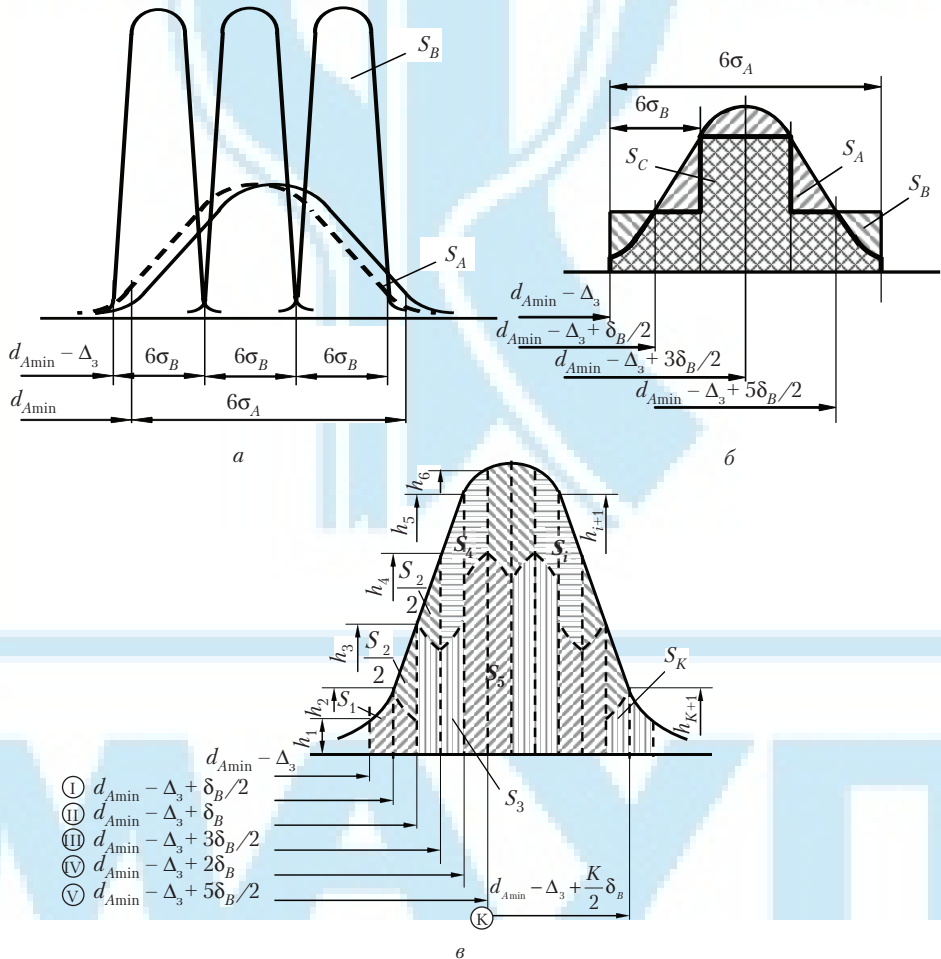


Рис. 10.13. Процес суміщення кривих розподілу розмірів спряжуваних елементів



На сьогодні в сучасних умовах складального виробництва процес зміщення має неорганізований характер, при якому необхідна кількість валів у кожній із допускових груп зміщення *не визначається* (кількість груп зміщення визначається відношенням  $\delta_A / \delta_B$  або  $6\sigma_A / 6\sigma_B$ ). У зв'язку з цим кількість деталей, що утворюють придатні з'єднання, обмежена. Це призводить до збільшення обсягу незакінченого виробництва і знижує ймовірність підбору пари пристроями автоматичного пошуку.

У тих же випадках, коли кількість валів у кожній із груп зміщення визначено, забезпечити повне суміщення кривих розподілу не вдається внаслідок несиметричності площі під ділянками кривої розподілу розмірів отворів під вал у кожній із груп зміщення (рис. 10.13, б).

Як видно з рис. 10.13, б, кількість деталей, які утворюють придатні з'єднання у цьому випадку, обмежується площею  $S_C$  при накладанні  $S_A$  і  $S_B$ .

Зазначену невизначеність у процесі селективного підбору складальних комплектів можна усунути зміною в технології управління процесом преселективної обробки валів.

Можна запропонувати підхід, який зумовлює рівність полів розсіяння розмірів для однакових партій спряжуваних деталей, що досягається вимушеним збільшенням поля розсіяння розмірів при виготовленні валу від його мінімального значення  $\delta_B$ , яке зумовлене довідною операцією, до максимального  $\delta_A$ , економічно доцільного для отворів, шляхом штучного суміщення поля розсіяння розмірів валу (рис. 10.13, а).

При настроюванні довідного безцентрово-шліфувального верстата (зокрема при фінішній операції обробки валів) необхідно забезпечувати повне суміщення кривих розподілу для однакових партій спряжуваних деталей. Пропонується зміщення настроювання здійснювати не на величину  $\delta_B$  розсіяння розмірів валу, а на величину  $\delta_B/2$  (рис. 10.13, в). Тобто кількість переналаджувань довідного верстата дорівнює  $6\sigma_A/3\sigma_B$ .

Для визначення кількості оброблюваних валів у кожній із  $K$  груп зміщення настроювання необхідно розрахувати площу під кривою Гауса із середнім квадратичним відхиленням  $6\sigma_A$ , яка відповідно матиме вигляд:

$$S = \frac{1}{\sigma\sqrt{2\pi}} \int_{d_{Amin}}^{d_{Amax}} e^{-\frac{z^2}{2\sigma^2}} dz. \quad (10.14)$$

Такий розрахунок є точнішим, проте потребує обчислень інтегралу. Якщо спростити розрахунок до допустимих меж міжопераційного заділу виробів, то можна скористатись апроксимованою кривою через ламану, якою подано розподіли на рис. 10.13, в. При цьому елементарні площі  $S_i$  приймаються обмеженими прямолінійними трапеціями з висотою  $\delta_B/2$  та основами  $h_i$  та  $h_{i+1}$ .

Тоді кількість валів, які обробляються в кожній з груп зміщення настроювання безцентрово-шліфувального верстата, розраховують за такими залежностями [19]:

- для першого настроювання  $d_{Amin} - \Delta_3 + \delta_B/2$  обробляється кількість валів під площею

$$S_1 = (h_1 + h_2) \delta_B/2, \quad (10.15)$$

і в цій групі настроювання залишається незаповненою площа  $S_1/2$ ;

- для другого настроювання  $d_{Amin} - \Delta_3 + \delta_B$  обробляється партія валів, визначена площею

$$S_2 = \left\{ \left[ 2(h_3 + h_2)/2 \right] \delta_B/2 \right\} - S_1/2 = (h_3 - h_1) \delta_B/2, \quad (10.16)$$

і для цього зміщення настроювання незаповненою залишається площа  $S_3/2$ ;

- у третьому настроюванні  $d_{Amin} - \Delta_3 + 3\delta_B/2$  партія валів визначається площею

$$S_3 = \left\{ \left[ 2(h_4 + h_3)/2 \right] \delta_B/2 \right\} - S_2/2 = (h_4 + h_1) \delta_B/2, \quad (10.17)$$

причому залишається необробленою кількість валів під площею  $S_4/2$ ;

- для четвертого настроювання  $d_{Amin} - \Delta_3 + 2\delta_B$  обробляється кількість валів під площею

$$S_4 = \left( \left[ \frac{2(h_5 + h_4)}{2} \right] \delta_B / 2 \right) - S_3 / 2 = (h_5 - h_4) \delta_B / 2, \quad (10.18)$$

і в цьому зміщенні настроювання залишаються необробленими вали під площею  $S_5/2$ , і т. д.

Таким чином, для кожної з  $k$  груп зміщення настроювання кількість оброблюваних валів визначається площею

$$\left. \begin{aligned} S_k^H &= (h_{k+1} + h_1) \delta_B / 2 - \text{для непарного } k \\ S_k^H &= (h_{k+1} - h_1) \delta_B / 2 - \text{для парного } k \end{aligned} \right\} \quad (10.19)$$

Згідно із залежностями (10.15) – (10.19) алгоритм роботи *підсистеми управління преселективною обробкою* (ПУПО) має такі основні етапи:

- настроювання верстата на мінімальний розмір вала, який є різницею між розміром отвору  $d_{Amin}$  і зазором  $\Delta_3$  у спряженні;
- настроювання верстата на розмір першої групи зміщення;
- визначення за допомогою відповідного модуля-обчислювача в ПУПО обсягу партії з цим зміщенням;
- стабілізація процесу обробки валів у межах розміру розсіювання  $\delta_B$ ;
- автоматичне настроювання верстата на розмір наступної групи зміщення після обробки усіх валів першого настроювання і т. д.

Весь процес преселективної обробки закінчується після  $k$ -го зміщення, де  $k = \delta_A (\delta_B / 2) - 1$ .

Таким чином, запропонований підхід дає змогу заздалегідь визначити кількість валів у кожній із груп зміщення настроювання довідного безцентрово-шліфувального верстата практично при будь-якій формі кривої, що обмежує елементарні площі  $S_i$  (у тому числі і гаусової кривої). Ступінь можливих спрощень у цьому випадку зумовлюється, з одного боку, порівнянням витрат на розрахунки (що у деяких випадках можна виконати тільки за допомогою складних обчислювальних пристроїв), а з другого — обсягом незавершеного виробництва, що визначається неповним збігом кривих розподілу при спрощеному розрахунку. Водночас *перенастроювання безцентрово-шліфувального верстата на обробку поточної групи відбувається в режимі реального часу реалізації складального процесу, коли в результаті здійснення принципу преселективної обробки для базової поточної деталі (отвору А) шліфується приєднана деталь (вал В)*.

**Приклад 10.2. Нечітке відображення належності до певного агрегатного стану системи взаємодії промислового робота з навколишнім середовищем.** Механізми захватного пристрою (ЗП) є одними з основних елементів маніпуляційної системи ПР, які визначають технологічні можливості ПР. Специфічними вимогами до ЗП є надійність захоплення та утримування об'єкта маніпулювання (ОМ), які визначаються умовами взаємодії ЗП з ОМ, тобто умовами накладання утримуючих і ненадвірних зв'язків на ОМ у процесі його захоплення.

*Виникнення нового агрегатного стану при взаємодії захватного пристрою з об'єктами маніпулювання.* При захопленні деталі ЗП може виникнути непередбачена ситуація через можливе проковзування ОМ між губок ЗП при недостатній силі захоплення внаслідок розсіяння розмірів деталей або зміні сили тертя через замаслювання чи запилення поверхонь «ОМ — губки ЗП».

Зокрема, як було вище зазначено, для найпоширенішої пари, якою є з'єднання вала із втулкою, у виробничих умовах технологія виготовлення отворів передбачає наявність максимального діапазону розсіяння похибок (що описуються нормальною кривою розподілу)  $6\sigma_A$  для всієї партії отворів, тобто розміри яких лежать у межах від мінімального  $d_{Amin}$  до максимального  $d_{Amax}$  у просторі  $6\sigma_A$  (рис. 10.13).

Зазначена непередбачена ситуація, внаслідок якої система взаємодії ЗП з ОМ переходить до нового АС, може бути наочно описана за допомогою застосування *принципів фаззі-логіки* (див. розділи 3 і 4), наприклад, так:

- ФН  $\mu_{M1}(x_1)$  нечіткої множини M1 для величини  $x_1$  зусилля захоплення ОМ через лінгвістичні невизначеності: «мале», «середнє», «велике» (рис. 10.14, а);
- ФН  $\mu_{M2}(x_2)$  нечіткої множини M2 для величини  $x_2$  ступеня проковзування ОМ через лінгвістичні невизначеності: «трохи проковзує», «сильно проковзує» (рис. 10.14, б);
- ФН  $\mu_{M3}(x_3)$  нечіткої множини M3 для ступеня можливого збільшення зусилля затиску, яке не перевищує величину  $x_3$  допустимого навантаження на поверхні ОМ, через лінгвістичні невизначеності: «деяке збільшення», «значне збільшення» (рис. 10.14, в);
- ФН  $\mu_{M4}(y)$  нечіткої множини M4 для необхідної керуючої дії  $y$ , яку треба подавати на привід ЗП, через лінгвістичні невизначеності: «трохи збільшити зусилля затиску», «значно збільшити зусилля затиску» (рис. 10.14, г).

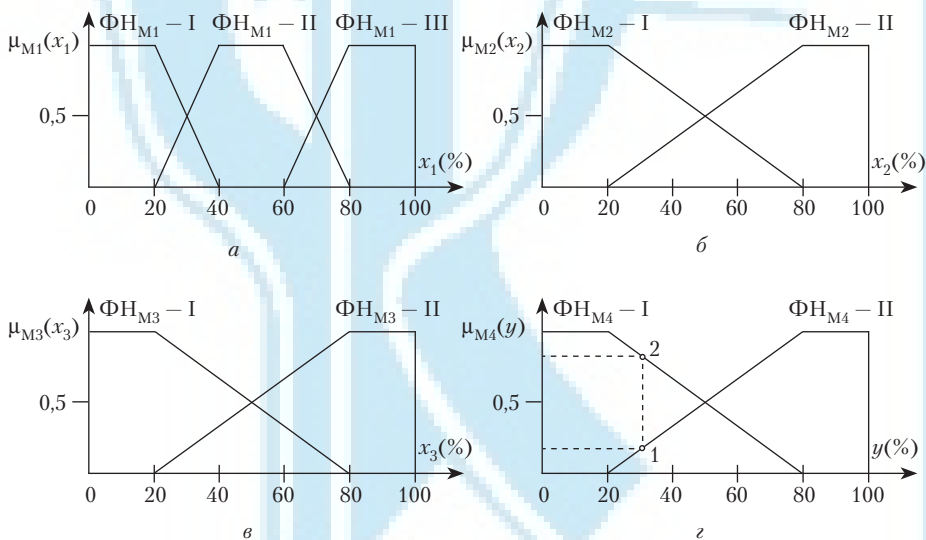


Рис. 10.14. Варіанти ФН для лінгвістичних величин і залежності від величин: а –  $x_1$  зусилля захоплення ОМ; б –  $x_2$  ступеня проковзування ОМ; в –  $x_3$  ступеня можливого збільшення зусилля затиску; г – необхідної керуючої дії  $y$  ФН<sub>M1</sub> – I

Фрагмент фаззі-системи та штучної нейронної сітки для розробки алгоритму управління зусиллям затиску. Тоді фрагмент продукційної системи, що описує алгоритм управління зусиллям затиску ОМ при можливому його проковзуванні між губок ЗП через недостатню силу захвату, можна подати у вигляді такого набору правил:

- Правило 1:** ЯКІЦО ( $x_1$ = «мале»)  
 І ( $x_2$ = «трохи проковзує»),  
 І ( $x_3$ = «деяке збільшення»),  
 ТО ( $y$ = «трохи збільшити зусилля затиску»).
- Правило 2:** ЯКІЦО ( $x_1$ = «мале»)  
 І ( $x_2$ = «сильно проковзує»),  
 І ( $x_3$ = «значне збільшення»),  
 ТО ( $y$ = «значно збільшити зусилля затиску»).

Графічне подання варіанта реалізації фаззі-правил 1 і 2 наведено на рис. 10.15.

Тоді можна навести *структуру фаззі-контролера* (рис. 10.16), де на його вхід через нормуючий елемент подаються сигнали  $x_1, x_2, x_3$  (принцип функціонування фаззі-контролера див. у розд. 4).

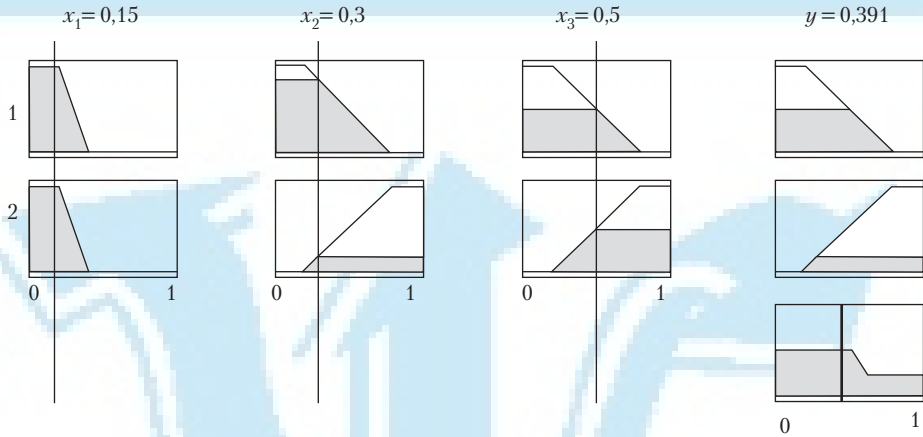


Рис. 10.15. Графічне подання одного з варіантів реалізації фаззі-правил 1 і 2

Обробка вхідної інформації в лінгвістичній формі здійснюється двома каналами, які працюють за наведеними вище правилами 1 і 2 у три кроки: 1) фаззіфікація, тобто F-перетворення фізичних величин у нечітку описову формулу; 2) логічне виведення, тобто пошук нечіткої величини керуючого впливу контролера; 3) дефаззіфікація, тобто зворотне D-перетворення нечіткого логічного виведення на другому кроці у вихідну фізичну величину керуючого впливу контролера.

Сполучення «І» нечітких висловлювань у частині правила «ЯКЩО» реалізується набором мінімуму поточних значень величин істинності  $x_1, x_2, x_3$  (правила 1 і 2):

$$\mu_{M_1,1}(y) = \min[\mu_{M_1,1}(x_1), \mu_{M_2,1}(x_2), \mu_{M_3,1}(x_3)], \quad (10.20)$$

$$\mu_{M_1,2}(y) = \min[\mu_{M_1,2}(x_1), \mu_{M_2,2}(x_2), \mu_{M_3,2}(x_3)], \quad (10.21)$$

де  $\mu_{M_i,k}(x_i)$  – значення ФН фаззі-множин  $M_i$  у  $k$ -му правилі,  $\mu_{M_i,k}(y)$  – значення ФН фаззі-множини  $M_i$  у  $k$ -му правилі,  $i = 1, 2, 3, k = 1, 2$ .

Результуюча ФН фаззі-правил має вигляд:

$$\mu^{(y)} = \max\{\mu_{y,1}(y), \mu_{y,2}(y)\}. \quad (10.22)$$

При дефаззіфікації за методом центра ваги вихідною величиною  $y_{ц}$  виступає значення абсциси положення центра ваги площі  $S$ , що лежить під графіком вихідної ФН (рис. 10.16, заштрихована площа на етапі дефаззіфікації).

Розрахунок абсциси центра ваги  $y_{ц} = S(x_s, y_s)$  ділянки площі, що охоплюється функцією  $y = f(x)$ , у багатьох застосуваннях при дефаззіфікації спрощується завдяки тому, що ФН для частин правил «ТО» на монотонних ділянках, відмінних від нуля, можуть стискатися. Завдяки введенню позначення  $f_i$  для  $n$  обмежених монотонних величин в окремих правилах абсциса центра ваги може бути обчислена, відповідно до (3.15) (див. підрозд. 3.4), за формулою

$$x_s = \frac{\sum_{k=1}^n (x_i f_i)}{\sum_{k=1}^n f_i}. \quad (10.23)$$

Щодо застосованих у фаззі-правилах лінгвістичних змінних, а також їхніх структурних взаємозв'язків у фаззі-нейросистемі задається структура задачі й тим самим виконується вимога до чітко *структурованої локальної* штучної НС (ШНС).

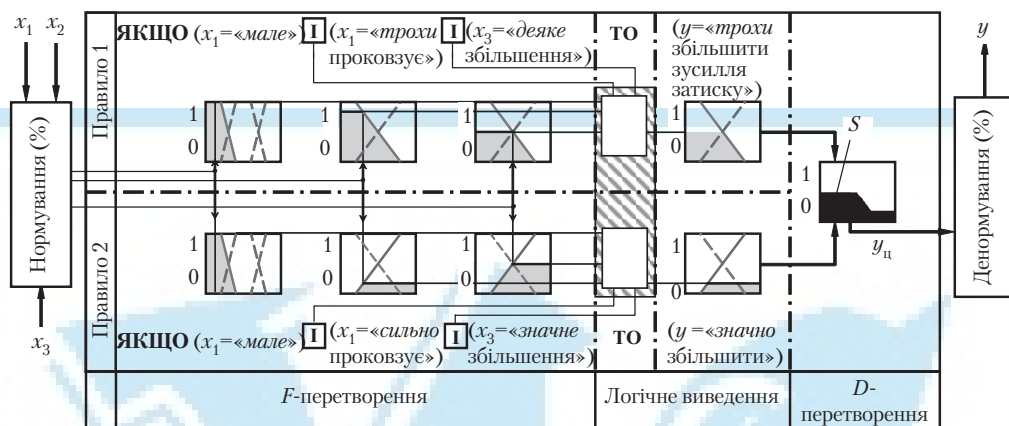


Рис. 10.16. Структура фаззі-контролера в системі управління зусиллям затиску ОМ ЗП

Згідно із структурою побудованого фрагменту фаззі-системи структура ШНС може мати вигляд, наведений на рис. 10.17.

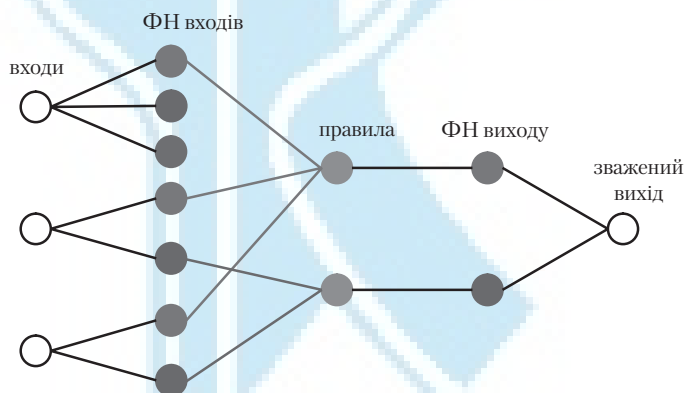


Рис. 10.17. Структура ШНС

Залежності вихідної величини фаззі-системи від вхідних у вигляді тривимірної поверхні при постійному значенні однієї з вхідних величин подано на рис. 10.18. Залежності виходу ШНС (зі структурою, поданою на рис. 10.17, згідно з фаззі-системою, наведеною на рис. 10.16) від входів у вигляді тривимірної поверхні при постійному значенні одного з входів та результати порівняння з даними фаззі-системи наведено на рис. 10.19 і рис. 10.20 відповідно.

Таким чином, здійснено формалізацію процедур фаззіфікації, логічного виведення на основі фаззі-правил 1, 2 і дефаззіфікації, побудовано фаззі-систему для управління зусиллям затиску ОМ ЗП, яка відображена на ШНС спеціальної структури.

При застосуванні нейросіткового підходу з використанням приблизних значень фаззі-системи забезпечується більш плавна поведінка системи (рис. 10.19), ніж та, яка виходить при фаззі-реалізаціях (рис. 10.18), що використовують для ФН форми трикутника чи трапеції (як у розглядуваному випадку).

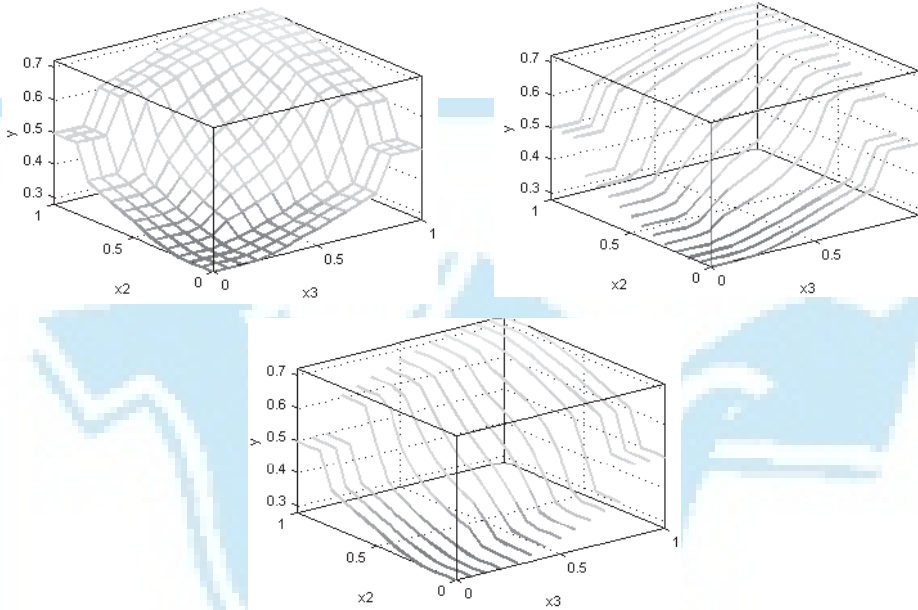


Рис. 10.18. Експериментальні залежності необхідної керуючої дії  $y$  від ступеня проковзування ОМ  $x_2$  та ступеня можливого збільшення зусилля затиску  $x_3$  при постійному значенні зусилля захоплення ОМ  $x_1 = 0,15$  у фаззі-системі

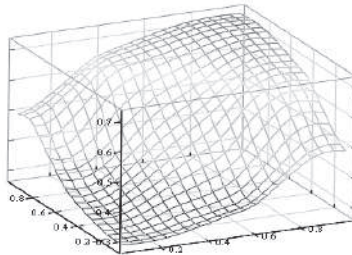


Рис. 10.19. Експериментальні залежності виходу  $y_{nc}$  від входів  $x_2$  і  $x_3$  при постійному значенні  $x_1 = 0,15$  у ШНС

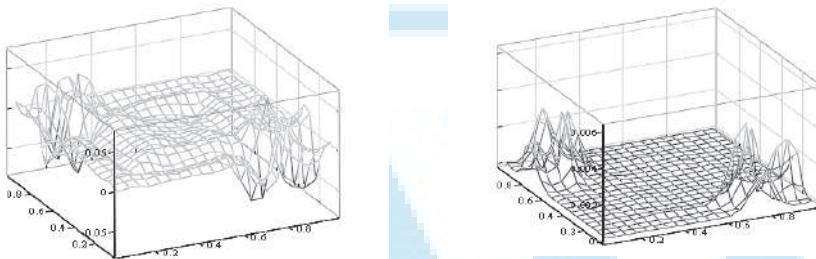


Рис. 10.20. Відхилення залежностей виходу  $y_{nc}$  ШНС від входів  $x_2$  і  $x_3$  від відповідних залежностей виходу  $y_{fuz}$  фаззі-системи від входів  $x_2$  і  $x_3$

### Контрольні завдання

1. Розкрити зміст поняття реального часу.
2. Сформулювати передумови використання ІС в задачах реального часу.
3. Прокоментувати основні вимоги щодо ІС реального часу.
4. Навести апаратні та програмно-методичні заходи щодо підвищення швидкодії ІС реального часу.
5. Пояснити функції таймера у складі ІСРЧ і розкрити відмінності у принципах збирання інформації, які базуються на підходах: з періодичним опитуванням датчиків, незалежним надходженням даних та мішаним (комбінованим).
6. Дати порівняльну оцінку основних методів організації виведення в ІСРЧ та розкрити структуру асинхронного модуля логічних виведень у ІСРЧ.
7. Розкрити зміст основних компонент структури робота з елементами ШІ і пояснити роль модулів «планувальник» та «система прийняття рішень».
8. Схарактеризувати особливості функціонування ІР у реальних середовищі та часі і дати означення інтелектуалізованого робота.
9. Пояснити, що входить до завдань алгоритмічного та декларативного підходів щодо керування складними робототехнічними пристроями.
10. Сформулювати принципи самокерування і розкрити роль інтелектуалізованої компоненти при цьому.
11. Дати означення інтелекту ГКІС.
12. Розкрити зміст двох складових ІР — моделі знань і системи планування цілеспрямованих дій.
13. На прикладах фреймової структури прокоментувати систему знань про зовнішнє середовище для ІР — сцену робота.
14. Аргументувати прикладом доцільність формування знань про операції ІР в його робочому просторі на основі фреймового подання.
15. Сформулювати приклад типових операцій нульового, першого, другого та третього рівнів при поданні процесу встановлення інтелектуальним роботом об'єкта на площині.
16. Дати загальну характеристику нештатних ситуацій у ГКІС і навести означення нештатної ситуації.
17. Дати означення агрегатного стану об'єкта та прокоментувати властивості просторово-агрегатної розподіленості ГКІС.
18. Навести і прокоментувати класифікацію нештатних режимів роботи ГКІС.
19. Розкрити зміст відображення процесу обробки надзвичайної операції у гіперпросторі АС системи.
20. Навести означення агрегату і розкрити складові його схеми.
21. Розкрити означення системи як сукупності агрегатів із зв'язками між ними.
22. Розтлумачити функції та навести означення абстрактного графа управління об'єктом і проілюструвати його просторово-часовою інтерпретацією імітаційно-керуючої моделі.
23. Сформулювати основні принципи розв'язання задачі оптимізації управління в нештатних режимах функціонування.
24. На прикладах преселективної обробки та обслуговування роботом ОР розкрити підходи до розв'язання невизначеності в умовах функціонування складних ОК в реальному часі.

## ІНТЕЛЕКТУАЛІЗОВАНЕ ДИСКРЕТНО-ПОДІЙНЕ МОДЕЛЮВАННЯ ГНУЧКИХ ВИРОБНИЧИХ СИСТЕМ

---

---

*Якщо наслідки тієї чи іншої речі погані,  
тоді прийнятніша та, з якої погане випливає меншою мірою.*

Аристотель

*Якщо не висловлено протилежних думок,  
то нема з чого вибирати найкращої.*

Геродот Галікарнаський

*Найнищівніші завжди —  
середні шляхи.*

Нікколо Макіавеллі

### 11.1. Особливості математичного, імітаційного та статистичного моделювання процесів АТСС

**Особливості подання ГКІС як об'єкта моделювання.** При моделюванні ГКІС розглядається як складна система, яка являє собою композицію окремих компонентів. Саме взаємодія компонентів ГКІС і становить основний аспект моделювання системи і має забезпечувати її функціонування відповідно до вимог технологій, що в ній реалізуються.

Потрібно зазначити, що дискретно-подійні системи, до яких належать і ГКІС [37], як часові моделі у загальному випадку є аналітично важкоотримуваними і комп'ютерно-розрахунково нездійснюваними через численність раптово-випадкових подій на можливих непередбачуваних інтервалах станів, що розв'язуються деякою мірою комбінаторно. Проте процеси у ГКІС, які підпорядковуються часо-синхронізуючому впливу, добре моделюються подійно-часовими графами (ПЧГ) — своєрідними модифікаціями сіток Петрі [74; 191], в яких часова затримка асоціюється з кожною позицією, що має тільки по одному вхідному та вихідному переходу [125; 155; 161]. Іншою модифікацією СП, створеною власне для моделювання динамічних перебігів подій у системі, є гнучкі СП [26; 59; 161]. Проте для дослідження динамічної поведінки дискретно-подійних систем з часовим навантаженням можна з успіхом застосувати діод- та макс-алгебри [145]. В останньому випадку дослідження ГКІС може здійснюватися визначенням періодичності системних матричних властивостей і розрахунком їх вагових оцінок. Використовуючи інструментарій макс-алгебри, можна, підраховуючи неконтрольовану поведінку ПЧГ-моделі, знаходити подання деякого нового бажаного режиму функціонування і визначити, чи може це подання бути реалізоване за допомогою набору контрольованих подій [65; 93; 180].



При виборі формального апарату для розв'язання поставленої задачі насамперед керуються притаманними особливостями і властивостями композиційних систем із сукупністю компонентів, які взаємодіють між собою. Тому формальний апарат, який планується для використання при описі властивостей ГКІС, має забезпечувати розвинені набори операцій композиції, компактне синтаксичне подання розрізаних і поєднаних системою компонентів [59]. При цьому постає питання, якою моделлю користуватись для відображення особливостей функціонування процесів у ГКІС.

**Методи моделювання ГКІС.** Як вже зазначалося в підрозд. 6.5, для розв'язання задачі моделювання використовуються методи імітаційного моделювання, теорії автоматів, марковських процесів, теорії черг, макс/мін-плюс алгебри, сітки Петрі, процесні алгебри. Наведемо узагальнену характеристику методів, окресливши основну специфіку задач моделювання взаємодії компонентів ГКІС: композиційність, синтаксичний опис процесу взаємодії, відображення внутрішньої будови ГКІС (рівень компонентів та/або їх внутрішньої будови).

*Теорія автоматів* [9; 22] забезпечує розвинений апарат опису та аналізу переходів між станами системи, але не відображає причинно-наслідкових відношень. Крім того, композиція моделей та їх ієрархічне подання в межах цього апарату ускладнені.

*Марковські процеси* [106] і *теорія черг* враховують стохастичний аспект поведінки системи і дають їх кількісні характеристики відносно певних проміжків часу чи наборів завдань, але вони абстрагуються від внутрішньої будови системи.

*Алгебри (max/min)* [144] за суттю абсолютно аналогічні з точністю до операції max або min і знаходять застосування в задачах розрахунку часу виконання визначених послідовностей дій у системі (якщо спостерігається досить регулярна структура, яка може бути подана матрицями) і проходження потоку вздовж шляху (маршруту, для якого оцінюється тривалість), що можна легко подати матричними операціями. Тоді, наприклад, деяка подія відбувається, «як тільки» (або «відразу ж, як») виконується певна умова (тому «min»); аналогічно, якщо треба очікувати завершення усіх передумов (тобто коли найпізніша завершиться), то виникає «max», тощо. Деякі аспекти застосування розширення <max/min+>-алгебри розглянуто у розд. 12.

*Процесні алгебри (ПА)* [128; 149] не призначені для опису чи подання будь-чого на певному рівні ієрархічного моделювання у ГКІС. Використання цього апарату забезпечує композиційність у формально-математичному, а не тільки інтуїтивному змістові (тобто надається можливість застосовувати математичні операції на рівні опису системи із застосуванням знаків, символів тощо, перетворювати вирази за певними правилами, як це робиться в алгебрі взагалі, але в цьому випадку кожний вираз визначає систему, а точніше, процес, що нею реалізується). На відміну від традиційної теорії автоматів, ПА більшою мірою спроможні подавати паралелізм, комунікаційність тощо, проте не мають достатньо ефективних засобів подання причинно-наслідкових відношень і обмежені у відображенні істинного паралелізму.

Деякі властивості стандартної ПА для виявлення основних характеристик цього формального апарату як системи подання знань і як інструментарію до моделювання взаємодії компонентів ГКІС розглянуто в підрозд. 2.8.

Що ж до *сіток Петрі*, застосування яких як апарат ієрархічного подання дискретно-подійних процесів ГКІС детально розглянуто в розд. 8, то порівняно з ними ПА краще забезпечують зручний синтаксис, проте гірше подають паралелізм. Можна стверджувати, що ПА спрямовані своїм змістом на моделювання часу та синхронізацію і різняться своїми семантичними та синтаксичними відмінностями.

*Теорія розкладів* [49; 118] використовує як поняття виконуваного процесу (оператор, операція, задача, робота), так і використовуваного ресурсу (машина, проце-

сор, прилад, пункт, людина). Задача може полягати, наприклад, у досягненні мінімуму сумарного часу виконання набору операцій шляхом оптимального закріплення їх за ресурсами системи. Самі операції можуть бути залежними та незалежними. Інша постановка задачі може полягати в оптимізації порядку проходження складових розв'язання через систему.

Більш складні реальні моделі процесів у ГКІС із застосуванням апаратів ПА, асинхронних протоколів (АП), алгебр (max/min,+) розглядаються, зокрема, в роботах [79; 145], де запропоновано нові стратегії управління розподіленими робочими потоками ГКІС з уникненням тупикових ситуацій. Стратегія синхронізуючих зон базується тільки на ресурсних потребах системи, поданих *графом потреб ресурсів* (ГПР). При цьому інформація про виробничий маршрут зводиться до «розділення» послідовності виготовлення об'єкта виробництва, в результаті чого відокремлені зони розглядаються як «паростки» в системі.

Стратегія *графа розподілу заявок* (ГРЗ) базується на розгляді сегментів ГКІС, які синхронізуються, і забезпечує локальними правилами керування робочими потоками та потребує визначення набору таких сегментів у ГРЗ-моделі функціонування ГКІС. Така стратегія пов'язана з визначенням усієї ГРЗ-моделі системи, а оскільки проблема дослідження циклів є NP-повною задачею, обчислення виконується в режимі off-line, тоді як керування окремим циклом (сегментом або петлею графа), що аналізується, здійснюється в режимі on-line. І нарешті, стратегія, яка базується на протоколах розподілу потужностей (ПРП), використовує ідею самосинхронізації робочих потоків у результаті визначення буферних потужностей і формування правил диспетчеризування стосовно цього набору виробничих маршрутів. У цьому випадку протоколи розглядаються як процедури, що координують конкуренцію процесів за доступ до необхідним чином розподілених і обмежених ресурсів.

Інше застосування механізмів ПА щодо дослідження проблеми управління обмеженими перепускними спроможностями композиційних складових ГКІС з періодичним характером операцій наведено в [73]. Композиційна складова тут розглядається як робоче місце з конкуруючими за доступ до загальних обробляючих ресурсів робочими потоками, які об'єднують технологічне устаткування, вхідний/вихідний буфери обмеженої потужності та промислові роботи, які обслуговують їх. Розв'язання задачі полягає у визначенні достатніх для формування правил розподіленого диспетчеризування робочих потоків та умов, які базуються на оціночних критеріях системної циклової продуктивності та коефіцієнтах використання ресурсів ГКІС. З цією метою запропоновано *логістичний підхід з використанням поняття асинхронних протоколів*, які і забезпечують локальний механізм для означених вище показників ефективності.

В роботі [98] на основі алгебраїчної моделі з використанням модифікованого апарату min-плюс-алгебри ГКІС розглядається як розподілена система, яка складається з автономних повторюваних процесів (транспортних потоків), які утворюють загальний набір і конфігурацію шляхопровідних секцій з певними відстанями між їх перетинами. Задача розробки розкладу обслуговування обробляючих ресурсів транспортною системою, за яким стабілізується функціонування ГКІС, супроводжується одночасно умовою щодо забезпечення і мінімального циклового часу (обернена величина — перепускна спроможність системи). Для розв'язання цієї задачі запропоновано підхід, який базується на використанні *принципу «віртуальних асинхронних транспортних світлофорів»*, які на маршрутних перетинах забезпечують механізми синхронізації з локальним обмеженням потоку індивідуальних транспортних засобів-роботарів. У результаті «світлофорне регулювання»

завершується усталеним циклічним режимом функціонування системи. Детально цей підхід розглядатиметься в розд. 12 на прикладі створення об'єктно-орієнтованого середовища для моделювання розподілених транспортних потоків ГКІС.

**Об'єктно-орієнтований вибір відповідностей «формалізми – об'єкти».** З урахуванням викладеного великого значення набуває задача розробки загальних підходів щодо ідентифікації процесів у ГКІС, їх композиційного моделювання на основі комбінованих методів гіперпросторової семантики (див. підрозд. 9.5). Розробка таких підходів дає змогу комплексно описувати процеси, які відбуваються у ГКІС, ще на стадії їх проектування [37]. Саме проектування при цьому реалізується згідно з визначеною системою формально обґрунтованих математичних правил, забезпечуючи аналітичне прогнозування складних динамічних і тупикових ситуацій, заціклювань, загальних блокувань системи тощо, пропонуючи засоби для попередження проблемних ситуацій, а за необхідності – їх розв'язання.

Ефективним й універсальним шляхом створення такого апарату, що належним чином реалізує взаємодію формалізмів, є такий їх опис на семантичному рівні, коли способи подання одних моделей одночасно слугують семантичним базисом для подання інших у просторах гіперпросторової моделі ГКІС. При цьому перетворення та маніпуляції над моделями визначають відповідні зміни семантики на всьому гіперпросторі моделювання.

Використання різних методик, які підтримуються у межах загального семантично узгодженого середовища, пов'язано з проблемою невизначеності вибору «найкращої» для розв'язання цієї задачі методики. Кожний формалізм, як показано вище, володіє рядом показників, які характеризують його можливості. Водночас вибір різних технік моделювання відбувається у рамках інтегрованого об'єктно-орієнтованого середовища, тому конкретні значення показників характеристик залежать також від розв'язуваних у процесі моделювання задач. Отже, визначення пріоритетності використання того чи іншого формалізму при роботі з різними об'єктами середовища, що використовується, полягає у виявленні значущості показників методу для розв'язання задач і вибору на основі цих показників найкращого методу.

Цю невизначеність застосовності різних методів зручно подати у вигляді гіперпросторової моделі (рис. 11.1) уживаності формалізмів (інструментальних засобів формального опису або подання знань про відображуваний об'єкт – компонент ПС).

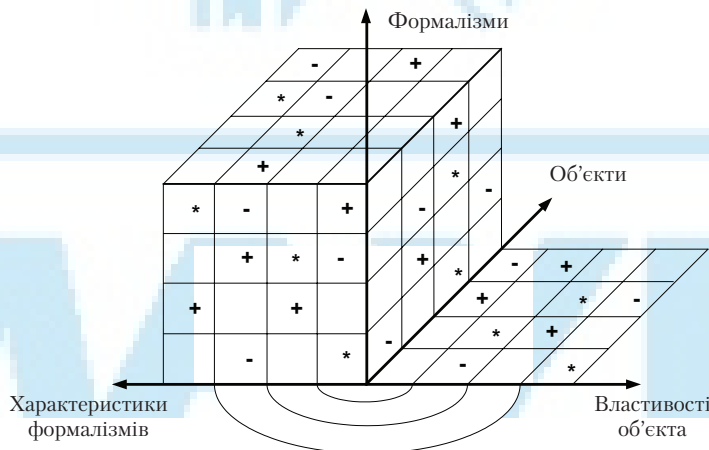


Рис. 11.1. Гіперпросторова модель уживаності формалізмів

Вимірами такої моделі є: *площина показників* цих семантик, що характеризують основні особливості та можливості відображення властивостей поданих об'єктів; *площина методів*, що становлять перелік формалізмів, які використовуються при моделюванні ГКІС; *площина об'єктів* з відображенням їх характерних властивостей та особливостей функціонування (динамічних, логічних і логістичних). Адекватним поданням значення комірки такої гіперпросторової моделі може бути оцінка придатності методу для розв'язання конкретної задачі з точки зору заданої функції відображення сукупної або локальної властивості об'єкта.

На початкових етапах вибору придатного формалізму з альтернативних варіантів семантик невизначеність зумовлена як об'єктивними, так і суб'єктивними факторами. До *об'єктивних факторів* належать неповнота множини вихідних даних, відсутність кваліфікованих спеціалістів, а також обмеження, зумовлені зовнішнім середовищем. *Суб'єктивні фактори* визначаються досвідом та інтуїцією експертів, їх становленням та особистою зацікавленістю у варіантах вибору. Аналітичними методами визначити вплив таких причин на кінцевий вибір досить складно через необхідність запровадження занадто громіздкої системи класифікаційних ознак, їх градацій і взаємовпливів, якими описуються можливості експертів, з одного боку, та ступенями ризику прийняття неадекватних рішень — з іншого. Водночас, використовуючи апарат нейросіток і методи теорії нечітких множин (див. розд. 3 і 4), таку невизначеність можна суттєво зменшити, приймаючи раціональні рішення щодо вибору семантики, яка задовольняє вихідні вимоги.

Наведена у табл. 11.1 характеристика формалізмів, які використовуються при моделюванні ГКІС, відображає їх застосовність лише якісно, указуючи, чи реалізує цей метод деяку властивість (+ або –). При цьому ряд параметрів реалізується частково (\* у табл.), інформація про деякі показники відсутня. Очевидно, що відобразити наявну інформацію можна на основі апарату нечіткої логіки. При цьому може використовуватися як якісна інформація у вигляді лінгвістичних змінних, які реалізують деяке значення з набору терм-множини, так і кількісні дані у вигляді оцінок застосовності, визначеної на деякій шкалі. Джерелом інформації для формування оцінок і параметрів термів лінгвістичних змінних на початковому етапі (навчання) виступають думки експертів. Надалі, в процесі використання, значення можуть коригуватися за допомогою різних адаптивних алгоритмів. Зважаючи на нечіткий характер змісту гіперпросторової моделі застосовності формалізмів і специфіку формування, прийняття рішення щодо вибору методу в умовах лінгвістичної невизначеності необхідно здійснювати на основі нечітких підходів. Застосуванню цього підходу для роботи з невизначеностями сприяє також гіперпросторова розподіленість ГКІС, коли за нечітких умов у задачах керування приймаються рішення переходу в новий агрегатний (під впливом зовнішніх факторів) або дискретний (відносно вектора керування) стани.

**Рейтингове оцінювання альтернативних варіантів семантик.** Усі математичні моделі прийняття рішень мають вигляд рівнянь, в яких загальний критерій функціонування (оптимізації) системи в цілому прирівнюється до деякого співвідношення, яке зв'язує між собою множини змінних, якими визначається поведінка системи. У загальному вигляді ці вирази можуть становити систему аналітичних або статистичних рівнянь або нерівностей.

**Означення 11.1.** *Критерій оптимальності* (критерій ефективності) являє собою кількісну оцінку (міру) формального відображення властивостей об'єкта дослідження.

Таблиця 11.1

## Характеристика методів і формалізмів моделювання ГКІС

	Формалізм	Опис динаміки процесу	Кількісні характеристики	Якісні характеристики	Аналіз переходів	Компактний синтаксичний опис	Логіка внутрішнього функціонування	Прогнозування	Простота опису складних систем	Опис ієрархії
1	Теорія автоматів (* стохастичні, часові, нечіткі)	+	*	*	+		-	-	-	+/-
2	Теорія графів			+					-	+
3	Сітки Петрі (* досяжність, безпека, живучість)	+		*	+	-			-	
	Часові СП (* відображає часові аспекти поведінки системи)	+	*	*		-	+		-	
	Розфарбовані СП (* за допомогою багатомірності атрибутів)	+	*	*	*	-	+		-	
	Об'єктні СП	+		+			+		+	+
4	Системи масового обслуговування	-	+						-	
	Марковські процеси (* імовірнісні, інтенсифіковані)	-	*	*	-		-	+		
	Теорія черг	-	*	*	-		-			
5	Процесні алгебри	-		+	+/-	+	-			-
6	Діючі (max, min+) алгебри		*	*					+	
7	Рівнянні	+				+				
8	Методи формальних специфікацій	+		*						

При обґрунтуванні критерію ефективності необхідно враховувати як характер розв'язуваної проблеми і специфіку проблемної області, так і обмеженість самих критеріїв. Зокрема, для оцінки якості формального відображення властивостей досліджуваного об'єкта можуть використовуватися адитивні критерії  $F = \sum_{j=1}^n \lambda_j F_j$ , в яких враховуються вагові коефіцієнти  $\lambda_j$  частинних критеріїв  $F_j$ , причому  $\sum_{j=1}^n \lambda_j = 1$ .

Як вже наголошувалося вище, рейтингову оцінку критеріїв проводять експерти, які оцінюють кожний критерій певною кількістю балів виходячи з їх важливості. Розглянемо два поширених методи рейтингової експертної оцінки критеріїв якості: метод ранжування і метод парних порівнянь [3].

**Метод ранжування.** Нехай є  $m$  експертів, які оцінюють  $n$  критеріїв. Кожний експерт проводить ранжування критеріїв шляхом присвоювання номерів від 1 до  $n$  відповідно до зменшення ступеня важливості цих критеріїв, і при цьому заповнюється така таблиця:

		критерії			
		1	2	...	$n$
експерти	1	$x_{11}$	$x_{12}$		$x_{1n}$
	2	$x_{21}$	$x_{22}$		$x_{2n}$
	...				
	$m$	$x_{m1}$	$x_{m2}$		$x_{mn}$

Тут  $x_{ij}$  – номер, який  $i$ -й експерт надав  $j$ -му критерію. У подальшому зручно перерахувати цю таблицю, надаючи більш важливому критерію більше число  $y_{ij} | i = 1, 2, \dots, m; j = 1, 2, \dots, n$ . Вводимо нові ваги  $y_{ij} = n - x_{ij}$ . Тепер найбільш важливий критерій має оцінку  $n - 1$ , а найменш важливий – нуль. Вагові коефіцієнти в адитивному критерії визначаються так:

$$\lambda_k = \sum_{i=1}^m y_{ij} / \left( \sum_{i=1}^m \sum_{j=1}^n y_{ij} \right), k = 1, 2, \dots, n.$$

Ступінь узгодженості результатів роботи експертів визначається коефіцієнтом

$$w = 12S / \left[ m^2 (n^3 - n) \right],$$

де  $S = \sum_{j=1}^n d_j^2; d_j = 0,5m(n+1) - \sum_{i=1}^m x_{ij}; j = 1, 2, \dots, n$ .

Значення  $w$  змінюються в межах  $w \in [0; 1]$  за правилами:

ЯКЩО  $w = 1$ , ТО всі експерти дають однакові оцінки;

ЯКЩО  $w = 0$ , ТО думки експертів неузгоджені.

Залежно від величини коефіцієнта  $w$  результати оцінки критеріїв або приймаються, або відкидаються.

**Метод парних порівнянь.** Цей метод застосовують тоді, коли експертам складно оцінити важливість критеріїв у балах. У цьому випадку кожний  $k$ -й експерт заповнює таку таблицю:

	1	2	$j$	$n$
1	$a_{11}$	$a_{12}$		$a_{1n}$
2	$a_{21}$	$a_{22}$		$a_{2n}$
$i$				...
$n$	$a_{n2}$	$a_{n2}$	...	$a_{nn}$

Тут  $a_{ij}^k = 1 \mid i = 1, 2, \dots, n; j = 1, 2, \dots, n; k = 1, 2, \dots, m$ , якщо  $i$ -й критерій перекладений над  $j$ -м критерієм, і  $a_{ij} = 0$  у протилежному разі.

Для кожного експерта будується така таблиця, тобто матимемо  $m$  таблиць. Далі відбувається об'єднання таблиць в одну шляхом підсумовування чисел, які стоять у відповідних клітинках.

$$b_{ij} = \sum_{k=1}^m a_{ij}^k.$$

Нарешті визначаються вагові коефіцієнти:

$$\lambda_k = \sum_{j=1}^n b_{kj} / \left( \sum_{i=1}^n \sum_{j=1}^n b_{ij} \right), k = 1, 2, \dots, m.$$

У цьому методі теж треба визначити ступінь неузгодженості експертів:

$$\gamma = \left[ \left( 2 \sum_{i=1}^n \sum_{j=1}^n C_{bij}^2 \right) / \left( C_m^2 C_n^2 \right) \right] - 1,$$

де  $C_v^r = \frac{v!}{r!(v-r)!}$  — кількість сполучень з  $v$  по  $r$ . Мінімальне значення  $\gamma_{\min} = -1/(m-1)$  для  $m$  парного і  $\gamma_{\min} = -1/m$  для  $m$  непарного.

Отже, для таблиці 11.1 після проведення ранжування та визначення ступеня узгодженості, наприклад,  $w = 0,85$ , можемо записати ранжування адекватності методів формального відображення властивостей ГКІС (табл. 11.2).

**Моделі взаємодії процесів ГКІС.** При моделюванні КІВС розглядається як складна система, яка являє собою композицію окремих компонентів. Саме взаємодія компонентів КІВС і становить основний аспект моделювання системи і має забезпечувати функціонування останньої відповідно до вимог технологій, що в ній реалізуються.

При цьому постає питання, якою моделлю користуватись для відображення особливостей функціонування процесів у КІВС. Для розв'язання означеної задачі необхідно вибрати модель, характеристики якої відповідатимуть особливостям реальної КІВС. Розглянемо основні типи існуючих моделей взаємодії, серед яких виокремимо *вкладену, однорівневу та мішану моделі* взаємодії [161]. Послідовно розглянемо їх основні характеристики.

**Вкладена модель взаємодії.** Вкладена модель базується на тому, що взаємодія між вкладеними матеріальними (товарів, послуг, енергії, інформації тощо) потоками керування може розглядатися як спеціальний випадок синхронного зв'язку [190], який веде себе подібно до процедурного виклику. Тобто мається на увазі, що компоненти, виконуючи запит, залишаються у блокованому стані доти, поки запит не буде задоволений. Ступінь невизначеності у вкладених моделях обмежена тим фактом, що компонент не сприймає зовнішніх (чужі) подій через очікування відповідної на попередній запит. Вкладені моделі характеризуються гнучкістю. Завдяки цьому *спрощується аналіз систем на базі таких моделей*.

**Однорівнева модель взаємодії.** Компоненти таких моделей проектуються відповідно до диспетчерського циклу. На відміну від попередньої моделі, запити можуть оброблятися, доки попередні операції ще не завершені. При цьому послідовність подій у системі стає більш випадковою, аніж у вкладених моделях. В однорівневій моделі можуть використовуватися різноманітні підходи до зв'язку. Однорівневою моделлю *забезпечується безтутиковість станів та реагування системи на події*, що є необхідною умовою для КІВС.

## Оцінка адекватності методів подання властивостей ГКІС

		Опис динаміки процесу	Кількісні характеристики	Якісні характеристики	...	Синтаксичний опис
1	Теорія автоматів	0,8	0,4–0,8	0,6–0,8		0,2
2	Теорія графів	0,5	0,2	0,8		0,3
3	Сітки Петрі	0,9	0,3	0,7		0,3
	Часові СП	0,7	0,8	0,5–0,7		0,2
	Розфарбовані СП	0,7	0,6–0,8	0,4–0,7		0,1
	Об'єктні СП	0,8	0,3	0,7		0,4
4	Системи масового обслуговування	0,4	0,7	0,3		0,1
	Марковські процеси	0,3	0,5–0,8	0,5		0,1
	Теорія черг	0,2	0,4–0,7	0,3		0,1
5	Процесні алгебри	0,3	0,2	0,8		0,8
6	Діючі (max, min +) алгебри	0,2	0,7	0,5–0,7		0,3
...						0,8
<i>n</i>	Методи формальних специфікацій	0,6–0,9	0,4	0,4–0,8		0,4

Однак при тестуванні однорівневі моделі є більш громіздкими, ніж вкладені моделі. Це пояснюється тим, що задіяний компонент може мати прямі зв'язки із процесами, які ще перебувають у стадії розвитку. Ще важче прогнозувати поведінку подійно-керованих систем, бо незалежні процеси можуть запускатися, конкуруючи в одному компоненті та непрогнозовано взаємодіючи між собою.

Проблема для семантичного моделювання однорівневих систем полягає в ідентифікації поведінок, які, за припущенням, можуть бути утворені взаємодіями компонентів.

**Мішані моделі.** Незважаючи на недоліки однорівневої моделі, практика іноді потребує її використання в системному проектуванні. Щоб частково уникнути непрогнозованих взаємодій між процесами, що характерно для однорівневої моделі, використовують мішану модель. Вона характеризується застосуванням блокування ресурсів і транзакцій.

**Означення 11.2.** Блокування ресурсів — це стан системи, в якому два процеси, що використовують ресурси, блокують один одного через те, що перший процес захопив ресурс *A* і очікує звільнення ресурсу *B*, тоді як другий захопив *B* і очікує звільнення ресурсу *A*. В такому стані системи обидва процеси виявляються взаємно заблокованими (прикладом такої ситуації може бути наведений на рис. 7.3 у підрозд. 7.1 спрощений приклад тупикової ситуації обслуговування роботами *R1* і *R2* об'єктів *A* і *B*, розташованих на позиціях-ресурсах *T1* і *T2*, відповідно).

**Означення 11.3.** Транзакція — це перехід процесу з поточного до попереднього стану в точках синхронізації.

Взаємне використання механізмів транзакцій та блокування ресурсів надає системі більш прогнозованої поведінки. Отже, блокування та транзакції в однорівневій системі можуть породжувати систему, яка інколи виявляє властивості вкладених



моделей. З іншого боку, багатопроцесність у системі, що має вкладений матеріальний потік, може також породжувати поведінку, яка характерна як для першої, так і для другої моделі. Ось чому моделі з використанням зазначених механізмів називаються *мішаними*.

У загальному випадку при використанні однорівневого підходу до проектування системи необхідно визначити всі заблоковані стани. *Щоб уникнути необхідності точного визначення усіх заблокованих станів, використовують багатопроцесність із застосуванням механізму транзакцій.*

До переваг мішаної моделі належать підтримка розподіленого проектування, коли різні компоненти в системі проектуються різними людьми, які можуть використовувати як вкладену, так і однорівневу модель.

Розгляд зазначених моделей дає можливість вибору, згідно з характеристикою реальної КІВС, відповідної моделі та підходів у визначенні зв'язків компонентів системи.

**Метод аналогій як основа системного аналізу складних об'єктів.** Раніше (див. розд. 8 та 9) обговорювалися загальні засади створення систем автоматизованого імітаційного моделювання (САІМ) і розкривалася роль, яку відіграють при цьому штучні інтелектуальні компоненти — пошукові модулі (ПМ). В цьому розділі йтиметься про застосування наведених у розд. 9 загальних підходів до синтезу САІМ для реальних умов системного проектування і дослідження складних об'єктів.

У системному аналізі основну роль відіграє *метод аналогій*, який дає змогу встановити відношення еквівалентності між двома системами, які розглядаються — реальною та абстрактною. Якщо система існує реально, то її можна вивчати і спостерігати, досліджуючи входи та виходи, і на основі цих досліджень будується деяка абстрактна система, в якій відношення еквівалентності визначають істотні властивості та аспекти поведінки, які в обох системах мають бути однакові.

**Означення 11.4.** *Моделлю* будемо називати систему, яка використовується для дослідження.

Встановлення міри відповідності між системою та моделлю визначають поняттями *ізоморфізму* (при існуванні однозначної відповідності між ними) та *гомоморфізму* (за наявності менш тісних зв'язків між системою та моделлю). В останньому випадку система та модель стають ізоморфними тільки в разі спрощення системи, тобто скорочення множини її властивостей та аспектів поведінки, які визначають простір станів системи.

Визначення *математичної* моделі наведено в підрозд. 6.1. Математичне моделювання для дослідження характеристик процесу функціонування систем можна поділити на аналітичне, імітаційне і комбіноване.

При *аналітичному моделюванні* процеси функціонування елементів системи записують у вигляді деяких функціональних залежностей або логічних умов. Якщо відомі явні залежності, які зв'язують шукані характеристики з початковими умовами, параметрами та змінними системи, можна виконати повне дослідження процесу функціонування системи. Проте одержати такі залежності неможливо. Суттєве спрощення вихідної моделі, яке допускається в аналогічних ситуаціях, призводить до втрати точності, що у разі проектування ГКІС може призвести до значних розбіжностей розрахункових і реальних показників ефективності. Напри-

клад, використання апарату *теорії масового обслуговування* (ТМО) припустиме на стадії розробки вихідного варіанта ГКІС, проте застосування положень ТМО на стадії оцінювання варіантів не можна вважати ефективним, оскільки розрахунки при цьому потребують введення не виконуваних на практиці жорстких припущень щодо характеру потоків заявок та інтенсивностей їх обслуговування, а також напівемпіричних апроксимацій. Тут не враховується поведінка системи в динаміці і залишаються відкритими питання оцінки ефективності вибраних стратегій керування.

*Імітаційне моделювання* пов'язує з конструюванням моделі існуючої або проєктованої системи та проведенням на моделі експериментів. Складовими імітаційної моделі (ІМ) є [121]:

- подання структури системи, яка моделюється (сукупного опису елементів і зв'язків між ними);
- створення засобів відтворення поведінки системи в моделі шляхом відображення станів та переходів між ними;
- відображення властивостей середовища, в якому функціонує досліджувана система, із завданням вхідних впливів середовища на модель.

До характерних для імітаційного моделювання можна віднести задачі розробки *моделі масового обслуговування*, пов'язані з розглядом черг. Власне ГКІС і належать до систем масового обслуговування і характеризуються наявністю підсистем — генераторів заявок на обслуговування та підсистем, які обслуговують їх, а також виникненням черг до транспортних засобів, позицій комплектації та обробного устаткування. При цьому внутрішня структура ГКІС може бути за необхідності деталізована до такої міри, щоб процеси функціонування усіх її компонентів піддавалися алгоритмізації та покроковому відтворенню на ЕОМ з прийнятною точністю з урахуванням їх взаємодії та впливів зовнішнього середовища. Саме програму, що реалізує такий спосіб імітації на ЕОМ процесу функціонування системи, і будемо розглядати як моделюючу програму (алгоритм), основа якої виступає як ІМ досліджуваного об'єкта [1].

Важливим моментом при виборі методу дослідження та подальшому формуванні структури моделі є визначення специфічних особливостей механізмів функціонування усіх її компонентів. Насамперед це стосується визначення того, випадковим чи детермінованим має бути процес функціонування кожного з елементів системи (див. підрозд. 8.4). Зазвичай розглядаються лише ті елементи системи, що формується, і процеси функціонування яких являють собою формалізовані відображення імовірнісних підпроцесів досліджуваного процесу. Функціонування всіх інших елементів системи вважаються детермінованими.

Розв'язання питання про те, імовірнісною чи детермінованою має бути створювана модель, рівнозначне вибору між високою точністю відтворення досліджуваного процесу на ЕВМ (за необхідності багаторазового «програвання» ІМ) та малою тривалістю роботи моделюючої програми (при можливих суттєвих похибках відтворення досліджуваного процесу). Враховуючи те, що не завжди можуть бути відомі закони розподілення випадкових величин та їх параметри, доцільно ввести припущення про детермінований характер процесу функціонування системи та передбачити у майбутній моделюючій програмі можливість швидкого переходу до варіанта ІМ, який відповідає імовірнісній системі. Припущення про детерміно-

ваність системи означає, що при прогнозуванні подій будуть використовуватися величини часових інтервалів, задані у вихідних даних і постійні на всьому інтервалі моделювання.

Зазначена інформація загалом має логіко-математичний характер та зображується у формі сукупності алгоритмів, які описують динаміку функціонування системи. Іншими словами, за ІМ здебільшого виступає її програмна реалізація на ЕОМ, а імітаційне моделювання зводиться до планування та проведення експериментів з моделлю шляхом прогонів програми на деякій множині даних, що є середовищем системи. Інакше кажучи, з *математичної точки зору ІМ може розглядатися як складена з рівнянь, які розв'язуються дослідженням еволюції їх розв'язків на певному відрізку часу. Ця еволюція визначається хронологічною послідовністю подій, які виникають у цій системі і цікавлять досліджувача, і відповідною послідовністю станів системи.* Доцільність в імітаційному моделюванні виникає як на *етапі проектування складної системи* для оцінки правильності прийнятих проектних рішень, так і при *експлуатації для оцінки змін*, які вносяться у систему в процесі переналагодження. Слід також зауважити, що на різних етапах проектування, налагодження та експлуатації у міру уточнення вихідних даних та виявлення нових істотних факторів, що впливають на систему, ступінь деталізації процесів у системі може збільшуватися, а отже, в моделі мають водночас бути присутні блоки з різним ступенем деталізації, що імітують одні й ті самі компоненти досліджуваної чи проектованої системи.

## 11.2. Загальна постановка задачі

Характерною ознакою ГКІС, яка відрізняє її від звичайних виробничих систем, є *принцип організації вантажопотоків*, який передбачає наявність автоматизованої транспортно-складської системи (АТСС). Аналіз роботи звичайних комплексів устаткування з числовим програмним керуванням показує, що значні витрати робочого часу пов'язані із несвоєчасним забезпеченням цього устаткування заготовками, інструментом, оснасткою, тобто з неоптимальністю диспетчеризування матеріальних потоків. Транспортні операції являють собою пряме вираження зв'язків між окремими фазами технологічного процесу, визначаючи тим самим залежність його ефективності від реалізації міжопераційного транспортування об'єктів і засобів виробництва. Отже, для успішного функціонування ГКІС необхідно ще на стадії їх проектування розв'язувати комплекс задач, серед яких однією з найважливіших є формування науково обґрунтованих вимог до технічних характеристик елементів АТСС та структури транспортних мереж (ТрМ), які забезпечують можливість оптимального керування адресуванням матеріальних потоків у ГКІС.

Еквівалентом загальносистемної оцінки якості функціонування АТСС та окремих її елементів є *тривалість простоїв*  $\tau_{\text{прі}}$  кожного модуля основного технологічного обладнання (ТО) (або середня тривалість простою  $\bar{\tau}_{\text{пр}}$  всіх модулів) через затримку транспортування оброблених деталей або відсутність носіїв із заготовками, тобто через неефективність АТСС.

Для розв'язання задачі мінімізації  $\bar{\tau}_{\text{пр}}$  при проектуванні АТСС найбільш ефективним є метод комбінованого моделювання, що ґрунтується на застосуванні *ком-*

**бінованого підходу** – аналітичного моделювання при синтезі вихідних варіантів АТСС та імітаційного моделювання при їх аналізі, уточненні характеристик і виборі найкращого рішення.

Задачу проектування АТСС можна сформулювати так: при заданих початкових даних і технічних обмеженнях створити варіант АТСС, який забезпечує мінімальний простій обробних ресурсів [46]:

$$\tau_{\text{пр}} = F \left( \begin{array}{l} N, m, m_{\text{ц}}, m_{\text{д}}, TM_j, T_{kj}, T_j, D, n_j, \\ p, C_{\text{тр}}, v_{\text{тр}}, M_{\text{тр}}, M_{\text{д}}, M_{\text{ц}}, M_{\text{н}}, z^*, R_{\text{в.н}}, \\ R_{\text{п.к}}, L, A_{\text{тр}}, A_{\text{п.к}}, A_{\text{в.ч}}, A_{\text{к.п}}, A_{\text{к.т}} \end{array} \right) \rightarrow \min \quad (11.1)$$

за умови

$$B_{\text{min}} \leq B_{\text{АТСС}} \leq B_{\text{max}}, \quad (11.2)$$

де  $N$  – кількість партій оброблюваних деталей;  $m, m_{\text{ц}}, m_{\text{д}}$  – відповідно кількість модулів основного ТО, централізованого та децентралізованого нагромаджувачів;  $TM_j$  – технологічні маршрути виготовлення виробів;  $T_{kj}$  – тривалість обслуговування деталей позиціями комплектування;  $T_j$  – поопераційні тривалості циклів виготовлення  $j$ -го виробу,  $j = 1, N$ ;  $D$  – втрати часу на переналагодження ТО;  $n_j$  – кількість партій запуску;  $p$  – компонування ТО;  $C_{\text{тр}}, v_{\text{тр}}, M_{\text{тр}}$  – відповідно кількість, швидкість і місткість транспортних пристроїв (ТрП);  $M_{\text{д}}, M_{\text{ц}}, M_{\text{н}}$  – місткості децентралізованих і централізованих нагромаджувачів і носіїв деталей (касет, полет);  $z^*$  – варіант закріплення ТрП за модулями основного та нагромаджувального ТО;  $L$  – типи позицій завантаження-розвантаження ТрП;  $R_{\text{в.н}}, R_{\text{п.к}}$  – відповідно кількість використовуваних носіїв деталей та позицій комплектування;  $A_{\text{тр}}, A_{\text{п.к}}$  – алгоритми диспетчеризування заявок на обслуговування ТрП та позиціями комплектування;  $A_{\text{в.ч}}, A_{\text{к.п}}$  – алгоритми формування вихідної черги запуску та ситуаційного календарного планування;  $A_{\text{к.т}}$  – алгоритм керування ТрП дискретного типу;  $B_{\text{min}}, B_{\text{АТСС}}, B_{\text{max}}$  – відповідно мінімально допустимі, фактичні та максимальні витрати на АТСС.

З урахуванням задачі проектування визначається *мета створення моделі* ГКІС: розробити інструмент моделювання, який можна використовувати на стадії технічного проектування при аналізі запропонованих варіантів ГКІС, а також на стадії експлуатації системи (для напрацювання заходів щодо підготовки до запуску нової номенклатури виробів).

Виходячи з поставленої задачі проектування ГКІС, мети розробки та умов використання моделі ГКІС, можна визначити такі основні вимоги, які має задовольнити створювана модель:

- можливість аналізу варіантів ГКІС, АТСС яких розрізняють як кількісно (кількістю модулів та їх характеристиками), так і якісно (типом засобів транспортування і принципом нагромадження матеріальних потоків);
- можливість оцінки впливу на продуктивність ГКІС будь-якого з параметрів системи.

Аналітичне моделювання АТСС виконується за такими основними етапами:

- 1) одержання початкової сукупності варіантів компонувань ТО та структури ТрМ з урахуванням обмежень площі виробничого приміщення;
- 2) розміщення ТО в межах вибраних на першому етапі варіантів з урахуванням вимог технології виробництва заданої номенклатури виробів (прийма-

те на цьому етапі рішення є найменш стійким до змін номенклатури виробів або розмірів партій їх запуску, що робить недоцільним пошук оптимального розв'язку задачі розміщення ТО);

- 3) розрахунок кількісних характеристик АТСС ( $C_{\text{тр}}, v_{\text{тр}}, M_{\text{н}}, R_{\text{п.к.}}$ );
- 4) визначення оптимальної дисципліни диспетчеризування заявок на обслуговування АТСС.

Результатом виконання комплексу заходів щодо аналітичного моделювання є ряд альтернативних варіантів АТСС, кожний з яких характеризується деякою сукупністю параметрів:

$$\langle C_{\text{тр}}, v_{\text{тр}}, E_{\text{н}}, N_{\text{п.к.}}, k, A_{\text{тр}}, A_{\text{п.к.}} \rangle. \quad (11.3)$$

Ці параметри дають інформацію про можливі характеристики АТСС, яка має такі особливості:

- точність одержаних результатів визначається адекватністю відображення застосованими математичними моделями властивостей досліджуваної системи. В загальному випадку вірогідність результатів не є достатньою для прийняття остаточного рішення через наявність ряду припущень, які не реалізуються на практиці, але зумовлені застосуванням аналітичного методу дослідження. Наприклад, розрахунки з використанням апарату теорії масового обслуговування потребують введення нездійснених на практиці жорстких припущень щодо характеру потоків заявок та інтенсивностей їх обслуговування, а також напівемпіричних апроксимацій без урахування при цьому динаміки виробничого процесу. Саме тому постає необхідність комплексного оцінювання процесу функціонування АТСС, що характеризується сукупністю різних параметрів;
- найбільш повні дослідження й оцінювання процесу функціонування ГКІС можуть бути проведені, якщо відомо явні аналітичні залежності, які пов'язують шукані характеристики з початковими умовами, параметрами та змінними системи. Проте дістати такі залежності, які б пов'язували цільову функцію (11.1) проектування АТСС хоча б з наведеними в (11.3) параметрами, неможливо. Застосування в аналогічних ситуаціях суттєвих спрощень первинної моделі або декомпозиція задач оцінювання ефективності функціонування АТСС на ряд підзадач з меншою кількістю змінних призводить до втрати точності проектування, що може зумовити появу значної розбіжності розрахункових і реальних показників системи. При цьому не враховується сукупний вплив зазначених в (11.1) параметрів на показники ефективності ГКІС, а також залишаються відкритими питання оцінювання ефективності вибраних стратегій керування системою, наприклад, алгоритмів формування вихідних черг запуску деталей, ситуаційного календарного планування, диспетчеризування заявок на обслуговування АТСС, керування ТрП та ін. Водночас внаслідок високого ступеня ризику, пов'язаного з великим обсягом капітальних вкладень, проектування систем такої складності, як ГКІС, висуває такі вимоги до його точності, які на цьому етапі перевищують можливості аналітичних методів їх проектування.

Таким чином, при проектуванні АТСС виникає задача оцінювання ефективності прийнятих на етапах аналітичного моделювання рішень. Ця задача розв'язується методами *імітаційного моделювання* (ІМ) АТСС, сутністю якого є комплексний аналіз одержаних аналітичних параметрів (11.3) з урахуванням всієї сукупності

(11.1) параметрів, які характеризують досліджувану систему, що виконується на етапах визначення раціональної місткості централізованого і децентралізованого нагромаджувачів  $M_{ц}$  і  $M_{д}$ , а також ряду інших характеристик АТСС ( $z^*$ ,  $L$ ,  $A_{тр}$ ,  $R_{в,н}$ ,  $R_{п,к}$ ,  $M_{тр}$ ); здійснення ітераційного процесу уточнення параметрів АТСС і вибір варіанта, який забезпечує необхідні (або досить близькі до необхідних) значення цільової функції (11.1).

### 11.3. Аналітичне моделювання процесів в АТСС

Розглянемо суть етапів аналітичного моделювання процесів в АТСС.

На **першому етапі** розробки технічного проекту АТСС здійснюється формування конкуруючих варіантів компоновки устаткування та структури ТрМ, з яких згодом за прийнятим критерієм вибирається найбільш раціональний варіант. Глобальний критерій (11.1) проектування АТСС трансформується на цьому етапі в необхідність вибору раціональних варіантів компоновки, які забезпечують мінімум середнього шляху  $\bar{S}_a = S_a / \nu$  при використанні одноадресної транспортної операції, де  $S_a$  – сумарне переміщення транспортного модуля за всіма одноадресними операціями при  $a$ -й структурі ТрМ та компоновці устаткування;  $\nu = m(m-1)$  – кількість одноадресних операцій при заданій кількості  $m$  елементів устаткування – пунктів адресування транспортного модуля.

Вибір як основного критерію забезпечення мінімуму середнього шляху  $\bar{S}_a$  переміщення транспортного модуля зумовлено тим, що при постійній швидкості руху (за винятком дільниць розгону та гальмування) саме середній шлях, який є функцією структури ТрМ та компоновки устаткування, визначає тривалість транспортних операцій. Остання ж безпосередньо впливає на завантаження устаткування та продуктивність ГКС загалом.

Виділяють два основних варіанти побудови ТрМ: 1) розімкнена ТрМ з двобічним переміщенням вантажів; 2) замкнена ТрМ з одно- чи двобічним переміщенням вантажів. Цим варіантам організації ТрМ відповідають типові компоновки *транспортно-складської системи* (ТСС) (рис. 11.2).

Кожна з них забезпечує певний середній шлях  $\bar{S}_a$  виконання одноадресної операції, який може бути виражений через середню довжину  $\bar{l}$  переміщення транспортного модуля між двома сусідніми модулями устаткування. Зокрема, для компонок з лінійним розташуванням устаткування відповідно з розімкненою (рис. 11.2, а) та із замкненою (рис. 11.2, б, в, д) ТрМ сумарне переміщення визначається так:

$$S_1 = \begin{cases} 2 \sum_{\sigma=1}^{m-m/2} \left( \sum_{k=1}^{m-\sigma} k\bar{l} + \sum_{k=1}^{\sigma-1} k\bar{l} \right), & \text{якщо } m \text{ парне;} \\ 2 \sum_{\sigma=1}^{m-(m+1)/2} \left( \sum_{k=1}^{m-\sigma} k\bar{l} + \sum_{k=1}^{\sigma-1} k\bar{l} \right) + 2 \sum_{k=1}^{m-(m+1)/2} k\bar{l}, & \text{якщо } m \text{ непарне} \end{cases} \quad (11.4)$$

та

$$S_2 = S_1 + 2q_1q_2(l_{1-2} - \bar{l}), \quad (11.5)$$

відповідно, причому  $q_1, q_2$  – кількість модулів устаткування в першому та другому рядах відповідно;  $l_{1-2}$  – відстань між рядами;  $k$  – кількість адрес операції (або позицій у ній);  $\sigma = 1, m$  – поточна кількість модулів ТО.

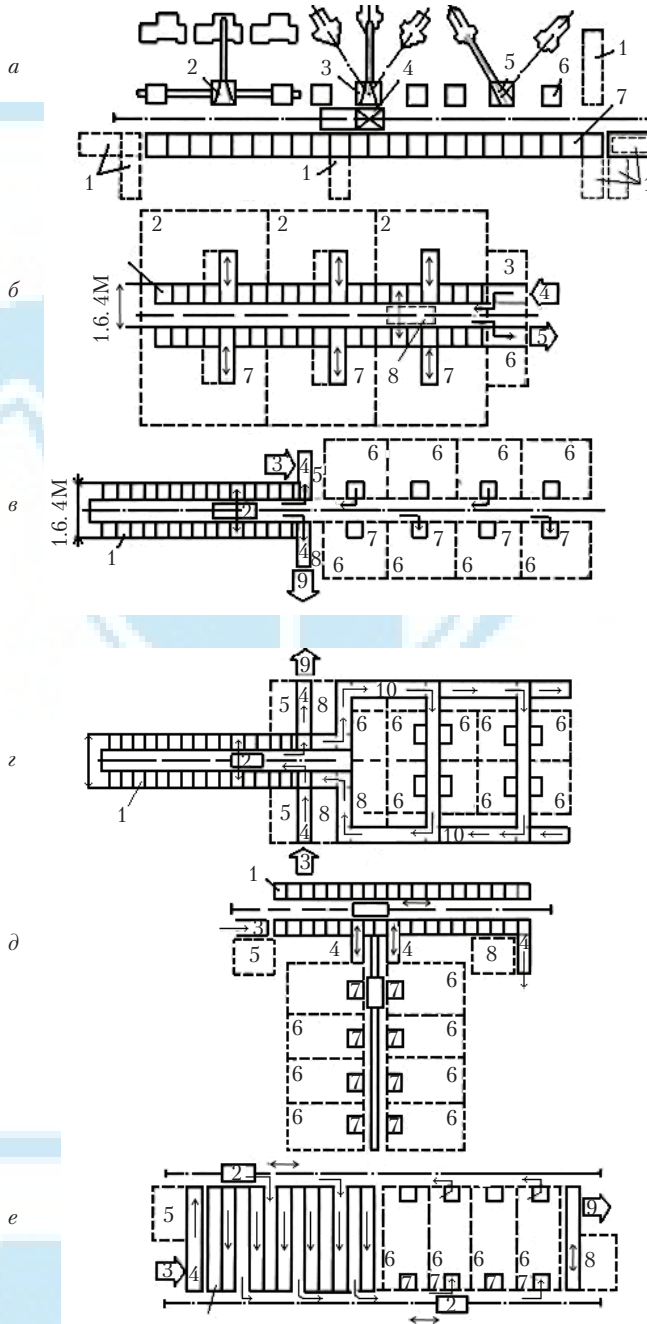


Рис. 11.2. Типові компонування ТСС

Аналогічно обчислюються й сумарні переміщення транспортного модуля для випадків лінійних багаторядних із розімкненою (рис. 11.2, б, в, д) і замкнутою (рис. 11.2, е) та кільцевою (рис. 11.2, з) ТрМ (розрахункові залежності для цих випадків наведено у праці [99]).

Значення у загальному випадку визначається розмірами модулів устаткування, відстанню між ними та видом зв'язку устаткування з ТрМ, який може бути внутрішнім або зовнішнім. Для внутрішнього зв'язку  $\bar{l}_{\text{вн}} = \bar{l}_{\text{н.с}} + \bar{l}_{\text{сер}}$ , де  $\bar{l}_{\text{н.с}}$  – середня для кожного типу устаткування норма відстані між транспортними модулями;  $\bar{l}_{\text{сер}}$  – середня довжина орієнтованого до ТрМ боку модуля. Для зовнішнього зв'язку  $\bar{l}_3 = \bar{l}_{\text{н.с}} + \bar{l}_{\text{сер}} + \bar{l}_{\text{н.т}}$ , де  $\bar{l}_{\text{н.т}}$  – середня для цього типу устаткування норма відстані між пристроями стикування та транспортними лініями.

Суть **другого етапу** розробки технічного проекту АТСС полягає в розміщенні устаткування в межах вибраних компонок з урахуванням вимог, які накладаються вихідною номенклатурою виробів, та обмежень можливості встановлення  $d_c$ -го устаткування на  $k$ -ту позицію.

Перші відображаються матрицею частоти:

$$\mathbb{C} = \left\| \mathbb{C}_{d_b d_c} \right\| \left( (b, c = \overline{1, m}; b \neq c) \right) \quad (11.6)$$

одноадресних транспортних операцій, зумовлених необхідністю переміщення об'єктів виробництва операціями їх технологічних маршрутів.

Другі формуються у вигляді матриці умов розміщення:

$$\mathbb{P} = \left\| \mathbb{P}_{g d_b} \right\| \left( (b, g = \overline{1, m}) \right), \quad (11.7)$$

де  $\mathbb{P}_{g d_b} \neq 0$  означає заборону розміщення  $d_b$ -го модуля на  $g$ -й позиції.

Позначимо через  $S_{gh}$  відстань між  $g$ -ю та  $h$ -ю позиціями розміщення устаткування. Введемо змінні  $x_{g d_b}$  таким чином:  $x_{g d_b} = 1$ , якщо  $d_b$ -й модуль устаткування розміщено на  $g$ -й позиції;  $x_{g d_b} = 0$ , якщо цього модуля там немає. Тоді задача зводиться до знаходження такого розміщення модуля, при якому досягається мінімум цільової функції:

$$Q = \sum_{b, c, g, h}^m S_{gh} \mathbb{C}_{d_b d_c} x_{g d_b} x_{h d_c} \quad (11.8)$$

за умови

$$\sum_{g=1}^m x_{g d_b} = 1; \quad \sum_{b=1}^m x_{g d_b} = 1; \quad x_{g d_b} = \{0, 1\}; \quad \mathbb{P}_{g d_b} = 0; \quad d = \overline{1, m}, \quad g = \overline{1, m}.$$

Функція (11.8) відображає квадратичну задачу призначень. Її розв'язують використанням алгоритмів пошуку в ширину і глибину (див. підрозд. 9.1).

На **третьому етапі** при розрахунку кількісних характеристик АТСС ( $C_{\text{тр}}$ ,  $v_{\text{тр}}$ ,  $M_{\text{н}}$ ,  $R_{\text{п.к}}$ ) критерій мінімізації простоїв основного ТО через неефективність АТСС набуває вигляду:

$$\bar{t}_{\text{тр}} = F_1(C_{\text{тр}}, v_{\text{тр}}, M_{\text{н}}, R_{\text{п.к}}) \rightarrow \min, \quad (11.9)$$

де

$$F_1(C_{\text{тр}}, v_{\text{тр}}, M_{\text{н}}, R_{\text{п.к}}) = \bar{T}_{\text{п1}} + \bar{T}_{\text{п2}} + \bar{T}_{\text{п.к}} - \bar{T}_{\text{ц}} \quad (11.10)$$

за умови (11.2). Тут  $\bar{T}_{\text{п1}}$ ,  $\bar{T}_{\text{п2}}$  – середні тривалості постачання одного носія відповідно від модуля основного ТО на позицію комплектації носіїв і назад;  $\bar{T}_{\text{п.к}}$  – середній час обслуговування одного носія на цій позиції;  $\bar{T}_{\text{ц}}$  – середня тривалість циклу обробки заготовок на одному носії.

Розглянемо послідовність визначення параметрів  $C_{\text{тр}}$ ,  $v_{\text{тр}}$ ,  $M_{\text{н}}$ ,  $R_{\text{п.к}}$ , виходячи із задоволення умови своєчасного забезпечення заготовками ТО

$$\bar{T}_{\text{ц}} > \bar{T}_{\text{п1}} + \bar{T}_{\text{п2}} + \bar{T}_{\text{п.к}}, \quad (11.11)$$

яка впливає з (11.9) і (11.10). Початковими даними при цьому є: кількість  $m$  елементів основного ТО; кількість  $N$  партій оброблених виробів, маршрути обробки



їх та значення  $T_{jf}$  тривалостей циклів обробки одного виробу  $j$ -ї партії на  $f$ -й операції технологічного маршруту; матриця  $S_{gh}$  відстаней між елементами технологічної підсистеми – пунктами адресування матеріальних потоків; середній час передавання одного носія з ТрП ( $\bar{t}_3$ ) і назад ( $\bar{t}_4$ ); середній час установаження на носій ( $\bar{t}_{y,j}$ ) та розвантаження з нього ( $\bar{t}_{p,j}$ ) одного виробу  $j$ -ї партії.

Отже, час, який витрачається на постачання одного носія від ТО до модуля нагромадження (МНК) і назад (відповідно,  $\bar{T}_{п1}$ ,  $\bar{T}_{п2}$ ), можна записати у вигляді:

$$\bar{T}_{п1} = \bar{T}_{п2} = \bar{t}_1 + \bar{t}_2 + \bar{t}_3 + \bar{t}_4, \quad (11.12)$$

де  $\bar{t}_1$  – середній час чекання ТрП;  $\bar{t}_2$  – середній час транспортування носія при виконанні одноадресної операції.

Тривалість обслуговування одного носія на МНК визначається виразом:

$$\bar{T}_{п.к} = \bar{t}_5 + (\bar{t}_6 + \bar{t}_7) M_n, \quad (11.13)$$

де  $\bar{t}_5$  – середній час чекання звільнення позиції комплектування;  $\bar{t}_6$ ,  $\bar{t}_7$  – відповідно середній час установаження на носій та розвантаження з нього виробу:

$$\bar{t}_6 = \left( \sum_{j=1}^N \bar{t}_{y,j} \right) / N; \quad \bar{t}_7 = \left( \sum_{j=1}^N \bar{t}_{p,j} \right) / N. \quad (11.14)$$

Аналіз виразів (11.12) та (11.13) показує, що шукані параметри АТСС впливають на  $\bar{t}_1$ ,  $\bar{t}_5$ ,  $\bar{t}_2$ ; при цьому  $\bar{t}_2$  в найпростішому випадку визначається як  $\bar{t}_2 = S_a / v_{тр}$ . Для визначення значень  $\bar{t}_1$ ,  $\bar{t}_5$  при фіксованих значеннях  $C_{тр}$ ,  $v_{тр}$ ,  $M_n$ ,  $R_{п.к}$  користуються положеннями ТМО, причому сукупність об'єктів «ТО–ТрП», «ТО–МНК» розглядаються при цьому як системи масового обслуговування типу  $M/M/C_{тр}$  і  $M/M/R_{п.к}$ .

Нехай на входи цих систем надходить, наприклад, пуасонівський потік заявок з інтенсивністю  $\lambda = 1/\bar{T}_{ц}$ , де

$$\bar{T}_{ц} = M_n \sum_{j=1}^N \left[ \left( \sum_{f=1}^{v_j} T_{jf} \right) / v_j \right] / N$$

середня тривалість циклу обробки заготовок на одному носії ( $v_j$  – кількість операцій технологічного маршруту  $j$ -ї партії виробів). Вважатимемо, що розподіл часу обслуговування заявок ТрП та позиціями комплектування носіїв змінюється за показниковим законом з параметрами  $\mu_{тр} = 1/\bar{T}_{тр}$ ,  $\mu_{п.к} = 1/\bar{T}_{п.к}$ , де  $T_{тр} = 2(\bar{t}_2 + \bar{t}_3 + \bar{t}_4)$  – середня тривалість обслуговування однієї заявки ТрП;  $T_{п.к} = \bar{t}_8 + M_n = \bar{t}_8 + M_n(\bar{t}_6 + \bar{t}_7)$  – середня тривалість обслуговування однієї заявки МНК ( $\bar{t}_8$  – середній час передавання носія з вхідного нагромаджувача МНК на робочий стіл і назад).

Ймовірність надходження  $K$  заявок на обслуговування ТрП можна визначити за формулами:

$$P_K^1 = a_1 (\lambda / \mu_{тр})^K P_0^1 \quad \text{при } 1 \leq K \leq C_{тр} \quad (11.15)$$

або

$$P_K^1 = a_2 (\lambda / \mu_{тр})^K P_0^1 \quad \text{при } C_{тр} \leq K \leq m. \quad (11.16)$$

Аналогічно визначається ймовірність надходження  $K$  заявок на обслуговування позицією комплектування носіїв:

$$P_K^2 = a_1 (\lambda / \mu_{п.к})^K P_0^2 \quad \text{при } 1 \leq K \leq R_{п.к} \quad (11.17)$$

або

$$P_K^2 = a_3 (\lambda / \mu_{п.к})^K P_0^2 \quad \text{при } R_{п.к} \leq K \leq m. \quad (11.18)$$

Імовірність того, що всі  $m$  елементів ТО працюють, записується як  $\sum_{k=n}^m P_k = 1$ , звідки

$$P_0^1 = \left[ 1 + \sum_{k=1}^{C_{\text{тр}}} a_1 (\lambda / \mu_{\text{тр}})^k + \sum_{k=R_{\text{тр}}+1}^m a_2 (\lambda / \mu_{\text{тр}})^k \right]^{-1}; \quad (11.19)$$

$$P_0^2 = \left[ 1 + \sum_{k=1}^{R_{\text{п.к}}} a_1 (\lambda / \mu_{\text{п.к}})^k + \sum_{k=R_{\text{п.к}}+1}^m a_3 (\lambda / \mu_{\text{п.к}})^k \right]^{-1}, \quad (11.20)$$

де  $a_1 = m! / [k!(m-k)!]$ ;  $a_2 = m! / [C_{\text{тр}}^{k-C_{\text{тр}}} C_{\text{тр}}!(m-k)!]$ ;  $a_3 = m! / [R_{\text{п.к}}^{k-R_{\text{п.к}}} R_{\text{п.к}}!(m-k)!]$ .

Тоді середню кількість елементів ТО, що чекають чергового обслуговування ТрП, можна визначити за формулою:

$$M_1 = \sum_{K=C_{\text{тр}}+1}^m (K - C_{\text{тр}}) P_K^1. \quad (11.21)$$

Формула для розрахунку кількості елементів *технологічного обладнання* (ТО), які чекають обслуговування позиціями комплектування носіїв, має вигляд:

$$M_2 = \sum_{K=R_{\text{п.к}}+1}^m (K - R_{\text{п.к}}) P_K^2. \quad (11.22)$$

Використовуючи вирази, які пов'язують середню довжину черги та середній час чекання обслуговування в черзі, можна визначити середній час чекання ТрП  $\bar{t}_1 = M_1 / \lambda$  та середній час чекання звільнення позиції комплектування  $\bar{t}_5 = M_2 / \lambda$ .

Якщо за знайденими значеннями  $C_{\text{тр}}$ ,  $v_{\text{тр}}$ ,  $M_{\text{н}}$ ,  $R_{\text{п.к}}$  умова (11.11) не виконується, то розрахунки треба повторити при нових значеннях  $C_{\text{тр}}$ ,  $v_{\text{тр}}$ ,  $M_{\text{н}}$ ,  $R_{\text{п.к}}$ , які задовольняють умову (11.2). Слід зауважити, що при використанні одномісних ТрП ( $M_{\text{тр}}=1$ ) та одномісних децентралізованих нагромаджувачів ( $M_{\text{дс}}=1$ ,  $\sigma=1$ ,  $m$ ) основна умова (11.11) набуває вигляду:

$$\bar{T}_{\text{п1}} + \bar{T}_{\text{п2}} + \bar{T}_{\text{ц}} > \bar{T}_{\text{п.к}};$$

до того ж після закінчення обробки чергового носія завжди виникатимуть простоювання ТО протягом часу  $T_{\text{пр}} = T_{\text{п1}} + T_{\text{п2}}$ . Коли  $M_{\text{тр}} > 1$  та  $M_{\text{дс}} = 1$ , якщо заявки на транспортне обслуговування подаються після закінчення обробки носіїв, неминучі простоювання ТО протягом часу  $\bar{t}_{\text{пр}} = \bar{t}_2 + \bar{t}_3 + \bar{t}_4$ .

Використання апарату ТМО дає змогу наближено визначити вплив динамічних характеристик (швидкості, прискорень) ТрП і вибрати більш прийнятні з них. Ця можливість реалізується при розрахунку середнього часу  $\bar{t}_2$  транспортування носія.

Усі транспортні операції в ГКІС за кількістю адрес ТО, що беруть участь у виконанні однієї заявки на обслуговування, поділяються на **одно-**, **дво-** й **багатоадресні**. Через те, що будь-яка багатоадресна операція може бути подана у вигляді сукупностей одноадресних, середній час обслуговування заявки можна визначити виходячи з тривалості виконання одноадресної операції. Прикладом останньої може бути постачання ТрП носія з централізованого нагромаджувача до МО.

Середній час транспортування носія при виконанні *одноадресної операції*

$$\bar{t}_2 = \bar{t}_{\text{р}} + \bar{t}_{\text{м}} + \bar{t}_{\text{г}} + \bar{t}_{\text{зн}},$$

де  $\bar{t}_{\text{р}}$  – час розбігу ТрП з прискоренням  $a_{\text{р}}$  до швидкості  $v_{\text{тр}}$ ;  $\bar{t}_{\text{м}}$  – час його руху з швидкістю  $v_{\text{тр}}$ ;  $\bar{t}_{\text{г}}$  – час гальмування ТрП з прискоренням  $a_{\text{г}}$  до зниженої швидкості  $v_{\text{зн}}$ . При цьому

$$\bar{t}_p = \bar{v}_{tp} / a_p; \bar{t}_m = S_m / v_{tp}; \bar{t}_r = (\bar{v}_{tp} - \bar{v}_{zn}) / a_r; \bar{t}_{zn} = S_{zn} / \bar{v}_{zn},$$

де  $S_m, S_{zn}$  — переміщення ТрП на маршовій та зниженій швидкостях відповідно.

Величини  $t_p, t_r$  залежать від динамічних параметрів ТрП. Цими ж параметрами визначається й значення  $t_{zn}$ . Справді, для переведення ТрП у режим гальмування використовують шляхові датчики, які встановлюють у реперних точках на заданій відстані  $S_{т.п}$  від елементів технологічної підсистеми ГКІС. Зазвичай ця відстань для всіх елементів однакова і тому

$$S_{zn} = S_{т.п} - S_r = S_{т.п} - (v_{tp}^2 - v_{zn}^2) / (2a_r),$$

де  $S_r$  — переміщення ТрП при гальмуванні.

Переміщення ТрП під час його руху з швидкістю  $v_{tp}$  визначається виразом:

$$S_m = S_{o.o} - S_p - S_{т.п},$$

де  $S_{o.o}$  — відстань, яку проходить ТрП при виконанні одноадресної операції.

Враховуючи викладене вище, можна визначити час транспортування носія для випадку  $\bar{S}^{(1)} > (S_p + S_{т.п})$  (одноадресної транспортної операції):

$$\begin{aligned} \bar{T}_{tp}^{(1)} = v_{tp} / a_p + (\bar{S}^{(1)} - S_p - S_{т.п}) / v_{tp} + \\ + (v_{tp} - v_{zn}) / a_r + S_{zn} / v_{zn} = \bar{S}^{(1)} / v_{tp} + c, \end{aligned} \quad (11.23)$$

де  $\bar{S}^{(1)}$  — середнє переміщення ТрП при виконанні цієї операції;  $c$  — стала, що залежить від динамічних характеристик ТрП, причому

$$c = v_{tp} / (2a_r) - S_{т.п} / v_{tp} + [2a_r S_{т.п} - (v_{tp} - v_{zn})^2] / (2a_r v_{zn}).$$

Для *двоадресної транспортної операції*, коли  $\bar{S}^{(2)} = S_p + S_{т.п}$ , час транспортування носія визначається виразом:

$$\bar{T}_{tp}^{(2)} = \sqrt{2(\bar{S}^{(2)} - S_{т.п})} / a_p + [2a_r S_{т.п} - (v_{tp} - v_{zn})^2] / (2a_r v_{zn}). \quad (11.24)$$

У випадку  $\bar{S}^{(b)} < (S_p + S_{т.п})$ , коли має місце *багатоадресна транспортна операція*, відповідний вираз для часу транспортування носія має вигляд:

$$\bar{T}_{tp}^{(b)} = \sqrt{2(\bar{S}^{(b)} - S_{т.п})} / a_p + [2a_r S_{т.п} - (v_p - v_{zn})^2] / (2a_r v_{zn}), \quad (11.25)$$

де  $v_p = \sqrt{2a_p(\bar{S}^{(b)} - S_{т.п})}$  — швидкість ТрП під час його розбігу;  $\bar{S}^{(2)}, \bar{S}^{(b)}$  — середні переміщення ТрП у транспортних операціях відповідних видів.

Суть *четвертого етапу* проектування АТСС полягає у виборі дисципліни диспетчеризування заявок на обслуговування. В умовах ГКІС кожна вимога на обслуговування під час прийняття рішення про вибір чергової заявки характеризується тривалістю її обслуговування та резервом виконання, тобто часом, протягом якого заявка може бути обслужена без простоювання з вини АТСС.

Нехай під час звільнення транспортного модуля (ЕРК) в черзі на транспортне обслуговування є  $n$  заявок, кожна з яких характеризується резервом часу  $T_R$  до початку простою джерела заявки та тривалістю її обслуговування  $\bar{t}_3^{ob} = \bar{t}_2 + \bar{t}_3 + \bar{t}_4 + t_{ЕРК}$ , де  $t_{ЕРК}$  — час переміщення транспортного модуля ЕРК від початкової адреси заявок до її кінцевої адреси.

Критерій ефективності (11.1) функціонування АТСС потребує організації обслуговування заявок у такій послідовності, щоб сумарна тривалість можливих простоювань адрес-джерел заявок була мінімальною:

$$\theta = \sum_{k=1}^n y_k \rightarrow \min, \quad (11.26)$$

де

$$y_k = \begin{cases} T_R - \sum_{z=i}^k t_z^{об} & \text{при } T_R - \sum_{z=i}^k t_z^{об} < 0; \\ 0 & \text{при } T_R - \sum_{z=1}^k t_z^{об} \geq 0. \end{cases}$$

Цю задачу можна віднести до класу задач теорії розкладу, тобто впорядкування для системи з  $n$  заявок й одного обслуговуваного приладу. Оскільки місткість  $M_{тр}$  транспортного модуля зазвичай значно менша від кількості заявок у черзі і існують можливості надходження нових конкуруючих заявок на час чергового звільнення ЕРК, немає необхідності впорядковувати всю чергу заявок. Згідно з цим для розв'язування задачі (11.26) застосовується алгоритм, який забезпечує вибір заявки, яка б очолювала впорядковану за (11.25) чергу без повного впорядкування початкової черги заявок.

#### 11.4. Імітаційне моделювання процесів в АТСС

Припущення щодо детермінованого характеру процесів функціонування ГКІС у цілому та АТСС зокрема означає, що при прогнозуванні подій мають використовуватися величини часових інтервалів, заданих у вихідних даних як постійні на всьому інтервалі моделювання. Інакше кажучи, *ГКІС та її складові розглядаються при цьому як дискретні динамічні системи з детермінованим часом подій*. Саме ж імітаційне моделювання використовується як **метод аналізу ГКІС**.

Етапами проектування АТСС, які виконуються на стадії її імітаційного моделювання, передбачається формування відсутніх параметрів  $\langle M_{тр}, M_{д}, M_{ц}, z^*, R_{в.п}, L, A_{к.т} \rangle$  останньої, оцінювання й уточнення раніше синтезованих варіантів і вибір найкращого з них з урахуванням усіх перелічених в (11.1) параметрів.

Таким чином, **метою імітаційного моделювання є розробка інструмента дослідження процесів в АТСС** для використання як на стадії технічного проектування ГКІС, так і в процесі переналадження системи на нову номенклатуру оброблюваних виробів.

Як **об'єкт імітаційного моделювання** виступає технологічна підсистема ГКІС, у якій на основі вибраної технології передбачається функціонування матеріального (заготовок, інструменту, оснастки) та інформаційного потоків.

Незважаючи на різновид існуючих ГКІС, які відрізняються одна від одної функціональним призначенням, рівнем автоматизації, продуктивністю, техніко-економічними та іншими показниками, можна виділити в них сукупність загальних ознак:

- структурно-технологічна підсистема складається із скінченної множини  $M = \{m_k | k = \overline{1, \rho}\}$  оснащених пристроями автоматичного подавання об'єктів виробництва та керованих від ЕОМ модулів ТО, що реалізують технологічні операції та допоміжні процеси;
- до складу ГКІС входять АТСС, функції яких полягають у забезпеченні такого автоматичного переміщення матеріальних потоків, за яким мінімізуються простоювання основного ТО;

- виробничий процес у ГКІС забезпечується скінченною множиною структурних зв'язків  $G = \{g_i | i = \overline{1, v}\}$  між модулями ТО. Ці зв'язки мають різну природу й забезпечують передавання об'єктів виробництва (заготовок, напівфабрикатів, готових виробів, інструменту, оснастки), а також довідкової та керуючої інформації. Носієм матеріальних зв'язків у ГКІС є АТСС;
- на матеріальні зв'язки множини  $M$  модулів ТО можуть накладатись обмеження у вигляді різних варіантів закріплення устаткування АТСС за окремими групами основного ТО, тобто

$$g_i \cap g_j = \emptyset, i \neq j, i \in M, j \in M;$$

- кожний модуль ТО має вхід і вихід для приймання й видачі заготовок, виробів, інформації тощо;
- функціонування ГКІС супроводжується виникненням черг заявок на обслуговування, довжина яких не обмежується;
- усі процеси обслуговування заявок є скінченними за часом, і їх переривання неприпустиме;
- рух предметів праці за обробними ресурсами здійснюється відповідно до множини їх технологічних маршрутів  $TM = \{TM_i | i = \overline{1, \vartheta}\}$ .

Згідно з метою створення й умовами використання ІМ її *моделюючий алгоритм* має задовольняти такі вимоги:

- відображати процес функціонування ГКІС в усій її складності;
- забезпечувати системні дослідження ГКІС з урахуванням взаємодії та взаємозв'язку всіх перелічених в (11.1) виробничих факторів. При цьому локальною вимогою є забезпечення аналізу проектів ГКІС, АТСС яких розрізняються як кількісно, так і якісно (типом засобів транспортування та принципом нагромадження матеріальних потоків);
- забезпечувати одержання таких результатів, які давали б змогу оцінювати початкові дані;
- реалізувати такі витрати часу проектувальника при індивідуальному настроюванні ІМ на конкретну структуру ГКІС, які не повинні перевищувати обсягу, потрібного для підготовки та введення заданої в (11.1) інформації;
- забезпечувати адаптацію ІМ до змін складу вхідних даних ГКІС, у тому числі перебудовою її структури.

При застосуванні ІМ як методу дослідження ГКІС **змістом моделювання роботи ГКІС є імітація руху матеріальних потоків за обробними ресурсами**, під яким розуміють як обслуговування (обробку транспортування) останніх, так і зберігання в перервах між обслуговуваннями. Згідно з цим при моделюванні необхідно імітувати роботу модулів ГКІС, які задіяні у матеріальних потоках. До таких модулів належать верстати, складальне устаткування, контрольно-вимірювальні прилади, склади, операційні нагромаджувачі, позиції комплектування, а також транспортні засоби – ЕРК в системах транспортування дискретного чи конвеєрного типу.

Процес функціонування будь-якого з перелічених модулів можна відобразити як єдиний, здійснюваний у логічній послідовності процес обробки взаємопов'язаних операцій: 1) приймання компонентів матеріального потоку із зовнішнього стосовно цього модуля ГКІС середовища; 2) зберігання компонентів матеріального потоку до і після обслуговування; 3) обслуговування компонентів матеріального потоку; 4) видача їх у зовнішнє середовище.

Різниця між модулями ГКІС полягає в кількісних їх характеристиках, а також у неповному збігу списку перелічених операцій, зумовленому якісною відмінністю їх складових між собою. Наприклад, із списку операцій, характерних для складально-го устаткування, можна виключити операції обслуговування заявок.

**Розробка моделюючого алгоритму.** Структуру ІМ можна зобразити у вигляді схеми, наведеної на рис. 11.3. Один з модулів тут є керуючим, а решта – базовими. Робота модульної програми зводиться до послідовного виконання своїх функцій базовими модулями, порядок запуску яких визначається модулем керування. Специфіка розв’язуваної при імітації задачі враховується комбінуванням існуючих програмних одиниць, які відображають те чи інше джерело багатоваріантності початкових даних.

**Прогнозування подій та керування ними.** Для імітації руху компонентів матеріального потоку за операціями їх технологічних маршрутів необхідно організувати в моделі послідовну обробку деякої сукупності подій. Цю сукупність утворюють події, які визначаються станами елементів технологічної підсистеми ГКІС при проходженні  $j$ -ю деталлю  $i$ -ї партії  $n$ -ї операції її технологічного маршруту, тобто закінченням таких операцій: *П1* – обслуговування на модулі основного ТО; *П2* – передавання в децентралізований вихідний нагромаджувач; *П3* – обміну між транспортним модулем і устаткуванням наступної групи; *П4* – переміщення; *П5* – зміни програми (обробки, складання, контролю, маркування тощо) та інструменту; *П6* – обслуговування на позиції комплектації; *П7* – передавання в робочу зону; *П8* – чекання (простоювання).

Події *П1–П7* є активними, а подія *П8* – пасивною. Обробка здійсненої активної події може спричинити здійснення пасивної події, але не навпаки, у зв’язку з чим прогнозування пасивних подій у моделі виконувати недоцільно. Подія *П7* також не прогнозується, проте тривалість передавання деталі в робочу зону враховується при прогнозуванні події *П1*.

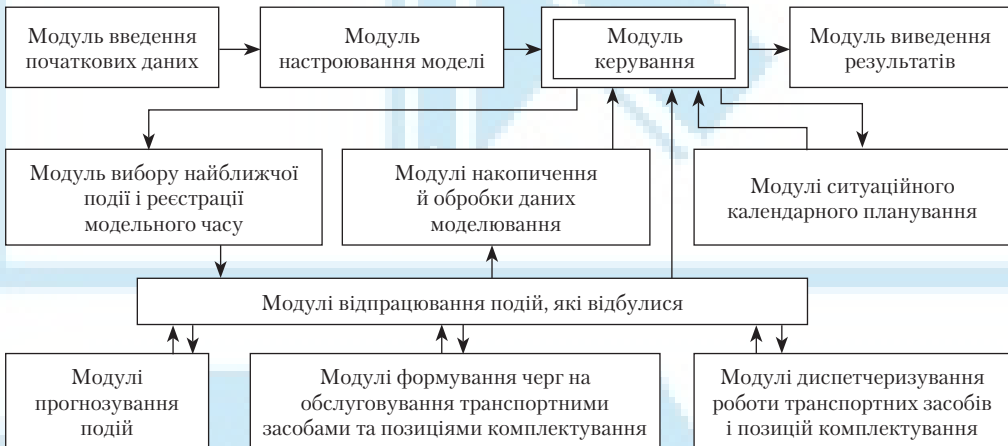


Рис. 11.3. Структура ІМ

Модульна організація процесу обслуговування подій у моделі, включаючи і їх прогнозування, впорядкування та обробку, повинна передбачати необмежену деталізацію списку подій за бажанням користувача, що забезпечує можливість адаптації моделі до зміни поставленої мети дослідження.

Механізм моделювання може ґрунтуватися на двох способах керування подіями — *безумовному* й *умовному*. Частіше використовують перший з них, відповідно до якого в момент настання деякої події можна визначити (спрогнозувати) критичний момент наступної події, логічно пов'язаний з появою цієї. Прогнозування подій реалізується спеціальними модулями, що здійснюють розрахунок моменту появи подій та формування списку її атрибутів. Після закінчення цих операцій вступає в дію модуль упорядкування подій, який розміщує цю подію в календарі згідно із значенням її критичного моменту.

**Означення 11.5.** *Календар* — це список упорядкованих за системним часом і пріоритетом подій.

Задавання пріоритету необхідне, коли на один і той самий час спрогнозовано здійснення кількох подій; при цьому перевагу має та подія, в якій більший пріоритет. При рівності критичних моментів і пріоритетів модуль вибору подій може працювати за правилом FIFO. Використання значення пріоритету як одного з атрибутів подій пов'язане з орієнтуванням ІМ на опис паралельно здійснених ТП в умовах послідовно виконуваного моделювання на ЕОМ.

**Механізм системного часу.** Функціонування ІМ відбувається в штучному (не в реальному та машинному) часі із забезпеченням потрібної за логікою роботи ГКІС послідовності подій та відповідних часових проміжків між ними. При цьому елементи машинної моделі діють послідовно, бо реалізуються вони за допомогою ЕОМ послідовної дії, в той час як модулі реальної ГКІС функціонують одночасно (паралельно). Через те, що в різних частинах об'єкта моделювання події часто виникають одночасно, для зберігання адекватності причинно-наслідкових часових зв'язків необхідно створювати «механізм» задавання часу для синхронізації дій елементів моделі.

Існує два основних методи задавання часу в ІМ: 1) метод фіксованого кроку — принцип  $\Delta t$ ; 2) метод кроку до наступної події — *принцип  $\Delta z$* . При моделюванні за першим принципом відлік системного часу проводиться через заздалегідь задані часові проміжки сталої довжини, тобто моделювання відбувається в звичайному часі з фіксованим кроком. У моделі, побудованій за другим принципом, час змінюється з появою кожної наступної події незалежно від проміжку часу між ними, тобто моделювання відбувається в часі подій зі змінним кроком.

При побудові ІМ доцільно використовувати принцип  $\Delta z$  (із змінним кроком або кроком до наступної події). Це пов'язано з існуванням *суттєвого недоліку принципу  $\Delta t$*  стосовно моделювання ГКІС, а саме:  $\Delta t$  має бути досить малим, щоб в умовах нерівномірного розподілу подій у часі дві неодноразові події не підпали під один і той самий часовий проміжок. Зменшення ж  $\Delta t$  призводить до збільшення витрат машинного часу на моделювання, бо значна частка його витрачається на відслідковування подій, яких у більшості часових проміжків може й не бути. При цьому навіть за досить великим  $\Delta t$  дві неодноразові події можуть підпасти під один і той самий часовий проміжок  $\Delta t$ , що створює помилкове уявлення про їх одночасність.

**Подання елементів ІМ.** Відповідно до вимог забезпечення оцінювання впливу будь-якого з перелічених у (11.1) параметрів на продуктивність та інші характеристики ГКІС в основу розробки ІМ необхідно покласти принцип імітації функціонування кожного елемента моделі. Для відображення змін характеристик модельованої ГКІС та стану елементів ІМ останні повинні подаватися в ній у вигляді *списків динамічних атрибутів*. Такі списки необхідні для переходу від абстрактного опису процесів взаємодії елементів до їх машинного подання.

Здійснювані при функціонуванні реальної ГКІС взаємодії різних компонентів ТО та матеріального потоку, які подаються зміною в часі їх станів, імітуються в моделі зміною атрибутів аналогів названих компонентів. Ці зміни, виходячи з логічних умов, відбуваються на множині подій з додержанням відповідних часових витримок.

**Черги заявок на обслуговування та їх організація.** Процес функціонування ІМ супроводжується генерацією необхідних для організації пріоритетного обслуговування чотирьох типів заявок:

1) заявки від ТО, яке завершило обробку чергової партії виробів і потребує визначення нового кандидата на обробку. Заявки цього типу обслуговуються модулями ситуаційного планування, реалізованими або на алгоритмах матричного моделювання календарних планів-графіків, або на комбінованих алгоритмах (з додаванням до попередніх ще й алгоритму мінімізації кількості повернутих носіїв). Аналогічні модулі застосовуються для розрахунку початкових черг запуску;

2) заявки на транспортне обслуговування. Черги таких заявок виникають у транспортних підсистемах дискретного типу, коли в окремі проміжки часу інтенсивність надходження заявок перевищує інтенсивність їх обслуговування. Заявка такого типу вибирається з черги відповідно до її пріоритету, який може бути або фіксованим (правило FIFO), або ситуаційним. В останньому випадку найбільший пріоритет мають заявки від обробних ресурсів системи (верстатів, складальних пристроїв тощо), які найменше забезпечені деталями на час вибору заявки з черги. Перелік пріоритету при цьому здійснюється відразу ж після вибору на транспортне обслуговування чергової заявки від обробного ресурсу, що забезпечує рівномірний розподіл транспортних ресурсів.

Заявки, завантажені на ТрП з місткістю нагромаджувача, більшою від одиниці, можуть обслуговуватись як за надходженням, та і з урахуванням можливості суміщення транспортних операцій. В останньому випадку після закінчення завантаження основних заявок згідно з їх пріоритетом і кількістю вільних місць на ТрП здійснюється додатковий пошук заявок (також з урахуванням їх пріоритету), які могли б бути обслужені завдяки суміщенню транспортних операцій;

3) заявки на обслуговування носіїв позицією комплектування. Причини виникнення черги таких заявок і правила вибору чергової заявки аналогічні заявкам другого типу, проте в разі ситуаційного пріоритету найвищий пріоритет мають заявки від ТО з мінімальним часом можливого чекання. Перелік пріоритету ТО також здійснюється після початку обслуговування чергової його заявки;

4) заявки, аналогічні заявкам третього типу, але за наявності конкуренції на вільні носії. При обмеженнях зайнятої  $i$ -ю партією деталей кількості носіїв  $R_{вн}^{(j)}$  під час обробки деталей на  $j$ -й операції технологічного маршруту додатково перевіряється умова  $r^{(i)k} < R^{(j)}$ , де  $r^{(i)k}$  — кількість носіїв  $k$ -го типу, яку займає  $i$ -та деталь під час прийняття рішення про вибір заявки.

Зазначені тут алгоритми пріоритетного обслуговування заявок оформляються у вигляді окремих програмних модулів і має забезпечуватись можливість їх доповнення або зміни.

**Склад реєстрових параметрів і розв'язуваних на ІМ задач.** Розподіл функцій між аналітичним та імітаційним моделюваннями ГКІС, характер розв'язуваних при цьому задач і отримувані результати можна звести у схему системного проектування ГКІС, зображену на рис. 11.4 [52].



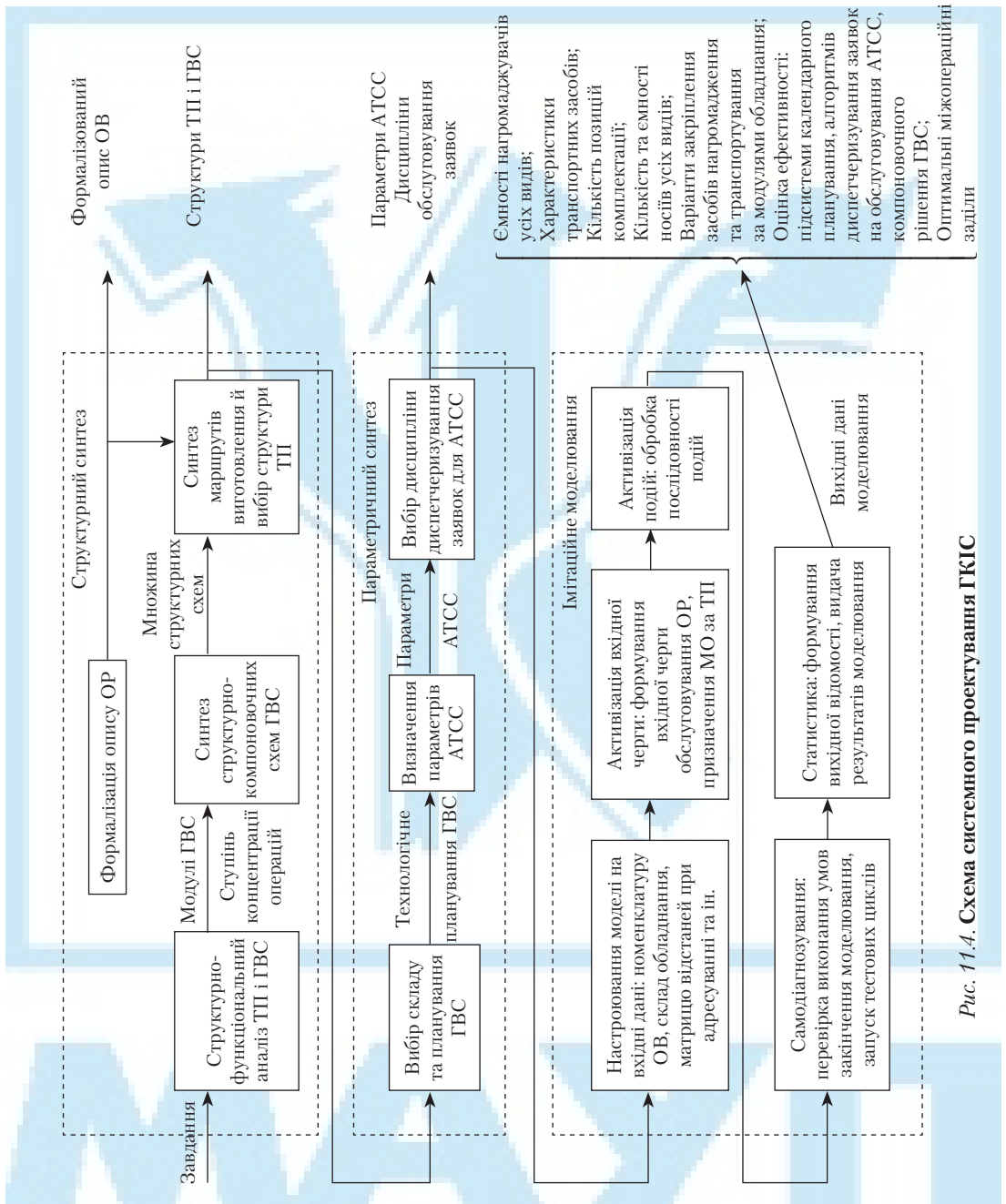


Рис. 11.4. Схема системного проектування ГКС

Наведена раніше (див. рис. 11.3) структура ІМ містить *модуль керування* та ряд базових модулів. Саме модулем керування автоматично здійснюється комбінування відповідних програмних одиниць, які відображають те чи інше джерело багатоваріантності початкових даних, відповідно до специфіки розв'язуваної при імітації задачі.

Схему спрощеного алгоритму модуля керування ІМ подано на рис. 11.5. Роботу цього модуля доцільно відобразити у вигляді послідовності взаємодіючих етапів.

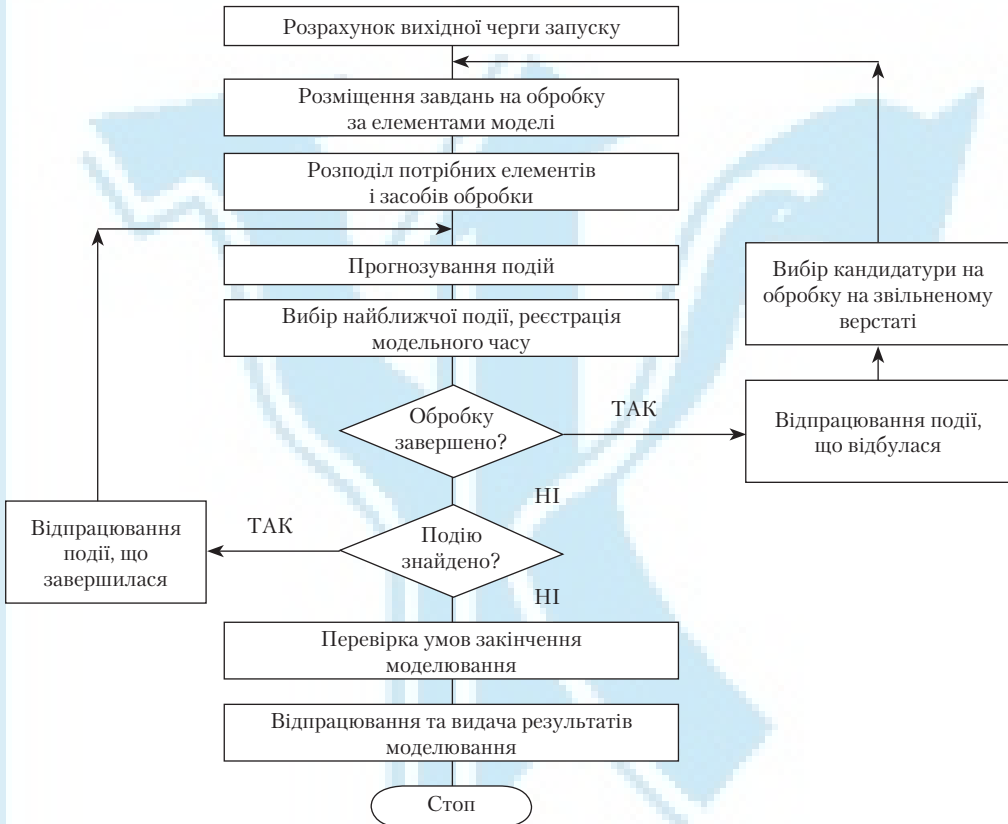


Рис. 11.5. Схеми алгоритму модуля керування ІМ

Характерною особливістю *першого етапу* є налаштування ТрМ на початкові дані. Для опису об'єкта моделювання (ГКІС) використовують такі початкові дані:

- номенклатуру деталей, які оброблюються за проміжок часу моделювання (містить дані про: тип і розмір партій; маршрути їх обробки із зазначенням типів або номерів ТО, тривалості обробки однієї деталі на кожній операції технологічного маршруту);
- склад основного ТО (тип і кількість верстатів кожного типу);
- склад складської підсистеми (тип елементів і їх ємності);
- тип транспортної системи, швидкість транспортування деталей, кількість і місткість ТрП дискретного типу;
- дані про закріплення засобів нагромадження на транспортування деталей за засобами обробки та елементами ГКІС у цілому (довільне та/або фіксоване);

- тип і кількість носіїв кожного типу, місткість їх за типами деталей та дані про закріплення носіїв за засобами обробки деталей;
- кількість позицій комплектування (якщо їх передбачено);
- витрати часу на рух носіїв між засобами транспортування та рештою елементів ГКІС, на тривалість циклів встановлення на носій та знімання з нього однієї деталі (середні або за типами деталей), на рух носія від вхідного нагромаджувача основного ТО та позицій комплектування до робочого стола й назад, на зміну програми обробки та інструменту (середні за елементами ТО);
- матрицю відстаней між пунктами адресування матеріальних потоків;
- індекси передбачених у моделі алгоритмів календарного планування ( $A_{к.п}$ ), диспетчеризування позицій комплектування ( $A_{п.к}$ ) та керування засобами транспортування ( $A_{тр}$ ).

Безпосередньо після введення початкових даних і згідно з ними виконується автоматичне настроювання ІМ: встановлення станів готовності елементів основного та транспортного обладнання; призначення вихідних положень ТрП дискретного типу (довільне чи відповідно до заданого адресою чекання завантаження); за необхідності імітації звернення деталей до позицій комплектування в ритмі ТП — роздільне розміщення деталей та їх носіїв на закріплених за позиціями комплектування модулями нагромадження; при моделюванні в режимі відсутності звернення до позицій комплектування — встановлення деталей на носіях і розміщення останніх у централізованих нагромаджувачах початкового положення; контроль відповідності верхніх границь масивів інформації початковим даним.

**Другий етап** роботи модуля керування ІМ є етапом активізації початкової черги запуску. Суть його така:

- програмною одиницею «*Формування початкової черги запуску*» визначається раціональна послідовність обробки партій деталей;
- відповідно до встановленої черги запуску програмною одиницею «*Вибір устаткування*» здійснюється пошук і призначення модуля устаткування на першу операцію технологічного маршруту  $j$ -ї партії деталей ( $j = \overline{1, N}$ ). Якщо необхідне устаткування вже зайняте, то перегляд черги запуску продовжується. В протилежному разі деталь  $j$ -ї партії з початкової черги виключається, а керування передається програмній одиниці «*Активізація претендента на обслуговування*»;
- функція програмної одиниці «*Активізація претендента на обслуговування*» зводиться до закріплення  $j$ -ї партії деталей за вибраним модулем устаткування, призначення адреси доставки її носіями, на яких розміщено цю партію, формування черг на їх обслуговування ТрП та позиціями комплектування, активізації вільних засобів транспортування та комплектування, прогнозування подій (закінчення операції передавання деталей централізованому вихідному нагромаджувачу, закінчення зміни програми та інструменту, закінчення операції обслуговування на позиції комплектування — див. події П2, П5 і П6, описані раніше), тобто до запуску  $j$ -ї партії деталей.

Після закінчення активізації початкової черги запуску з передаванням керування програмній одиниці «*Активізація подій*» модуль керування переходить до **третього етапу** імітації виробничого процесу. Кожне звернення до програми «*Активізація подій*» забезпечує обробку однієї події, після закінчення якої перевіряються такі умови:

- якщо в ході обробки події  $P_1$  (дивись вище) обслуговування  $j$ -ї партії деталей на  $\eta$ -й операції її технологічного маршруту закінчилося, а активізація початкової черги триває, то здійснюється перехід до другого етапу роботи модуля керування (див. другий пункт другого етапу імітації). Якщо ж початкову чергу до того часу вже вичерпано чи в ній немає претендентів на обслуговування модулем устаткування, який звільнився, то керування переходить до програмної одиниці «Вибір претендента на обслуговування». Якщо претендент відсутній, то керування передається програмі «Активізація подій». У протилежному разі цьому передує виклик підпрограми «Активізація претендента на обслуговування»;
- якщо час моделювання вичерпано, то здійснюється обробка й видача результатів, а керування передається програмній одиниці «Статистика». У протилежному разі перевіряється, чи виконувалась обробка подій при останньому зверненні до програми «Активізація подій». Якщо подію було знайдено й оброблено, то третій етап імітації виробничого процесу повторюється.

Якщо черга подій виявилася пустою, то модуль керування переводить ІМ на **четвертий етап** роботи — її самоконтроль. Цей етап полягає в перевірці виконання умов закінчення моделювання та безпосередньому самоконтролі функціонування моделі. До умов закінчення моделювання належать:

- відсутність інформації про невиконані операції технологічних маршрутів;
- відповідність адрес розташування деталей адресам повернення, заданим у початкових даних.

Умовою успішного завершення самоконтролю ІМ є відсутність неприпустимої інформації в масивах даних — аналогах модулів устаткування ГКІС. Кожна невиконана умова закінчення моделювання чи самоконтролю ІМ супроводжується повідомленням про це користувача.

Отже, критерієм закінчення моделювання є відсутність подій у календарі чи настання часу, заданого користувачем. При цьому критерієм вірогідності вичерпання черги подій є відсутність аварійних повідомлень на четвертому етапі роботи модуля керування ІМ.

Функціонування описаного алгоритму завершується передаванням керування підпрограмній одиниці обробки та видачі результатів «Статистика», яка реалізує **п'ятий етап** роботи модуля керування.

Отже, результатом моделювання, яке в цьому разі відтворює просторово-подійну розподіленість (див. підрозд. 11.5) подання процесів ГКІС, є визначення **критичного ресурсу** (критичних ресурсів)/**критичної операції** (критичних операцій), від якого (яких) залежить показник ефективності системи на верхньому рівні абстрагування — її циклова продуктивність.

### 11.5. Інтелектуалізація моделювання ГКІС з урахуванням динамічних властивостей її складових

**Загальна характеристика проблеми та постановка задачі.** Наведені в розділах 11.3 і 11.4 механізм синтезу аналітичної та імітаційної моделей АТСС ГКІС, а також структура системного проектування останньої дають змогу при правильно організованому модельному експерименті розв'язувати задачі її ефективного функціонування за рахунок оптимального використання обробляючих ресурсів шляхом

своєчасного постачання матеріальних потоків. Така задача може бути реалізована при лінійному просторово-подійному моделюванні (див. підрозд. 10.5) на основі відповідних модифікацій апарату сіток Петрі (див. розд. 7).

Проте зміст ГКІС охоплює не тільки технологічні операції й процеси, а й задачу аналізу й синтезу як окремих її компонентів, так і системи в цілому, причому питання проектування та переналадження з метою забезпечення умов повного використання обробляючих ресурсів є найбільш актуальними через те, що від них значною мірою залежить ефективність ГКІС. У свою чергу, ефективність використання ресурсів системи визначається повною мірою кількістю часу, необхідного для виконання тієї чи іншої операції при обмеженнях на енерговитрати, рівнем надійності, строком служби тощо.

Традиційно навантаження часовими атрибутами верхніх рівнів сіткових моделей дискретно-подійних об'єктів моделювання відбувається з урахуванням або очікуваних (бажаних) вірогідних значень тривалостей окремих складових процесів на конкретному обробляючому ресурсі, або на основі заданих технологією часових параметрів. При цьому верхній рівень ієрархії СП-моделі являє собою часову сітку Петрі (ЧСП, див. підрозд. 8.2) таку, де кожному переходу  $J$  ставиться у відповідність певний інтервал часу  $\tau_j$ , який дорівнює протяжності його виконання, тобто протяжності зіставленої з ним операції (так звана *просторово-подійна розподіленість з одновекторним виміром у моделюванні*, див. підрозд. 9.5). Нижні рівні моделі призначені для визначення часів спрацьовування переходів  $\tau_j$ , чим забезпечують так зване навантаження моделі часовими параметрами. У подальшому при настроюванні моделі верхнього рівня на необхідний показник оптимальності (як правило, значення циклової продуктивності) з'ясовується, що призначені значення часових атрибутів для деяких обробляючих ресурсів системи не забезпечують заданий показник оптимальності. Тоді для уточнення параметрів сіткових моделей верхнього рівня САІМ необхідно провести дослідження неперервних динамічних моделей тих ресурсів ГКІС, які «винні» в неоптимальному налаштуванні моделі (так звана *просторово-ієрархічна розподіленість з двовекторним виміром у моделюванні*). У цьому розділі розглянемо як приклад дослідження динамічних моделей нижчого рівня, спрямоване на визначення часу виконання технологічним устаткуванням зі складу ГКІС окремого елементарного руху (критичної операції), який би забезпечував заданий ступінь використання складових ресурсів системи. Синтез узагальненої моделі устаткування із складових ГКІС можна здійснювати на основі відомих підходів щодо дослідження динамічних властивостей [15; 16; 24; 57; 84; 86]. Коректне використання одержаних у цих роботах результатів дає змогу забезпечити адекватність моделі, достатню для проведення досліджень, пов'язаних з обчисленням *показника ступеня використання ресурсів* (ПСВР) при функціонуванні ГКІС. Тому питання визначення ступеня адекватності узагальненої динамічної моделі в межах подальших міркувань і доведень у прикладі, що наводиться далі, не розглядатиметься.

Слід також зазначити, що нижні рівні ієрархії можуть мати іншу семантику, використовувати інші формальні підходи щодо подання процесу, який відповідає переходові СП верхнього рівня. Для забезпечення автоматизації процесу перенавантаження переходів ЧСП верхнього рівня слід застосовувати якісь інтелектуалізовані структури, здатні аналізувати об'єкт моделювання, використовуючи відповідну семантику на нижньому рівні формалізації. Алгоритмічно такі структури можна реалізувати за допомогою застосування підходів, які базуються на введенні

пошукових інтелектуалізованих компонент, наведених у підрозд. 8.3. Далі коротко наводимо послідовність створення та дослідження ієрархічної сіткової моделі ГКІС з урахуванням динамічних властивостей функціонування її компонентів на нижньому рівні ієрархії, запропоновану в працях [25; 26].

**Обґрунтування вибору об'єкта дослідження.** З усього спектра устаткування, що входить до складу ГКІС, як об'єкт моделювання вибиратимемо найбільш складний пристрій для того, щоб результати, отримані при його дослідженні, легко могли бути поширені на будь-який інший компонент ГКІС. Таким пристроєм з урахуванням кінематичної складності й динамічних властивостей є промисловий робот. При цьому потрібно враховувати, що досягнутий для традиційного устаткування рівень оптимізації технологічних режимів обробки (складання), визначених у результаті розробки технологічного процесу, практично не залишає резерву для їх інтенсифікації, тоді як допоміжні операції, що виконуються ПР і займають значну частку циклу обробки (складання) об'єкта роботизації (ОР) (рис. 11.6), можна значно інтенсифікувати в результаті дослідження динаміки їх виконання.



Рис. 11.6. Структура циклу обробки (складання) ОР

Як базу приймемо досить поширену модельну лінійку роботів «PUMA» виробництва NOKIA (Фінляндія), що працює в полярній сферичній системі координат (рис. 11.7, а).

Функціональний склад такого робота показано на рис. 11.7, б.

Дослідження динаміки руху маніпуляційної системи (МС) ПР виконується згідно з її аналітичним описом — математичною моделлю (ММ).

**Означення 11.6.** Математична модель МС — функціональна залежність траєкторії руху МС від команд, що генеруються керуючим пристроєм ПР.

Математична модель означеного ПР повинна складатися з урахуванням кінематичних і динамічних характеристик МС, пружностей і нелінійностей механічних передач, типів і параметрів двигунів.

При визначенні часу виконання руху за заданою траєкторією необхідно розв'язати обернені задачі кінематики й динаміки, що потребуватиме опису відповідної функціональної залежності, яка визначається в кілька етапів [23]:

1. Перетворення траєкторії в абсолютних координатах у траєкторію в узагальнених координатах.
2. Визначення сил і моментів у зчленуваннях, а також сил і моментів, які розвиває двигун.
3. Визначення керуючих впливів приводу МС.

**Формування траєкторій в узагальнених координатах** (розв'язання оберненої задачі кінематики – 1-й етап синтезу ММ). Розглянемо завдання формування траєкторії вектора узагальнених координат при відомому законі руху захвату (обернена задача кінематики).

Вибрана модель ПР (рис. 11.7, а) з трьома ступенями рухомості (тобто трьома узагальненими координатами, що відповідають цим ступеням) задовольняє умову переміщення захвату в довільну точку простору.

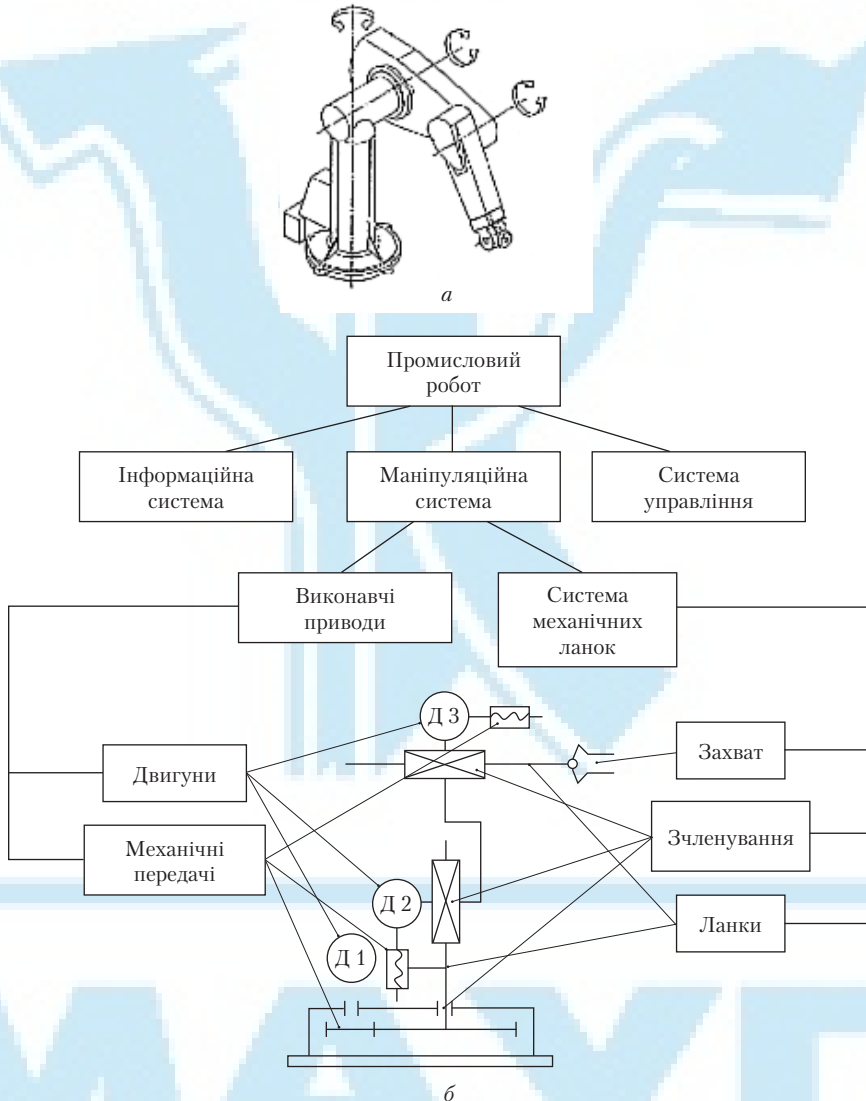


Рис. 11.7. Структура (а) і функціональний склад (б) ПР «PUMA»

Якщо маніпулятор має більше ступенів рухомості, то узагальнені координати, що відповідають зайвим ступеням, фіксуються й не змінюються в процесі руху.

При такій постановці задачі існує однозначний зв'язок між координатами захвату в абсолютній системі координат й узагальненими координатами маніпулятора (єдиний розв'язок прямої задачі кінематики)

$$\mathbf{x} = f(\mathbf{q}) = [\mathbf{f}_1(\mathbf{q}), \mathbf{f}_2(\mathbf{q}), \mathbf{f}_3(\mathbf{q})]^T, \quad (11.27)$$

де  $\mathbf{x}$  — радіус-вектор захвату в абсолютній системі координат, а  $\mathbf{q}$  — вектор узагальнених координат маніпулятора.

Відомо ряд методів точного розв'язання оберненої задачі кінематики, тобто розв'язання системи (11.27) щодо вектора  $\mathbf{q}$  [29], але на основі цих методів не вдається побудувати єдиний алгоритм, що забезпечує знаходження вектора узагальнених координат за заданим вектором положенням захвату в абсолютній системі координат для будь-якої частини випадку узагальненої кінематичної схеми МС ПР. У зв'язку з цим при створенні автоматизованої системи дослідження моделі доцільно використати один з наближених методів розв'язання цієї задачі.

Найпоширенішими є градієнтний метод і метод Ньютона [187].

**Градієнтний метод** базується на зведенні завдання знаходження розв'язку системи (11.27) до еквівалентної йому задачі мінімізації функціоналу неузгодженості між векторами заданого  $\mathbf{x}_3$  і поточного  $\mathbf{f}(\mathbf{q})$  положень:

$$p^2(\mathbf{q}) = |\mathbf{x}_3 - \mathbf{f}(\mathbf{q})|^2,$$

що розв'язується методами математичного програмування.

Можна скористуватися **методом Ньютона** як таким, що має меншу обчислювальну складність та забезпечує швидшу збіжність ітераційного процесу до шуканого розв'язку порівняно з градієнтним методом. При використанні методу Ньютона система (11.27) замінюється лінійною системою рівнянь

$$\mathbf{x}_3 - \mathbf{f}(\mathbf{q}_0) = \mathbf{J}(\mathbf{q}_0) \delta\mathbf{q}, \quad (11.28)$$

де  $\mathbf{J}(\mathbf{q}_0) = \frac{\partial \mathbf{f}(\mathbf{q}_0)}{\partial \mathbf{q}}$  — матриця Якобі при  $\mathbf{q} = \mathbf{q}_0$ .

З (11.28) можна визначити  $\delta\mathbf{q}$ :

$$\delta\mathbf{q} = \mathbf{J}^{-1}(\mathbf{q}_0)(\mathbf{x}_3 - \mathbf{f}(\mathbf{q}_0)), \quad (11.29)$$

причому  $\mathbf{q}_1 = \mathbf{q}_0 + \delta\mathbf{q}$  є першим наближенням значення вектора  $\mathbf{q}$ , що відповідає вектору  $\mathbf{x}_3$  заданого положення захвату в абсолютній системі координат.

Друге наближення знаходимо за формулою (11.29) шляхом заміни в ній  $\mathbf{q}_0$  новим значенням  $\mathbf{q}_1$  і т. д.

$$\mathbf{q}_i = \mathbf{q}_{i-1} + \mathbf{J}^{-1}(\mathbf{q}_{i-1})(\mathbf{x}_3 - \mathbf{f}(\mathbf{q}_{i-1})). \quad (11.30)$$

Ітераційний процес виконується доти, доки значення  $(\mathbf{x}_3 - \mathbf{f}(\mathbf{q}_i))$  не стане меншим, аніж заздалегідь фіксований малий вектор  $\Delta\mathbf{x}$ .

Таким чином, при відомій траєкторії руху захвату в абсолютній системі координат  $\mathbf{x}(t)$  можна обчислити значення вектора узагальнених координат  $\mathbf{q}(t_i)$  з довільним кроком  $\Delta t$ , необхідним для формування рівнянь динаміки МС ПР.

В задачах оперативного керування рухом ПР застосовується трохи інший підхід до розв'язання оберненої задачі кінематики. При використанні цього підходу розв'язок шукається у вигляді [17]:

$$\dot{\mathbf{q}}(t) = \mathbf{J}^{-1} \dot{\mathbf{x}}(t). \quad (11.31)$$

Синтез узагальнених траєкторій виконується таким чином. Визначається траєкторія, якою повинен рухатися захват у просторі:  $\mathbf{x}(t)$  і  $\dot{\mathbf{x}}(t)$ , починаючи від вихідного положення маніпулятора, якому відповідають узагальнені координати



$\mathbf{q}(0)$  і абсолютні координати  $\mathbf{x}(0)$ . Припускаючи, що для досить малих інтервалів  $\Delta t = t_i - t_{i-1}$  справедливе  $\mathbf{J}(\mathbf{q}(t_i)) \approx \mathbf{J}(\mathbf{q}(t_{i-1}))$ , можна стверджувати, що справедливе співвідношення

$$\mathbf{q}(t_i) \approx \mathbf{q}(t_{i-1}) + \dot{\mathbf{q}}(t_{i-1})\Delta t.$$

де  $\dot{\mathbf{q}}(t_{i-1})$  знаходиться з (11.31).

До переваг цього підходу можна віднести швидкість обчислення траєкторії, тому що кожне наступне значення вектора узагальнених координат обчислюється на основі попереднього за один крок, на відміну від ітераційної процедури, де на пошук розв'язку необхідно 10–20 ітерацій. Недоліком є залежність точності розв'язання задачі від вибраного значення, що не дає змоги розв'язувати обернену задачу з довільним кроком.

Для ПР «PUMA», у якого зчленування 1 – типу 2 (вісь обертання рухомої ланки збігається з осями обох ланок кінематичної пари), а зчленування 2 і 3 – типу 1 (вісь обертання рухомої ланки перпендикулярна осям обох ланок кінематичної пари), елементи матриці Якобі будуть мати вигляд [17]:

$$\begin{aligned} j_{1,1} &= -\sin q_1 \cos q_2 \sin q_3 l_3 - \sin q_1 \sin q_2 (\cos q_3 l_3 + l_2); \\ j_{1,2} &= -\cos q_1 \sin q_2 \sin q_3 l_3 + \cos q_1 \cos q_2 (\cos q_3 l_3 + l_2) + \cos q_1 l_1; \\ j_{1,3} &= -\cos q_1 \sin q_2 \sin q_3 l_3 + \cos q_1 \cos q_2 (\cos q_3 l_3 + l_2); \\ j_{2,1} &= \cos q_1 \cos q_2 \sin q_3 l_3 + \cos q_1 \sin q_2 (\cos q_3 l_3 + l_2); \\ j_{2,2} &= -\sin q_1 \sin q_2 \sin q_3 l_3 + \sin q_1 \cos q_2 (\cos q_3 l_3 + l_2) + \sin q_1 l_1; \\ j_{2,3} &= -\sin q_1 \sin q_2 \sin q_3 l_3 + \sin q_1 \cos q_2 (\cos q_3 l_3 + l_2); \\ j_{3,1} &= 0; \\ j_{3,2} &= -\cos q_2 \sin q_3 l_3 - \sin q_2 (\cos q_3 l_3 + l_2); \\ j_{3,3} &= -\cos q_2 \sin q_3 l_3 - \sin q_2 (\cos q_3 l_3 + l_2), \end{aligned}$$

де  $q_1, q_2, q_3$  – узагальнені координати в рухомостях ПР;  $l_1, l_2, l_3$  – довжини відповідних ланок ПР.

**Формування моментів у зчленуваннях (2-й етап синтезу ММ).** Після визначення траєкторій здійснюється визначення сил і моментів, що забезпечують рух по цих траєкторіях і прикладених у зчленуваннях МС ПР. Оскільки для визначення таких сил і моментів необхідно розглядати рух маніпулятора як рух системи, що складається з ланок з певними масо-інерційними характеристиками, то для роботи алгоритму необхідні такі вихідні дані:

- траєкторії узагальнених координат  $\mathbf{q}(t)$ , їхні швидкості  $\dot{\mathbf{q}}(t)$  і прискорення  $\ddot{\mathbf{q}}(t)$ ;
- маси всіх ланок маніпулятора  $\mathbf{m}$ ;
- моменти інерції всіх ланок маніпулятора  $\mathbf{J}$ .

Обчислення сил і моментів виконуватимуться на основі рівнянь Лагранжа II роду, за допомогою яких, як відомо, може бути описана будь-яка голономна система з  $n$  ступенями рухомості [25]

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = Q_k, \quad k = 1, \dots, n, \quad (11.32)$$

де  $L = E - \Pi$  – функція Лагранжа системи,  $E$  – кінетична енергія системи,  $\Pi$  – потенційна енергія системи,  $q_k$  – узагальнені координати,  $\dot{q}_k$  – узагальнені швидкості,  $Q_k$  – узагальнені сили. Для розв'язання поставленої задачі на цьому етапі необхідно знайти траєкторію вектора узагальнених сил  $\mathbf{Q}$ . Значення вектора  $\mathbf{Q}$  у кожний момент часу визначатиметься в три кроки:

- визначення кінетичної енергії маніпулятора;
- визначення потенційної енергії маніпулятора;
- обчислення вектора узагальнених сил  $\mathbf{Q}$  з (11.32).

Процедури, які супроводжують визначення складових (11.32), детально розглядаються у фаховій літературі, зокрема у [86], тому далі не наводяться.

*Визначення номінальних сил і моментів приводів.* Для забезпечення необхідних сил і моментів у зчленуваннях ПР необхідно реалізувати сили й моменти, що розвиваються виконавчими двигунами з урахуванням особливостей механізмів передачі руху.

*Механічна передача з урахуванням її динамічних особливостей.* Механізми передачі рухів досить складно піддаються аналітичному опису, що пов'язано, по-перше, з нелінійними залежностями між координатою двигуна й узагальненою координатою об'єкта маніпулювання (наприклад, редуктор зі змінним передавальним відношенням або механізм, у якому обертальний рух у зчленуванні забезпечується двигуном з лінійним переміщенням); по-друге, з особливостями, що помітно впливають на динаміку всієї системи, такими як: пружність валів, моментних датчиків і зубів шестірень, люфти в редукторі, в'язке й сухе тертя, зовнішні сили, що діють на механізм [84].

У загальному вигляді можна скористатися рівняннями динаміки механічної передачі, принципова схема якої наведена на рис. 11.8 і яка відповідає властивостям будь-якої з рухомих пар ПР «PUMA»:

$$\begin{cases} M_d = J_d \ddot{q}_d + M_n + |M_{с_{тд}}| \text{sign}(\dot{q}_d) + k_{в_{тд}} \dot{q}_d \\ M_n = J_n \ddot{q} + |M_{с_{тн}}| \text{sign}(\dot{q}) + k_{в_{тн}} \dot{q}_n + M_{с_{т}} + M_{зовн} \end{cases}, \quad (11.33)$$

де  $M_d$  – обертальний момент двигуна;  $M_n$  – момент навантаження;  $J_d, J_n$  – моменти інерції двигуна і навантаження відповідно;  $\dot{q}_d, \ddot{q}_d$  – швидкість і прискорення двигуна по узагальненій координаті  $q$  відповідно;  $M_{с_{тд}}, M_{с_{тн}}$  – моменти сухого тертя двигуна і навантаження відповідно;  $k_{в_{тд}}, k_{в_{тн}}$  – коефіцієнти в'язкого тертя двигуна і навантаження відповідно;  $M_{с_{т}}$  – статичний момент навантаження;  $M_{зовн}$  – сума зовнішніх моментів опору руху.

На принциповій схемі передачі позначено  $M_{тд}$  і  $M_{тн}$ , які визначаються з (11.33):

$$M_{тд} = |M_{с_{тд}}| \text{sign}(\dot{q}_d) + k_{в_{тд}} \dot{q}_d \quad (11.34)$$

$$M_{тн} = |M_{с_{тн}}| \text{sign}(\dot{q}_n) + k_{в_{тн}} \dot{q}_n \quad (11.35)$$

Одержавши математичну модель механічної передачі, можна вивести залежності моменту двигуна від моменту, що розвивається в зчленуванні, і координати двигуна від узагальненої координати, що описує положення зчленування (тобто одержати відповідь на питання про те, який момент повинен розвинути двигун і на який кут повинен повернутися його вал, щоб забезпечити заданий рух у зчленуванні).

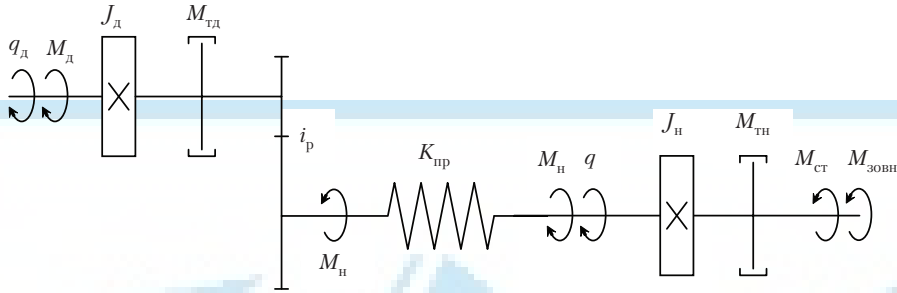


Рис. 11.8. Принципова схема механічної передачі

У ПР «PUMA» момент від двигуна до зчленування передається за допомогою прямих і конічних зубчастих передач і приводних валів, пружністю яких можна знехтувати. Таким чином, у разі відсутності зовнішніх навантажень для ПР «PUMA» справедливе:

$$\begin{cases} q_d = q_i \\ M_d = J_d \ddot{q}_d + Q + M_{тд} + M_{тн} \end{cases}$$

**Формування номінальних управляючих дій (третьої етапу синтезу ММ).** Завершальним етапом побудови ММ МС є знаходження керуючих впливів, дій, які забезпечують заданий рух, на виконавчі механізми (двигуни). Ці впливи можна відшукати, знаючи сили і моменти, які повинен розвивати двигун, та математичну модель самого двигуна.

**Універсальна математична модель привода.** Математичні моделі приводів у загальному випадку нелінійні. Однак можна досягти задовільних результатів, якщо припустити, що моделі приводів лінійні у просторі станів і нелінійні за вхідним сигналом. Зазначимо, що ММ приводів різних типів відрізняються видом і порядком їх диференціальних рівнянь, що ускладнює алгоритмізацію різних процесів, у яких ці приводи беруть участь. Тому постає необхідність у виборі універсальної ММ, за допомогою якої можна було б описати кожен з використовуваних у ПР приводів.

При виборі як вектора стану  $x = (q_d, \dot{q}_d, i_a)^T$  модель електромеханічних двигунів, які використовуються у ПР «PUMA», набуває вигляду [17]:

$$\dot{x} = A(\theta)x + b(\theta)u + f(\theta)M, \quad (11.36)$$

причому

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{k_{в\ тд}}{J_d} & \frac{k_M}{J_d} \\ 0 & -\frac{k_E}{L} & -\frac{R}{L} \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}, \quad f = \begin{bmatrix} 0 \\ -\frac{1}{J_d} \\ 0 \end{bmatrix}, \quad (11.37)$$

де  $k_M, k_E$  — коефіцієнти пропорційності моменту та електрорушійної сили відповідно;  $R, L$  — опір та індуктивність якоря двигуна відповідно.

**Визначення траєкторії керуючого сигналу.** Розглянемо процедуру знаходження траєкторії керуючого сигналу для моделі (11.36) третього порядку, тому що при моделюванні приводів роботів-маніпуляторів раціонально використовувати

системи не вище третього порядку [17]. Вектор стану моделі приводу вибирається таким чином, щоб  $x_1$  відповідав координаті двигуна (наприклад, куту повороту вала двигуна), а  $x_2$  – її першій похідній;  $x_3$  залежить від типу двигуна – це може бути струм ротора для двигуна постійного струму або різниця тисків у циліндрі для гідродвигуна. Невідоме значення  $x_3$  обчислюється в такий спосіб [17]:

$$x_3 = (\dot{x}_2 - a_{22}x_2 - f_2 M_{3n}) / a_{23}, \quad (11.38)$$

де  $a_{ij}$  і  $f_i$  означають відповідні елементи матриць  $A$  і  $f$  в описі (11.37).

Далі обчислюється номінальне програмне керування, тобто входні сигнали на кожен двигун, які забезпечать рушійні моменти й рух захвату ПР по заданій траєкторії:

$$u = (\dot{x}_3 - a_{32}x_2 - a_{33}x_3) / b_3. \quad (11.39)$$

У рівняннях (11.38) і (11.39) враховано вид матриць моделі двигуна  $A, b, f$  (11.37).

Обчислене програмне керування забезпечує рух по заданій траєкторії в ідеальних умовах, тобто за відсутності збуджень і за умови припущення, що математична модель МС ПР ідеально ідентифікована.

**Означення 11.7.** Під ідеальними умовами мається на увазі відсутність зовнішніх сил, не врахованих у моделі, і повна адекватність моделі об'єкту, що моделюється.

Очевидно, що досягти ідеальних умов на практиці неможливо, тому на етапі керування, як правило, використовуються системи автоматичного регулювання руху ПР, до завдання яких входить компенсація похибок, викликаних неідеальністю умов.

Таким чином, ММ електромеханічних приводів ПР «PUMA» визначаються відповідно до (11.37).

**Синтез повної моделі МС ПР.** На основі співвідношень, наведених у цьому розділі, можна скласти повну математичну модель МС ПР. Для цього необхідно знайти функціональну залежність керуючих впливів на двигуни МС від траєкторії руху захвату, заданої в абсолютних координатах:

$$u = u(x). \quad (11.40)$$

Процедуру складання ММ продемонструємо на прикладі тієї ж моделі ПР «PUMA», яка має три обертальні ступені рухомості (кутова система координат), що дає змогу також виконувати й транспортні операції. Четвертою ланкою є захват, і відповідний йому ступінь рухомості зафіксований:  $q_4 = 0$ .

Залежність (11.40) визначається в кілька етапів (рис. 11.9) з визначенням:

1. Узагальнених координат  $q_k$  за значеннями абсолютних координат  $x$ .
2. Координат двигунів  $q_d$  і моментів  $M_E$  зовнішніх навантажень на двигуни за узагальненими координатами  $q_k$ .
3. Струмів роторів за координатами і моментами двигунів.
4. Значень керуючих напруг  $u$  за струмами роторів і координатами двигунів.

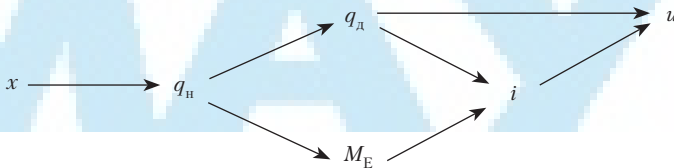


Рис. 11.9. Послідовність визначення залежності  $u = u(x)$

Визначимо співвідношення, необхідні для складання ММ, з урахуванням особливостей розглянутого ПР. Всі співвідношення будуть наведені для ступеня рухомості  $k$ -ї ланки, причому індекс буде упушений скрізь, де можливо. Також будуть упушені всі процедури інтегрування й диференціювання, тобто вважається, що якщо відома функція, то відомі всі її похідні й початкові функції. Це пов'язано з тим, що при моделюванні інтегрування й диференціювання виконуються в числовому вигляді, тому немає необхідності наводити аналітичні залежності. Початкові значення змінних інтегрування й диференціювання визначаються з умов:

$$\dot{q}(0) = 0; \quad \ddot{q}(0) = 0.$$

Враховуючи, що в ПР «PUMA» використовуються електромеханічні двигуни, одержуємо:

$$\begin{cases} u = \left( \dot{i}_я + \frac{k_E}{L} \dot{q}_д + \frac{R}{L} i_я \right) / \frac{1}{L} \\ i_я = \left( \ddot{q}_д + \frac{k_{в\tau д}}{J_д} \dot{q}_д + \frac{1}{J_д} M_{зн} \right) / \frac{k_M}{J_{\Phi\Phi}} \end{cases} \quad (11.41)$$

З урахуванням відсутності пружності передавального механізму одержуємо:

$$q_д = q_{н\text{п}}. \quad (11.42)$$

За умови відсутності зовнішніх навантажень одержуємо вираз для моменту зовнішнього навантаження двигуна:

$$M_{зн} = Q + M_{тн},$$

який з урахуванням (11.35), а також приймаючи, що  $Q = J_н \ddot{q} + M_{ст}$  є сумою трьох складових, з яких перша описує дію сил від інерційних навантажень, друга — дію відцентрової сили та сили Кориоліса, якою можна нехтувати через її малий вплив, третя — дію гравітаційних сил (тут  $J_н$  — ефективний момент інерції, віднесений до узагальненої координати), можна переписати як

$$M_{зн} = J_н \ddot{q} + M_{ст} + |M_{стн}| \text{sign}(\dot{q}_н) + k_{в\tau н} \dot{q}_н. \quad (11.43)$$

З (11.29), (11.30) випливає:

$$q_н = \sum_{i=0}^n J^{-1}(q_i)(x - f(q_i)), \quad (x - f(q_n)) < \Delta x. \quad (11.44)$$

Таким чином, **етап 1** визначення залежності (11.40) реалізується виразом (11.44), **етап 2** — (11.42) і (11.43), **етапи 3 і 4** — системою (11.41), що дає змогу обчислити повну ММ МС ПР з наступними матрицями й векторами:

- для координат  $R_i$  центра мас  $i$ -ї ланки в абсолютній системі координат по осі  $x_{3i}$  та матриці  $G^T$  прискорення  $g$  вільного падіння

$$R_i = (0, 0, x_{3i}^i, 1), \quad G^T = (0, 0, -g, 0);$$

- для векторного рівняння кінематики маніпулятора, що визначає залежність між узагальненими координатами  $q_i$  і декартовими координатами довільної точки маніпулятора

$$B_i^j = \begin{cases} A_1 \cdots A_{j-1} D_j A_j A_{j+1} \cdots A_i, & \text{якщо } j \leq i \\ 0, & \text{якщо } j > i; \end{cases}$$

- для матриці інерції  $H_i$

$$H_i = \begin{bmatrix} J_1^i & 0 & 0 & 0 \\ 0 & J_2^i & 0 & 0 \\ 0 & 0 & J_3^i & m_i x_{3i} \\ 0 & 0 & m_i x_{3i} & m_i \end{bmatrix};$$

- для матриць  $A_1 \dots A_4$  переходу від  $i$ -ї до  $1-i$ -ї системи координат

$$A_1 = \begin{pmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} \cos q_2 & 0 & \sin q_2 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin q_2 & 0 & \cos q_2 & l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} \cos q_3 & 0 & \sin q_3 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin q_3 & 0 & \cos q_3 & l_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & 0 & 0 & 1 \end{pmatrix};$$

- для проєкційних матриць  $D_1 \dots D_3$

$$D_1 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad D_2 = D_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Параметри МС ПР можна розділити на кінематичні, динамічні й параметри двигуна.

*До кінематичних параметрів* належать: довжина ланки  $l$ , координати центра мас  $x_{ц}$ , передаточне число редуктора  $i_p$ .

*Динамічні параметри*: маса ланки  $m$ , моменти інерції ланки  $J$ , момент сухого  $M_{ст}$  і коефіцієнт в'язкого  $k_{в'тн}$  тертя.

*Параметри двигуна*: струм якоря  $i_я$ ; опір ланцюга якоря  $R$ ; індуктивність ланцюга якоря  $L$ ; коефіцієнти пропорційності моменту  $k_m$  та ЕРС  $k_E$ , момент інерції ротора  $J_d$ , момент сухого  $M_{стд}$  й коефіцієнт в'язкого  $k_{в'тд}$  тертя.

Математична модель придатна для дослідження кінематики й динаміки МС ПР і легко піддається алгоритмізації.

**Дослідження моделі МС ПР.** Як було визначено на початку цього розділу, завданням дослідження ММ МС є пошук часу  $\tau_f$  за який ПР може перемістити певний вантаж по заданій траєкторії при заданому значенні ПСВР  $f$ . Розв'язання такої задачі дасть змогу збільшити продуктивність ГКІС і повніше використати її динамічні ресурси.

Введемо коефіцієнт, що коригує керуючий вплив відповідно до умов функціонування устаткування.

На етапі проектування або аналізу устаткування можна одержати значення  $\tau_f$  з точністю, що визначається адекватністю математичної моделі, оскільки при виконанні програмного руху в реальних умовах у разі впливу зовнішніх збурюючих факторів система повинна мати деякий ресурс для компенсації помилок, викликаних неточністю моделі.

**Означення 11.8.** Коефіцієнт компенсації неадекватності  $k_{кн}$  — коефіцієнт, що враховує необхідність компенсації похибок керування рухом робота-маніпулятора в реальних умовах, викликаних неадекватністю математичної моделі такого руху.

Крім того, слід враховувати режими роботи устаткування при виконанні заданого руху за проміжок часу  $\tau_f$ . Незначне збільшення швидкості виконання операції може призвести до підвищеного спрацювання устаткування, нестійкої його роботи, перевитрати електроенергії тощо.

**Визначення часу  $\tau_f$ .** Аналітичне визначення часу  $\tau_f$  ускладнене через складність математичної моделі МС ПР, зокрема через її нелінійність. Тому доцільно використати ітераційну процедуру пошуку розв'язку із залученням числових методів розв'язання рівнянь.

Розв'язавши систему (11.41) — (11.44) стосовно  $u$ , можна одержати траєкторію вектора керуючих впливів  $\mathbf{u}(t)$  у просторі станів. Крім того, відомі граничні значення керуючих впливів  $\mathbf{u}_{пр}$ , зумовлені можливостями виконавчих механізмів, тобто заданий рух може бути відпрацьовано виконавчим механізмом тільки в разі виконання умови

$$|u_i(t)| \leq u_{прi}, t \in [0, \tau], i = 1, \dots, n, \quad (11.45)$$

де  $n$  — кількість керованих виконавчих механізмів (у випадку ПР — керованих ступенів рухомості).

Розглянемо наведені значення керуючих впливів

$$\tilde{u}_i(t) = \frac{|u_i(t)|}{u_{прi}}, i = 1, \dots, n, \quad (11.46)$$

де  $\tilde{u}_i(t)$  — приведені значення керуючого впливу  $i$ -го виконавчого механізму, причому заданий рух може бути відпрацьований тільки при виконанні умови  $\tilde{u}_i(t) \leq 1$ .

Визначимо функцію, що відображає найбільше відносне навантаження серед усіх виконавчих приводів МС ПР у цей момент часу:

$$\tilde{u}_m(t) = \max_i \tilde{u}_i(t). \quad (11.47)$$

Максимум функції  $\tilde{u}_{\max}(\tau)$  у проміжок часу  $[0, \tau_f]$  визначає момент часу й значення ПСВР ПР при виконанні заданого руху

$$\tilde{u}_{\max} = \max_{\tau} \tilde{u}_m(t), t \in [0, \tau]. \quad (11.48)$$

Таким чином, якщо  $\tilde{u}_{\max} \leq 1$ , то ПР може виконати заданий рух.

Для пошуку часу, за який ПР спроможний виконати заданий рух, необхідно досліджувати функціональну залежність  $\tilde{u}_{\max}(\tau)$ .

**Твердження 11.1.** Функція  $\tilde{u}_{\max}(\tau)$  є незростаючою.

Формальне доведення цього твердження було б занадто складним, тому з метою скорочення обсягу роботи нижче наводяться такі міркування. Так, при збільшенні часу виконання руху певною траєкторією з певним профілем швидкості граничні швидкості переміщення, а отже, й прискорення, що дають змогу їх досягти, зменшуються. Зменшення прискорень, згідно з другим законом Ньютона, приводить до зменшення сил, що їх розвивають. Зменшення сил, які в ПР розвивають виконавчі приводи, відповідно приводить до зменшення значення керуючого впливу, а тому при зменшенні часу значення  $\tilde{u}_{\max}(\tau)$  відповідно зменшується. Однак, крім виконання руху, приводи повинні виконувати функцію компенсації сили тяжіння, що не залежить від швидкостей і прискорень переміщення МС, тому в разі зміни часу значення функції  $\tilde{u}_{\max}(\tau)$  може не змінюватися. Ці міркування дають змогу

припустити, що функція має незростаючий характер. Далі, ґрунтуючись на законах Ньютона й законі збереження енергії, можна стверджувати, що

$$\begin{cases} \tau \in (0, \infty) \\ \tilde{u}_{\max}(\tau) \in (0, \infty). \\ \lim_{\tau \rightarrow 0} \tilde{u}_{\max}(\tau) = \infty \end{cases} \quad (11.49)$$

Таким чином, можна стверджувати, що функція  $\tilde{u}_{\max}(\tau)$  є незростаючою, всі її значення лежать в першій чверті і при наближенні аргументу до нуля функція зростає до нескінченності.

Для визначення часу виконання операції  $\tau_f$  із заданим ступенем використання ресурсів динаміки устаткування й з урахуванням неадекватності математичної моделі необхідно знайти розв'язок рівняння

$$k_{\text{кн}} \tilde{u}_{\max}(\tau) = f. \quad (11.50)$$

З урахуванням виду функції  $\tilde{u}_{\max}(\tau)$  можна стверджувати, що рівняння (11.50) має або один розв'язок, тоді він і вибирається як  $\tau_f$ , або множину розв'язків, що лежать у певному діапазоні, тоді як  $\tau_f$  вибирається ліва границя цього діапазону, тобто найменший з можливих розв'язків.

Для розв'язання цього рівняння можна скористатися одним із числових методів, наприклад, січних — ітераційним методом пошуку розв'язку рівняння  $f(x) = 0$ , за яким кожне наступне наближення знаходиться за формулою:

$$\begin{cases} x^{(m+1)} = x^{(m)} - \frac{f(x^{(m)})}{s_m}, \quad m = 1, 2, \dots \\ s_m = \frac{f(x^{(m)}) - f(x^{(m-1)})}{x^{(m)} - x^{(m-1)}} \end{cases}, \quad (11.51)$$

де  $x^{(0)}, x^{(l)}$  вибираються так, щоб  $f(x^{(0)}) - f(x^{(l)}) < 0$ , а  $x^{(k)}$  вважалось розв'язком рівняння  $f(x) = 0$ , якщо  $|x^{(k)} - x^{(k-1)}| < \epsilon$ , де  $\epsilon$  — задана точність знаходження розв'язку.

Цей метод можна застосувати в розглянутому випадку, зробивши підстановку  $x = \tau_f(x) = k_{\text{кн}} \tilde{u}_{\max}(x) - f$ , і в результаті його застосування одержати значення  $\tau_f$ . Може виникнути ситуація, в якій рівняння (11.50) взагалі не має розв'язків. У такому випадку робиться висновок про те, що цей ПР у принципі не здатний виконати рух із заданими параметрами, тому що така ситуація може виникнути тільки за умови, що ПР не здатний компенсувати силу тяжіння, що діє на елементи його конструкції й перемішуваний вантаж. Слід зауважити, що кожна ітерація методу січних потребує обчислення функції  $\tilde{u}_{\max}(\tau)$ , тобто, по суті, постановки експерименту над математичною моделлю МС ПР. Одержавши шуканий час  $\tau_f$  і відповідну йому ММ, необхідно виконати перевірку здатності виконавчих двигунів реалізувати необхідний рух МС ПР.

Таким чином, виконавши на рівні просторово-ієрархічної розподіленості моделі ГКІС необхідні процедури щодо визначення прийняттого значення часу  $\tau_f$ , можна перенавантажувати верхній рівень просторово-подійно-розподіленої моделі ГКІС цим знайденим часовим інтервалом і досліджувати останню на відповідність заданому критерію якості (зокрема циклової продуктивності).



## 11.6. Структура динамічного навантаження моделі ГКІС

**Загальна мета дослідження моделі ГКІС з просторово-ієрархічною розподіленістю.** Результатом проведення імітаційного моделювання процесів ГКІС (див. розд. 11.4) стає, по-перше, визначення критичної технологічної операції, а по-друге, створення умов для переходу від просторово-подійної до просторово-ієрархічної розподіленості в моделюванні, яке враховує динамічні характеристики обробних ресурсів системи. Таким чином, підвищення ефективності роботи ГКІС реалізується на основі підходу, який полягає в керуванні системою в два етапи: на першому відшукуються критичні операції, на другому критичні операції інтенсифікуються шляхом постановки спрямованих серій експериментів над імітаційною моделлю устаткування, яке виконує критичну операцію.

**Означення 11.9.** *Критична операція* — технологічна операція, що входить до складу технологічного процесу (ТП), поліпшення показників ефективності функціонування якої приводить до підвищення показників ефективності функціонування ТП загалом.

Іншими словами, критична операція — це операція, що стримує ТП ГКІС, тобто є його «вузьким місцем».

**Означення 11.10.** *Спрямована серія експериментів* — серія експериментів, у якій параметри кожної наступної ітерації визначаються на основі числового аналізу результатів попередніх ітерацій відповідно до заданого критерію ефективності.

Як правило, задача підвищення ефективності розв'язується як задача пошуку оптимального керування, що забезпечує мінімальне/максимальне значення заданого критерію при певних обмеженнях. Критерієм ефективності роботи ГКІС можуть виступати такі величини, як час, витрата електроенергії, спрацьовування устаткування і т. д., або їх комбінація (наприклад, зважена сума).

У розглянутому в підрозд. 11.5 підході далі задача підвищення ефективності (чи інтенсифікації) розв'язується як задача забезпечення заданих часових показників функціонування ГКІС шляхом узгодження часу виконання критичних операцій з часом циклу виробничого процесу. При такій постановці шукати оптимальне за часом рішення недоцільно через те, що необхідно лише забезпечити певний час виконання операції, який задовольняє циклову продуктивність системи, тоді як оптимальне рішення передбачає роботу устаткування на граничних режимах експлуатації, що підвищує його знос і витрати енергії.

Оптимальні задачі можуть становити інтерес за такої постановки як задачі визначення запасу динамічного ресурсу системи.

Існує два шляхи інтенсифікації функціонування устаткування: перший пов'язаний з поліпшенням законів планування руху; другий — з використанням наявних ресурсів динаміки устаткування.

Як критерій ефективності роботи ГКІС при цьому може застосовуватися показник ПСВР — відношення максимального значення параметрів системи (швидкості, напруги живлення, моменту, сили тощо) при поточному керуванні до їх граничних значень.

Можна визначити таку послідовність заходів, спрямованих на інтенсифікацію роботи ГКІС шляхом дослідження її моделі з просторово-ієрархічною розподіленістю:

1. Побудова моделі компонентів ГКІС з урахуванням динамічних характеристик цих обробних ресурсів системи.

2. Синтез керування шляхом пошуку критичних операцій і подальшої постановки спрямованої серії експериментів над моделлю устаткування, яке виконує ці операції.
3. Впровадження отриманих результатів у процес оперативного керування ГКІС.

**Процедура керування експериментами при моделюванні ГКІС.** Зміст першого етапу розглянуто в підрозд. 11.5. Відповідно до етапу 2 заходів щодо інтенсифікації роботи ГКІС необхідно організувати серії експериментів над її моделлю, спрямовані на пошук критичної операції та інтенсифікацію роботи устаткування, яке виконує цю операцію.

Проведення експериментів з моделлю ГКІС за традиційною схемою передбачає участь людини-дослідника, яка за допомогою зовнішнього програмного забезпечення (ПЗ) інтерпретує результати і приймає рішення щодо керування ходом експериментів (рис. 11.10).

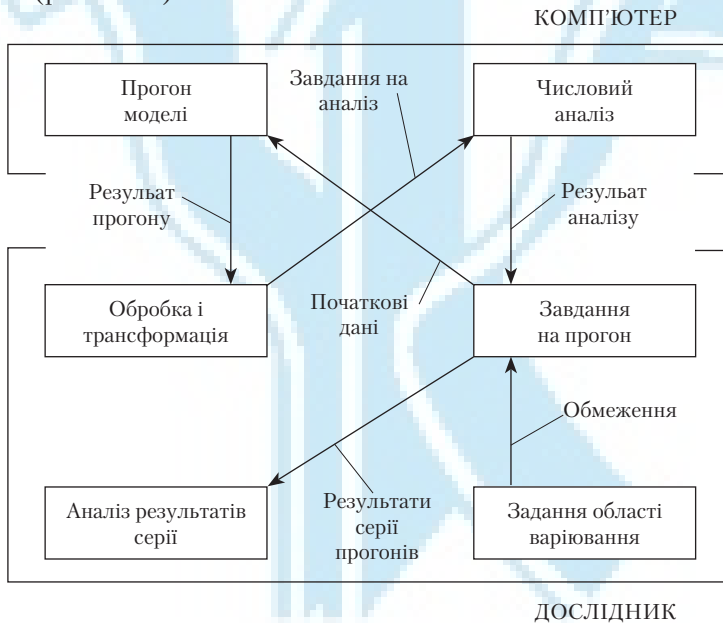


Рис. 11.10. Традиційна система проведення експериментів

Відповідно до цієї схеми ітераційна процедура постановки серії експериментів виконується в такий спосіб: дослідник на основі заданих обмежень і результатів числового аналізу даних про попередні експерименти задає вихідні дані для подальшого прогону моделі; результати прогону обробляються дослідником і трансформуються у вигляд, придатний для числового аналізу; виконується числовий аналіз результатів прогону, далі цикл замикається.

Наявність людини в контурі керування експериментами визначає ряд недоліків традиційної схеми моделювання. Серед таких недоліків можна зазначити: трудомісткість процедури моделювання, необхідність у висококваліфікованому фахівці на етапі моделювання, складність оцінки результатів моделювання, високу імовірність помилки, низьку продуктивність.

Усунути зазначені недоліки можна шляхом автоматизації імітаційного моделювання, принципи якої описано в розд. 6.

Для автоматизації процесу моделювання необхідно створити замкнутий програмно-відтворений контур керування імітаційною моделлю, в якому функції обробки і трансформації результатів експериментів і прийняття рішення про подальший хід експериментів замість людини-дослідника повинні виконувати спеціально введені до складу ПЗ пошукові модулі ПМ (рис. 11.11). У загальному випадку такі модулі повинні задавати набори даних, ініціювати прогони моделі, обробляти результати і приймати рішення про подальший хід експериментів відповідно до заданого алгоритму керування моделюванням. Такий алгоритм спрямований на пошук сполучення параметрів ІМ в області їх припустимих значень, що забезпечувало б оптимум заданому показнику якості (див. умову (6.20)).

Сукупність програмних засобів, що забезпечують процес автоматизованого моделювання, реалізується САІМ на основі тверджень (6.1) – (6.4). *Формальні основи автоматизованого моделювання* детально розглядалися в підрозд. 6.4 і були пов'язані з алгоритмами імітаційного моделювання (АлгІМ), які керують постановкою експериментів над ІМ, причому завданням АлгІМ є пошук в області припустимих значень параметрів ОМ такого їх сполучення, яке забезпечувало б задане значення показника ефективності роботи ГКІС.

На вхід АлгІМ надходить слово  $W_{вх}$ , що відповідає множині  $D_{W_{вх}}$ , яка являє собою область варіювання параметрів ОМ. Вихідне слово  $W_{вих}$  визначає основні результати експерименту.

Етапи процедури розв'язання проблеми автоматизації керування експериментами наведено на рис. 6.7 (підрозд. 6.4).

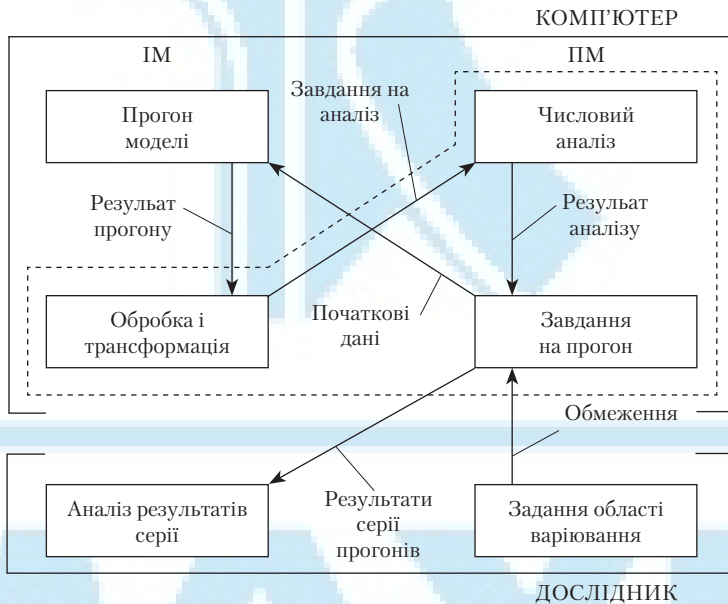


Рис. 11.11. Система автоматизованого управління експериментами

**Обґрунтування структури автоматизованої моделі ГКІС.** У підрозд. 6.4 викладено основи реалізації інтелектуальної компоненти САІМ – пошукового модуля ПМ, а також розглянуто варіанти вкладеності ІМ та ПМ, зумовлені проблемною галуззю, яка й визначає функціональний розподіл між ними. Тому нижче наводять-

ся тільки деякі уточнення щодо підходу створення САІМ об'єктів із складовими, в яких досліджуються їх динамічні властивості.

**Означення 11.11.** *Пошуковий модуль (ПМ)* – функціональна компонента САІМ, яка організовує певну послідовну серію експериментів з ІМ відповідно до конкретної стратегії.

Пошуковий модуль реалізує деякий метод пошуку, для якого ІМ виступає засобом розрахунку значень заданих показників. Умовно структуру функціонування контуру ПМ–ІМ та його взаємодії з дослідником зображено на рис. 11.12.

Пошуковий модуль та ІМ, які утворюють замкнений контур, і являють собою АІМ. На рис. 11.12 наведено спрощений варіант реалізації АІМ, яка у загальному випадку може мати складну ієрархічну структуру. Через те, що різниця в поняттях ПМ та ІМ визначається проблемною галуззю (ПМ може виступати в ролі ІМ на іншому рівні абстракції), то можна визначити множину типів ієрархічного підпорядкування компонент АІМ (див. підрозд. 6.4).

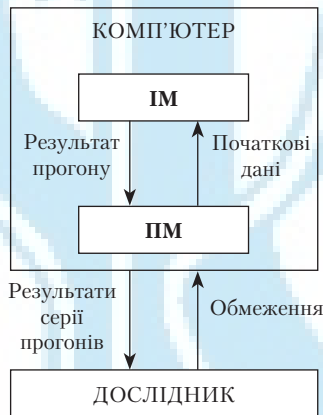


Рис. 11.12. Структура функціонування контуру ПМ–ІМ

У цьому розділі для розв'язання задачі інтенсифікації роботи ГКІС з урахуванням динамічних характеристик буде використовуватися АІМ, побудована на основі ієрархічної моделі ГКІС, в якій кожна підмодель охоплена ПМ. Використовуючи таку схему побудови АІМ, легко розв'язати, в межах моделі, задачу узгодження семантик відмінних рівнів ГКІС, бо взаємодія між рівнями здійснюється не безпосередньо, а через ПМ, причому функцією ПМ є визначення або уточнення параметрів ІМ більш високого рівня, наприклад, на рис. 11.13 модуль ПМ2 визначає параметри ІМ1, досліджуючи її підмодель ІМ2.

Застосовуючи наведену схему організації АІМ, можна запобігти надмірному ускладненню алгоритмів та обсягів обчислень ПМ через те, що ПМ розв'язує задачу ідентифікації якогось одного параметра, досліджуючи при цьому обмежену підмодель.

**Обґрунтування складових динамічної імітаційної моделі ГКІС.** Автоматизована ІМ ГКІС будується на основі ієрархічної моделі ГКІС, де на всіх рівнях ієрархії, крім нижнього, система описується за допомогою сіток Петрі, а на нижньому рівні – диференціальними рівняннями динаміки устаткування, що входить до складу ГКІС. Задачею пошукових модулів ПМ у складі АІМ ГКІС є уточнення параметрів моделі більш високого рівня. Можна виділити два типи ПМ: пошуковий модуль ПМ1, що проводить експерименти над СП-моделлю; пошуковий модуль

ПМ2, що проводить експерименти над динамічною моделлю устаткування. Модуль ПМ1, відповідно, є надбудовою над усіма рівнями моделі, крім нижнього, а модуль ПМ2 – над нижнім.

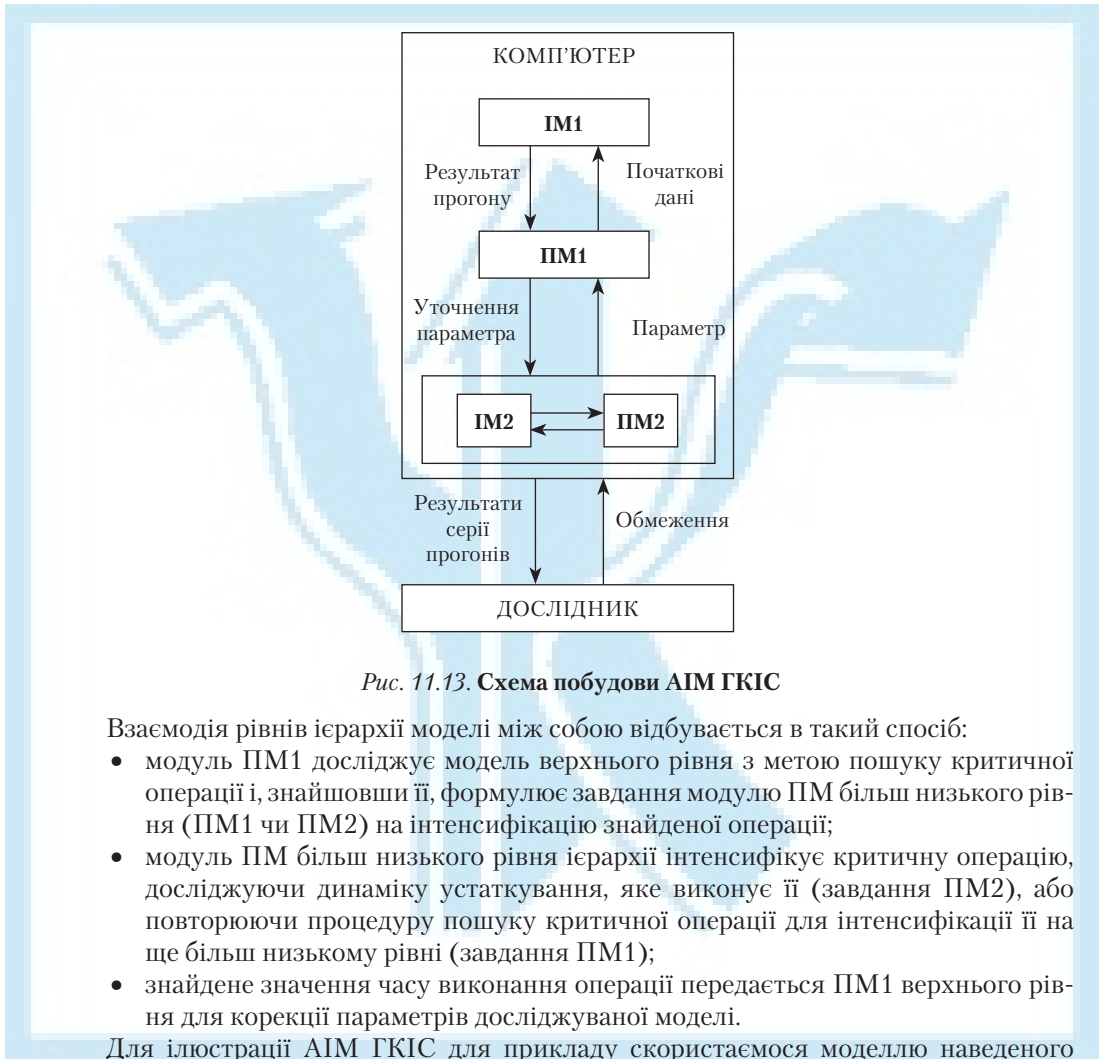


Рис. 11.13. Схема побудови АІМ ГКІС

Взаємодія рівнів ієрархії моделі між собою відбувається в такий спосіб:

- модуль ПМ1 досліджує модель верхнього рівня з метою пошуку критичної операції і, знайшовши її, формулює завдання модулю ПМ більш низького рівня (ПМ1 чи ПМ2) на інтенсифікацію знайденої операції;
- модуль ПМ більш низького рівня ієрархії інтенсифікує критичну операцію, досліджуючи динаміку устаткування, яке виконує її (завдання ПМ2), або повторюючи процедуру пошуку критичної операції для інтенсифікації її на ще більш низькому рівні (завдання ПМ1);
- знайдене значення часу виконання операції передається ПМ1 верхнього рівня для корекції параметрів досліджуваної моделі.

Для ілюстрації АІМ ГКІС для прикладу скористаємося моделлю наведеного у підрозд. 7.1 (рис. 7.1) тестового гнучкого виробничого модуля (ТГВМ) у складі ГКІС, який містить: нагромаджувач заготовок (НЗ); нагромаджувач виробів (НВ); промисловий робот, який виконує операції завантаження-розвантаження (Р1); двопозиційний стіл-маніпулятор виробів (МВ) з позиціями П1 і П2; промисловий робот, який виконує складання виробів (ПР2); нагромаджувач деталей (НД) (рис. 11.14). Нагадаємо, що всі нагромаджувачі мають ємність, рівну одиниці.

Автоматизована імітаційна модель ТГВМ складається з: 1) СП-моделей ТГВМ на різних рівнях абстракції; 2) модулів ПМ1, які досліджують СП-моделі; 3) узагальненої динамічної моделі устаткування, яке входить до складу ТГВМ; 4) модуля ПМ2, який досліджує модель і до завдання якого входить уточнення часових параметрів функціонування СП-моделей верхніх рівнів.

Якщо подати АІМ ТГВМ у вигляді програмного комплексу, то дані про параметри моделей ТГВМ та устаткування, а також про критерії ефективності зберігаються в загальній базі даних, а ПМ1 і ПМ2 являють собою підпрограми, або програмні модулі.

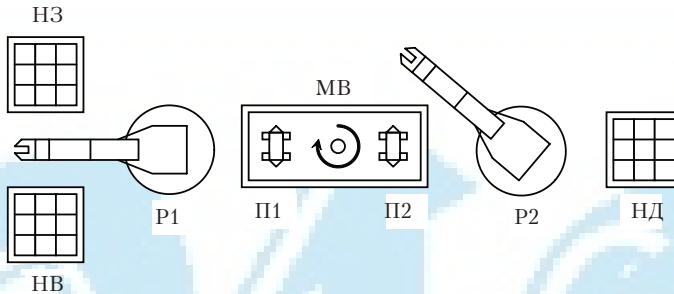


Рис. 11.14. Тестовий гнучкий виробничий модуль

Програмний модуль, у якому реалізовано функції ПМ, у загальному випадку повинен мати такі компоненти:

- *блок постановки експериментів* — у цьому блоці реалізовано алгоритми, що керують ходом постановки експериментів й інтерпретують їх результати;
- *блок імітації* — блок, у якому виконується розрахунок моделі за параметрами, заданими блоком постановки експериментів (тобто який безпосередньо проводить експеримент);
- *інтерфейсний блок* — блок, відповідальний за організацію взаємодії з іншими модулями програмного комплексу (модулями, що реалізують функції ПМ більш високого і більш низького рівнів ієрархії, диспетчерським модулем програмного комплексу, модулем взаємодії з користувачем тощо);
- *блок взаємодії з базою даних* — блок, який за запитами інших блоків здійснює вибірки необхідних даних з бази.

**Галузь застосування та завдання модуля ПМ1 при моделюванні ГКІС.** Як зовнішню процедуру, яка визначає функцію часів спрацьовування  $\tau$  для розв'язання задачі пошуку критичної операції, в АІМ застосовується інтелектуалізований модуль ПМ1.

**Означення 11.12.** *Пошуковий модуль ПМ1* — це інтелектуалізована компонента, що застосовується для інтенсифікації роботи ТГВМ шляхом пошуку найбільш ефективного алгоритму керування з корекцією параметрів ГСП-моделі за допомогою взаємодії з більш низькими рівнями ієрархії моделі (у тому числі і з рівнем динамічної моделі устаткування).

До завдання ПМ1 входить: організація серії експериментів, спрямованих на пошук найбільш ефективного керування ТГВМ; реалізація дослідження властивостей ГСП-моделі як об'єкта, над яким проводяться експерименти; організація взаємодії з рівнем подання ТГВМ у вигляді динамічних моделей устаткування; взаємодія з базою даних (зчитування параметрів системи, необхідних для функціонування ПМ1, запис результатів серії експериментів і скоригованих параметрів системи).

Отже, функція ПМ1 полягає в пошуку критичної операції, формулюванні завдання на інтенсифікацію цієї операції для ПМ більш низького рівня і коригуванні досліджуваної моделі з використанням результатів, отриманих ПМ більш низького

рівня ієрархії. Залежно від рівня абстракції, на якому розглядається модель ТГВМ, можна виділити кілька способів виконання технологічного процесу: *рівнобіжний* – на рівні розгляду ТГВМ як множини взаємодіючих модулів і систем, де всі технологічні операції виконуються відповідно до циклового часу виробничого процесу; *паралельно-послідовний* – на рівні моделювання роботи окремого модуля, де за час циклу може виконуватися кілька рівнобіжних послідовностей переходів операцій; *послідовний* – на рівні моделювання окремої технологічної одиниці, що виконує свої функції тільки послідовно.

У випадку паралельного виконання операцій критичною вважатиметься найдовша за часом операція; у випадку паралельно-послідовного – операція зі складу найдовшої в часі послідовності технологічних переходів, яка має найбільший ресурс динаміки; у випадку послідовного – функція устаткування з найбільшим ресурсом динаміки.

Остаточне завдання на функціонування ПМ1 можна визначити як пошук часу виконання критичних операцій із заданим ступенем використання ресурсів динаміки устаткування.

Розглянемо організацію алгоритму функціонування ПМ1 за умови коректно сформованих наборів вхідних даних.

**Структура і склад алгоритму ПМ1.** Відповідно до завдань, виконуваних ПМ1, визначимо структуру і склад алгоритму його роботи (АлгПМ1). Алгоритм АлгПМ1 складається з блоків: взаємодії з базою даних; інтерфейсного; постановки експерименту; синтезу законів керування.

Взаємодія цих блоків із зовнішніми структурами і між собою організується згідно зі структурною схемою, наведеною на рис. 11.15.

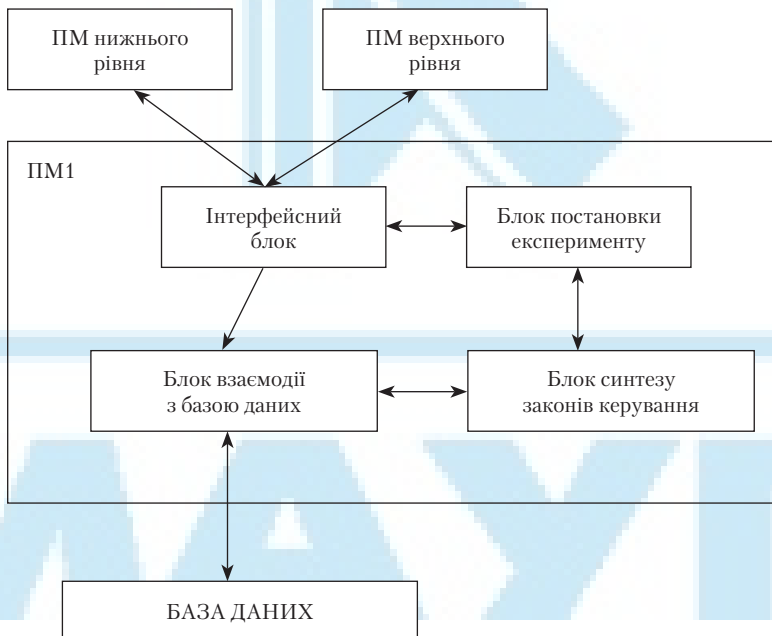


Рис. 11.15. Структура алгоритму ПМ1

*Блок взаємодії з базою даних.* Вихідні дані для роботи АлгПМ1 зберігаються у загальній базі даних. Залежно від того, яка ТГВМ моделюється у поточний момент часу і яке завдання аналізу виконується, необхідно з бази даних витягти певну підмножину інформації про об'єкт дослідження відповідно до вхідного слова АлгПМ1. Завданням блоку взаємодії з базою даних є вибірка такої підмножини.

Вибірка інформації повинна здійснюватися відповідно до формату вхідного слова інтерфейсного блоку. У такий спосіб досягається універсальність, наприклад, можливість застосування одного ПМ для розв'язання задачі однотипних досліджень усіх ТГВМ, що входять до складу гнучкого автоматизованого виробництва.

Вхідними даними блоку взаємодії з базою даних є вхідне слово інтерфейсного блоку і вся інформація, що зберігається в базі даних. Вихідні дані – вибірка інформації, необхідної для функціонування АлгПМ1.

*Інтерфейсний блок.* Взаємодія ПМ1 з іншими модулями організовується за допомогою інтерфейсного блоку, до завдань якого входить інтерпретація вхідного слова  $W_{\text{вх}}$  і формування вихідного слова  $W_{\text{вих}}$ .

Для ПМ1  $W_{\text{вх}}$  містить у собі інформацію про ГСП-модель і заданий показник якості функціонування ТГВМ.

Вихідне слово  $W_{\text{вих}}$  являє собою знайдене значення показника якості і необхідну інформацію про час виконання критичних операцій.

Завдання, виконувані інтерфейсним блоком, є ключовими у процесі функціонування ієрархічної моделі, тому що саме вони реалізують «ієрархічність» моделі шляхом узгодження семантик різних рівнів ієрархії і різних формальних підходів до опису об'єкта дослідження.

Вхідними даними інтерфейсного блоку є інформація, що передається від ПМ1 зовнішнім процедурам або навпаки. Вихідні дані являють собою відповідним чином інтерпретовані в термінах АлгПМ1 вхідні дані.

*Блок синтезу законів керування.* Безпосереднє функціонування імітаційної моделі забезпечується блоком синтезу законів керування. До завдання цього блоку входить пошук серед усіх послідовностей спрацьовування переходів ГСП такого, при якому час функціонування ГСП мінімальний. Через те, що на рівні подання ГКІС у вигляді ГСП послідовність спрацьовування переходів може інтерпретуватися як закон керування системою, то в результаті роботи блоку синтезу законів керування буде визначено найбільш ефективний спосіб керування ГКІС.

Для розв'язання поставленої задачі на основі даних про параметри ГСП-моделі ГКІС побудується дерево досяжності ГСП до моменту першої появи в ньому вершини, маркування якої аналогічне маркуванню початкової вершини. Шлях від початкової вершини до знайденої і буде найбільш ефективним законом керування.

Вхідними даними для блоку синтезу законів керування є параметри ГСП, а вихідними – послідовність спрацьовування переходів з найбільш ефективним функціонуванням ГСП з погляду часу виконання поставленої задачі.

*Блок постановки експерименту.* Розв'язання задачі, поставленої перед ПМ1, здійснюється шляхом постановки спрямованої серії експериментів над ГСП-моделлю ГКІС, що здійснюється блоком постановки експерименту. Задачею, виконуваною блоком, є постановка серії експериментів над ГСП-моделлю ГКІС, спрямованої на пошук закону керування і часу виконання ТП ГКІС при заданому значенні ПСВР. Для розв'язання цієї задачі використовується алгоритм пошуку критичної операції.



**Вихідні дані:** ГСП, що описують роботу досліджуваної системи; задані часи виконання елементарних операцій  $t_{ei}$ ; функції ПСВР для кожної операції  $f_i(t)$ , що показують залежність ступеня використання ресурсів устаткування від часу виконання операції, причому при  $f_i(t) = 0$  ресурси не використовуються, а при  $f_i(t) = 1$  ресурси використовуються на 100 % значень цільової функції, які необхідно забезпечити в ході розв'язання задачі  $f$ .

**Алгоритм 11.1.**

*Крок 1.* Прийняти  $t_{ei}$  як поточні значення часових параметрів  $t_i$ .

*Крок 2.* Побудувати дерево досяжності і на його основі визначити послідовність критичних операцій  $\theta$  при реалізації ТП.

*Крок 3.* Обчислити значення ПСВР ГСП у цілому

$$f(t) = \frac{\sum_{\theta} f_i(t_i)t_i}{\sum_{\theta} t_i}. \quad (11.52)$$

*Крок 4.* Якщо  $f(t) < f$ , то перейти до кроку 5, у протилежному разі — до кроку 3.

*Крок 5.* Визначити критичну операцію  $j$ , для якої  $f_j(t_j) = \min f_i(t_i)$ .

*Крок 6.* Визначити нове значення ПСВР для операції  $f_j(t_j) \stackrel{\theta}{=} f$ .

*Крок 7.* За відомим  $f_j(t_j)$  відшукати нове значення  $t_j$  шляхом формулювання завдання і виклику ПМ більш низького рівня.

*Крок 8.* Замінити час реалізації операції  $j$  новим значенням  $t_j$ .

*Крок 9.* Повернутися до кроку 2.

*Крок 10.* Обчислити час виконання ТП:  $T_{\Pi} = \sum_{\theta} t_i$ .

*Крок 11.* Закінчити роботу алгоритму.

При синтезі ГКІС можлива модифікація алгоритму шуканого розв'язку у вигляді  $f(T_{\Pi}) > f$  у разі виконання умови  $\forall i, j: f_i(t_i) = f_j(t_j)$ . В цьому випадку все устаткування, що входить у ГКІС, працюватиме з однаковою віддачею, однак підвищиться обчислювальна складність алгоритму.

Зазначимо, що завдання для ПМ1 найвищого рівня формулюється зовнішньою процедурою, наприклад, системою керування АІМ шуканого розв'язку задачі узгодження часу роботи досліджуваної системи з цикловим часом ГКІС.

**Галузь застосування, завдання та принципи функціонування ПМ2.** Дослідження динамічної моделі устаткування в межах АІМ проводиться за допомогою ПМ2 — пошукового модуля визначення умов динамічного навантаження (ПМДН).

**Означення 11.13.** *Пошуковий модуль ПМ2* — це інтелектуалізована компонента, що застосовується для інтенсифікації роботи ГВС шляхом пошуку найбільш ефективного алгоритму керування технологічним устаткуванням, спрямованого на мінімізацію часу виконання заданих рухів.

Завданням ПМ2 є визначення часу  $\tau_f$ , який відповідає заданому значенню ПСВР і поставлений у відповідність певному параметру моделі верхнього рівня. Крім того, ПМ2 повинен визначити умови функціонування об'єкта, за яких досягається необхідний час. Зазначимо, що залежно від об'єкта, який моделюється, і необхідної точності й адекватності способи виконання поставленої задачі можуть бути різними за змістом і складністю: від простого пошуку часових параметрів у БД до розв'язання складних систем диференціальних рівнянь високих порядків, що описують процеси, які відбуваються в об'єкті. Таким чином, при поданні об'єкта у вигляді АІМ досягається повна універсальність подання для різних типів об'єктів,

що пояснюється незалежністю моделі від процесів, які відбуваються в об'єкті моделювання: досить лише використати відповідний ПМ2, завдяки чому можна досягати як різного ступеня абстракції, так і точності тієї самої моделі.

**Структура та склад ПМ2.** Структуру ПМ2, що виконує задачу визначення часу виконання переміщення із заданим значенням показника ступеня використання ресурсів (ПСВР), можна подати схемою на рис. 11.16.

До складу ПМ2 входять такі основні блоки: взаємодії з БД; інтерфейсний; постановки експерименту; перетворення траєкторії; синтезу номінального керування; інтерпретації прийнятих рішень.

**Блок взаємодії з базою даних.** Для роботи ПМ2 необхідно, відповідно до вхідного слова алгоритму роботи ПМ2, зробити вибірку даних із загальної БД АІМ. Передбачається, що інформація з усіх складових і траєкторій системи, що модулюється, міститься в єдиній БД АІМ, де кожному переходу ГВС відповідає певна підмножина даних.

Завданням цього блоку є пошук потрібної інформації з БД і збереження її у внутрішніх змінних ПМ2. Такий підхід робить ПМ2 універсальним засобом розв'язання певної задачі для всіх складових системи, що моделюється, тобто ПМ2 не є атрибутом якогось певного переходу ГВС, а доступний для будь-якого її переходу.

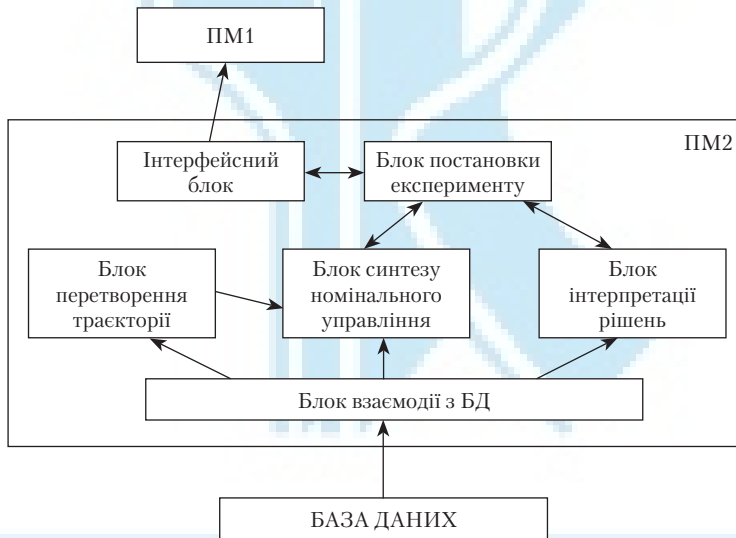


Рис. 11.16. Структурна схема ПМ2

**Інтерфейсний блок.** Інтерфейсний блок розв'язує задачу взаємодії ПМ2 з іншими модулями АІМ.

Для ПМ2  $W_{вх}$  містить у собі інформацію про модель досліджуваного об'єкта — показник ступеня використання його ресурсів.

Вихідне слово  $W_{вих}$  являє собою знайдене значення показника — необхідну інформацію про критичну операцію.

Фактично взаємодія з моделлю ГВС (за посередництва ПМ1) полягає у навантаженні її переходів часовими значеннями, тобто в присвоєнні відповідним змінним моделі числових значень.

Дані між різними ПМ передаються через спільну БД, так що взаємодія між різними модулями не залежить від типу модуля або характеру інформації, що передається.

**Блок постановки експерименту.** Модуль ПМ2 реалізує процес пошуку у вигляді постановки серії експериментів над моделлю досліджуваного об'єкта. Вихідні дані для кожного нового експерименту визначаються результатами попередніх експериментів і заданою точністю критерію оцінки результату. Критерій вибирається залежно від задачі, розв'язуваної при моделюванні (у розглянутому випадку це пошук найменшого часу виконання руху). Алгоритм планування експериментів, що лежить в основі функціонування ПМ2, можна подати в такий спосіб (вихідні дані: початкове наближення часу виконання руху  $t_{\text{п}}$ , необхідне значення ПСВР  $f$ ).

**Алгоритм 11.2**

*Крок 1.* Траекторію перетворити у вигляд, придатний для обробки алгоритмами побудови ММ МС  $q(t) = f(x(t))$ .

*Крок 2.* Задати початкове наближення часового параметра  $\tau_f = t_{\text{п}}$ .

*Крок 3.* Знайти значення керуючих впливів  $u(t) = f(q(t))$ ,  $t \in [0, \tau_{\text{мін}}]$ .

*Крок 4.* За отриманим значенням розрахувати наступне наближення часового параметра  $\tilde{t}_{\text{макс}}(\tau_f)$ .

*Крок 5.* Якщо значення часового параметра задовольняють критерій оцінки результату пошуку, тобто  $k_{\text{кп}} \tilde{t}_{\text{макс}}(\tau) \leq f$ , перейти до кроку 6, у протилежному разі — повторити кроки 3–5.

*Крок 6.* Перевірити знайдене значення на відповідність можливостям технологічного обладнання (ТО).

*Крок 7.* Якщо значення часового параметра не відповідає можливостям ТО, зробити пошук прийняттого розв'язку.

*Крок 8.* Закінчити роботу алгоритму.

Співвідношення, необхідні для пошуку часового параметра, що відповідає можливостям ТО, а також критерій оцінки результату розглянуто в підрозд. 11.5 (11.45) – (11.51).

**Блок перетворення траекторії.** Цей блок призначений для формування траекторії вектора узагальнених координат, що забезпечує заданий рух МС, залежно від того, яким чином задана його траекторія. Розглянемо приклад роботи блоку у випадку, коли прямолінійна траекторія задана за допомогою: а) координат початкової точки траекторії  $\mathbf{x}_{\text{п}} = \mathbf{x}(0)$ ; б) координат кінцевої точки траекторії  $\mathbf{x}_{\text{к}} = \mathbf{x}(\tau)$ ; в) профілю швидкості (часу руху  $\tau$ , часу розгону  $t_p$ , часу гальмування  $t_r$ ) (рис. 11.17). Використовуючи значення цих параметрів, обчислюють залежність  $\mathbf{x}(t)$ :

$$\mathbf{x}(t) = \mathbf{x}_{\text{п}} + \text{sign}(\mathbf{x}_{\text{к}} - \mathbf{x}_{\text{п}}) \begin{cases} 0,5\mathbf{a}_1 t^2, & t_0 \leq t < t_1 \\ \mathbf{V}(t-t_1) + \mathbf{S}_1, & t_1 \leq t < t_2, \\ \mathbf{S}_3 + 0,5\mathbf{a}_2(t-t_3)^2, & t_2 \leq t \leq t_3 \end{cases} \quad (11.53)$$

де  $t_0 = 0$ ,  $t_1 = t_p$ ,  $t_2 = t - t_m$ ,  $t_3 = \tau$ ;  $\mathbf{S}_3 = \mathbf{x}_{\text{к}} - \mathbf{x}_{\text{п}}$ ;  $\mathbf{a}_2 = \mathbf{S}_3 \frac{2}{t_2^2 + t_1(t_3 - t_2) - t_3^2}$ ;  $\mathbf{V} = \mathbf{a}_1 t_1$ ;

$$\mathbf{S}_1 = \mathbf{a}_1 \frac{t_1^2}{2}; \quad \text{sign}(\mathbf{x}) = \left( \frac{x_1}{|x_1|}, \frac{x_2}{|x_2|}, \frac{x_3}{|x_3|} \right)^T.$$

Після визначення залежності  $\mathbf{x}(t)$  для будь-якого моменту часу обчислюються значення  $\mathbf{q}(t)$  (див. підрозд. 11.5).

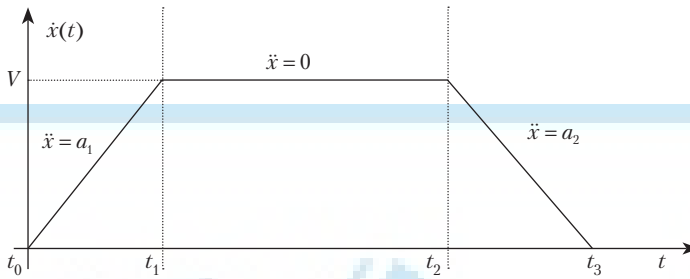


Рис. 11.17. Профіль швидкості PR

Блок синтезу номінального керування. У ПМ2 номінальне керування використовується для оцінки часу, необхідного для виконання заданого руху.

**Означення 11.14.** Номінальне керування — таке керування, що забезпечує рух МС по заданій траєкторії при ідеальних умовах.

Для синтезу номінального керування необхідна така інформація: траєкторія руху маніпулятора в узагальнених координатах  $\mathbf{q}(t)$ ; відстані від осей зчленувань до центрів мас ланок  $a_{ij}$ ; динамічні параметри маніпулятора (маси  $m_i$  і моменти інерції ланок  $\mathbf{J}_i$ ); динамічні параметри вантажу (маса  $m$  і моменти інерції  $\mathbf{J}_i$ ); параметри приводів (наприклад, у вигляді матриць підсистеми  $\mathbf{A}$ , перетворення навантаження  $\mathbf{J}$  і перетворення вхідного сигналу  $\mathbf{b}_i$ ); параметри механічних передач (нелінійності, коефіцієнти пружності  $k_p$ , передаточні числа редукторів  $i_{pi}$  і т. д.). На основі цієї інформації проводиться дослідження динаміки маніпулятора і обчислюється вектор керуючих впливів на приводи робота  $\mathbf{u}(t)$ , що забезпечують рух МС по заданій траєкторії  $\mathbf{x}(t)$ , за математичною моделлю (11.41)–(11.44).

*Блок інтерпретації рішення.* Коли знайдено час  $\tau_f$  і отримано відповідні керуючі впливи  $\mathbf{u}(t)$ , необхідно зробити перевірку можливості відпрацювання двигунами заданого руху. Наприклад, для електродвигунів необхідно перевірити, чи не відбувається порушення теплового балансу і чи не виходить значення швидкості переміщення за припустимі значення і т. ін.

Після проведення перевірки і корекції часу  $\tau_f$  інтерфейсним блоком формується вихідне слово  $W_{\text{вих}}$ , що містить інформацію про знайдений розв'язок поставленої задачі. Сформоване слово передається на більш високий рівень ієрархії, на чому роботу алгоритму ПМ2 можна вважати завершеною.

**Результат роботи алгоритму.** У розглянутому випадку в результаті роботи ПМДН можуть бути отримані два типи розв'язків: або мінімальний час  $\tau_{\text{min}}$ , за який PR може перемістити певний вантаж по заданій траєкторії, або висновок про те, що PR не в змозі виконати цю операцію.

Якщо рух не може бути виконаний або отримано незадовільне значення  $\tau_{\text{min}}$ , то зовнішнє стосовно ПМДН алгоритмічне забезпечення повинно прийняти рішення або про перепланування траєкторії руху, або про заміну PR, або про зміну параметрів технологічного процесу. У будь-якому разі для такої системи необхідна інформація про ММ МС, а також про закони її руху при відпрацюванні заданої траєкторії. Крім того, можуть знадобитися інші дані, отримані в ході функціонування ПМДН, наприклад, інформація про невикористані ресурси або перевантаження двигунів. Тоді зовнішній стосовно ПМДН алгоритм визначає дії для інтенсифікації виконання заданої технологічної операції і задає нові умови пошуку розв'язку. Отже, формується ітераційний процес пошуку розв'язку на вищому рівні.

Якщо знайдено задовільне значення  $\tau_{\min}$ , ним навантажуються часовий перехід часової СП (ЧСП), яка визначає послідовність і тривалість технологічних операцій у технологічному процесі.

### Контрольні завдання

1. Дати аналітичний огляд методів моделювання ГКІС.
2. Розкрити принципи об'єктно-орієнтованого підходу до вибору відповідностей «формалізм — об'єкт дослідження».
3. Пояснити, на чому ґрунтується оцінювання альтернативних варіантів семантик і в чому полягає критерій їх ефективного вибору.
4. Дайте загальну характеристику семантичного середовища як сукупності формальних методів подання властивостей об'єкта моделювання.
5. Пояснити, в чому полягають методи ранжування та попарних порівнянь при експертній оцінці критеріїв якості вибору варіантів семантик.
6. Схарактеризувати вкладену, однорівневу та мішану моделі взаємодії процесів у ГКІС.
7. Розкрити зміст понять «блокування ресурсів», «транзакція».
8. Розкрити роль методу аналогій у системному аналізі складних об'єктів.
9. Схарактеризувати сутність загального (комбінованого) підходу щодо постановки задачі моделювання процесів АТСС.
10. Сформулювати і прокоментувати основні етапи аналітичного моделювання процесів створення АТСС.
11. Розкрити мету імітаційного моделювання процесів в АТСС і схарактеризувати технологічну підсистему ГКІС як об'єкт імітаційного моделювання.
12. Навести загальну структуру ІМ технологічної підсистеми ГКІС і схарактеризувати роль основних модулів її складу.
13. Пояснити принципи формування черг заявок та організацію їх обслуговування при моделюванні процесів функціонування АТСС.
14. Розкрити зміст чотирирівневої структури системного проектування ГКІС і прокоментувати склад реєстрових параметрів та розв'язуваних на ІМ задач при моделюванні процесів ГКІС.
15. Навести і прокоментувати загальний алгоритм модуля керування ІМ.
16. Схарактеризувати загальну проблему постановки задач інтелектуалізації моделювання ГКІС.
17. Дати обґрунтування вибору ПР моделі «PUMA» як об'єкта інтелектуалізованого моделювання.
18. В загальних рисах прокоментувати складові математичної моделі МС ПР «PUMA» і обґрунтувати послідовність етапів розв'язання обернених задач кінематики і динаміки МС.
19. Розкрити процедуру керування експериментами при моделюванні процесів ГКІС і дати порівняльну оцінку традиційної та автоматизованої схем управління експериментами.
20. Детально проаналізувати умови функціонування взаємозв'язаного контуру ПМ–ІМ і розкрити роль інтелектуалізованих компонент — пошукових модулів ПМ1 і ПМ2 у структурі АІМ ГКІС.

## НОВІ ПІДХОДИ ДО МОДЕЛЮВАННЯ ТА УПРАВЛІННЯ В ГНУЧКИХ КОМП'ЮТЕРНО- ІНТЕГРОВАНИХ СИСТЕМАХ

---

---

*Уява важливіша за знання. Знання обмежене.*

*Уява – безмежна.*

Альберт Ейнштейн

*Теоретик – це людина, яка знає як проїхати,  
проте не знає, як водити автомобіль.*

Кеннет Тінан

### 12.1. Семантично-узгоджене середовище гіперпросторового моделювання ГКІС

**Особливості створення семантично-узгодженого середовища гіперпросторового моделювання.** Процеси в організаційно-технічних, технологічних, економічних, екологічних та інших складних об'єктах управління, які в сучасному техногенному суспільстві трансформувались у ГКІС, характеризуються значною кількістю параметрів, що залежать як від стану і властивостей їх компонентів, так і зовнішніх факторів, причому *модульність* і *гетерогенність* таких систем реалізується в розподіленій у просторі і часі формі.

**Означення 12.1.** *Гетерогенність системи* – це інтегрування різних за фізичною природою і складністю комплектуючих її компонент (підсистем, модулів тощо), кожна з яких, у свою чергу, також може поєднувати різні за своїми властивостями і принципом функціонування складові.

Серед інших визначальних властивостей ГКІС, що позначаються на ефективності їх розробки та використання, слід враховувати: *дискретність* процесів, які в них проходять; *конвеєрний характер руху* матеріальних (інформаційних, енергетичних, транспортних) потоків; наявність так званого *істинного паралелізму* в таких системах при проходженні процесів або подій, пов'язаних з їх виконанням; можлива наявність *циклічних утворень* у послідовностях технологічного функціонування, що потребує *синхронізації* та *координації* процесів і явищ, які відбуваються в системі; *ієрархічність побудови*, пов'язана з відповідними рівнями цільового абстрагування функцій ГКІС; *агрегатованість станів*, які змінюються під впливом сукупної дії внутрішніх і зовнішніх факторів; нарешті, *гнучкість системи*, яка проявляється в її спроможності шляхом параметричних і структурних змін, що не виходять за межі вихідного визначення цієї системи, реагувати на ініційовані ззовні зміни поточного завдання або умов (середовища) його виконання при дотримуваних прийнятих обмежень на часові та інші параметри.

Все наведене утруднює ідентифікацію об'єктів такої складності та розмірності, коли вимагається наявність математичного апарату для відображення природи

явищ, які проходять в них, з комплексним урахуванням специфіки функціонування – *семантично-узгодженого комплексно-інтегрованого середовища (СУКІС)*.

При ідентифікації та моделюванні ГКІС розглядаються як віртуально утворені *дискретно-подійні системи (ДПС)*, зміна станів яких відбувається в дискретні, непередбачувані моменти часу при здійсненні подій.

Реальні ГКІС, як правило, реалізуються у «розподіленій» формі (рис. 12.1), модульна основа якої зберігається в усіх просторових вимірах.

**Означення 12.2.** *Розподілена ГКІС* – це система із сукупністю автономних компонентів – обробляючих ресурсів, пов'язаних між собою через спеціальним чином організовані інформаційні й транспортні потоки і спрямованих на генерацію певного загального виходу (напівфабрикату чи готового виробу).

Це означення дає уявлення про ситуаційний, подійний характер інформаційно-матеріальних перетворень, яких зазнає об'єкт виробництва за час дискретного проходження процесу на окремих обробляючих ресурсах системи.

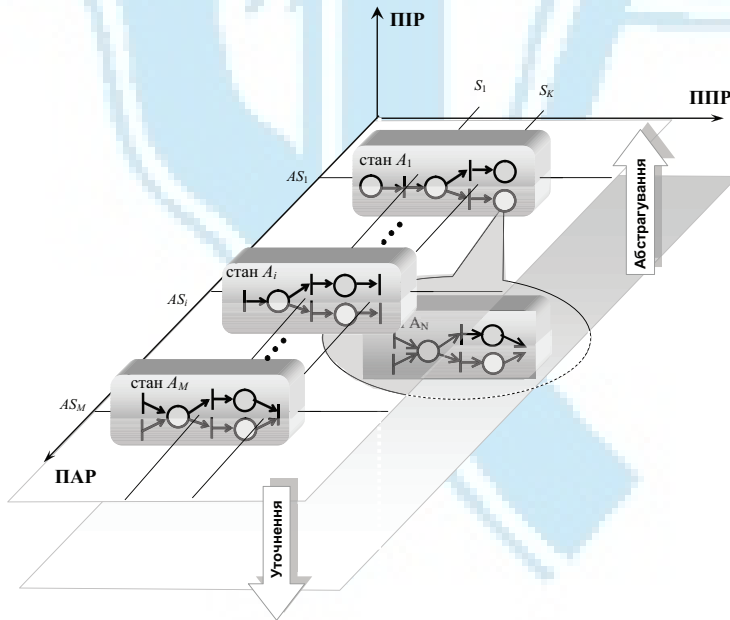


Рис. 12.1. Гіперпросторова модель дискретно-подійної ГКІС

Як зазначалося в підрозд. 9.5, така послідовність чергування подій у «плані» являє собою *планарну композицію* процесу і свідчить про ППР системи, яка ілюструється рис. 12.2.

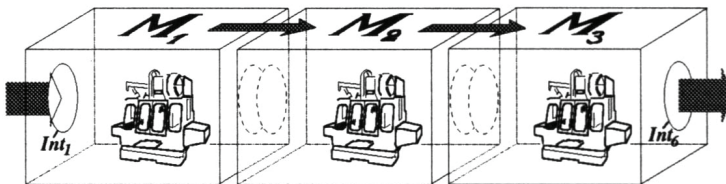


Рис. 12.2. Приклад планарної композиції системи ( $M_1, M_2, M_3$  – модульні компоненти системи;  $int_1, \dots, int_6$  – інтерфейсні зв'язки між модулями)

Головною метою ієрархічного композиційного моделювання є або «уточнення» будь-якої з компонент ДПС шляхом заміни її групою підмоделей нижчого рівня, або «абстрагування» декількох компонент ДПС шляхом заміни їх однією підмоделлю на вищому рівні подання системи. Така структурна міжрівнева трансформація ієрархічної ДПС свідчить про її ПІР, ілюстрацію якої наведено на рис. 12.3.

Утворені в результаті композиційного моделювання підмоделі після об'єднання стають внутрішніми компонентами моделі, а інтерфейси успадковуються з об'єднаних або заміщених компонент.

Нарешті, як показано в підрозд. 9.5, можливість переходу ГКІС під впливом збурюючих факторів з одного агрегатного стану в інший (в тому числі й нештатний режим функціонування) викликає необхідність такого подання в ДПС, яке відображає вплив цих факторів на якісні/кількісні показники функціонування реальної системи. Множина таких описів характеризує *композиційність станів* і утворює ПАР системи (див. рис. 9.9).

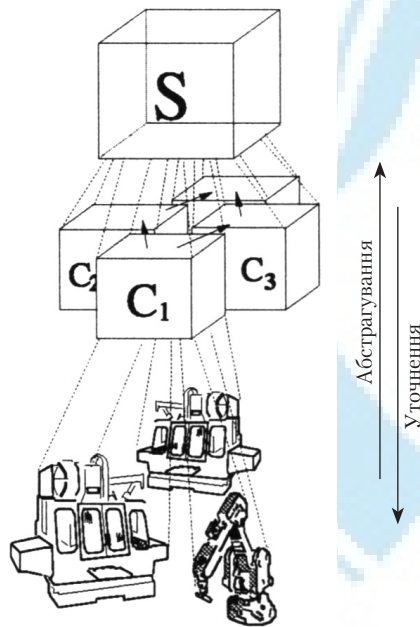


Рис. 12.3. Приклад ієрархічної композиції системи ( $S$  – системний компонент вищого рівня;  $C_1, C_2, C_3$  – підмоделі нижчого рівня)

Реалізація ДПС сама по собі є непростим завданням, яке посилюється гетерогенною природою ГКІС і складними схемами синхронізації та координації. Існуючі методи синтезу/аналізу, верифікації, ідентифікації та моделювання дуже спеціалізовані і не підтримують автоматичної послідовності «синтез – імітація – верифікація – перегляд» у межах одного семантичного середовища. Так, для відображення особливостей функціонування ГКІС використовуються методи: теорії автоматів, марковських процесів, теорії черг, макс-плюс і процесорних алгебр, імітаційного моделювання, апарату сіток Петрі, диференційних рівнянь тощо. Але з переліченого тільки апарат СП задовольняє першочергову вимогу щодо засобів подання та синтезу/аналізу ДПС – ефективно відображувати асинхронність та істинний паралелізм у системі і має потужні можливості для аналізу її поведінки. Проте недос-



татні в СП розвиненість синтаксичного опису і композиційні спроможності часто вимагають відповідних розширень цього апарату на конкретну об'єктну область.

**Процедурна реалізація гіперпросторових моделей ГКІС.** З урахуванням сказаного створення СУКІС сприятиме підвищенню ефективності розв'язання задач ідентифікації процесів і композиційного моделювання на основі комбінованих методів гіперпросторової семантики на стадії створення системи та управління нею і забезпечує: *по-перше*, визначеність формально обґрунтованих математичних правил; *по-друге*, аналітичне передбачення складних динамічних і тупикових ситуацій і взаємних блокувань компонент, зацикловань і т. ін.; *по-третє*, набір засобів розв'язування і попередження проблемних ситуацій. Особливого значення у зв'язку з гіперпросторовою розподіленістю ГКІС набувають методи вибору й оптимізації ефективних значень параметрів настроювання ДПС, що пов'язано з необхідністю прийняття рішень при невизначеності і нечіткості умов переходу системи в новий агрегатний (під дією зовнішніх факторів) або дискретний (відповідно до вектора управління) стани [60]. Необхідні вимоги, що накладаються специфікою ДПС на умовну цільову СУКІС, ілюструються цільовим середовищем моделювання (рис. 12.4). Насамперед базовий формалізм 1, на основі якого будуються складові моделі  $M_k, M_m, M_n, \dots$  семантичного середовища і який може відображати асинхронність, паралелізм, тупикові ситуації при функціонуванні ДПС. Цей формалізм має композиційні властивості і забезпечений набором відповідних операторів алгебраїчного типу для довільних сполучень 2 і 3 підмоделей  $M_k - M_m, M_m - M_n$ , відповідно, що дає змогу реалізувати часткове залучення підмоделей (так, композиція підмоделей  $M_m$  і  $M_n$ , де  $M_n$  вважається частково «охопленою» оператором, що на рис. 12.4 умовно відображено її невизначеним положенням 4). Можуть моделюватися як однозначно визначені 5, так і з різною (в тому числі нечітко поданою) залежністю 6 елементів міжмодульні матеріальні та інформаційні потоки. При цьому незалежно від паралельності комбінування і дроблення структури на підмоделі поведінка досліджуваної ГКІС 7 однозначно визначена ДПС. Моделююче середовище ДПС враховує взаємодію 8 із зовнішніми методами прийняття рішень і підтримується відповідними методами 9.

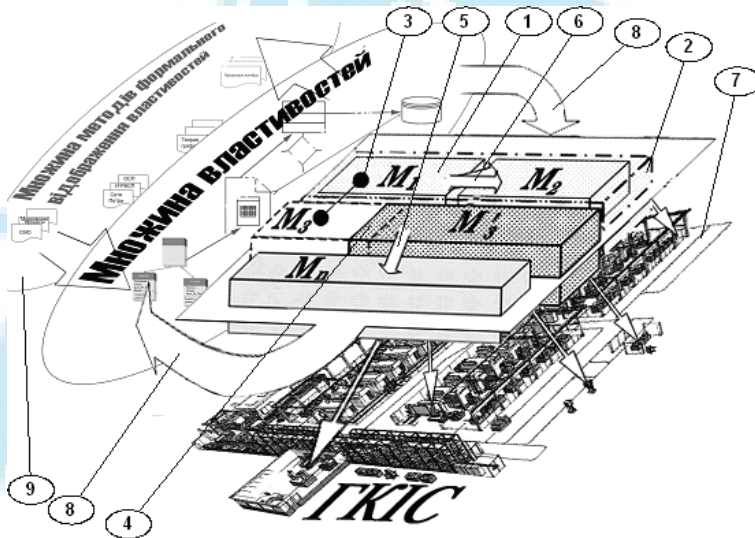


Рис. 12.4. Цільове середовище моделювання ГКІС

З урахуванням сказаного можна визначити таку процедурну послідовність формалізації гіперпросторових ГКІС [65]:

*Процедура 1.* Будується інформаційна модель виробу, яка подається множиною  $A$  властивостей і станів конструктивно-технологічних, структурних, економічних, інформаційних тощо параметрів. Якщо поділити множину  $A$  на неперетинні підмножини  $a_{ij}$  різних ієрархічних рівнів і врахувати структуру об'єкта дослідження (виробу) та реальні зв'язки між його компонентами, то такий об'єкт як система подається інформаційною моделлю у вигляді графа. Проміжні вершини цього графа визначають властивості окремих складових виробу, а корінь дерева — узагальнену властивість виробу; дуги графа вказують на взаємозв'язок усіх складових об'єкта. Матриця суміжності граф-моделі дає змогу врахувати всі кількісно-якісні характеристики процесу виготовлення і по суті формувати технологічний процес.

*Процедура 2.* На основі структурно-функціонального аналізу граф-моделі, а також множини  $A$  і підмножин  $a_{ij}$  компонентів виробу на кожному з ієрархічних рівнів формується система засобів технологічного забезпечення ГКІС і структура останньої [19].

*Процедури 3–5.* Обґрунтовується вибір СУКІС — набір ефективних взаємоузгоджених семантичних засобів для: відображення логіки диспетчеризування і динаміки ситуаційного моделювання транспортних та інформаційних потоків в обробляючих ресурсах ГКІС (тобто формується простір ППР) на різних рівнях абстракції з метою уточнення динамічних властивостей і показників ефективності у заданих обмеженнях умовах (простір ППР); одночасно враховується вплив зовнішніх збуджень і внутрішніх критичних факторів, які мають імовірнісний характер, і відображується можливість виникнення нештатних ситуацій у системі (формується простір ПАР).

*Процедури 6–7.* На основі вибраної СУКІС будується гіперпросторова модель ГКІС з високорівневою сітковою семантикою у вигляді апарату модифікованих СП (див. розд. 8), а також низкорівневих підмоделей ГКІС із залученням відповідних синтаксичних подань та операційної семантики, в тому числі розширень апарату СП (див. розд. 7 і 8), рівнянь динаміки (див. підрозд. 11.5) і процесних алгебр (див. розд. 2).

*Процедура 8.* На базі об'єднання імітаційного моделювання (див. розд. 10) з методами задоволення обмеженням реалізуються принципи обробки нечітких ситуацій композиційного моделювання і прийняття рішень в умовах невизначеності (див. розд. 4 і 9) при інтелектуалізованому управлінні ходом технологічного процесу, що реалізується у ГКІС. Одночасно на основі спеціальних програм планування і проведення експерименту інтелектуалізованими компонентами системи (див. підрозд. 6.4 і 11.6) здійснюється прогін моделі та оптимізуються параметри компонент і ГКІС загалом [140].

## 12.2. Нечітке метакерування у ГКІС

**Концепція нечіткого метакерування.** Підхід, який далі розглядається, не є протиставленням більш загальним схемам розподіленого штучного інтелекту, які детально розглядалися в розд. 4 і 9 і які ефективно розв'язують складні та універсальні задачі. В розглядуваному випадку ідея полягає в тому, щоб, розглядаючи *нечіткі керуючі компоненти* (НКК) як складні системи, використати переваги нечіткого підходу до керування роботою самих НКК. При цьому завдання нечіткого метакерування полягає в динамічній побудові з існуючих функціонально-спеціалізованих

блоків таких конфігурацій, які якнайкраще відповідають поточному стану об'єкта управління – ГКІС (рис. 12.5, контур метакерування виділений жирною лінією).

Реалізація цієї ідеї можлива за допомогою побудови *гнучкої інтелектуалізованої мультимодульної конфігурації* (ГІММК), особливостями якої є:

- використання модулів різних типів;
- високий ступінь паралелізму;
- децентралізоване структурне та параметричне метакерування в рамках НКК.

Модульна структура НКК формується з множини  $M = \{M_1, \dots, M_n\}$  пов'язаних між собою модулів. На вхід НКК (рис. 12.5) надходить множина  $V^{(x)} = \{V(X_1), \dots, V(X_k)\}$  значень вхідних змінних  $X = \{X_1, \dots, X_k\}$  від деяких зовнішніх джерел (визначають інформаційне поле агрегатних станів системи). Фаззі  $F$ -перетворювач («чіткий  $\rightarrow$  нечіткий») трансформує  $V^x$  у множину факторів  $F^{(x)} = \{F_1^{(x)}, \dots, F_l^{(x)}\}$  (є нечіткими множинами, заданими на значеннях вхідних змінних) з відповідними факторами достовірності (ступенями належності)  $C^{(x)} = \{C_1^{(x)}, \dots, C_l^{(x)}\}$ . Дефаззі  $D$ -перетворювач («нечіткий  $\rightarrow$  чіткий») трансформує множини своїх вхідних факторів  $F^{(y)} = \{F_1^{(y)}, \dots, F_p^{(y)}\}$  і відповідних факторів достовірності  $C^{(y)} = \{C_1^{(y)}, \dots, C_p^{(y)}\}$  у множину  $W^{(x)} = \{W(Y_1), \dots, W(Y_k)\}$  значень керуючих змінних  $Y = \{Y_1, \dots, Y_m\}$  (детально процедури фаззі-перетворення розглядалися у розд. 4 і 10).

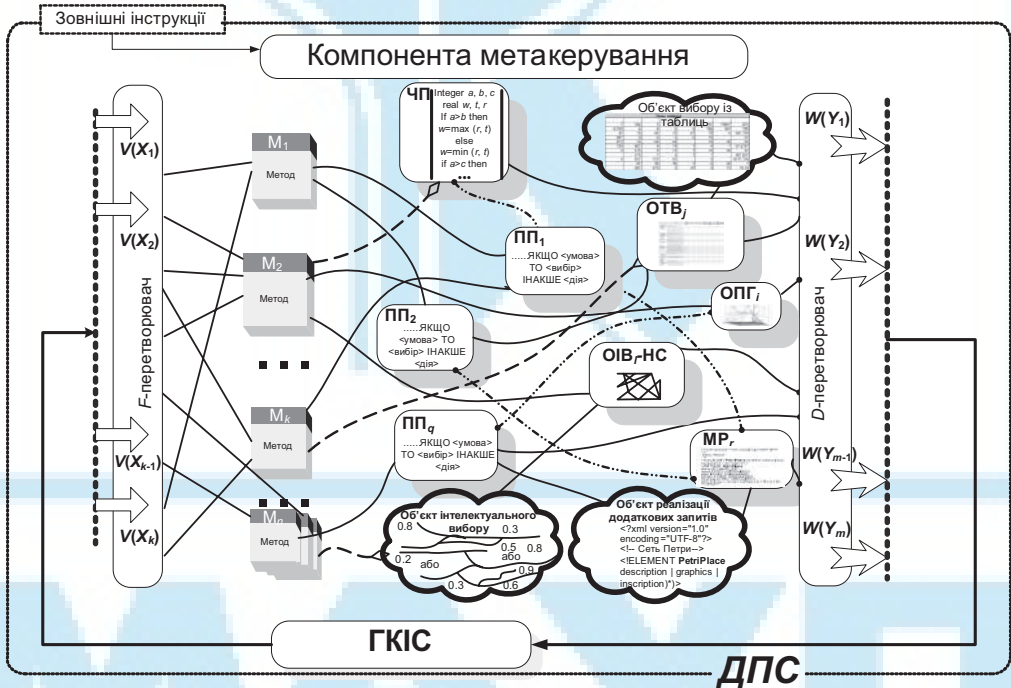


Рис. 12.5. Загальна модульна структура ГІММК

Компоненти ГІММК і їх взаємодія в процесі функціонування. Функціонування ГІММК формується із загального внеску її компонент (на рис. 12.5 вони символічно подані модулями з продукційних правил (ПП), експертних систем (ЕС),

нейросітками (НС) – об'єктами інтелектуалізованого вибору (ОІВ), числовим процедурами (ЧП), об'єктами побудови графіків (ОПГ) для візуалізації результатів моделювання та поліпшення умов проведення експертного аналізу, а також модулем реалізації (МР), якщо експерт робить висновок про необхідність запровадження додаткових правил, обмежень тощо. Базуючись на вхідних даних і меті керування, ГІММК виробляє на виході у прийнятних часових межах набір керуючих значень  $W(Y_1) - W(Y_m)$ .

Прийнявши за  $H, G$  відповідно набори (імена) вхідних і вихідних змінних, а за  $Z$  – набір з усіх змінних (включаючи вхідні і вихідні), що беруть участь у міжмодульному інформаційному обміні  $Z \supset H, Z \supset G$ , а також враховуючи, що структура ГІММК формується з множини  $M = \{M_1, \dots, M_n\}$  пов'язаних між собою модулів, тоді, використовуючи множину компонент із сенсорів  $S_i \subset S \cup M$  (вхідних стосовно цього модуля  $M_i$ ) і дій  $A_i \subset A \cup M$  (вихідних стосовно  $M_i$ ), формальний опис  $i$ -го модуля  $M_i$  можна подати у вигляді:

$$M_i = (T_i, D_i, V_i, W_i, S_i, A_i, B_i, E_i),$$

де  $T_i$  – тип модуля (ЕС, НС, ЧП і т. п.);  $D_i$  – тип умови для активації (наприклад,  $D_1$  – зміною вхідних даних модуля;  $D_2$  – естафетною умовою, тобто завершенням поточного прогону визначених попередніх модулів;  $D_3$  – незалежною активацією, в тому числі згідно з часовим законом;  $D_4$  – у відповідь на запит від іншого модуля, і так далі);  $V_i, W_i$  – набори вхідних і вихідних змінних модуля  $M_i$  відповідно;  $B_i, E_i$  – набори розташованих вище стосовно  $M_i$  (чий вказівки він виконує) і підпорядкованих (стосовно  $M_i$ ) модулів відповідно.

При цьому:

$$\forall M_i \in M : (V_i \cap W_i = \emptyset) \wedge (S_i \neq \emptyset) \wedge (A_i \neq \emptyset) \wedge (V_i \neq \emptyset, V_i \subset H \cup (\cup W_j)) \wedge \\ \wedge (W_i \neq \emptyset, W_i \subset G \cup (\cup V_j)),$$

а ситуація  $|B_i| > 0$  є допустимою. Крім того, для відображення факту можливості активзації  $M_i$  розв'язком інших модулів, вважаємо, що значення спеціальної «перемикальної» змінної  $\bar{\theta} \in V_i \setminus (H \cup G)$  визначає стан («активний», «очікування») модуля  $M_i$ .

Зазначимо також, що модулі ГІММК можуть характеризуватися: використовуваною моделлю –  $N_1$  (тобто НС, ПП, ЧП тощо); типом подання –  $N_2$  (наприклад, числовий, символічний або більш детально: фреймовий, семантичний і т. д. Зрозуміло, що  $N_1$  і  $N_2$  тісно пов'язані); особливостями виконання –  $N_3$  (як окремі фізичні (мікросхеми, комп'ютери) або віртуальні об'єкти (в межах тієї ж програми), і так далі).

Тип  $N_2$  передбачає необхідність проміжних перетворень даних, інтеграцію множин числових значень і формування символічних умов для логічного виведення, а також навпаки – перетворення фактів (виведень) у числові значення (у разі нечітких подань це відповідає перетворенням «чіткий  $\rightarrow$  нечіткий» і «нечіткий  $\rightarrow$  чіткий» відповідно). Ці перетворення можуть бути або «вбудованими» у загальне функціонування модуля, або виконуються спеціальними процедурами трансформацій. Сам модуль, у свою чергу, може також мати модульну структуру. Зокрема, для реалізації перетворень «нечіткий  $\rightarrow$  чіткий», наприклад, за наявності невизначеності при ідентифікації агрегатних станів ГКІС такий модуль може виконуватися у вигляді фаззі-контролера, спрощену структуру якого подано рис. 9.16, а принцип функціонування детально розкрито у розд. 4.

Кожний такий модуль  $M_i$  формується з множини специфікацій  $SP$  якісних правил  $SP_i = \{R_{i1}, R_{i2}, \dots, R_{ik}\}$ , наприклад, у формі «ЯКЩО» вихідному АС відповідає множина  $X$  значень вхідних змінних, «ТО» відкликовій реакції вектора керування у фазовому просторі відповідає множина  $Y$ , через що для математичної обробки таких правил необхідні операції взаємодії між окремими нечіткими множинами, які належать до частини «ТО». Як наголошувалося в розд. 4, для фаззі-контролерів продукування розв'язків у таких випадках характеризується взаємодією вхідних і вихідних функцій належності (ФН), причому операції «І» та «АБО» використовуються як зв'язування у правилах, якщо вони застосовуються у множинах, елементи яких належать до різних основних множин.

Після операцій фаззіфікації, логічного розв'язання, дефаззіфікації (див. підрозд. 4.4) визначається результуюча функція ФН (4.8), яку для нашого загального випадку запишемо як

$$\mu_G = \max_{k=1, \dots, n} \left\{ \min[\mu_{I_k}, \mu_{Nk}(G)] \right\},$$

де  $n$  — кількість правил;  $\mu_{I_k}$  — величина істинності частини «ЯКЩО» у  $k$ -му правилі;  $\mu_{Nk}(G)$  — функція ФН частини «ТО»  $k$ -го правила.

При дефаззіфікації можна використовувати будь-який з методів зведення до чіткості, розкритих у підрозд. 3.4, проте найчастіше застосовують «центроїдний метод», який полягає у визначенні значення абсциси положення  $G_{ц}$  центра ваги площі, що лежить під графіком вихідної ФН (див. рис. 3.4 і рис. 9.16, етап  $D$ -перетворення).

**Міжмодульна взаємодія у ГІММК.** Взаємодія між модулями характеризується їх типом і статусом і в загальних рисах пояснювалася в підрозд. 6.6 і 6.4.

Для взаємодії модулів різних типів характерними є:

- *випадок E1*, коли модуль  $M_k$  (являє собою автономну прогнозуючу ЕС) звертається до модуля  $M_l$  (відповідає за числову процедуру, яка активізується як права частина деякої продукції у  $M_k$ ) для того, щоб розрахувати імовірність деякої події:  $M_k \in B_l$ ,  $M_l \in E_k$ ,  $T_k = \langle \text{ЕС} \rangle$ ,  $T_l = \langle \text{ЧП} \rangle$ ,  $D_i = \langle D_4 \rangle$ ,  $W_l \subset V_k$ ,  $T_i = \{M_k\}$ ;
- *випадок E2*, коли модуль  $M_i$  (є числовою оптимізаційною процедурою) звертається до  $M_j$  (є продукційною ЕС) за допомогою у визначенні найбільш вигідного напрямку руху до наступної точки;
- *випадок E3*, коли модуль  $M_p$  (являє собою ПІД-контролер) безперервно виробляє керуючі значення, а модуль  $M_r$  є НС і призначений для розпізнавання деяких ситуацій і виклику за необхідності відповідного модуля для їх обробки.

Для взаємодії ідентичних (або подібних) за типом моделей відбувається:

- *випадок E4*, коли  $M_p$ ,  $M_n$  — однотипні ЕС, які одночасно розв'язують дві незалежні задачі у загальному процесі керування;
- *випадок E5*, коли деякий, такий, що ґрунтується на продукційних правилах, модуль  $M_k$  здійснює подальшу обробку даних, які є результатом виведення у двох попередніх і також побудованих на продукційних правилах модулів  $M_i$  і  $M_j$ , тоді як останні вже зайняті обробкою наступної «порції» даних.

За статусом модулів взаємодія може бути *рівнозначною* (див. E3, E4, E5) і *скоординованою* у вигляді «базовий — підпорядкований» (див. E1, E2).

В разі необхідності модифікації ГИММК шляхом додавання нового модуля  $M_i$  потрібно враховувати відношення  $\Delta_i$  інформації на виході цього модуля до даних на виході інших модулів. Якщо  $\Delta_i = W_i \cap (\cup_{j \neq i} W_j) = \emptyset$ , то жодні додаткові дії не потрібні.

Послідовні модифікування шляхом введення нових модулів можуть суттєво ускладнити систему, збільшити витрати на обчислення, призвести до зайвих компонентів і т. ін., проте односпрямована структура та прогнозованість характеру взаємодії забезпечують можливість послідовного визначення загальної залежності «вхід — вихід» у ГИММК шляхом просування уздовж головного потоку даних і з урахуванням характеристик окремих модулів.

Таким чином, враховуючи прагнення забезпечити у ГИММК в основному односпрямований потік активацій, допускаються два типи відхилень від цього правила (рис. 12.6):

1. Звертання (запити) до інших (підпорядкованих) модулів (випадки  $E3, E4$ , які на рис. 12.6 позначені подвійними вертикальними стрілками між модулями  $M_3, M_5$  або  $M_i, M_m$ ).

2. Циклічні повторювання деякого фрагменту (пунктирна лінія на рис. 12.6, модулі  $M_{i+1} - M_{i+2} - M_{i+3}$ ).

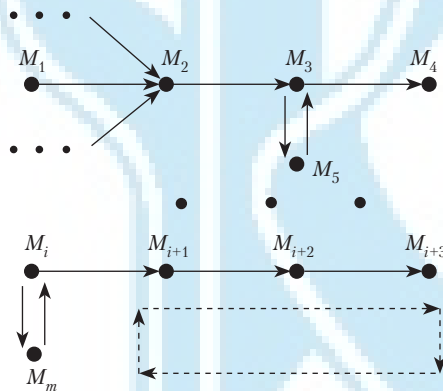


Рис. 12.6. Потіки активації модулів у ГИММК

При такій міжмодульній взаємодії в структурі ГИММК кожний базовий модуль, передавши свої дані на виході іншим модулям, готовий до обробки чергової «порції» даних, не очікуючи на завершення роботи кінцевих (тобто тих, для яких  $W_i \subseteq G$ ) модулів (винятком є модулі, які належать до ітеративно повторюваних фрагментів). За такої інтерпретації різні «шари» модулів одночасно беруть участь в обробці задач, «запущених» вхідними модулями у різні моменти часу; як наслідок, фрагменти ГИММК можуть мати різні середні частоти активації.

Можна розв'язувати й обернену задачу — при відомому загальному законі керування реалізується пошук (синтез) більш простої структури з таким самим виходом. Отже, еволюція ГИММК складається з послідовних і переміжних фаз кількісних (зростаюча конфігурація) та якісних (заміна оптимізуючою структурою) змін.

**Сутність метакерування.** Метакерування у ГИММК має розподілений характер і охоплює такі дві функції:

1. Базовий модуль (позначимо їх множину через  $M^{(B)}$ ) приймає рішення про необхідність підключення до розв'язання задачі деяких інших базових модулів (наприклад, випадок  $E3$ ).

2. Базовий модуль динамічно приймає рішення про «важливість» деякого іншого базового модуля стосовно третього модуля (наприклад, модуль  $M_i$  динамічно підстроює відповідне вагове значення  $\beta_{jk}$  деякого модуля  $M_j$  для модуля  $M_k$ , причому  $\beta_{jk} \in V_k$ ,  $\beta_{jk} \in U \setminus (H \cup G)$ ). Зокрема,  $M_i$  може для себе оцінити важливість модуля  $M_j$ , що спостерігається, коли  $M_k$  отримує дані про ті самі параметри від декількох модулів, і, таким чином, продукує «інтегральний» результат (це дає змогу підтримувати однозначність керуючих значень на виході).

**Означення 12.3.** Модулі з функціями метакерування — це модулі, які спроможні приймати рішення щодо активації інших модулів або щодо їх «важливості» («ваги»).

Множини таких модулів позначаються через  $M^{(M)}$ , причому  $M^{(M)} \subseteq M^{(B)}$ . Як правило, саме модулі  $M^{(M)}$  є «справжніми» інтелектуальними компонентами у ГІМ-МК (тобто ЕС, НС і т. д.), а не числовими оптимізаційними процедурами. Подібні процедури можуть виконуватися інтелектуальними надбудовами у ДПС — пошуковими модулями із задачами, розкритими у підрозд. 6.4 та 11.6.

**Означення 12.4.** Гнучкою інтелектуалізованою мультимодульною конфігурацією є така мультимодульна конфігурація, яка містить модулі  $M^{(M)}$  з функціями метакерування, які реалізують механізм розподіленого динамічного визначення «важливості» інших модулів з різною природою, формують різні закони керування, забезпечують паралельність роботи модулів різнорідних «шарів» і дають змогу гнучко реагувати на зміни стану зовнішнього середовища шляхом піднастроювання загального виходу згідно з керуванням, яке найбільше задовольняє поточну ситуацію.

**Сіткові модулі взаємодії керуючих потоків у модульних структурах НКК.** Прийняття рішень у реальному часі в умовах складних швидкозмінюваних обставин, які ведуть до зміни АС, можна реалізувати на принципах ситуаційного планування. Для цього структури НКК повинні формуватися із сукупності модулів, які взаємодіють між собою і з зовнішнім середовищем і відображають системи правил для прийняття рішень щодо характерних для того чи іншого АС сполучень та значень інформаційних ознак зовнішнього середовища. Модульність може бути наслідком функціональної кластеризації бази знань, групування правил за пріоритетом їх використання і т. д. Як вже зазначалося, використання правил може мати різну природу: від нейросітки або виділеної частини продукційної бази знань до значень експертної системи. Кожний модуль приймає рішення виходячи з вхідних (сенсорних) даних і/або передумов, які встановлюються суміжними модулями. Стратегії ініціювання модулів можуть бути часовими, подійними або комбінованими, а ініціюючими подіями — зовнішні (зміна сенсорних даних) та внутрішні.

З урахуванням вимог семантичного узгодження засобів опису процесів ГКІС бажано, щоб і механізм міжмодульної взаємодії в структурах НКК реалізовувався на модифікаціях СП (див. розд. 7 і 8). У такому випадку кожному окремому модулю  $M_i$  відповідає сітка  $N_i$  (на рис. 12.7, а виокремлена пунктирною лінією), у якій спрацьовування переходу  $t_i$  відповідає прогону  $M_i$ , а його позиції на вході й виході — з'єднанням з відповідними модулями на вході й виході. Загальна сітка, що відповідає механізму координації модулів у ГІММК, реалізується шляхом сполучення вхідних і вихідних позицій таких підсіток. Можливість запиту до іншого модуля модулюється структурою на рис. 12.7, б; принцип метаактивації інших модулів зображено на рис. 12.7, в (⊙ — позначення модуля з метафункцією); циклічне повторювання фрагмента, охопленого метафункцією, подано структурою на рис. 12.7, з.

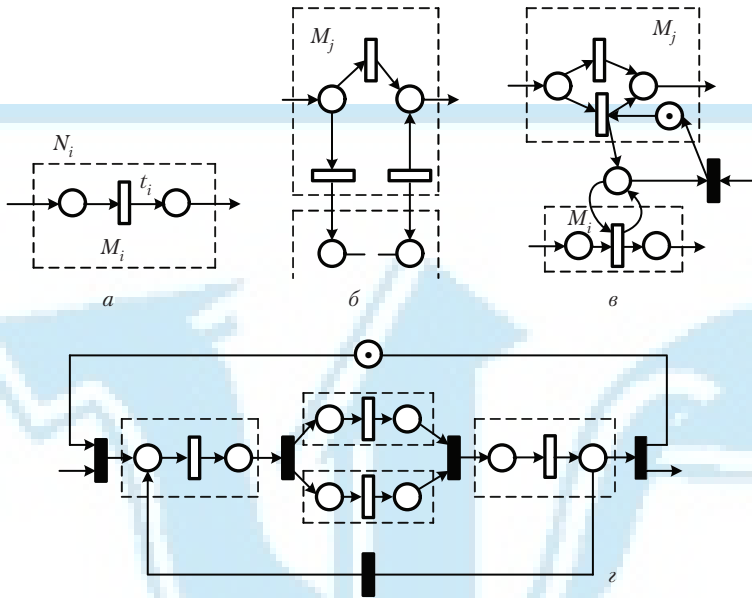


Рис. 12.7. Подання фрагментів НКК за допомогою СП

**Сіткова семантика потоків ініціалізації.** Skorистаємося більш зручною формою сіткового подання блоків у НКК. Нехай кожен модуль ініціюється елементарною СП-сіткою  $N_p$ , яка складається з однієї позиції  $p_i$  і одного переходу  $t_i$  (рис. 12.8, а). Спрацьовування  $t_i$  відповідає спрацьовуванню модуля  $M_i$ . Вхідні дуги позиції  $p_i$  ведуть від переходів  $t_w, t_v$ , які відповідають вхідним модулям  $M_p$ , а дуги на виході з'єднують  $t_i$  з позиціями  $p_j, p_q$ , які відповідають модулям – споживачам даних на виході з  $M_i$ . Цей випадок визначається умовами внутрішньої ініціалізації подій в НКК.

Якщо модуль  $M_j$  має зовнішню ініціюючу подію, то наставання останньої моделюється спрацьовуванням спеціального «відкритого» переходу  $t^{\text{від}}$ , причому  $\{t^{\text{від}}\} \in T^{\text{від}}$ , де  $T^{\text{від}}$  – множина всіх таких переходів. Позиції, суміжні з відкритими переходами, також є відкритими і позначаються  $p^{\text{від}}$ , причому  $\{p^{\text{від}}\} \in P^{\text{від}}$  (рис. 12.8, б).

Поведінка модульної структури НКК може інтерпретуватися та аналізуватися у термінах властивостей сітки  $N$  (живучість, еквівалентність тощо), які впливають із спеціальних правил спрацьовування переходів [61].

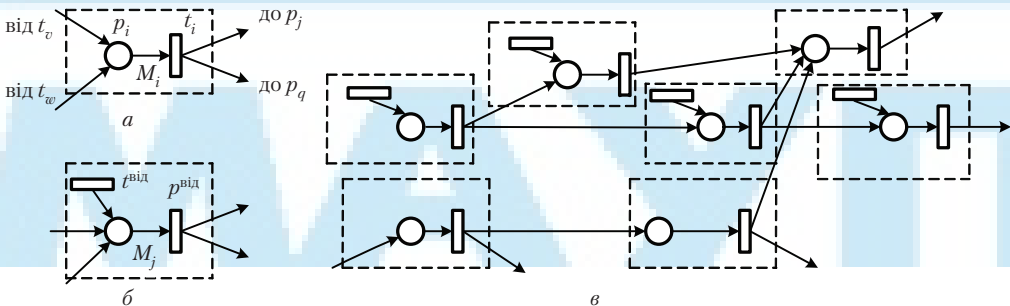


Рис. 12.8. Сіткова інтерпретація модулів НКК



### 12.3. Розподілене керування «критичними» ресурсами у циклічних ГКІС

**Розподілене метакерування у ДПС.** Як зазначалося раніше (див. розд. 7 і 8), теорія класичних СП набула подальшого розвитку у вигляді об'єктно-орієнтованих модифікацій через необхідність адекватної інтерпретації явищ, які відбуваються при функціонуванні ГКІС, і відображення ієрархічності побудови інтегрованих у них підсистем (технологічної, керуючої, підготовки виробництва, планувальної та інформаційної), а також паралельності процесів, які відбуваються в ГКІС.

Ці розширення базового СП-методу згруповано у два напрямки. *Перший* пов'язаний із структурними перетвореннями сіток на основі типових сіткових блоків, типових складових елементів процесів та конструкцій і спрямований на підвищення виразних можливостей СП, а головне, полегшення процесів аналізу коректності (проблема досяжності, безпеки і живучості) і синтезу власне сіток й об'єктів моделювання (розв'язання типових конфліктів). *Другий* напрямок має на меті розширення меж об'єктної орієнтації до розв'язання задач щодо відображення та поліпшення якісних і кількісних показників ГКІС, а саме, продуктивності та надійності [139; 140]. Вимоги до цих показників можна забезпечити, проте лише методами *розподіленого керування* (РК), де рішення приймаються локальними, автономними контролерами, поєднаними в комп'ютерні мережі. Такий підхід потребує, однак, деяких нових стратегій, орієнтованих на розподілені процедури запобігання тупиковим ситуаціям.

Прикладом задачі розподіленого метакерування в ДПС при моделюванні процесів ГКІС може служити керування так званими критичними ресурсами. Групові технології, які реалізуються в ГКІС, забезпечують обслуговування широкої номенклатури виробів із змінними маршрутами прямування через об'єднані транспортними зв'язками системні ресурси. При цьому самі ГКІС розглядаються як такі, що складаються з автономних циклічних процесів розподілених систем з вільним вибором віртуальних (визначених маршрутом прямування виробів) конфігурацій та функціональних керувань.

Циклічні процеси ділять загальний набір ресурсів, доступ до яких регламентується їх перепускною спроможністю та пріоритетністю обслуговування. В обох випадках небезпеку становлять конфліктні ситуації, наслідком яких можуть бути перестоювання через залучення в циклічний процес ресурсів з різними для поточної технології виготовлення виробів часовими завантаженнями.

Ураховуючи, що на другому, модульному рівні ієрархічного абстрагування ГКІС як цільову функцію прийнято використовувати показники циклової продуктивності  $Q_{ц}$  коефіцієнта технологічного використання устаткування  $\eta_{ТВ}$ , якісно оцінимо можливість зниження собівартості  $C$  продукції, яка виступає, поряд з фактичною продуктивністю  $Q_{ф}$  системи, показником ефективності на її третьому, системному ієрархічному рівні. Для цього розглянемо як компоненту модуля ГКІС гнучкий виробничий модуль ГВМ, поданий на рис. 12.9, який містить обслуговуючі роботи ПР, вхідний/вихідний буферні нагромаджувачі  $B4, B6$  і технологічні машини  $M5$  і  $M6$  — у цьому випадку роботизовано складальні центри, які мають різні перепускні спроможності щодо конкретного технологічного процесу (наприклад,  $Q_{M5} > Q_{M6}$ ).

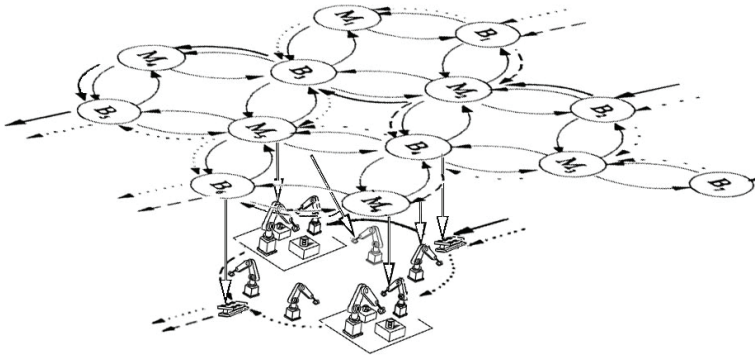


Рис. 12.9. Компонентний модуль ГКІС – ГВМ

Тут «критичний» ресурс –  $M_6$  з більш низькою продуктивністю  $Q_{M_6}$ . Традиційно «вузьке» місце «розширяється» або встановленням додаткового ресурсу, або вжиттям заходів щодо інтенсифікації процесів у ресурсі. У першому випадку собівартість  $C$  виробу збільшується через збільшення капітальних витрат, а в другому – через зниження показника надійності (ймовірність  $P$  безвідмовної роботи) обробляючого ресурсу та збільшення у зв'язку з цим витрат на його відновлення та вимушені перестоювання.

Якісна оцінка ситуації та можливих рішень впливає з графіків, які відображають сказане вище і наведені на рис. 12.10.

Із збільшенням перепускної спроможності ресурсу над обумовленою нормальними умовами експлуатації (коефіцієнт завантаження  $K_3 \rightarrow 1$ ) різко знижуються показники надійності  $P = f(Q_{II})$  (ділянка 1'), що спричиняє різке підвищення витрат на відновлення та втрати від перестоювань, а отже, пов'язаної з цим частки собівартості  $C_1 = f(P)$  виготовлюваної продукції (ділянка 2').

Проте із збільшенням циклової продуктивності  $Q_{II}$  складова  $C_2 = f(Q_{II})$  знижується. Таким чином, існує оптимальне значення  $C = f(P, Q_{II})$ . З урахуванням викладеного можна підвищувати перепускню спроможність «критичного» ресурсу ГВМ до рівня оптимальної  $C_{min}$  собівартості виробу, а недовикористані потужності інших задіяних у циклічному процесі цього ГВМ ресурсів використовувати для розв'язання конфліктних ситуацій у ресурсах суміжних ГВМ (див. рис. 12.9).

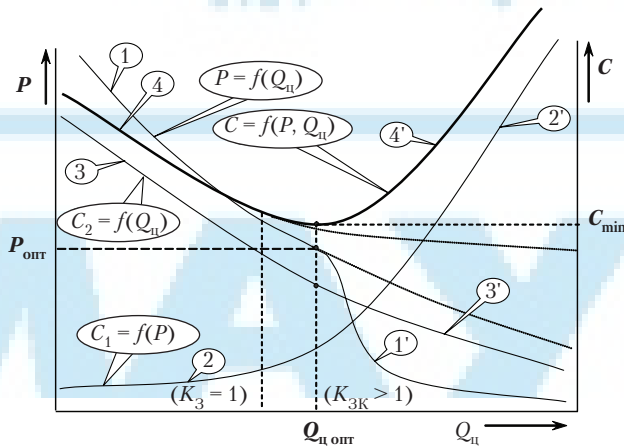


Рис. 12.10. Якісна оцінка критичного завантаження ресурсів ГВМ

Алгоритм функціонування інтелектуальної компоненти, яка входить до складу ДПС і керує призначенням ресурсів, зводиться до таких етапів:

#### Алгоритм 12.1

**Етап 1.** Визначення коефіцієнтів завантаження  $K_{3i}$  у  $i$ -тих ресурсів ГВМ та виявлення «критичного»  $K_{3к}$  ресурсу щодо конкретного виду оброблюваного виробу.

**Етап 2.** Розрахунок функцій  $P = f(Q_{1i})$ ,  $C_1 = f(P)$ ,  $C_2 = f(Q_{1i})$  для нормальних ( $0 \leq K_3 \leq 1$ ) (відповідно ділянки 1, 2 і 3) та інтенсифікованих ( $K_3 > 1$ ) (ділянки 1', 2', 3') умов експлуатації «критичного» ресурсу; визначення ( $K_3 > 1$ ) для оптимального значення  $C = C_1 + C_2 \Rightarrow \min$ .

**Етап 3.** При виявленні нового кандидата на «критичний» ресурс ГВМ у зв'язку з підвищенням перепускної спроможності модуля до  $Q_{\min}$  — перехід до етапу 2 для цього «критичного» ресурсу; якщо ні, то перехід до етапу 4.

**Етап 4.** Визначення запасу перепускної спроможності  $\Delta K_{3(i-1)}$  для кожного з решти  $(i-1)$ -тих ресурсів ГВМ.

**Етап 5.** Підбирання «робіт» (технологічних переходів) з переліку тих, що беруть участь у конфліктних ситуаціях в інших ГВМ системи.

**Етап 6.** Встановлення пріоритетів на обслуговування з переліку «робіт», які беруть участь у конфліктних ситуаціях за доступ до обмежених ресурсів з перевіркою їх завантаження починаючи з етапу 1 алгоритму.

Конкурентне обслуговування веде до втрати продуктивності складовими циклічних процесів у зв'язку з очікуванням на доступ до «робіт» загального ресурсу. Мірою ефективності підбору «робіт» на етапах 5 і 6 може бути коефіцієнт  $\eta$  продуктивності ГКІС, який обчислюється за формулою

$$\eta = 1 - \sum_{i=1}^n (T - m_i T_i) / nT,$$

де  $n$  — кількість обробних ресурсів;  $T, T_i$  — цикловий час ГВМ та  $i$ -го ресурсу відповідно;  $m_i$  — кількість циклів у  $i$ -го ресурсу за період  $T$ . Величина  $\eta$  сходиться до одиниці при ефективному використанні ресурсів ГВМ (виходячи із запланованої для ГКІС продуктивності  $Q_{1i}$ ).

**Етап 7.** Перевірка на безпеку, живучість і відсутність тупиків за деревом досяжності СП-моделі ГКІС.

#### Стратегії розподілу робочих потоків ГКІС з уникненням тупикових ситуацій.

Запропонований вище підхід дає змогу формувати здійснювані правила запобігання тупиковим ситуаціям і розв'язання «вузких місць» за рахунок використання інформації про ресурси, необхідні для кожної з операцій виробничого процесу. При цьому *стратегіями розподіленого керування* для запобігання тупикам є [79]:

- синхронізуючі зони;
- граф розподілу заявок;
- протоколи розділення потужності.

Отже, керування за такими стратегіями стає розподіленим між зонами, сегментами і локально діючими правилами диспетчеризування відповідно.

Зауважимо тільки, що компоновальні структури ГКІС для дрібносерійного виробництва будуються за принципом міжопераційного транспортування деталей «верстат — склад — верстат» (рис. 12.11) і поділяються на централізовані та децентралізовані. Централізована компоновальна структура визначається наявністю централізованого міжопераційного складу-накопичувача з одним або декількома автономними транспортними модулями — електроробочими (ЕРК).

Децентралізована компоновальна структура характеризується наявністю у кожного верстата накопичувача, який обслуговується також одним або декількома ЕРК. Ці приверстатні накопичувачі, як і міжопераційний склад, забезпечують зберігання запасу заготовок або напівфабрикатів на довгостроковий період роботи ГКІС. Переважаючим правилом побудови компоновань ГКІС для дрібносерійного виробництва є поєднання принципів централізації та децентралізації, тобто використання централізованого міжопераційного складу-накопичувача разом із приверстатними накопичувачами обмеженої місткості.

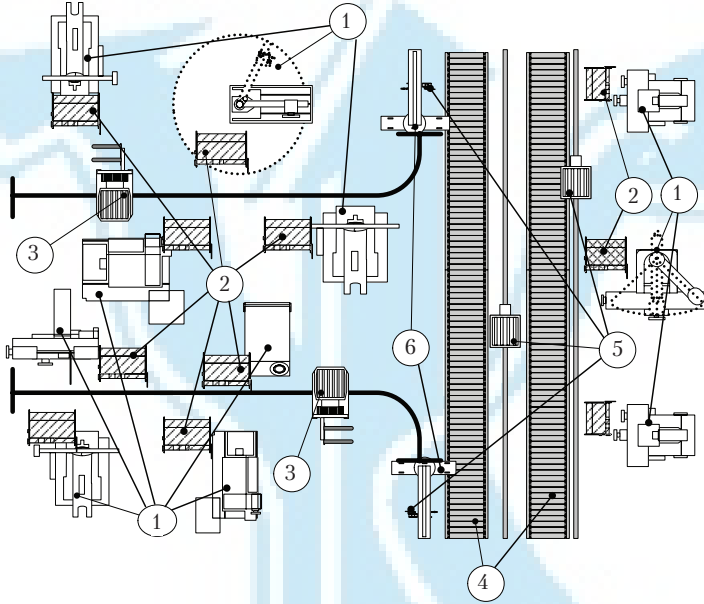


Рис. 12.11. Компоновальна структура ГКІС: 1 – ГВМ; 2 – завантажувально/розвантажувальні позиції ГВМ; 3 – ЕРК; 4 – АС заготовок/деталей/інструменту; 5 – робот-штабелер; 6 – завантажувально/розвантажувальні позиції складу

Для подальшого розгляду особливостей означених вище стратегій РК скористаємося фрагментом ГКІС – гнучким модулем ГВМ (рис. 12.12). До складу ГВМ входять: верстати  $M_1, M_2$ , промислові роботи  $R_1, R_2$ , магазин змінних захватів  $G_1, G_2$ , вхідні  $IB_1, IB_2$  та вихідні  $OB_1, OB_2$  накопичувачі.

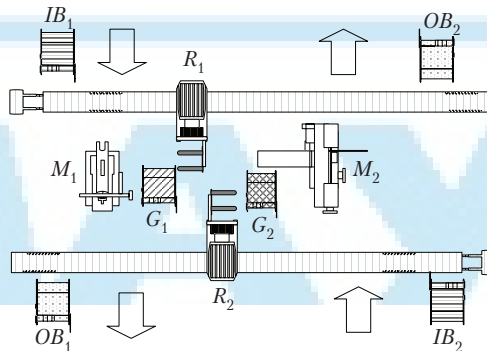


Рис. 12.12. Гнучкий виробничий модуль

При надходженні на  $IB_1(IB_2)$  партії сировини ініціюється цикл механообробки, за яким робот  $R_1(R_2)$ , оснащений захватом  $G_1(G_2)$ , бере партію з нагромаджувача  $IB_1(IB_2)$  і завантажує її на верстат  $M_1(M_2)$  для виконання операцій обробки. Оброблена партія розвантажується з верстата  $M_1(M_2)$  роботом  $R_2(R_1)$  із захватом  $G_2(G_1)$  і встановлюється на вихідний нагромаджувач  $OB_1(OB_2)$ .

Приймаємо потужність (операційний час) кожного з ресурсів за одиницю і подамо процеси так:

$$\begin{aligned} PR_1 &= (O_{1,1}, \{R_1, G_1\}), (O_{1,2}, \{M_1\}), (O_{1,3}, \{R_2, G_2\}); \\ PR_2 &= (O_{2,1}, \{R_2, G_2\}), (O_{2,2}, \{M_2\}), (O_{2,3}, \{R_1, G_1\}), \end{aligned} \quad (12.1)$$

де  $(O_{i,u} \{Res_j | j \in J_{i,u}\})$  –  $u$ -та операція в  $i$ -му виробничому маршруті, завершення якого потребує піднабору  $\{Res_j | j \in J_{i,u}\}$  ресурсів ГВМ; набір ресурсних потужностей такий, що кожен його елемент дорівнює  $1/0$ .

Очевидно, синхронне виконання процесів може призвести до їх взаємного блокування з відтворенням тупикової ситуації з маркуванням  $\mu = (1,1,0,1,1,0,1,1,1,1,1,1)$  у сітковій моделі ГВМ, наведеній на рис. 12.13, а. Більше за те, з аналізу графа досяжності випливає, що або перехід  $t_1$  у  $\mu' = (0,1,0,1,1,0,0,0,1,1,1,1)$ , або перехід  $t_5$  у  $\mu'' = (1,1,0,0,1,0,1,1,1,1,0,0)$  повинен бути заборонений для спрацьовування з метою запобігання тупиковій ситуації у разі маркування  $\mu$ .

Існуючі методи керування тупиковою ситуацією можна класифікувати відповідно до таких підходів:

- виявлення тупика та відновлення системи. Дає змогу виявити тупик, тобто визначити процес і ресурс, які спричиняють блокування, усуває причину;
- заблокованості процесу шляхом перерозподілу ресурсних агрегатів і перевіряє наслідки усунення блокування;
- запобігання тупиковій ситуації. Використовує протоколи, які регламентують запити на ресурси, і це не дає змоги задовольнити необхідні для блокування умови;
- уникнення тупика. Базується на допоміжній апріорній інформації про шляхи використання ресурсів кожним з процесів і на дозволі виконувати відбір заявок на ресурси, який гарантує перехід системи з одного безпечного стану до іншого. Цей підхід забезпечує найвищий рівень використання ресурсів.

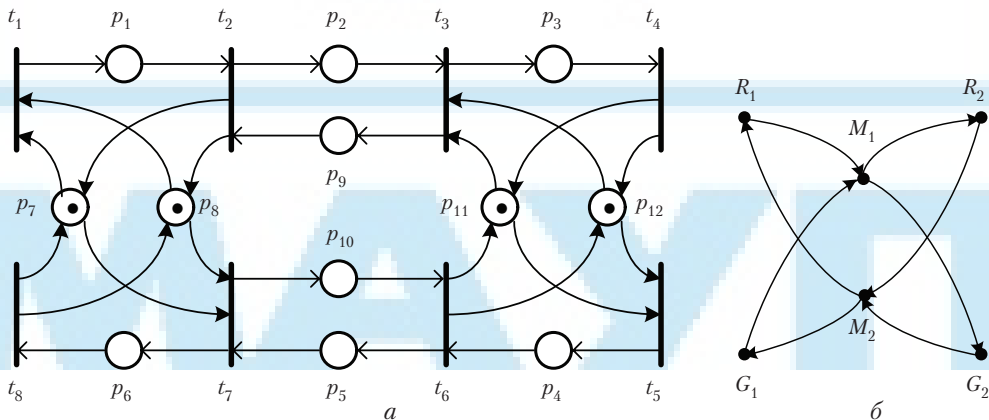


Рис. 12.13. Графічне подання: а – моделі СП; б – графа потреб ресурсів

Розглянемо детальніше можливості уникнення тупикових ситуацій на основі сформованих послідовностей інформації про кожну операцію в системі за умови, що всі активні операції можуть конкурувати одна з одною за ресурси при існуванні відмінності в ресурсних конфліктах між операціями з однаковими виробами і операціями з відмінними виробами.

*Стратегія синхронізуючих зон.* Підхід базується на припущенні, що інформація про виробничий маршрут може розділятися на послідовності виготовлення, або зони, які розглядаються як індивідуальні «паростки» в системі. Така стратегія є прогресивнішою за існуючі алгоритми уникнення тупика для комп'ютерних операційних систем, бо вона базується тільки на ресурсних потребах. Як впливає з ГВМ на рис. 12.13, а всередині такої структури можна спостерігати окремі задовільні стани, дослідження яких потребуватиме прийнятеного часу і реалізація яких завершуватиметься уникненням запусків, що призводять до тупиків.

Для того щоб ці умови проявилися, розглянемо таке означення графа  $H$  потреб ресурсів:

$$H = (\text{Res}, \mathbb{N}), \mathbb{N} \in \text{Res} \times \text{Res}, \quad (12.2)$$

де  $\text{Res}_i, \text{Res}_j \in \mathbb{N}$ , якщо  $\text{Res}_i$  і  $\text{Res}_j$  належать двом послідовним елементам виробничого маршруту  $\text{Res}_i \in \text{crd}_2(\text{crd}_k(PR_n))$  і  $\text{Res}_j \in \text{crd}_2(\text{crd}_{k+1}(PR_n))$ , що справедливі для  $i \neq j; n \in \{1, 2, \dots, v\}, k \in \{1, 2, \dots, I_{n-1}\}$ .

Граф  $H$ , адекватний опису (12.1) процесів, наведено на рис. 12.13, б. З аналізу тупиків формується означення стану блокування процесів.

**Означення 12.5.** Стан  $S$  системи є станом блокування процесів, якщо і тільки якщо у графі  $H$  існує цикл такий, що всі його ресурси розподілено до моменту появи цього стану.

Застосовуючи це означення до системи, де кожний ресурсний запас дорівнює одиниці, можна сформулювати таке правило:

**Правило 12.1.** Стан  $S$  є безпечним, якщо підтримується умова

$$N(S, G) \leq L(G) - 1, \quad (12.3)$$

де  $N(S, G)$  — кількість партій, оброблених за маршрутами  $PR$  до стану  $S$ , який містить розподіл поточних ресурсів;  $L(G)$  — довжина найкоротшого елементарного циклу у графі  $H$ .

Тому для уникнення тупика в системі можуть одночасно оброблятися не більше  $L(G) - 1$  процесів (партій). Недоліком такого розв'язання є низький рівень використання ресурсів системи, що спостерігається, наприклад, у процесах, де існує два і більше суміжних цикли.

Новими, менш обмежувачими, але достатніми для уникнення тупиків умовами враховується інформація про порядок використання ресурсів, і ці умови мають такий вигляд:

$$\begin{aligned} N(S, H_{i,k}) &\leq L(H_{i,k}) - 1 \text{ для } L(H_{i,k}) \neq 1 \text{ або } N(S, H_{i,k}) \leq 1 \text{ для } L(H_{i,k}) = 1, \\ N(S, Z_{i,k}) &\leq L(Z_{i,k+1}), \\ N(S, H_{i,k+1}) + L(H_{i,k}) &< L(H) - 1 \text{ для всіх зон} \\ \{Z_{i,k} \mid i \in \{1, 2, \dots, v\} \&\ k \in \{1, 2, \dots, I\}\}, \end{aligned} \quad (12.4)$$

де  $H_{i,k}$  — підграф, сформований ресурсами відповідних розділених зон  $Z_{i,k}$ , які передують нерозділеній зоні  $Z_{i,k+1}$ ;  $N(S, H_{i,k})$  — кількість партій, оброблених розділеними операціями у стані  $S$ ; розділена (нерозділена) зона — порядок прямування



між серверами ресурсів на віртуальному рівні керуючих комп'ютерів, який відображає реальну картину виробничого процесу. Така процедура повинна виконуватися після кожної модифікації стану системи.

*Стратегія, що базується на протоколах розподілу потужностей (ПРП).* Концепція ПРП щодо синхронізації робочих потоків використовує стратегію, яка потребує визначення буферних потужностей і розподілу правил диспетчеризування стосовно цього набору виробничих маршрутів. Умови призначення буферних потужностей, як і умови, що визначають розподілення правил диспетчеризування (протоколів) і гарантують циклічний сталий стан реалізації процесів (тобто реалізацію процесів без недозавантаження устаткування і безупинковий її характер), відіграють першорядну роль. Протоколи можуть розглядатися як процедури, що координують конкуренцію процесів за доступ до необхідним чином розділених ресурсів. Отриману в результаті систему можна розглядати як таку, що самосинхронізується і яка здатна повернутися до єдиного сталого стану з будь-якого стану, в який вона у вимушеному порядку була переведена в результаті випадкового збурення.

Для ілюстрації наведеного підходу зупинимося на поданому наступними виробничими маршрутами наборі послідовних циклічних процесів, які взаємодіють відповідно до рис. 12.14, б:

$$P_1 = (M_4^1, M_1^1, M_2^1); P_2 = (M_1^2, M_6^2, M_3^2); P_3 = (M_3^3, M_6^3, M_2^3), \quad (12.5)$$

де  $M_k^i$  –  $i$ -й процес розподілу до  $M_k$ -го ресурсу.

Стан  $S_0 = (1, 0, 2, 0, 3, 2'')$ , що містить у собі початковий розподіл процесів, може бути спрямований до стану тупика  $S_T = (1, 3, 2, 0, 2', 0)$ . Проте використання набору локальних правил диспетчеризування

$$PRes_{M_1} = [P_1, P_2]_{M_1}; PRes_{M_2} = [P_1, P_3]_{M_2}; PRes_{M_3} = [P_3, P_2]_{M_3} \quad (12.6)$$

може гарантувати виконання безупинкового стану. Це означає, що існує початковий стан і набір правил типу (12.6) диспетчеризування, які гарантують виконання безупинкового стану. Більше за те, прийнявши розподіл додаткової потужності агрегату за ресурсами  $M_1$  або  $M_2$ , або  $M_3$ , поведінка системи виявляється безупинковою в обох випадках: як для початкового стану розподілу процесів, так і для набору розподілу правил диспетчеризування (так звана *умова збалансованості потоку*).

## 12.4. Розподілене керування із самосинхронізацією потоків ГКІС

**Принципи розподіленого світлофорного керування.** Сучасні виробничі системи передбачають максимальну гнучкість автоматизованих розподілених транспортних систем (АРТС), які у ГВС служать зв'язуючим ланцюгом для позицій обробляючих ресурсів. Такі системи складаються, як правило, з окремих автономних транспортних засобів — *електророботкарів* (ЕРК) з бортовими контролерами і формуванням у реальному часі керування, функціями якого є:

- маршрутизація і розподіл шляху;
- диспетчеризування ЕРК і загальне упорядкування транспортних потоків в умовах невизначеності навколишнього середовища і нештатними ситуаціями, що виникають у системі (перевантаження, відмови тощо), або неповноти та динаміки щодо реального зовнішнього оточення.

У першому випадку визначальною проблемою є розподіл ресурсів та планування завдань їх обслуговування транспортними засобами (про що йшлося в підрозд. 12.3), а в другому — забезпечення останніх можливістю переадресування і каналами



сенсорної чутливості (див. підрозд. 5.6–5.7). За наявності початкових умов, законів, які лежать в основі тих чи інших деструктивних явищ, ці знання можуть бути поданими у вигляді відповідних і в принципі розв'язуваних математичних залежностей, на підставі яких можна передбачити поведінку окремих складових і системи в цілому. Інший метод дослідження ґрунтується на визначенні і використанні емпіричних регуляцій при дослідженні системи імітаційними експериментами (див. у підрозд. 11.2–11.4).

Метою підходу, що розглядається далі, є створення алгебраїчної моделі для визначення структури рівнянь, які дають змогу розрахувати показники функціонування системи без трудомістких і тривалих досліджень.

Слід зазначити, що дискретно-подійні системи і часові моделі у загальному випадку є аналітично важкоотримуваними і комп'ютерно-розрахунково нездійснюваними через численність раптово-випадкових подій на можливих непередбачуваних інтервалах станів, що розв'язується деякою мірою комбінаторно.

Процеси у ГКІС, які підпорядковуються часо-синхронізуючому впливу, добре моделюються *подійно-часовими графами* (ПЧГ) — своєрідними модифікаціями СП, в яких часова затримка асоціюється з кожною позицією, що має тільки по одному вхідному та вихідному переходу; іншою модифікацією СП, створеною власне для моделювання динамічних перебігів подій у системі, є гнучкі СП (детально розглядалися в підрозд. 8.6). Проте для дослідження динамічної поведінки дискретно-подійних систем з часовим навантаженням можна з успіхом застосувати діюд- та макс-алгебри [145; 146]. В останньому випадку дослідження динамічної поведінки класу АРТС-систем може здійснюватися визначенням періодичності системних матричних властивостей і розрахунком їх вагових оцінок. Використовуючи інструментарій макс-алгебри, можна, підраховуючи неконтрольовану поведінку ПЧГ-моделі, знаходити подання деякого нового бажаного режиму функціонування і визначити, чи може це подання бути реалізованим за допомогою набору контрольованих подій [180].

Як об'єкт досліджень приймемо ГКІС з періодичним характером функціонування, яким повторно виконуються ідентичні обмежені набори виробничих програм (дрібносерійне виробництво), причому характеристикою такого функціонування є перепускна спроможність або обернена її величина — цикловий час системи. Розклад обслуговування обробляючих ресурсів транспортною системою, за яким стабілізується функціонування ГКІС, забезпечує одночасно і мінімальний цикловий час. Отже, будемо розглядати АРТС як розподілену систему, яка складається з автономних повторюваних процесів (або транспортних потоків), які утворюють загальний набір і конфігурацію шляхопровідних секцій з визначеними відстанями між їх перетинами. Запропонований підхід базується на використанні принципу «віртуальних асинхронних транспортних світлофорів» (ТС), які на маршрутних перетинах забезпечують механізм синхронізації з локальним обмеженням потоку ЕРК і, таким чином, «світлофорне регулювання» завершується усталеним циклічним режимом системи.

**Означення 12.6.** *Світлофорне розподілене керування* — це керування із самосинхронізуючими властивостями, за якими циклічні процеси індивідуальних транспортних засобів локально синхронізовані таким чином, щоб досягти єдиного усталеного циклічного стану АРТС у цілому.

Проте використання макс-алгебри формалізмів [193] для розв'язання поставленої задачі обмежене типом моделей, які подаються маркованими графами. Це відбувається через інтерпретацію останніми процесів перетину ЕРК з необмеженим

напрямок руху, що не відповідає реаліям функціонування ГКІС, коли, по-перше, цей напрямок може бути на поточний момент зайнятий іншими ЕРК, а по-друге, перетин перехрестя може здійснюватися більше ніж одним ЕРК одночасно. Тому бажано було б удосконалити модель макс-алгебраїчного подання, яка б запобігала наявності взаємовиключних протоколів керування перетинами ЕРК на шляхоперехрестеннях.

Таким чином, *новизна запропонованого підходу полягає в заміні класичного підходу складання транспортних розкладів на реалізацію ідеї розподіленого часового призначення останніх, що може розв'язуватися аналітично.* Іншою важливою перевагою є те, що цей підхід гарантує вільне від недозавантажень і зіткнень керування ЕРК у разі непередбачуваних порушень штатних режимів функціонування.

**Формулювання проблеми.** Розглядається клас АРТС з петлеподібною структурою (рис. 12.15).

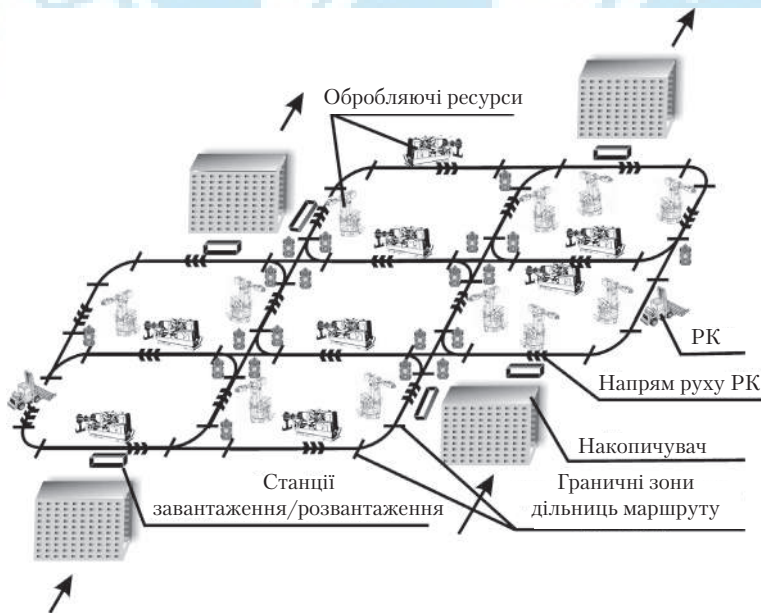


Рис. 12.15. Конфігурація петлеподібною АРТС

У наведеній структурі:

- навігаційні траси дозволяють односпрямований рух груп ЕРК;
- кожний з шляхів перетину перехрестя керується ТС;
- час перетину перехрестя незначний (не враховуючи час очікування на дозволяючий сигнал «Рух»);
- маршрут вважається пройденим при проходженні усіх перехресть на шляху маршруту.

Крім того, визначимо такі маршрутні особливості:

- заявки на будь-який ЕРК є періодичними, з постійними проміжками часу між ними;
- повний транспортний шлях ЕРК складається тільки з проїздодійснюваних примушень, тобто кожне транспортне завдання повинно завершуватися до наступної заявки того ж завдання;



ціонування АРТС, необхідною стає модифікація класичного формалізму макс-алгебри з метою забезпечення математичними засобами точного прогнозування перебігу подій в АРТС, а саме, визначення:

- найменшої часової відстані між двома заданими ЕРК уздовж шляхової секції петлеподібної структури АРТС;
- часу руху ЕРК уздовж заданого маршруту АРТС;
- найкоротшого з усіх можливих маршрутів, які поєднують дві задані позиції у петлеподібній структурі шляхів АРТС.

**Математична модель АРТС.** Перед поданням формалізмів моделювання введемо деякі позначення:

$n$  – кількість перетинів (перехресть) у структурі петлеподібної АРТС, причому вихідну позицію позначимо індексом 0, а пункт призначення через  $n+1$  (на рис. 12.16  $n = 4$ );

$\Delta t_{ij}$  – середній час руху ЕРК між двома безпосередньо зв'язаними  $i$ -м та  $j$ -м пунктами в АРТС, який обчислюється як частка від довжини та середньої швидкості руху ( $i, j = 0, \dots, n+1$ );

$T_i$  – час, необхідний власне ТС для заміни його стану на  $i$ -му перетині ( $i = 1, \dots, n$ );

$r_{ij}$  – тривалість сигналу «Рух» на  $i$ -му перетині з напрямку  $j$ -го пункту системи ( $i = 1, \dots, n; j = 0, \dots, n+1$ );

$(ph)_{ij}$  – фаза ТС на  $i$ -му перетині з урахуванням стосовно часового моменту  $t_0 = 0$ , відрахованого від початку сигналу «Стоп», спостереженого з  $j$ -го пункту структури АРТС ( $i = 1, \dots, n; j = 0, \dots, n+1$ );

$x_0$  – час початку руху, відрахований від  $t_0 = 0$ .

**Сіткова модель петлеподібної секції АРТС.** Для визначення основних сіткових характеристик АРТС (живучість, обмеженість, безтупиковість) будуюмо сітку Петрі, якою інтерпретуються процеси петлеподібної секції на узагальненому рівні абстракції (рис. 12.17).

Опишемо цю СП:

1. Початкова розмітка:  $\mu_0 = (1\ 0\ 0\ 0\ 0\ 0)$ .

2. Матриця інцидентної функції  $F$ :

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

Матриця інцидентної функції  $H$ :

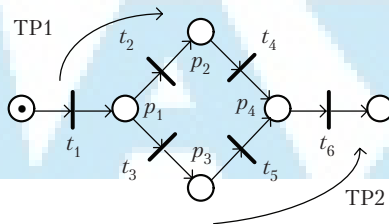
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$


Рис. 12.17. Сітка Петрі, яка інтерпретує транспортні потоки секції АРТС

Пояснення станів СП:

- $\langle\langle \mu_0(100000) \rangle\rangle$  – вихідне маркування;
- $\langle \mu_6(010000) \rangle$  – спрацював перехід  $t_1$ ;
- $! \mu_{6-1}(001000) \rangle$  – спрацював перехід  $t_2$ ;
- $! \mu_{6-1-2}(000100) \rangle$  – спрацював перехід  $t_4$ ;
- $\otimes \mu_{6-1-2-3}(000010) \rangle$  – спрацював перехід  $t_6$ . Тупик;
- $! \mu_{6-5}(000001) \rangle$  – спрацював перехід  $t_3$ ;
- $\rightarrow \mu_{6-5-4}(000100) \rangle$  – спрацював перехід  $t_5$ . Переведення на  $\mu_6$ .

Для цієї СП будемо граф досяжних маркувань (станів) у вигляді дерева, а також проводимо дослідження сітки на обмеженість, живучість, наявність тупиків.

**А.** Досліджуючи СП на *обмеженість*, розв'язуємо систему

$$\begin{cases} AX = 0 \\ X^T M_k = X^T M_0 \end{cases} \Rightarrow \begin{vmatrix} 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \end{vmatrix}; X^T M_k = \begin{vmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{vmatrix}.$$

З результату розв'язання видно, що отриманий  $p$ -ланцюг є повним через те, що присутня інформація про усі позиції, в яких кількість міток дорівнює «1». Отже, ця сітка є обмеженою.

**В.** Досліджуючи далі СП на *живучість*, розглянемо цілочисловий вектор  $Y$  (Т-інваріант), який дорівнює:

$$Y^T = |-1 -1 0 1 1 0|.$$

Оскільки не всі компоненти вектора  $Y$  є нульовими, то послідовність станів сітки не має повернень, що підтверджується графом станів. Таким чином, сітка не є живучою.

**С.** Для дослідження СП на *наявність тупиків* розглянемо різницю матриць:

$$F^T = H - A = \begin{cases} \mu_1 = 0 \\ \mu_2 = 0 \\ \mu_3 = 0 \\ \mu_4 = 0 \\ \mu_5 = 1 \\ \mu_6 = 0. \end{cases}$$

Отримане маркування (000010) є *тупиком*, що відповідає графу досяжних маркувань. Але оскільки ми перевіряємо сітку тільки на досяжність ЕРК від початкової позиції до кінцевої, а маркування (000010) і являє собою досягнення кінцевої позиції, то сітка є обмеженою та безпечною.

**Алгебраїчна модель АРТС.** Скористаємося основними поняттями та визначеннями, які дають змогу використовувати апарат макс-алгебри для системних матричних перетворень [98].

**Означення 12.7.** Мін-плюс-алгебра  $(R_{\min}, \oplus, \otimes)$  є визначеною з огляду на наведене [193]:

- $R_{\min} = R \cup \{\infty\}$ , де  $R$  – набір дійсних чисел;

- $\oplus$  — це мінімум-оператор, який узгоджується зі ступенем  $R_{\min}$  та містить нейтральний елемент  $\varepsilon = \infty$ ;
- $\otimes$  — це звичайний оператор складання з нейтральним елементом  $e = 0$  і поглинання  $\varepsilon$ .

Від звичайної макс-плюс-алгебри наведена модифікація відрізняється визначенням набору  $R_{\max} = R \cup \{-\infty\}$  та оператора складання  $\oplus$  як максимум узгодженості зі ступенем  $R_{\max}$ . Можна стверджувати, що обидві алгебри є ізоморфними. Як наслідок з останнього випливає, що всі доведені властивості макс-плюс-алгебри зберігаються і в мін-плюс-алгебрі.

Щодо процедури підсумовування та добутку матриць у мін-плюс-алгебрі, то вони є подібними до тих же процедур класичної лінійної алгебри, беручи до уваги розбіжності в означенні операторів додавання до добутку.

**Означення 12.8.** Сума  $\oplus$  матриць  $\mathbf{A} = (A_{ij})$  та  $\mathbf{B} = (B_{ij})$  тих самих розмірностей визначається виразом:

$$(\mathbf{A} \oplus \mathbf{B})_{ij} = A_{ij} \oplus B_{ij}. \quad (12.7)$$

**Означення 12.9.** Добуток  $\otimes$   $m \times n$ -матриці  $\mathbf{A} = (A_{ij})$  та  $n \times p$ -матриці  $\mathbf{B} = (B_{ij})$  визначається виразом

$$(\mathbf{A} \otimes \mathbf{B})_{ij} = \bigoplus_{k=1}^n (A_{ik} \otimes B_{kj}). \quad (12.8)$$

Надалі головним чином будемо використовувати матриці з невід'ємними елементами і нульовими діагональними елементами, коли справедливим є таке твердження.

**Твердження 12.1.** Якщо матриця  $\mathbf{A}$  є невід'ємною матрицею з діагональними елементами, які дорівнюють нулю, то:

$$\exists k_0 \forall k \geq k_0; \mathbf{A}^{k+1} = \mathbf{A}^k, \quad (12.9)$$

де  $k$  — оцінювальна функція навантаження дуг  $i-j$  (для  $i \neq j$ ) у графовій інтерпретації матриці;  $k_0$  — деяке бажане значення  $k$ ;  $\mathbf{A}^k = \mathbf{A} \otimes \mathbf{A}^{k-1}$ ;  $\mathbf{A}^0 = \mathbf{E}$ ;  $\mathbf{E} = (E_{ij})$  — одинична матриця в значенні мін-плюс-алгебри, тобто  $E_{ij} = 0$  і  $E_{ij} = \infty$  для  $i \neq j$ .

**Перевірка.** Якщо матриця  $\mathbf{A}$  не є від'ємною, тоді вона асоціюється з графом невід'ємних навантажень дуг. Оцінка елемента  $A_{ij}$  в матриці є мінімальною на дузі відповідно до однореберного шляху не більше ніж між  $i$ -ю та  $j$ -ю вершинами. Отже, за означенням 12.9  $(\mathbf{A}^k)_{ij}$  означає вагу найкоротшого можливого шляху між  $i$ -м та  $j$ -м перетинами для шляхів з не більше ніж з  $k$  дуг. Для обмеженої кількості  $n$  вершин (перетинів) графа кожний шлях, який не містить циклів, може мати не більше як  $n-1$  дуг. Отже, є верхня границя для бажаного  $k_0$ .

З використанням особливостей макс-алгебри поведінка часоподібного графа (тобто системної моделі) підпорядковується співвідношенням:

$$\mathbf{x}(k) = \bigoplus_{i=0}^M (\mathbf{A}(k, k-1) \otimes \mathbf{x}(k-1)); \mathbf{x}(0) = \mathbf{x}_0, \quad (12.10)$$

де  $\mathbf{x}(k)$  — часова послідовність спрацьовування подій у  $k$ -му проміжку часу;  $\mathbf{x}_0$  — послідовність початкових часових спрацьовувань;  $A_{ij}(k, k-1)$  — функція затримування на інтервалі між подіями в  $(k-1)$ -му та  $k$ -му часових інтервалах;  $\otimes, \oplus$  — оператори макс-алгебри [145].

**Моделювання.** Нехай граф конфігурації АРТС подається матрицею  $\mathbf{A} = [A]_{i,j=0}^{n+1}$  розмірністю  $(n+1) \times (n+2)$  з  $A_{ij} = t_{ij}$ , діагональними елементами  $A_{ij} = 0$  та позадіагональними елементами  $A_{ij} = \infty$ , якщо нема безпосереднього зв'язку між  $i$ -м та  $j$ -м перетинами.

Для визначення формули до розрахунку мінімального часу руху з початкової позиції до пункту призначення скористаємося наведеною лемою.

**Лема 12.1.** Якщо  $\tau$  є мінімальний час руху з початкової позиції до пункту призначення, то

$$\exists k_0: \tau = x_{n+1}(k_0) = \left( \mathbf{A}^{k_0} \otimes \mathbf{x}(0) \right)_{n+1}, \quad (12.11)$$

де

$$\mathbf{x}(0) = \begin{bmatrix} x_0 \\ \infty \\ \dots \\ \infty \end{bmatrix} \in (n+2) \times 1 \text{ — вектор з індикацією } i = 0, \dots, n+1. \quad (12.12)$$

*Перевірка.* Справді,  $i$ -й елемент вектора  $\mathbf{x}(1) = \mathbf{A} \otimes \mathbf{x}(0)$  репрезентує найкоротший час прибуття ЕРК на  $i$ -те перехрестя уздовж шляху з щонайбільше одним відгалуженням, причому  $i = 1, \dots, n$  (для пункту призначення  $i = n+1$ ). Якщо ж елемент дорівнює  $\infty$ , це означає, що такого односпрямованого шляху не існує. Тому мінімальний маршрут з  $k$  односпрямованих відрізків може бути поданий вектором

$$\mathbf{x}(k) = \mathbf{A} \otimes \mathbf{x}(k-1) = \mathbf{A}^k \otimes \mathbf{x}(0), \quad (12.13)$$

і аналогічно

$$\mathbf{x}(k+1) = \mathbf{A}(k+1) \otimes \mathbf{x}(0). \quad (12.14)$$

Матриця  $\mathbf{A}$  асоціюється з графом з невід'ємними вагами дуг, які дорівнюють тривалостям часу на подолання ЕРК відстані між перехрестями. Тоді, опираючись на *твердження* 12.1 і на рівняння (12.13), (12.14), наступний вираз є справедливим:

$$\exists k_0 \forall k \geq k_0: \mathbf{x}(k) = \mathbf{x}(k+1), \quad (12.15)$$

і немає жодного сенсу обчислювати стани після  $\mathbf{x}(0)$ . Отже, *теза* (12.11) є доведеною.

При виведенні формули для обчислення часу руху ЕРК до операторів  $\oplus$  та  $\otimes$  додається новий оператор  $\oslash$  [145].

**Означення 12.10.** Оператор  $\oslash$  модуля визначається для деяких  $a, t \in [0, \infty]$ ,  $a \oslash t = b$ , якщо і тільки якщо існує  $k \in \mathbb{N}$  таке, що  $a = t^k \otimes b$ , де  $\mathbb{N}$  — набір натуральних чисел, і наступні властивості задовольняються:

$$\forall a \in R_{\min}: a \oslash \infty = a; \forall a \in R_{\min}: \infty \oslash a = \infty. \quad (12.16)$$

Якщо  $\mathbf{A} = (A_{ij}) \in m \times n$  — матриця, елементи якої є невід'ємними, і  $\mathbf{b} = (b_{ij}) \in m \times 1$  — вектор з невід'ємними елементами, тоді  $\mathbf{B} = \mathbf{A} \oslash \mathbf{b} \in m \times n$  — матрицею, елемент якої дорівнює

$$B_{ij} = A_{ij} \oslash b_i. \quad (12.17)$$

**Приклад 12.1.** Розглянемо приклад, яким ілюструється метод обчислень мінімального часу руху. Нехай секція петлеподібної структури АРТС подається рис. 12.16. Для спрощення обчислень прийемо, що середня швидкість руху ЕРК між перехрестями дорівнює 1. Тоді системна матриця  $\mathbf{A}$  набирає форми

$$\mathbf{A} = \begin{array}{c} \text{Пункт} \\ \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ \begin{bmatrix} 0 & 5 & \infty & \infty & \infty & \infty \\ 5 & 0 & 10 & 15 & \infty & \infty \\ \infty & 10 & 0 & \infty & 15 & \infty \\ \infty & 15 & \infty & 0 & 20 & \infty \\ \infty & \infty & 15 & 20 & 0 & 5 \\ \infty & \infty & \infty & \infty & 5 & 0 \end{bmatrix} \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \end{array} \quad (12.18)$$

Легко побачити, що найтриваліші шляхи без циклів у цій системі мають чотири дуги. Отже,  $\tau = x_3(4) = (A^4 \otimes x(0))_5 = 35$ , де матриця  $A^4$  найкоротших часових з'єднань між пунктами  $k = 0, \dots, 5$  траси має вигляд

$$A^4 = \begin{bmatrix} 0 & 5 & 15 & 20 & 30 & 35 \\ 5 & 0 & 10 & 15 & 25 & 30 \\ 15 & 10 & 0 & 25 & 15 & 20 \\ 20 & 15 & 25 & 0 & 20 & 25 \\ 30 & 25 & 15 & 20 & 0 & 5 \\ 35 & 30 & 20 & 25 & 5 & 0 \end{bmatrix},$$

і вектор початкового стану  $x(0)$  відповідає виразу (12.12) зі стартовим часом  $x_0 = 0$ . Розглядаючи часи руху між характерними пунктами системи, легко побачити, що маршрут  $0 \rightarrow TC1 \rightarrow TC2 \rightarrow TC4 \rightarrow 5$  (див. трасу TP1 на рис. 12.16) забезпечує мінімальний час руху.

Але в наведеному прикладі ЕРК не очікує на перетин перехрестя, що не завжди виконується в реальній АРТС. Тому в наступному підрозділі розглянемо випадок із затримками ЕРК через очікування дозволу на рух від ТС.

**Синтез керуючої структури.** При практичному використанні АРТС постають питання щодо таких призначень ТС-регулювань, які утворюють самосинхронізуюче керування з метою підтримки як високих швидкостей руху ЕРК у пікові періоди виробництва, так і усталеного живлення матеріальними потоками високомістких виробничих процесів без їх переривання.

**Визначення маршрутів ЕРК.** Параметри перехресть було наведено в пункті «Моделювання» і передбачають такі їх форми для ТС:

- $\mathbf{T} = [T_i]_{i=0}^{n+1}$  – вектор періоду ТС – ТС – перемикавання, де  $T_0 = T_{n+1} = \infty$ ;
- $\mathbf{R} = [R_{ij}]_{i,j=0}^{n+1} - (n+2) \times (n+2)$  – матриця з позадіагональними елементами  $R_{ij} = r_{ij}$ , якщо існує безпосередній зв'язок між  $i$ -м та  $j$ -пунктами;  $R_{ij} = 0$  в інших випадках;
- $(\mathbf{Ph}) = [(Ph)_{ij}]_{i,j=0}^{n+1} - (n+2) \times (n+2)$  – матриця з позадіагональними елементами  $(Ph)_{ij} = (ph)_{ij}$ , якщо існує безпосередній зв'язок між  $i$ -м та  $j$ -м пунктами;  $(Ph)_{ij} = 0$  в інших випадках.

У разі петлеподібної структури АРТС, наведеної на рис. 12.16, означені вище параметри зображуються такими матрицями:

$$\begin{array}{c} \text{Період} \quad \text{Тривалість сигналу «Стоп»} \quad \text{ТС-фази від } t_0 = 0 \\ \mathbf{T} = \begin{bmatrix} \infty \\ 8 \\ 20 \\ 16 \\ 12 \\ \infty \end{bmatrix}; \quad \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 6 & 6 & 0 & 0 \\ 0 & 15 & 0 & 0 & 15 & 0 \\ 0 & 12 & 0 & 0 & 12 & 0 \\ 0 & 0 & 9 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad (\mathbf{Ph}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 1 & 3 & 0 & 0 \\ 0 & 2 & 0 & 0 & 7 & 0 \\ 0 & 6 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (12.19) \end{array}$$

У матрицях  $\mathbf{A}$ ,  $\mathbf{R}$ ,  $(\mathbf{Ph})$  та у векторі  $\mathbf{T}$  враховуються тільки такі параметри системи, які стосуються заданого маршруту.

Наступною лемою забезпечується формула, яка дає змогу обчислювати час, необхідний для пересування ЕРК уздовж заданого маршруту.



**Лема 12.2.** Час пересування ЕРК уздовж заданого маршруту задається як

$$\tau = x_{n+1}(n+1), \quad (12.20)$$

де  $x_{n+1}$  — це  $(n+1)$ -та компонента вектора стану, вивідного з рівнянь

$$\mathbf{X}(k) = \mathbf{B}(k-1) \otimes \mathbf{x}(k-1); \quad (12.21)$$

$$\mathbf{B}(k-1) = \mathbf{C} + \mathbf{Z}(k-1); \quad (12.22)$$

$$\mathbf{C} = \mathbf{A} + \mathbf{R}; \quad (12.23)$$

$$\mathbf{Z}(k-1) = -\left[ (\mathbf{A} + (\mathbf{P}\mathbf{h}) + \mathbf{e} \otimes \mathbf{x}^T(k-1)) \oslash \mathbf{T} \right] \oplus \mathbf{R}, \quad (12.24)$$

де  $\mathbf{x}(0)$  задається виразом (12.12);  $\mathbf{e}$  — це  $(n+2) \times 1$  — вектор з усіма елементами, які дорівнюють нулю.

**Приклад 12.2.** Проходження (проїзд) ЕРК через одне перехрестя. Припустимо, що маршрут руху має тільки один перетин. Нехай  $l_{10}$  — відстань між пунктом початку руху та перетином, а  $V_{10}$  — середня швидкість руху уздовж  $l_{10}$ . Аналогічно  $l_{21}$  і  $V_{21}$  — відстань між перетином і пунктом призначення і середня швидкість на цій частині маршруту відповідно. Тоді час руху:

$$t_{i,j-1} = l_{i,j-1} / V_{i,j-1} \text{ для } i = 1, 2 \quad (12.25)$$

Відповідно до припущень, наведених у пункті «Математична модель АРТС», час руху через перехрестя  $\tau_1 = 0$ . Нехай початковий час руху зі стартової позиції буде часом початку відліку:  $x_0 = 0$  (рис. 12.18), а  $T_1$  — період перемикання сигналу ТС на перехресті; крім того, нехай  $t_{10}$  — тривалість часу сигналу «Стоп», коли ЕРК прибуває на перехрестя.

Якщо  $(ph)$  — гнучкий стосовно початку формування сигналу «Стоп» проміжок часу, можна сказати, що стан ТС, коли ЕРК прибуває на перехрестя, характеризується фазою  $(ph)$ . Тоді початковий момент сигналу «Стоп» перебуває у фазі нуль, а початковий момент сигналу «Рух» перебуває у фазі  $r_{10}$ . Якщо точку початку відліку  $t_0 = 0$  встановлено на момент активації сигналу «Стоп», тоді фаза  $(ph)$  на деякий момент часу  $t$  може обчислюватися за формулою

$$(ph) = t \oslash T_1, \quad (12.26)$$

де  $T_1$  — цикловий час ТС.

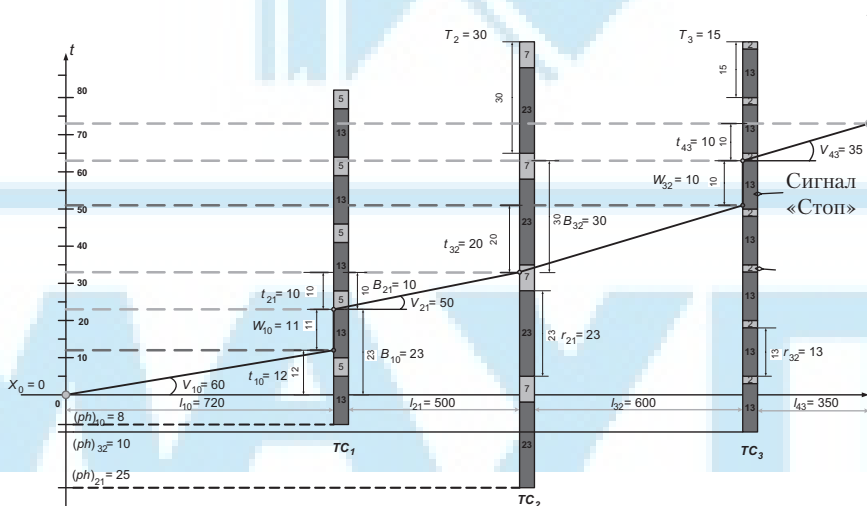


Рис. 12.18. Часова діаграма маршруту з трьома перехрестями

Проте відлікова (відправна) точка з часом  $t_0 = x_0 = 0$  не обов'язково збігається з фазою прибуття ЕРК на перехрестя в момент активації ТС, тобто не обов'язково фаза  $(ph)_{10}$  дорівнюватиме нулю. Отже, у загальному випадку стан ТС визначається фазою  $(ph)$ :

$$(ph) = ((ph)_{10} + (t \oslash T_1)) \oslash T_1 = ((ph)_{10} + t_{10}) \oslash T_1. \quad (12.27)$$

Зокрема, на момент, коли ЕРК після проходження дистанції  $l_{10}$  прибуває на перехрестя, стан ТС визначається таким значенням фази  $(ph)$ :

$$(ph) = ((ph)_{10} + x_0 + t_{10}) \oslash T_1 = ((ph)_{10} + t_{10}) \oslash T_1. \quad (12.28)$$

Якщо  $(ph) \geq r_{10}$ , то ЕРК встигає на перехрестя при положенні ТС «Рух» і має можливість пройти перетин без затримки; в іншому випадку ЕРК має очікувати на дозвіл  $(r_{10} - (ph))$  одиниць часу. З цього випливає, що час очікування ЕРК на перехресті дорівнюватиме

$$W_{10} = r_{10} - ((ph) \oplus r_{10}). \quad (12.29)$$

Отже, час очікування ЕРК на перетині перехрестя задається формулою

$$x_1 = (t_{10} + W_{10}) \oslash x_0 = (c_{10} + z_{10}(x_0)) \oslash x_0, \quad (12.30)$$

де

$$c_{10} = t_{10} + r_{10}; \quad z_{10}(x_0) = -((t_{10} + (ph)_{10} + x_0) \oslash T_1) \oplus r_{10}. \quad (12.31)$$

Тоді загальний час руху ЕРК:

$$\tau = t_{21} + x_1. \quad (12.32)$$

Для системи з одним (тобто  $n = 1$ ) перехрестям, відповідно до прийнятих вище позначень формалізмів моделювання, можна записати:

$$\mathbf{A} = \begin{vmatrix} 0 & \infty & \infty \\ t_{10} & 0 & \infty \\ \infty & t_{21} & \infty \end{vmatrix}; \quad \mathbf{T} = \begin{vmatrix} \infty \\ T_1 \\ \infty \end{vmatrix}; \quad \mathbf{R} = \begin{vmatrix} 0 & 0 & 0 \\ r_{10} & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}; \quad (\mathbf{Ph}) = \begin{vmatrix} 0 & 0 & 0 \\ (ph)_{10} & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}; \quad \mathbf{x}(0) = \begin{vmatrix} x_0 = 0 \\ x_1 = \infty \\ x_2 = \infty \end{vmatrix}. \quad (12.33)$$

Аналогічно до (12.31) визначимо ті ж параметри у матричній формі

$$\left. \begin{aligned} \mathbf{C} &= \mathbf{A} + \mathbf{R}; \\ \mathbf{Z}(0) &= -[(\mathbf{A} + (\mathbf{Ph}) + \mathbf{e} \otimes \mathbf{x}^T(0) \oslash \mathbf{T}) \oplus \mathbf{R}] \end{aligned} \right\} \quad (12.34)$$

де  $\mathbf{e}$  – це  $(n+2) \times 1$  – вектор усіх елементів, які дорівнюють нулю.

Тоді

$$\begin{aligned} \mathbf{x}(1) &= (\mathbf{C} + \mathbf{Z}(0)) \otimes \mathbf{x}(0) = \\ &= \left( \begin{vmatrix} 0 & \infty & \infty \\ t_{10} + r_{10} & 0 & \infty \\ \infty & t_{21} & 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 & 0 \\ (t_{10} + (ph)_{10}) \oslash T_1 \oplus r_{10} & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \right) \otimes \begin{vmatrix} 0 \\ \infty \\ \infty \end{vmatrix} = \end{aligned} \quad (12.35)$$

$$= \begin{vmatrix} 0 \\ t_{10} + r_{10} - [(t_{10} + (ph)_{10}) \oslash T_1] \oplus r_{10} \\ 0 \end{vmatrix}.$$

Обчисливши з (12.34)  $\mathbf{Z}(1)$  відповідно до (12.24), можна визначити

$$\mathbf{x}(2) = (\mathbf{C} + \mathbf{Z}(1)) \otimes \mathbf{x}(1) = \begin{vmatrix} 0 \\ t_{10} + r_{10} - [(t_{10} + (ph)_{10}) \oslash T_1] \oplus r_{10} \\ t_{10} + r_{10} - [(t_{10} + (ph)_{10}) \oslash T_1] \oplus r_{10} + t_{21} \end{vmatrix}. \quad (12.36)$$

Наступні перехрестя не змінюють стану вектора. Відповідні елементи цього вектора являють собою оцінки стартового часу, часу проходження перехрестя і часу надходження ЕРК на пункт призначення. Значення цих оцінок визначаються виразами (12.29) та (12.30), а правило для обчислення наступних станів векторів визначаються згідно з тезами лєми 1.

**Приклад 12.3.** Проходження (проїзд) ЕРК через  $n$  перехресть. Розглянемо більш загальний випадок проїзду ЕРК через  $n$  перехресть, проте методологія розв'язання проблеми не змінюється: це повторювання одноперехресного маршруту  $n$  разів. Час перетину ЕРК  $i$ -го перехрестя стає стартовим часом для маршруту до  $i+1$ -го перехрестя.

Для ілюстрації такої задачі розглянемо випадок, коли задано маршрут через три перехрестя (рис. 12.18) з параметрами:  $l_{10} = 720$ ;  $l_{21} = 500$ ;  $l_{32} = 600$ ;  $l_{43} = 350$ ;  $V_{10} = 60$ ;  $V_{21} = 50$ ;  $V_{32} = 30$ ;  $V_{43} = 35$ ;  $r_{10} = 13$ ;  $r_{21} = 23$ ;  $r_{32} = 13$ ;  $(ph)_{10} = 8$ ;  $(ph)_{21} = 25$ ;  $(ph)_{32} = 10$ ;  $T_1 = 18$ ;  $T_2 = 30$ ;  $T_3 = 15$ . Тоді  $t_{10} = l_{10}/V_{10} = 12$ ;  $t_{21} = l_{21}/V_{21} = 10$ ;  $t_{32} = l_{32}/V_{32} = 20$ ;  $t_{43} = l_{43}/V_{43} = 10$ , і основні параметри у матричній формі подання для цього випадку мають вигляд:

$$\mathbf{T} = \begin{vmatrix} \infty \\ 18 \\ 30 \\ 15 \\ \infty \end{vmatrix}; \mathbf{A} = \begin{vmatrix} 0 & \infty & \infty & \infty \\ 12 & 0 & \infty & \infty \\ \infty & 10 & 0 & \infty \\ \infty & \infty & 20 & 0 \\ \infty & \infty & \infty & 10 \end{vmatrix}; \mathbf{R} = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 13 & 0 & 0 & 0 \\ 0 & 23 & 0 & 0 \\ 0 & 0 & 13 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}; (\mathbf{Ph}) = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix},$$

з чого випливає:

$$\mathbf{C} = \mathbf{A} + \mathbf{R} = \begin{vmatrix} 0 & \infty & \infty & \infty \\ 25 & 0 & \infty & \infty \\ \infty & 33 & 0 & \infty \\ \infty & \infty & 33 & 0 \\ \infty & \infty & \infty & 10 \end{vmatrix}.$$

Наступні ітерації з урахуванням (12.34), (12.21) та (12.22)

$$\mathbf{x}(0) = \begin{vmatrix} 0 \\ \infty \\ \infty \\ \infty \\ \infty \end{vmatrix}; \mathbf{B}(0) = \begin{vmatrix} 0 & \infty & \infty & \infty \\ 23 & 0 & \infty & \infty \\ \infty & 10 & 0 & \infty \\ \infty & \infty & 20 & 0 \\ \infty & \infty & \infty & 10 \end{vmatrix}; \mathbf{x}(1) = \begin{vmatrix} 0 \\ 23 \\ \infty \\ \infty \\ \infty \end{vmatrix}; \mathbf{B}(1) = \begin{vmatrix} 0 & \infty & \infty & \infty \\ 23 & 0 & \infty & \infty \\ \infty & 10 & 0 & \infty \\ \infty & \infty & 20 & 0 \\ \infty & \infty & \infty & 10 \end{vmatrix};$$

$$\mathbf{x}(2) = \begin{vmatrix} 0 \\ 23 \\ 33 \\ \infty \\ \infty \end{vmatrix}; \mathbf{B}(2) = \begin{vmatrix} 0 & \infty & \infty & \infty \\ 23 & 0 & \infty & \infty \\ \infty & 10 & 0 & \infty \\ \infty & \infty & 30 & 0 \\ \infty & \infty & \infty & 10 \end{vmatrix}; \mathbf{x}(3) = \begin{vmatrix} 0 \\ 23 \\ 33 \\ 63 \\ \infty \end{vmatrix}; \mathbf{B}(3) = \begin{vmatrix} 0 & \infty & \infty & \infty \\ 23 & 0 & \infty & \infty \\ \infty & 10 & 0 & \infty \\ \infty & \infty & 20 & 0 \\ \infty & \infty & \infty & 10 \end{vmatrix}; \mathbf{x}(4) = \begin{vmatrix} 0 \\ 23 \\ 33 \\ 63 \\ 73 \end{vmatrix},$$

звідки  $\tau = x_4(4) = 73$ .

**Вибір маршруту з найкоротшим часом руху.** Тепер розглянемо основну проблему — вибір маршруту з найкоротшим часом руху серед багатьох можливих маршрутів між заданими пунктами петлеподібної структури АРТС.

**Теорема 12.1.** Найкоротший з можливих час руху  $\tau$  між заданими ділянками петлеподібної структури АРТС може бути обчислений з урахуванням лемми 1 за таким виразом:

$$\exists k_0 : \tau = x_{n+1}(k_0) = \left( \left( \bigotimes_{j=0}^{k_0-1} \mathbf{B}(j) \right) \otimes \mathbf{x}(0) \right)_{n+1}, \quad (12.37)$$

де  $\mathbf{B}(j)$  та  $\mathbf{x}(0)$  визначаються згідно з виразами (12.22) та (12.12) відповідно.

*Перевірка:* Доведення є аналогічним класичному випадку визначення найкоротшого шляху на графі (*твердження 1*) з урахуванням перевірки невід'ємності матриці  $\mathbf{B}(k)$ .

Але на підставі (12.24) для  $i, j = 0, \dots, n+1$  маємо

$$Z_{ij}(k) \geq -R_{ij}, \quad (12.37)$$

тобто

$$B_{ij}(k) = [A_{ij} + R_{ij} + Z_{ij}(k)] \geq (A_{ij} + R_{ij} - R_{ij}) = A_{ij}, \quad (12.38)$$

і оскільки матриця  $\mathbf{A}$  часових відстаней між перехрестями в структурі АРТС не є від'ємною, то й матриця  $\mathbf{B}(k)$  за будь-яких  $k$  не буває від'ємною.

**Приклад 12.4.** Для ілюстрації застосування виразу (12.37) розглянемо усі перетини в секції структури АРТС, зображеній на рис. 12.16.

Параметри  $R_{ij}$ ,  $(Ph)_{ij}$ ,  $T_i$  задано матрицями (12.19), а початковий стан  $\mathbf{x}(0)$  визначається з (12.12). Відповідними ітераціями є:

$$\begin{aligned} \mathbf{x}(0) = \begin{vmatrix} 0 \\ \infty \\ \infty \\ \infty \\ \infty \\ \infty \end{vmatrix}; \quad \mathbf{B}(0) = \begin{vmatrix} 0 & 5 & \infty & \infty & \infty & \infty \\ 9 & 0 & 10 & 15 & \infty & \infty \\ \infty & 10 & 0 & \infty & 15 & \infty \\ \infty & 15 & \infty & 0 & 20 & \infty \\ \infty & \infty & 15 & 20 & 0 & 5 \\ \infty & \infty & \infty & \infty & 5 & \infty \end{vmatrix}; \quad \mathbf{x}(1) = \begin{vmatrix} 0 \\ 9 \\ \infty \\ \infty \\ \infty \\ \infty \end{vmatrix}; \quad \mathbf{B}(1) = \begin{vmatrix} 0 & 5 & \infty & \infty & \infty & \infty \\ 9 & 0 & 10 & 15 & \infty & \infty \\ \infty & 24 & 0 & \infty & 15 & \infty \\ \infty & 15 & \infty & 0 & 20 & \infty \\ \infty & \infty & 24 & 20 & 0 & 5 \\ \infty & \infty & \infty & \infty & 5 & 0 \end{vmatrix}; \\ \mathbf{x}(2) = \begin{vmatrix} 0 \\ 9 \\ 33 \\ 24 \\ \infty \\ \infty \end{vmatrix}; \quad \mathbf{B}(2) = \begin{vmatrix} 0 & 5 & \infty & \infty & \infty & \infty \\ 9 & 0 & 12 & 19 & \infty & \infty \\ \infty & 24 & 0 & \infty & 15 & \infty \\ \infty & 15 & \infty & 0 & 20 & \infty \\ \infty & \infty & 24 & 20 & 0 & 5 \\ \infty & \infty & \infty & \infty & 5 & 0 \end{vmatrix}; \quad \mathbf{x}(3) = \begin{vmatrix} 0 \\ 9 \\ 33 \\ 24 \\ 44 \\ \infty \end{vmatrix}; \quad \mathbf{B}(3) = \begin{vmatrix} 0 & 5 & \infty & \infty & \infty & \infty \\ 9 & 0 & 12 & 19 & \infty & \infty \\ \infty & 24 & 0 & \infty & 24 & \infty \\ \infty & 15 & \infty & 0 & 20 & \infty \\ \infty & \infty & 24 & 20 & 0 & 5 \\ \infty & \infty & \infty & \infty & 5 & 0 \end{vmatrix}; \end{aligned}$$

$$\mathbf{x}(4) = \begin{vmatrix} 0 \\ 9 \\ 33 \\ 24 \\ 44 \\ 49 \end{vmatrix},$$

з чого випливає, що  $\tau = x_5(4) = 49$ .

При порівнянні двох альтернативних маршрутів ТР1 і ТР2 на рис. 12.16 легко побачити, що з найкоротшим часом руху буде маршрут ТР2 ( $0 \rightarrow \text{ТС}_1 \rightarrow \text{ТС}_3 \rightarrow \text{ТС}_4 \rightarrow 5$ ). На відміну від прикладу 12.1, де  $\tau = x_5(4) = 35$  і де найкоротший за шляхом маршрут  $0 \rightarrow \text{ТС}_1 \rightarrow \text{ТС}_2 \rightarrow \text{ТС}_4 \rightarrow 5$  був і найшвидшим, після введення результатів з ураху-

ванням фаз  $P_{ij}$  перемикання ТС, маршрут через перехрестя із світлофором  $ТС_3$  загальною за часом прибуття ЕРК на позицію 5 структури АРТС стає швидшим, аніж маршрут через перехрестя із світлофором  $ТС_2$ .

Отже, теорема 1 задає найкоротший за загальним часом маршрут руху між початковим і кінцевим пунктами АРТС-структури при керуванні рухом ЕРК за допомогою ТС. Втім, для задоволення вимог ефективного функціонування ГКІС як складної розгалуженої системи значної кількості обробляючих ресурсів необхідно розв'язати задачу синхронізації роботи усіх системних ТС. Виходячи з того, що кожен субнаступний стан системи обчислюється за виразом (12.13) з матрицею  $\mathbf{A}$  із постійних елементів, а форма керування в разі АРТС-структури з ТС-керуваними перехрестями — за виразом (12.21), до того ж системна  $N \times N$  матриця  $\mathbf{B}(k)$  змінюється упродовж кожної ітерації згідно з рівнянням (12.22) з кількістю дій для її обчислення в межах  $mN^2$ , то загальна кількість дій, необхідна для розв'язання такої поліноміально складної проблеми, визначається порядком  $mN^6$ , тому проблему пошуку найкоротшого шляху у графовій АРТС-структурі з ТС-керуваними перехрестями розв'язують за алгоритмом Дейкстри [160], і при цьому порядок не перевищує  $N^3$ , де  $N$  — кількість вершин графа.

**Синхронізація ТС.** Тепер уявімо, що АРТС (рис. 12.15) задано набором транспортних маршрутів для виконання задач обслуговування розгалуженої сітки окремих структур (рис. 12.16), об'єднаних у загальну ГКІС. Задачу поширимо на випадок наявності зустрічних транспортних потоків, що неминуче призведе до накладання додаткових умов на формування кожного з ЕРК-маршрутів і ненульове ТС-регулювання. Іншими словами, цикловий час ЕРК, з одного боку, не може бути меншим за операційний час, а з другого — передбачати ненульовий резерв часу на очікування перемикання ТС, пов'язаного з необхідністю урахування «інтересів» зустрічних потоків ЕРК. Розв'язання цієї задачі впливає з наведеної нижче теорему.

**Теорема 12.2.** *Задано:*

- АРТС петлеподібної структури з  $m$  транспортними ЕРК, які прямують приписаними шляхами;
- цикловий час на подолання ЕРК відповідного шляху з обчислюваним набором ТС-фаз.

*Тоді період  $T$  стану усталеного режиму функціонування АРТС дорівнює найменшій загальній складовій з набору  $\{T_\Psi : \Psi = 1, \dots, m\}$ , де  $T_\Psi$  — часовий цикл  $\Psi$ -го ЕРК.*

*Перевірка:* Цикловий час кожного із шляхів можна обчислити за формулою (12.20), забезпеченою лемою 12.2. При цьому зауважимо, що будь-який цикловий шлях можна подати як такий, що складається з двох окремих маршрутів (обох напрямків руху). Тоді для обчислення часу руху за зворотним (друга частина циклу) напрямом, досить врахувати новий стартовий (початковий) час  $x_0$  і первісну ТС-фазу, одержану з обчислень для прямого (перша частина циклу) напрямку руху. Слід тільки зауважити, що часи руху обчислюються для умови усталеного режиму функціонування АРТС, що гарантується циклічним характером перемикання ТС, а отже, і цикл АРТС має визначатися найменшою загальною кратною компонентою циклових процесів.

Тоді, беручи до уваги здійснюваність приписаних маршрутів, фази перемикання ТС можна визначити за такою структурно-послідовно-пошуковою процедурою:

*Крок 1.* Обчислення циклового часу для кожного окремого маршруту, складеного із сегментів таким чином, щоб початковий і кінцевий вузли (вершини) кожного із сегментів відповідали позиціям, пов'язаним із технологічними позиціями системи (розвантаження/завантаження ЕРК, інші обробляючі ресурси ГКІС, які обслуговуються транспортом).

*Крок 2.* Обчислення часу прибуття ЕРК на перехрестя у кожній окремій частині сегментів маршруту.

*Крок 3.* Обчислення циклового часу усієї системи як найменше загальне кратне циклів складових її процесів. Починаючи з довільно заданого моменту часу, перевіряються (всередині заданого максимального періоду) існуючі ситуації, коли більше ніж один ЕРК займає ту саму дільницю секції або перехрестя.

*Крок 4.* В разі відсутності збігів місцезнахожень в ЕРК за часом, для кожної дільниці уздовж маршруту впроваджується регулювання ТС таким чином, щоб сигнал «Стоп» переводився на сигнал «Рух» на час прибуття ЕРК. Регулювання окремого ТС повинно задовольняти вимогу, за якою цикловий час дільниці є кратним цикловим часом ТС з урахуванням динамічних змін інших потоків як на перехрестях, так і інших перекриттях різних маршрутів.

*Крок 5.* Якщо перекривання у розкладі ЕРК відсутні, то або повторюється повна процедура, починаючи з альтернативного набору можливих маршрутів, або «вилучаються» перекривання крок за кроком. В останньому випадку відбираються дільниці з перекриванням у русі ЕРК і перевіряється можливість встановлення нових часових моментів прибуття ЕРК на відповідні перехрестя при додержанні графіка обслуговування робочої позиції. Якщо це можливо, то далі перевіряється збіжність досягнутого розв'язку з вимогою кратності циклового часу ТС з цикловим часом на дільниці.

Наведена процедура працює на окремому наборі системних параметрів, для яких прагнуть відшукати здійснюваний розклад для цього маршруту. В загальному ж випадку проблема складання АРТС-розкладу є NP-повною.

**Висновки.** 1. Часові призначення ведуть до робастного розподіленого керування матеріальними потоками у петлеподібних АРТС-структурах, що забезпечує реалізацію руху ЕРК в режимі без зіткнень і очікувань.

2. Наведений підхід виключає необхідність трудомістких і довготривалих складань розкладів руху ЕРК і гарантує вільний від пульсацій рух і виконання якісної (без перевантажень) та кількісної (оптимальної за продуктивністю) функцій АРТС.

3. Викладена методика обчислення параметрів АРТС забезпечує можливість передбачення точної поведінки усіх транспортних компонентів як в умовах усталеного режиму функціонування, так і у випадках дії дестабілізуючих зовнішніх порушень.

4. З введенням модуль-оператора  $\emptyset$  формалізм макс-плюс-алгебри дає змогу враховувати пріоритети при поданні процесів, які відбуваються в перехрестях петлеподібних АРТС-структур. А це, у свою чергу, забезпечує керування ЕРК у таких АРТС без зупинок, тобто з постійними (усталеними) циклами пересування. Іншими словами, наведене алгебраїчне розширення являє собою інструмент для моделювання мультиподійної синхронізації між конкуруючими та взаємодіючими процесами.

### Контрольні завдання

1. Дати тлумачення терміна гетерогенності системи та проаналізувати властивості розподілених складних систем з урахуванням гетерогенності їх складових.
2. Обґрунтувати необхідність створення семантично-узгодженого комплексно-інтегрованого середовища, що враховує специфіку функціонування гетерогенних систем.
3. Розкрити зміст гіперпросторового композиційного моделювання ГКІС і проаналізувати процедурну послідовність реалізації гіперпросторових моделей останніх.
4. Розтлумачити концепцію нечіткого метакерування у ГКІС і роль нечіткої керуючої компоненти при цьому.
5. Розкрити особливість побудови і взаємодії компонент гнучкої інтелектуалізованої мультимодульної конфігурації для реалізації нечіткого метакерування.
6. Дати тлумачення метакерування й схарактеризувати особливості модулів з функціями метакерування.
7. Схарактеризувати особливості взаємодії керуючих потоків у модульних структурах НКК і прокоментувати сіткову інтерпретацію модулів НКК.
8. Дати роз'яснення поняття «критичний ресурс» і прокоментувати якісну оцінку критичного завантаження ресурсів системи.
9. Розкрити стратегію розподілення робочих потоків ГКІС з уникненням типових ситуацій та охарактеризувати значення формування: синхронізуючих зон, графа розподілення заявок і протоколів розподілення потужності при цьому.
10. Дати означення і розкрити зміст світлофорного розподіленого керування.
11. Навести приклад структури петлеподібної АРТС і пояснити на ньому принципи світлофорного диспетчеризування матеріальних потоків.
12. Обґрунтувати доцільність застосування модифікованого формалізму макс-алгебри для відображення транспортних операцій в системі із світлофорним регулюванням.
13. Навести сіткову СП-модель петлеподібної секції АРТС і дати математичне тлумачення її функціонування.
14. Розкрити зміст алгебраїчної моделі АРТС на основі модифікованого апарату макс-алгебри – мін-плюс-алгебри.
15. Охарактеризувати процедуру матричного моделювання потоків у петлеподібній АРТС.
16. Розкрити зміст етапів синтезу керуючої структури в основних фазах функціонування транспортних потоків у петлеподібній структурі АРТС із світлофорним регулюванням.
17. Пояснити, як вибирається маршрут транспортних засобів з найкоротшим часом руху та прокоментувати структурно-послідовно-пошукову процедуру визначення фаз переміщення ТС з метою їх синхронізації.
18. Дати узагальнені міркування щодо проблеми складання АРТС-розкладу як NP-задачі.

## ПІСЛЯМОВА

*Техніка дійде до такої довершеності,  
що людина зможе обійтися без себе.*

Станіслав Єжи Лец

*Людство вступило в настільки високу стадію розвитку,  
що про його майбутнє говорять із впевненістю тільки астрологи.*

Автор невідомий

*І тільки той оцінить труд наш,  
кто зрушить цей камінь.*

Ось і закінчилася наша подорож у світ штучного інтелекту. Ми ознайомилися з основними поняттями, методами досліджень, напрямками розвитку та системами реального використання в цій новій галузі науково-технічних розробок і переконалися в тому, що вже сьогодні системи з ШІ є потужним знаряддям, здатним не тільки полегшити розв'язання складних погано структурованих задач і тих задач, які важко формалізуються й виникають у сферах діяльності людини, а й взагалі звільнити людину від виконання деяких видів розумової праці. Завершити цей перший підручник з проблем створення та використання ШІ хотілося б окресленням можливих наслідків їх розв'язання.

Відтепер є реальний шлях до розв'язання найскладніших задач в інтелектуальних сферах діяльності людини — це узагальнення та формалізація знань, якими володіють спеціалісти-експерти певного профілю, перетворення їх у машинну форму подання із залученням ЕОМ для формування обґрунтованих висновків. У подальшому можна сподіватися на процес самонавчання й удосконалення набутих знань самою машиною.

Отже, про ШІ ведуться досить гострі дискусії на сторінках періодичних видань, причому основне питання їх полягає в тому, чи можна створити технічну систему — штучний мозок, який повністю відтворював би функції природного людського мозку з його емоціями, творчістю, інтуїцією тощо. Це питання має технічний, психологічний, філософський і соціальний аспекти і навряд чи можна сподіватися на однозначну відповідь на нього найближчими роками. Але немає жодного сумніву в тому, що проблематика ШІ вже давно перейшла із суто наукових досліджень у практичне русло, де одним з найпріоритетніших напрямів є суміщення можливостей людського мозку із ШІ в одній біотехнічній системі з розподілом функцій між людиною й ЕОМ в ЕС.

Ми вже з'ясували, що більшість ОК можна зарахувати до слабковизначених, які мають ряд неочікуваних для традиційного управління властивостей, таких як унікальність, відсутність формалізованої мети існування та оптимальності, наявність високої динамічності, неповнота опису і, нарешті, індивідуальність (обмеженість) поведінки того, хто приймає рішення щодо корекції ОК в умовах невизначеності. Так з'явилася ідея необхідності автоматизації інтелектуальної діяльності людини, а для її реалізації потрібне виконання принаймні трьох основних умов:

- ідея не повинна суперечити відомим законам науки;
- має визріти соціальне замовлення на її реалізацію;



- повинен бути досягнутий такий рівень технології суспільного виробництва, який забезпечує ефективну реалізацію закладених в ідею технічних принципів.

Якщо перші дві умови в нашому випадку безперечно виконуються, то для оцінки третьої слід трохи відхилитися від обговорюваної теми і торкнутися проблеми ІІІ як такої.

Відповідно до двох напрямів розвитку робіт у галузі ІІІ (комп'ютерного та евристичного моделювання) виділяють дві тези, що визначають кінцеву мету створення інтелектуальних систем: *сильну тезу ІІІ*, яка передбачає моделювання за допомогою комп'ютера інтелекту людини в цілому, і *слабку тезу ІІІ*, що має практичну орієнтацію на розв'язання специфічних задач. Інакше кажучи, ці дві тези немовби визначають програму-максимум і програму-мінімум, між якими і лежить сьогодні межа досліджень ІІІ.

Характерною галуззю, де стан робіт з ІІІ підтверджує викладене, є інтегровані виробничі системи (ІВС). Розвинені, функціонально завершені підсистеми, що є компонентами ІВС різного типу, можна поділити на шість основних груп: І – підсистеми збирання та обробки інформації про зовнішнє середовище і стан ОК; Д – підсистеми дії на об'єкти зовнішнього світу (власне обробні ресурси, пов'язані між собою конкретною технологією, тобто технологічні підсистеми); Н – підсистеми нагромадження і транспортування об'єктів виробництва; У – підсистеми планування, моделювання та управління ІВС; П – підсистеми технологічної підготовки виробництва; К – підсистеми, що забезпечують комунікацію компонентів ІВС з людиною-оператором і між собою в умовах спілкування різних рівнів ієрархії.

Будь-яка конкретна ІВС, яка містить певне поєднання таких підсистем, належить до тої чи іншої групи. Це дає змогу класифікувати ІВС за ступенем інтеграції набору підсистем, які до них входять. Інтеграція першого ступеня зумовлює утворення шести класів – І, Д, Н, У, П, К; інтеграція другого ступеня – 15 класів (наприклад, ДІ, ДН тощо); інтеграція третього ступеня – 19 класів (наприклад, ІДН, НУП, УПК тощо), а інтеграція шостого ступеня – тільки один клас ДІНУПК.

Отже, ІВС – це системи другого – шостого ступенів інтеграції, призначені для автономного функціонування в умовах впливу внутрішніх і зовнішніх факторів. Під *автономним функціонуванням* у такому випадку розуміють здатність (здобуту як наслідок програмування або навчання) самостійного, без втручання людини, виконання певних дій, спрямованих на досягнення поставленої мети. Ефективність роботи ІВС суттєво залежить від обсягу та способу подання в такій системі знань про збудуючі фактори середовища й можливі корективи управління для компенсації наслідків дії цих факторів. Саме такі аспекти інтелектуалізації виробничих систем розглянуто в підрозд. 5.1, 5.2, 6.4, 7.4 – 7.6 книги.

Щодо спеціальностей, на які насамперед орієнтовано цей підручник, то проблемна галузь знань породжується саме ІВС і процесами, які в них відбуваються. Ключовим елементом таких систем є роботи, які й повинні створити умови для безлюдного (автономного) функціонування ІВС. Реалізація цих умов пов'язана з наданням роботу таких двох властивостей, притаманних біологічним організмам:

1. Навчання на основі накопиченого досвіду (тобто інформації про навколишнє середовище) та перебудови (адаптація) закладених або вироблених роботом методів розв'язання задач залежно від предмета вивчення.

2. Самовідтворення як у фізичному змісті, так і в разі накопичення знань і на основі цього проведення повного навчання наступних поколінь роботів.

Щодо *першої властивості*, то вона є дуже схожою до проблеми самопрограмування ЕОМ, оскільки в такому разі йдеться про напрям створення ШІ, коли моделювання інтелектуальної діяльності здійснюється за допомогою обчислювальних машин. Тепер уже відомо, що програму для ЕОМ можна скласти так, щоб вона модифікувалася певним чином при накопиченні інформації про зовнішнє середовище. Наприклад, деякі системи можуть генерувати нові правила розв'язання задач, які виникають, удосконалюючи початковий алгоритм. Фактично при розв'язуванні запрограмованої задачі система може навіть сформулювати досить загальні концепції (це простежується на прикладах, наведених у підрозд. 4.6, 5.1, 5.2, 7.4, 7.6 книги).

Однак теорема неповноти стверджує, що всередині будь-якої формальної системи, яка може досліджувати та «пізнавати» себе якоюсь мірою, існують області потенційного знання, недоступні для системи. Такі твердження справді мають сенс. Виявляється, що фундаментальною характеристикою будь-якого інтелекту є створення значної кількості окремих взаємодіючих рівнів обробки інформації. Вони класифікуються, починаючи з найнижчих рівнів (машинна мова), на яких двійкова інформація через логічні ключі зчитується й записується на тимчасовий носій, і, проходячи більш високі рівні специфікації математичних або символічних операцій, а в подальшому утворення рядків слів і чисел, підпорядкованих правилам граматики та синтаксису, використання символічних і загальних концепцій.

Більшість з того, що робить інтелект, можна зарахувати до переміщення даних і символів угору й униз рівнями ієрархії. При цьому справді найвищий рівень — самосвідомості та обробки символів — не «оцінює» деталей внутрішньої структури програмування на машинному рівні тією мірою, як це може зробити тільки людина, розшифровуючи свій генетичний код. Але це не є суттєвим бар'єром до самопізнання. Все, що при цьому необхідно, — це визначити існування більш низьких рівнів і розробити непрямі (побічні) засоби декодування (розшифрування) знань. Від здатності декодування до здатності перепрограмування є відстань, але не така вже й велика.

Визначення здатності навчання та розв'язання різних задач, непередбачених початковим алгоритмом функціонування, приводить до позитивної відповіді щодо можливості набуття інтелектуальними роботами *другої властивості* — відтворення ними подібних до себе, у тому числі й завдяки перенавчанню наступних поколінь роботів. Проте можна лише стверджувати, що така комбінація — інтелекту та самовідтворення — для роботів є досить віддаленою перспективою. Але вже тепер ясно одне — не треба шкодувати зусиль і коштів на просування до реалізації цієї багатобічючої мети.

### Контрольні завдання

1. Прокоментувати три необхідні та достатні умови, що сприяли виникненню ідеї автоматизації інтелектуальної діяльності людини.
2. Розкрити сильну і слабку тези ШІ з урахуванням об'єктної галузі застосування — ГКІС.
3. Прокоментувати дві основні властивості, які повинні мати інтелектуалізовані РТК.

## СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. *Автоматизированные системы технологической подготовки робототехнического производства* / Л. С. Ямпольский, О. М. Калинин, М. М. Ткач. — К.: Вища шк. Головное изд-во, 1987. — 271 с.
2. *Автоматное управление асинхронными процессами в ЭВМ и дискретных системах* / Под ред. В. И. Варшавского. — М.: Наука, 1986. — 400 с.
3. *Автоматизированное управление технологическими процессами: Учеб. пособие* / Под ред. В. Б. Яковлева. — Л.: Изд-во Ленингр. ун-та, 1988. — 224 с.
4. *Айвазян С. А., Бежаева З. И., Староверов О. Л.* Классификация многомерных наблюдений. — М.: Статистика, 1974. — 239 с.
5. *Алиев Р. А.* Интеллектуальные роботы с нечеткими базами знаний. — М.: Радио и связь, 1994. — 177 с.
6. *Алиев Р. А., Абдикеев Н. М., Шахназаров М. М.* Производственные системы с искусственным интеллектом. — М.: Радио и связь, 1990. — 264 с.
7. *Апериодические автоматы* / Под ред. В. И. Варшавского. — М.: Наука, 1976. — 424 с.
8. *Батков А. М.* Методы оптимизации в статистических задачах управления. — М.: Машиностроение, 1974. — 240 с.
9. *Бауэр В.* Введение в теорию конечных автоматов. — М.: Радио и связь, 1987.
10. *Блажко О. В., Лісовиченко О. І., Пуховський Є. С., Ямпольський Л. С.* Моделивання процесів взаємодії компонентів комп'ютерно-інтегрованих систем із застосуванням апарату процесних алгебр // Адаптивні системи автоматичного управління. — 2003. — № 6 (26). — С. 9–28.
11. *Браверман З. М., Мучник И. Б.* Структурные методы обработки эмпирических данных. — М.: Наука, 1983. — 464 с.
12. *Бусленко Н. П.* Моделирование сложных систем. — М.: Наука, 1978. — 400 с.
13. *Васильев В. И.* Распознающие системы. — К.: Наук. думка, 1983. — 436 с.
14. *Верхаген К., Дейн Р.* и др. Распознавание образов: Состояние и перспективы. — М.: Радио и связь, 1985, — 104 с.
15. *Воробьев Е. И., Попов С. А., Шевелева Г. И.* Механика промышленных роботов: В 3 кн. Кн. 1: Кинематика и динамика. — М.: Высш. шк., 1988. — 304 с.
16. *Вукобратович М., Стокич Д.* Управление манипуляционными роботами: теория и приложения. — М.: Наука, 1985. — 384 с.
17. *Вукобратович М., Стокич Д., Кирчански Н.* Неадаптивное и адаптивное управление манипуляционными роботами: Пер. с англ. — М.: Мир, 1989. — 376 с.
18. *Гавриш А. П., Ямпольский Л. С.* Гибкие робототехнические системы: Учебник. — К.: Вища шк. Головное изд-во, 1989. — 407 с.
19. *Гибкие автоматизированные производственные системы* / Л. С. Ямпольский, О. М. Калинин, М. М. Ткач и др.; Под ред. Л. С. Ямпольского. — К.: Техника, 1985. — 280 с.
20. *Гибкое автоматическое производство* / Под общ. ред. С. А. Майорова, Г. В. Орловского, С. М. Халпиокова. — М.; Л.: Машиностроение, Ленингр. отд-е, 1985. — 454 с.
21. *Глибовець М. М., Олецкий О. В.* Штучний інтелект. — К.: Вид. дім «КМ Академія», 2002. — 366 с.
22. *Глушков В. М.* Синтез цифровых автоматов. — М.: Физмашгиз, 1962.

23. *Гнучкі комп'ютеризовані системи: проектування, моделювання і управління: Підручник / Л. С. Ямпольський, П. П. Мельничук, Б. Б. Самотокін, М. М. Поліщук, М. М. Ткач, К. Б. Остапченко, О. І. Лісовиченко.* — Житомир: ЖДТУ, 2005. — 680 с.
24. *Динамика управления роботами / В. В. Козлов, В. П. Макарычев, А. В. Тимофеев, Е. И. Юревич.* — М.: Наука. 1984. — 336 с.
25. *Дубина Д. А., Сильвестров А. Н, Ямпольский Л. С.* Интеллектуализация моделирования гибких производственных систем // *Адаптивні системи автоматичного управління.* — 2000. — № 3 (23). — С. 41–60.
26. *Дубина Д. А., Ямпольский Л. С., Пуховский Е. С.* Использование модифицированных сетей Петри при моделировании гибких производственных систем // *Автоматизация вироб. процесів.* — 2001. — № 2 (13). — С. 60–67.
27. *Дуда Р., Харт П.* Распознавание образов и анализ сцен. — М.: Мир, 1976. — 386 с.
28. *Дюк В. А.* Компьютерная психодиагностика. — СПб.: Братство, 1994.
29. *Дьяконов В. П., Круглов В. В.* Математические пакеты расширения MATLAB: Специальный справочник. — СПб.: Питер, 2001. — 480 с.
30. *Заде Л.* Понятие о лингвистической переменной и ее применении к принятию приближенных решений. — М.: Мир, 1976. — 167 с.
31. *Зайченко Ю. П.* Исследование операций. — К.: Выща шк., 1988. — 552 с.
32. *Зарипов Ш. Х.* Машинный поиск вариантов при моделировании творческого процесса. — М.: Наука, 1983. — 232 с.
33. *Захаров В. Н., Хорошевский В. Ф.* Программные и аппаратные средства. Кн. 3. — М.: Радио и связь, 1990.
34. *Ивахненко А. Г.* Индуктивный метод самоорганизации моделей сложных систем. — К.: Наук. думка, 1982. — 296 с.
35. *Ивахненко А. Г., Мюллер И. А.* Самоорганизация прогнозирующих моделей. — К.: Техника, 1985. — 221 с.
36. *Ивахненко А. Г., Степанко В. С.* Помехоустойчивость моделирования. — К.: Наук. думка, 1985. — 216 с.
37. *Интегрированная семантически согласованная среда гиперпространственного моделирования гибких компьютеризированных производственных систем / О. И. Лисовиченко, В. И. Костюк, А. А. Лавров, Л. С. Ямпольский // Автоматизация вироб. процесів.* — 2001. — № 2 (13). — С. 86–100.
38. *Искусственный интеллект: В 3 кн. Кн. 1. Системы общения и экспертные системы: Справочник / Под ред. Э. В. Попова.* — М.: Радио и связь, 1990. — 464 с.
39. *Искусственный интеллект: В 3 кн. Кн. 2. Модели и методы: Справочник / Под ред. Д. А. Поспелова.* — М.: Радио и связь, 1990. — 304 с.
40. *Искусственный интеллект: В 3 кн. Кн. 3. Программные и аппаратные средства: Справочник / Под ред. В. Н. Захарова, В. Ф. Хорошевского.* — М.: Радио и связь, 1990. — 368 с.
41. *Использование модифицированных сетей Петри при моделировании гибких производственных систем / Д. А. Дубина, Л. С. Ямпольский, Е. С. Пуховский // Автоматизация вироб. процесів.* — 2001. — № 2 (13). — С. 60–67.
42. *Исследование операций: В 2 т.* — М.: Мир, 1981.
43. *Итоги науки и техники. Сер. Физ. и мат. модели нейронных сетей.* — М.: ВИНТИ, 1990. — Т. 1. — 384 с.
44. *Итоги науки и техники. Сер. Физ. и мат. модели нейронных сетей.* — Т. 2. — 132 с.
45. *Каган Б. М.* Электронные вычислительные машины и системы. — М.: Энергия, 1979. — 528 с.

46. *Калин О. М., Ямпольский С. Л.* Моделирование гибких производственных систем. — К.: Техника, 1991. — 180 с.
47. *Клейнрок Л.* Теория массового обслуживания / Пер. с англ. И. И. Грушко; Под ред. В. И. Нейман. — М.: Машиностроение, 1979. — 432 с.
48. *Колесов Ю. Б., Сениченков Ю. Б.* Визуальное моделирование. — СПб.: Мир и семья, 2001. — 256 с.
49. *Конвей Р. В.* Теория расписаний: Пер. с англ. / Р. В. Конвей, В. Л. Максвелл, Л. В. Миллер. — М.: Наука, 1975. — 359 с.
50. *Кондрашина Е. Ю., Литвинцева Л. В., Поспелов Д. А.* Представление знаний о времени и пространстве в интеллектуальных системах. — М.: Наука, 1989. — 128 с.
51. *Конструктивные* полиномиальные алгоритмы решения индивидуальных задач из класса NP / Под ред. А. А. Павлова. — К.: Тэхника, 1993. — 128 с.
52. *Костюк В. И., Ямпольский Л. С.* Гибкие робототехнические системы: системный подход. — К.: Выща шк., 1988. — 72 с.
53. *Котов В. Е.* Алгебра регулярных сетей // Кибернетика. — 1980. — № 5. — С. 10–18.
54. *Котов В. Е.* Сети Петри. — М.: Наука, 1984. — 160 с.
55. *Кофман А., Дебазей Г.* Сетевые методы планирования и их применение. — М.: Прогресс, 1968. — 168 с.
56. *Круглов В. В., Борисов В. В.* Искусственные нейронные сети. Теория и практика. — М.: Горячая линия; Телеком, 2001. — 382 с.
57. *Кулешов В. С., Лакота Н. А.* Динамика систем управления манипуляторами. — М.: Энергия, 1971. — 304 с.
58. *Куссуль Э. М.* Ассоциативные нейроподобные структуры. — К.: Наук. думка, 1992. — 144 с.
59. *Лавров О. А.* Моделювання гнучких дискретно-подійних систем на основі методів з комбінованою семантикою: Дис. ... д-ра техн. наук. — К.: НТУУ (КПІ), 2000. — 363 с.
60. *Лавров А. А., Лисовиченко О. И., Ямпольский Л. С.* Обучение и двухуровневая нечеткая оптимизация в разработке компьютерно-интегрированных систем // Радиотехніка, інформатика, управління: Вісник ЗДГУ, Запоріжжя, 1999. — С. 114–120.
61. *Лавров А. А., Ямпольский Л. С.* Сетевые модели управляющих потоков в модульных реактивных системах // Проблемы создания новых машин и технологий: Науч. тр. Кременчуг. ГТУ. Вып. 2. — Кременчуг, 1999. — С. 248–252.
62. *Лавров А. А., Ямпольский Л. С., Ямпольский С. Л.* Модуль оперативного планирования ГАУ со встроенной имитационной моделью // Механизация и автоматизация управления. — 1990. № 4. — С. 28–31.
63. *Лазарев В. Г., Пийль Е. И.* Синтез управляющих автоматов. — М.: Энергоатомиздат, 1989. — 328 с.
64. *Лебедев А. Н.* Моделирование в научно-технических исследованиях. — М.: Радио и связь, 1989. — 223 с.
65. *Лисовиченко О. І., Ямпольський Л. С.* Семантично-узгоджене середовище гіперпросторових моделей складальних комп'ютерно-інтегрованих систем // Адаптивні системи автоматичного управління: Міжвід. наук.-техн. зб. — Дніпропетровськ: Системні технології, 2000. — Вип. 3 (23). — С. 47–56.
66. *Максимай И. В.* Имитационное моделирование на ЭВМ. — М.: Радио и связь, 1988. — 232 с.

67. Меликов А. З., Пономаренко Л. А. Оптимизация цифровой сети интегрального обслуживания с конечным числом пользователей и блокировками // Автоматика и телемеханика. — 1992. — № 6. — С. 34–38.
68. Меликов А. З., Пономаренко Л. А., Рюмиш П. А. Математические модели многопоточковых систем обслуживания. — К.: Техника, 1991. — 265 с.
69. Методы параллельного микропрограммирования / Под общ. ред. О. Л. Бандман. — Новосибирск: Наука, Сиб. отд. АН СССР, 1981. — 180 с.
70. Минский М. Вычисления и автоматы. — М.: Мир, 1971. — 431 с.
71. Минский М. Фреймы для представления знаний. — М.: Энергия, 1979. — 151 с.
72. Многоуровневые модели сборочных роботизированных производств / Л. С. Ямпольский, А. А. Лавров, Ю. В. Самохатко, В. Л. Семенченко // Новые технологии в машиностроении. — Харьков: ХАИ, 1992. — С. 71–75.
73. Моделювання розподіленого управління перепускними спроможностями циклічних виробничих систем / З. А. Банашак, О. І. Лісовиченко, Д. І. Гергі, Л. С. Ямпольський // Труды филиала МГТУ им. Н. Э. Баумана в г. Калуге. Спецвыпуск: Материалы междунар. науч.-техн. конф. «Приборостроение-2001». — Калуга, 2001. — С. 158–163.
74. Модифікація апарату сіток Петрі і моделювання складних комп'ютерно-інтегрованих систем з ієрархічною семантикою подання процесів / П. В. Кузьмін, О. А. Лавров, О. І. Лісовиченко, К. Б. Остапченко, Л. С. Ямпольський // Вісник ЖТІ. — 1998. — Вип. 8. — С. 80–92.
75. Нейлор К. Как построить свою экспертную систему: Пер. с англ. — М.: Энергоатомиздат, 1991. — 286 с.
76. Нейронні мережі в системах автоматизації / В. І. Архангельський, І. М. Богаєнко, Г. Г. Грабовський, М. О. Рюмиш. — К.: Техніка, 1999. — 364 с.
77. Нечеткие множества в моделях управления и искусственного интеллекта / Под ред. Д. А. Поспелова. — М.: Наука, 1986. — 311 с.
78. Нильсон Н. Дж. Принципы искусственного интеллекта. — М.: Радио и связь, 1985. — 373 с.
79. Нові підходи до моделювання і управління в гнучких комп'ютеризованих системах / П. В. Кузьмін, О. А. Лавров, К. Б. Остапченко, З. Банашак, Л. С. Ямпольський // Адаптивні системи автоматичного управління: Регіональний міжвузівський зб. наук. пр. — Вип. 7 (27) (до 100-річчя КПП). — Дніпропетровськ, 1998. — С. 47–63.
80. Оре О. Теория графов. — М.: Наука, 1980. — 336 с.
81. Петрушин В. А. Экспертно-обучающие системы. — К.: Наук. думка, 1992. — 196 с.
82. Петухов С. И., Новиков В. А., Коваленко И. Н. Теория массового обслуживания. — М.: Сов. радио. 1969. — 340 с.
83. Питерсон Дж. Теория сетей Петри и моделирование систем: Пер. с англ. — М.: Мир, 1984. — 264 с.
84. Пол Р. Моделирование, планирование траекторий и управление движением робота-манипулятора. — М.: Наука, 1976. — 104 с.
85. Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. Математическая теория оптимальных процессов. — М.: Наука, 1983. — 392 с.
86. Попов Е. П., Верещагин А. Ф., Зенкевич С. Л. Манипуляционные роботы: динамика и алгоритмы. — М.: Наука, 1978. — 400 с.
87. Попов Э. В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. — М.: Наука, 1987. — 288 с.

88. Попов Э. В., Фридман Г. Ф. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. — М.: Наука, 1976. — 183 с.
89. Поспелов Г. С. Искусственный интеллект — основа новой информационной технологии. — М.: Наука, 1988. — 280 с.
90. Поспелов Д. А. Модели и методы. Кн. 2. — М.: Радио и связь, 1990.
91. Построение экспертных систем: Пер. с англ. / Под ред. Ф. Хейеса-Рота, Д. Уотермана, Д. Лената. — М.: Мир, 1987. — 442 с.
92. Представление и использование знаний: Пер. с яп. / Под ред. Х. Чжо, М. Исидзуки. — М.: Мир, 1989. — 220 с.
93. Проблема автоматизации моделирования и управления в гибких гетерогенных распределенных сборочных системах / Л. С. Ямпольский, А. А. Лавров, О. И. Лисовиченко, А. Л. Сигал // Міжвід. наук.-техн. зб. «Адаптивні системи автоматичного управління». — Дніпропетровськ: ДНВП «Системні технології», 1999. — Вип. 2(22). — С. 53–73.
94. Промышленная робототехника / Л. С. Ямпольский, В. А. Яхимович, Е. Г. Вайсман и др.: Под ред. Л. С. Ямпольского. — К.: Тэхника, 1984. — 264 с.
95. Промышленная робототехника / Под ред. Я. А. Шифрина. — М.: Машиностроение, 1982. — 415 с.
96. Промышленные роботы: Конструирование, управление, эксплуатация / В. И. Костюк, А. П. Гавриш, Л. С. Ямпольский, А. Г. Карлов. — К.: Выща шк., 1985. — 264 с.
97. Проценко В. С., Чаленко П. И., Сорока Р. А. Техника программирования. — К.: Наук. думка, 1990. — 183 с.
98. Реалізація концепції розподіленого керування з самосинхронізацією потоків транспортних засобів ГВС / З. А. Банашак, О. І. Лисовиченко, Г. М. Ткач, Л. С. Ямпольський // Міжвід. наук.-техн. зб. «Адаптивні системи автоматичного управління». — Дніпропетровськ: ДНВП «Системні технології», 2001. — Вип. 4(24). — С. 88–108.
99. Робототехніка: Підручник / В. І. Костюк, Г. О. Спиноу, Л. С. Ямпольський, М. М. Ткач.— К.: Вища шк., 1994. — 447 с.
100. Робототехника и гибкие автоматизированные производства: В 9 кн. Кн. 1. И. М. Макаров. Системные принципы создания гибких автоматизированных производств: Учеб. пособие для вузов. — М.: Высш. шк., 1986. — 175 с.
101. Робототехника и гибкие автоматизированные производства: В 9 кн. Кн. 5. Моделирование робототехнических систем и гибких автоматизированных производств / С. В. Пантюшкин, В. М. Назаретов, О. А. Тягунов и др.; Под ред. И. М. Макарова. — М.: Высш. шк., 1986. — 175 с.
102. Робототехника и гибкие автоматизированные производства: В 9 кн. Кн. 6. Техническая имитация интеллекта / В. М. Назаретов, Д. П. Ким; Под ред. И. М. Макарова. — М.: Высш. шк., 1988. — 144 с.
103. Робототехника и гибкие автоматизированные производства. В 9 кн. Кн. 7. Гибкие автоматизированные производства в отраслях промышленности: Учеб. пособие для вузов / Под ред. И. М. Макарова. — М.: Высш. шк., 1986. — 176 с.
104. Розенблум Л. Я. Сети Петри // Изв. АН СССР: Техническая кибернетика. — 1983. — № 3. — С. 12–40.
105. Руденко О. Г., Бодянский Е. В. Основы теории искусственных нейронных сетей. — Харьков: ТЕЛЕТЕХ, 2002. — 317 с.
106. Саати Г. А. Элементы теории массового обслуживания и ее приложения. — М.: Советское радио, 1971. — 520 с.

107. САПР систем логического управления / Под ред. В. А. Горбатова. — М.: Энергоатомиздат, 1988. — 232 с.
108. Семенченко В. Л. Самонагружаемые сетевые модели робототехнических систем сборочного производства: Дис. ... канд. техн. наук. — К.: Киевский политехнический институт, 1993. — 155 с.
109. Семенченко В. Л., Ямпольский Л. С. Вероятностное планирование эксперимента в самонагружающихся сетевых моделях РТС // Вестн. Киев. политехнического ин-та. Техн. кибернетика. — 1993. — Вып. 17. — С. 12–14.
110. Сирл С., Госман У. Матричная алгебра в экономике. — М.: Статистика, 1974. — 374 с.
111. Скурихин В. И., Мальшев Н. Г. Автоматный подход к конструированию нормативной алгоритмической модели производственной организации // Управляющие системы и машины. — 1974. — № 2. — С. 1–5.
112. Скурихин В. И., Шифрин В. Б., Дубровский В. В. Математическое моделирование. — К.: Тэхника, 1983. — 270 с.
113. Слепцов А. И., Юрасов А. А. Автоматизация проектирования управляющих систем гибких автоматизированных производств / Под ред. Б. Н. Малиновского. — К.: Тэхника, 1986. — 110 с.
114. Словарь по кибернетике / Под ред. В. С. Михалевича. — К.: УСЭ, 1989. — 751 с.
115. Соболев И. М., Статников Р. В. ЛП-поиск и задачи оптимального конструирования // Пробл. случайного поиска. — Рига: Зинатне, 1972. — № 1. — С. 117–135.
116. Таль А. А., Юдицкий С. А. Иерархия и параллелизм в сетях Петри 1, 2 // Автоматика и телемеханика. — 1982. — № 7. — С. 113–122.
117. Таль А. А., Юдицкий С. А. Иерархия и параллелизм в сетях Петри 1, 2 // Автоматика и телемеханика. — 1982. — № 9. — С. 83–88.
118. Теория расписаний и вычислительные машины / Под ред. Э. Г. Коффмана. — М.: Наука, 1984. — 336 с.
119. Тимофеев А. В. Роботы и искусственный интеллект. — М.: Наука, 1978. — 192 с.
120. Толковый словарь по искусственному интеллекту / Авт.-сост.: А. Н. Аверкин, М. Г. Гаазе-Рапопорт, Д. А. Поспелов. — М.: Радио и связь, 1992. — 256 с.
121. Томашевський В. М. Моделювання систем: Підручник. — К.: Вид. група ВНУ, 2005. — 352 с.
122. Український радянський енциклопедичний словник: У 3 т. / Редкол.: А. В. Кудрицький (відп. ред.) та ін. — 2-ге вид. — К.: Головн. ред. УРЕ, 1986. — Т. 1. — 752 с.
123. Управление ГПС: Модели и алгоритмы / Под ред. С. В. Емельянова. — М.: Машиностроение, 1987. — 368 с.
124. Управление гибким производством и операциями / Э. С. Минаев, Н. Г. Агеева, А. Аббата Дага. — М.: ИНФРА-М, 2000. — 256 с.
125. Управление дискретными процессами в ГПС / Л. С. Ямпольский, З. Банашак, К. Хасегава и др. — К.: Тэхника; Вроцлав: Вроцлав. Техн. ун-т; Токио: Токосё, 1992. — 251 с.
126. Управляющие системы промышленных роботов / Под ред. И. М. Макарова, В. А. Чиганова. — М.: Машиностроение, 1984. — 288 с.
127. Харари Ф. Теория графов. — М.: Мир, 1973. — 362 с.
128. Хоар Ч. Э. Р. Взаимодействующие последовательные процессы / Пер. с англ. А. А. Бульонковой; Под ред. А. П. Ершова. — М.: Мир, 1989. — 264 с.
129. Цаленко М. Ш. Моделирование семантики в базах данных. — М.: Наука, 1989. — 288 с.



130. Шостак В. Ф. Модели и методы управления сложными технологическими комплексами в нештатных (экстремальных) режимах работы в АСУТП // Автоматика и телемеханика. — 1994. — № 10. — С. 158–164.
131. *Экономические* вопросы подготовительной стадии производства новых орудий труда / С. М. Ямпольский, А. М. Майданович, С. В. Козаченко и др. — К.: Наук. думка, 1978. — 216 с.
132. *Экспертные* системы: Принцип работы и примеры: Пер. с яп. / Под ред. Р. Форсайта. — М.: Радио и связь, 1987. — 185 с.
133. *Элти Дж., Кунмбс М.* Экспертные системы: Концепция и примеры. — М.: Финансы и статистика, 1987. — 190 с.
134. Юдицкий С. А. Конвейерные дискретные процессы и сети Петри // Автоматика и телемеханика. — 1983. — № 6. — С. 141–147.
135. Юдицкий С. А., Магерзгум В. З. Логическое управление дискретными процессами: модели, анализ, синтез. — М.: Машиностроение, 1987. — 176 с.
136. Юревич Е. И., Аветиков Б. Г. Устройство промышленных роботов. — М.: Машиностроение, 1980.
137. Юценко А. С. Принципы интерактивного управления роботами // Робототехника: новый этап развития (Кибернетика — неограниченные возможности и возможные ограничения). — М.: Наука, 1993. — С. 129–139.
138. Ямпольский Л. С. Проблема автоматизации моделирования и управления в сборочном производстве // Робототехника: новый этап развития (Кибернетика — неограниченные возможности и возможные ограничения). — М.: Наука, 1993. — С. 60–73.
139. Ямпольский Л. С., Банашак З. Автоматизация проектирования и управления в гибком производстве. — К.: Тэхника; Варшава: Науч.-техн. изд-во, 1989. — 214 с.
140. Ямпольський Л. С., Лавров О. А. Штучний інтелект у плануванні та управлінні виробництвом: Підручник. — К.: Вища шк., 1995. — 255 с.
141. Ямпольский Л. С., Полищук М. Н. Оптимизация технологических процессов в гибких производственных системах. — К.: Тэхника, 1988. — 175 с.
142. Ямпольський Л. С., Поліщук М. М., Ткач М. М. Елементи робототехнічних пристроїв і модулі ГВС / За ред. Л. С. Ямпольського. — К.: Вища шк., 1992. — 431 с.
143. Aarts F., Korst J. Simulates Annealing and Boltzmann Machines. — Chichester, 1990.
144. Akian M., Vapat R. B., and Gaubert S. Asymptotics of the Perron eigenvalue and eigenvector using max-algebra. — 1998.
145. A max-algebra approach to the robust distributed control of repetitive AGV system / M. B. Zaremba, A. Obuchowicz, Z. A. Banaszak, and K. J. Jedrzejek // Production Research. — 1997. — Vol. 35. — №10. — P. 2667–2687.
146. Synchronization and linearity: An Algebra for Discrete Event Systems / F. Bacceli, G. Cohen, G. J. Olsder, and J. P. Quadrat. — Chichester, UK: Wiley, 1992. — 128 p.
147. Baeten J. C. M., Weijland W. P. Process algebra. — Cambridge Univ. Press, 1990. — 248 p.
148. Barthelot G., Roucairol G. Reduction of Petri nets // Lecture Notes in Computer Science. — Berlin: Springer-Verlag, 1980. — Vol. 45. — P. 277–290.
149. Butler M. J. A Practical Approach to Combining CSP and B. Declarative Systems and Software Engineering Group: Technical Report DSSE-TR-99-2, February, 1999.
150. Best E. Semantics of sequential and parallel programs. — Prentice Hall, 1996.
151. Best E., Fernandez C. Nonsequential processes: A Petry net view. — Berlin: Springer-Verlag, 1988. — 112 p.

152. *Brinksma E. A., D'Argenio P., Hermanns H., and Katoen J. P.* Stochastic Process Algebras: linking process descriptions with performance // IFIP WG10.4 Stenungsund, 2001.
153. *Brookes S. D.* On the relationship of CCS and CSP in Lecture Notes in Computer Science 4154: Automata, Languages, and Programming. The 10-th Colloquium. — Springer-Verlag, 1983. — P. 83–96.
154. *Broomhead D. S., Low D.* Multivariable Funkhion Interpolation and Adaptive Networks // Complex System. — 1988. — 2. — P. 321–355.
155. *Burns A.* Programming in Occam 2. — Addison-Wesley, 1988.
156. *Cardoso J., Valette R., Dubois D.* Fuzzy Petry net: An overview // Proc. 13<sup>th</sup> IFAC World congress. — Vol.J. — San Francisco (USA), 1996. — P. 443–448.
157. *Cohen G., Dubois D., Quadrat J. P., Viot M.* A linear system theoretic view of discrete event processes and its use for performance evaluation in manufacturing // IEEE Transaction on automatic control. — 1985. — Vol. 30. — P. 210–220.
158. *Craigen D., Gerhart S., and Ralston T.* Formal Methods Reality Check: Industrial Usage // IEEE Transactions on Software Engin. — 1995. — Vol. 21. — № 2. — P. 90–98.
159. *Diaz M.* Modelling and analysis of communication and cooperation protocols using Petry nets basd models. Protocol specific. testing and verific // Proc. IFIP WG 0.1.2. Intern. Workshop. — Amsterdam, 1998. — Vol. 2. — P. 456–510.
160. *Dijkstra E. W.* A note on two problems in connections in graphs // Numeriche Matematik. — 1959. — №1. — P. 269–271.
161. *Flater D.* Specification of Integrated Manufacturing Systems. — NISTIR 6484, 2000.
162. *Fukushima K.* Neocognitron: Self-Organizing Neural Network Model for Mechanism of pattern Recognition Uneffected by shift in Position // Biological Cubernetics. — 36. — 1980. — P. 193–202.
163. *Harel D.* Statecharts: a visual formalism for complex systems // Sci. Comput. Program. — 1987. — Vol. 8. — P. 251–321.
164. *Hebb D. O.* The Organisation of Behavior // A Neuro psychological Theory. — New York: Wiley, 1949. — 430 p.
165. *Hinton G. E., Rumelhart D. E., Williams R. J.* Learning Representations by Backpropagation Errors // Nature 323. — 1986. — P. 533–536
166. *Hoare C. A. R.* Communicating sequential processes. — Prentice Hall, 1985. — 256 p.
167. *Hollerbach J.* A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity // IEEE Trans. Syst. Man Cybern. — 1986. — Vol. SMC10. — P. 730–736.
168. *Hopcroft J. E., Ullman J. D.* Introduction to automata theory, languages, and computation. — Addison-Wesley, 1979.
169. *Hopfield J. J.* Natural Networks and phisical systems with emergdent collective computational abilities // Proc. Of the Nat. Academy of Sciences, 79. — Washington, 1982.
170. *Huber P., Jensen K., Shapiro R.* Hierarchies in Coloured Petri Nets // Lecture Notes of Computer Science. — Berlin: Springer-Verlag, 1990.
171. *Hybrid Automata: An algorithmic approach to the specification and verification of hybrid systems / R. Alur, Courcoubetis C., Henzinger T. A., Ho P.-H. // Hybrid systems / Grossman R. L., Nerode A., Rischel H., Ravn A. P. (Eds.) — Springer Verlag, 1993. — LNCS. — Vol. 736. — P. 209–229.*
172. *Intelligent Dynamic Models of Computer-Integrated Manufacturing Systems / Yampolsky L. S., Lavrov A. A., Semenchenko V. L., Kuzmin P. V // Proceedings of the 1993 European Simulation Sumposium. — Delft, 1993. — P.161–163.*

173. *Jensen K.* Coloured Petri Nets. A way to Describe and Analyse Real World Systems-Without Drowning in Unnecessary Details//Proceedings of the 5-th International Conference on Systems Engineering. – Dayton, 1987. – P. 395–401.
174. *Jensen K.* Coloured Petri Nets / Petri Nets: Central Models and Their Properties, Advances in Petri Nets // Lecture Notes in Computer Science. – Berlin: Springer-Verlag, 1987. – P. 248–299.
175. *Kandel A.* Fuzzy switching and automata: theory and applications. – N.Y.: Russak, 1979.
176. *Kinnebrock W.* Neuronale Netze // R. Oldenbourg Verlag München Wien 2. Verbesserte Auflage, 1944. – 176 s.
177. *Kohonen T.* Self-organization and Associative Memory // Series in Information Sciences. – Berlin: Springer Verlag. – 1984. – Vol. 8. – 2nd. ed. – B.: Springer, 1988.
178. *Krogh B. H., Screenivas R. S.* Essentially Decision Free Petri Nets for Reak-Time Resource Allocation // The Proceedings of the IEEE International Conference on Robotics and Automation. IEE Computer Society Press. – Washington: DC, 1987. – P. 1005–1010
179. *Lakos C. A., Keen C. D.* Simulation with Object-Oriented Petri Nets // The Proceedings of 6`th Australian Software Engineering Conference. Sydney, Australian. – Berlin: Springer-Verlag, 1991. – P. 203–216.
180. *Lavrov Alexander A., Yampolsky Leonid S.* Configuration Synthesis and Refinement: an Approach to Simulation of Heterogeneous Distributed Systems // УСИМ. – 2000. – № 3. – С. 11–17.
181. *Lemmer K., Schnieder E.* Modeling and Control of Complex Logistic Systems for Manufacturing // Application and Theory of Petri Nets. K. Jensen (ED). – Berlin: Springer-Verlag, 1992. – P. 373–378.
182. *Low D.* Adaptive Radial Basis Function Nonlinearities and the Problem of Generalization / Proc. of IEE Int. Conf. of Artificial Neural Networks. – London, UK. – 1989. – P. 171–175.
183. *Milner R.* Lecture Notes in Computer Science 492: A Calculus of Communicating Systems. – Berlin: Springer-Verlag, 1980.
184. *Minsky M. L., Papert S. A.* Perseptrans // Cambridge (Mass): MIT Press, 1969. – 321 p.
185. *Minsky M. L., Papert S. A.* Perseptrans. An Introduction to Computational Geometry. Expandel Edition // Cambridge (Mass): MIT Press, 1988.
186. *Murray J. J.* Computational Robot Dynamics. PhD thesis, Carnegie Mellon University, 1984.
187. *Ramamoorthy C. V., Ho G. S.* Performance evaluation of asinchronous concurrent systems using Petry nets. // IEEE Trans. on Software Engineering. – 1980. – Vol. SE-6. – № 5. – P. 440–449
188. *Ramadge P. J., Wonham W. M.* Supervisory control of a class of discrete event processes // SIAM J. Control and optimization. – 1987. – Vol. 25. – No. 1. – P. 206–230.
189. *Ramadge P. J., Wonham W. M.* The control of descrete event systems // Proc. IEEE: Special issue on discrete event dynamic systems. – 1989. – Vol. 77. – P. 81–98.
190. *Raynel M. and Helary J.* – M. Synchronization and Control of Distributed Systems and Programs. – John Wiley & Sons, 1990.
191. *Reising W.* Petri nets: an introduction. – Berlin: Springer-Verlag, 1985. – 161 p.
192. *Reising W.* Petrinetze-Eine Einführung, überarbeitete und erweiterte Auflage. – Berlin: Springer, 1986. – 218 p.

193. *Synchronization and linearity: An Algebra for Discrete Event Systems* / Bacceli F., Cohen G., Olsder G. J., and Quarat J. P. – Chichester, UK: Wiley, 1992. – 128 p.
194. *The theory of timed automata* / R.Alur, Courcoubetis C., Henzinger T. A., Ho P.-H. // *Theoretical computer science.* – 1994. – Vol. 126. – P. 183–235.
195. *Thomas J. Scriber. Simulation using GPSS.* – New York: John Wiley, 1974. – 275 p.
196. *Valavanis K. P., Saridis G. N. Probabilistic modeling of intelligent robotic system* // *IEEE Trans. Rob. and Autom.* – 1991. – 7, N 1. – P. 164–171.
197. *Valette R. Analysis of Petry nets by sterwise refinements* // *Journ. Comp. and Syst. Sci.* – 1979. – Vol.18. – № 1. – P. 35–46.
198. *Widrow B., Hoff M. E. Adaptive Switching Circuits* / IRE WESCON Convention Record. – New York: IRE, 1960. – P. 96–104.
199. *Widrow B., Lehr M. 30 years of Adaptive Neural Networks: Perseptron, Madaline and Backpropagation* // *Proc. IEEE.* – 1990. – 78. – № 9. – P. 1415–1442.
200. *Woo A. C., Peppard L. E. System-level modeling in VHDL* // *Microelectron. J.* – 1992. – 23. – №3. – P. 223–230.
201. *Yudrow B., Winter R. Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition* // *IEEE Computer.* – 1988. – 21. – № 3. – P. 25–39.
202. *Zadeh L. Fuzzy Sets* // *Information and Control.* – 1965.
203. *Zadeh L. et al. Fuzzi Logic for the management of uncernteinty* // John Willey & Sons. – New York, 1992.



## ПРЕДМЕТНИЙ ПОКАЖЧИК

### А

Автомат скінченний 6.5  
Автономність робота 9.4  
Агрегат 9.5  
Аксиома 1.7, 2.1  
Аксиоматичне твердження 2.1, 2.2,  
2.3, 2.4, 2.5, 2.6, 2.7, 2.8  
Алгебра булева 3.3  
– макс/мін 10.1  
– макс-плюс 11.4  
– мін-плюс 11.4  
– процесна 2.8, 6.5, 10.1  
Алгоритм 9.4  
– МГУА багаторядний 6.2  
– комбінований 6.2  
– однорядний 6.2  
– МПЕ 7.5  
– нечіткого виведення 3.4  
– оперативного керування ІВК 8.4  
– організації блочного виведення 7.1  
– пошукового модуля 10.6  
– тренування нейро-фаззі-систем 4.5  
– нечіткої нейросітки 4.5  
– управління організаційно-технічною системою 9.5  
– адаптуючий 6.1  
– у нештатному режимі 9.5  
– функціонування імовірнісної СНСП-моделі 7.5  
– числовий 6.3  
– Mamdani 3.4  
– Tsukamoto 3.4  
Алфавіт формальної системи 1.7, 2.1,  
2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8  
Аналіз 5.1, 9.4  
– вхідного слова 7.4  
– кластерний 5.1  
Ансамбль нейронний 2.6  
Антецедент 2.2  
Асоціативність 2.1, 3.6  
Атрибут 2.3

### Б

База даних 1.2  
– інтелектуальна 1.7  
– знань 1.2, 1.7

Блок активний 7.1  
– виконаний 7.1  
– виток 7.1  
– живучий 7.1  
– умовно виконаний 7.1  
– живучий 7.1  
– типовий сітковий 8.4  
– конвеєрний 8.4  
– паралельний 8.4  
– умовний 8.4  
– циклічний 8.4  
Блокування ресурсів 10.1

### В

Взаємодія безпосередня 6.3  
– непряма 6.3  
– міжмодульна у ПІММК 11.2  
– модулів рівнозначна 11.2  
– скоординована 11.2  
Вибір об'єктно-орієнтований 10.1  
Виведення 1.1, 2.2  
– блочне 7.1  
– висхідне (пряме) 1.8, 2.2, 2.5  
– низхідне (зворотне) 1.8, 2.2, 2.5  
– логічне 2.1, 3.4, 4.2  
– нечітке 3.2, 3.4  
– матричне 7.1  
– на ЛСП 7.1  
– неточне 3.2  
– повне 7.1  
– пряме  
– часткове 7.1  
Висловлювання 2.1, 2.3  
Висновок логічний 2.1  
Висота нечіткої множини 3.3  
Відношення інтенціональне 1.2, 2.3  
– екстенціональне 1.2, 2.3  
– квантифіковане 2.3  
– лінгвістичне 2.3  
– логічне 2.3  
– нечітке 2.4  
– теоретико-множинне 2.3  
Властивість 2.3

### Г

Гетерогенність системи 11.1

- Граф вхідного слова 6.4  
 –“– екстремальних ситуацій 9.5  
 –“– І-АБО 8.1  
 –“– надлишковий абстрактний 9.5  
 –“– ОМ 6.4  
 –“– орієнтований 8.1  
 –“– розподілу заявок 10.1, 11.3  
 –“– станів задачі 8.1  
 –“– подійно-часовий 11.4
- Д**
- Демон 2.4  
 Дерево досяжності 7.1  
 –“– –“– локальне 7.6  
 Десигнат 2.4  
 Дефазифікація (D-перетворення) 3.4, 4.4, 9.5, 11.2  
 Джерело знань 2.5  
 Дискретизація 5.4  
 –“– рівномірна 5.4  
 Дистрибутивність 2.1, 3.6  
 Дія 6.4  
 Доведення формальне 1.7
- Е**
- Евристика 1.1  
 Експеримент активний 6.2  
 –“– пасивний 6.2
- З**
- Забезпечення програмне 10.6  
 Закон алгебраїчний 2.8  
 –“– вилучення третього 2.1  
 –“– відокремлення 2.1  
 –“– Гауса 9.5  
 –“– контрапозиції 2.1  
 –“– суперечності 2.1  
 –“– силогізму 2.1  
 Запит 2.4  
 –“– на існування 2.4  
 –“– –“– перелічення  
 Зведення до чіткості 3.4  
 Зміна лінгвістична 3.3  
 –“– нечітка 3.3  
 Зміст моделювання ГКТС 10.4  
 Знання 1.1  
 –“– глибинні 1.1, 8.2, 8.3  
 –“– декларативні 1.1, 2.3  
 –“– процедурні 1.1, 2.3  
 –“– недостовірні 3.2
- “– ненадійні 3.5  
 –“– непевні 3.2  
 –“– нечіткі 3.2  
 –“– системи 9.5  
 Зображення дискретне 5.3  
 Зона синхронізуюча 11.3
- І**
- Ідемпотентність 2.1, 3.6  
 –“– часткова 3.6  
 Ідентифікація 5.1, 5.6, 9.4  
 Імовірність апостеріорна 3.5  
 –“– збудження переходу 7.5  
 –“– події 3.5  
 –“– успіху експерименту 7.4  
 Інженерія знань 1.5  
 Інтелект *Вступ*  
 –“– ГКІС 9.4  
 –“– машинний *Вступ*  
 –“– штучний *Вступ, передмова, післямова*  
 Інтелектуалізація моделювання 10.5  
 Інтерпретація 1.7  
 –“– ГКІС скінченно-автоматна 6.5  
 –“– ТПСП часова 8.4  
 Інформатизація 1.1  
 Інформація 1.1  
 Істинність 1.7  
 –“– універсальна 2.1
- К**
- Карта Кохонена 2.6  
 Керування програмне номінальне 10.5  
 –“– світлофорне розподілене 11.4  
 Клас 5.1  
 Класифікація 5.1, 5.6, 9.4  
 Клауза 2.1  
 Когнітрон 2.6  
 Код ланцюговий Фрімена 5.3, 5.6  
 –“– функціональної відповідності 8.4  
 Коефіцієнт компенсації неадекватності 11.5  
 Комплекс інженерно-виробничий 9.4  
 –“– програмний 10.6  
 Композиційність станів 11.1  
 Композиція 3.4  
 –“– планарна 11.1  
 –“– max-min 3.4  
 –“– max-prod 3.4  
 –“– max-\* 3.4  
 Компонента керуюча нечітка 11.2

Компонента інтелектуалізована 10.5	–“– евристичний прийняття рішень 6.1
Комутативність 2.1, 3.6	–“– експертних оцінок 6.1
Конорма трикутна 3.3	–“– –“– оцінювань 8.4
Консеквент 2.2	–“– зведення до чіткості 3.4
Константа 1.7	–“– зіставлення з еталоном 5.1
Конфігурація мультимодульна гнучка інтелектуалізована 11.2	–“– імовірнісний 5.1
Координата узагальнена 10.5	–“– індукції 6.2
Критерій балансу змінних 6.2	–“– –“– математичної повної 6.2
–“– комбінований 6.2	–“– скінчених автоматів 8.2
–“– мінімуму зсуву 6.2	–“– контролю синхроімпульсний 5.7
–“– оптимальності узагальнений 6.1	–“– логічного згладжування 5.4
–“– селекції 6.2	–“– математичної індукції 6.2
–“– оптимальності 10.1	–“– максимуму критеріального 3.4
Крок дискретизації 5.4	–“– –“– першого 3.4
	–“– –“– середнього 3.4
	–“– моделювання 11.1
<b>Л</b>	–“– найменших квадратів 6.1
Логіка імовірнісна 3.5	–“– Ньютона 11.5
–“– –“– байєсівська 3.5	–“– парних порівнянь 10.1
–“– епістемічна 3.2	–“– повного перебирання 8.1
–“– модальна 3.2	–“– подання об’єктів наближений 5.5
–“– немонотонна 3.5	–“– програмування динамічного 6.1
–“– нечітка 3.4, 3.5	–“– –“– квадратичного 6.1
–“– –“– 3.6	–“– –“– лінійного 6.1
–“– –“– 3.6	–“– –“– опуклого 6.1
–“– –“– 3.6	–“– –“– цілочислового 6.1
–“– –“– 3.6	–“– ранжування 10.1
–“– умовчання 3.5	–“– резолюцій 2.1
–“– (числення) висловлювань 2.1, 3.4	–“– розпізнавання
–“– –“– предикатів 1.8, 1.11, 2.1, 7.1	–“– самоорганізації моделей 6.2
	–“– статистичного моделювання 6.1
<b>М</b>	–“– структурний 5.1
Маркер кольору спрацювання переходу 8.4	–“– формалізації опису ознак 5.6
Маркування ГСП 7.6	–“– Ф-функцій 5.5
Маршрут технологічний 7.1	–“– Хебба навчання нейросіток 2.6
Матриця виведення 7.1	–“– центроїдний 3.4
–“– допоміжна 7.1	Методика моделювання функцій ІВК 8.4
Метазнання 1.1	–“– проектування алгоритму керування 8.4
Метакерування 11.2	Механізм логічного виведення 1.4, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8
–“– –“– нечітке 11.2	–“– системного часу 11.4
–“– –“– розподілене 11.2, 11.3	–“– створення складних розгалужень 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8
Метод аналітичного аналізу об’єктів 5.5	Мислення модельне 7.5
–“– аналогій 11.1	Міра можливості 3.5
–“– висотної дефазифікації 3.4	–“– необхідності 3.5
–“– градієнтний 11.5	Місце робоче автоматизоване 8.4
–“– групового урахування аргументів 6.2	Множина матриць перестановки 7.5
–“– Делфі 8.4	–“– –“– розширених 7.5
–“– детерміністський 5.1	–“– нечітка нормальна 3.3
–“– дискримінантний 5.1	

- “–“– порожня 3.3
  - “–“– субоптимальна 3.3
  - “–“– непорожня 3.3
  - “–“– унімодальна 3.3
  - “– характеристична 6.4
  - “– нечітка 3.3, 9.5
  - “– чітка 3.3, 9.5
  - Мова формалізована технологічна 5.5
  - “– декларативного типу 1.10
  - “– подання знань 1.9
  - “– процедурного типу 1.10
  - “– розробки ЕС 1.4
  - Модель 10.1
  - “– АРТС алгебраїчна 11.4
  - “–“– математична
  - “– взаємодії вкладена 10.1
  - “–“– мішана 10.1
  - “–“– однорівнева 10.1
  - “– гіперпросторова 10.1
  - “– ГКІС гіперпросторова 11.1
  - “–“– імітаційна динамічна 10.6
  - “–“– концептуальна 7.1
  - “– дошки оголошень 1.11, 2.5
  - “– імітаційна автоматизована 6.4, 10.6
  - “– координатна математична 5.5
  - “– логічна 1.8, 1.11, 2.1
  - “– математична МС 10.5
  - “–“– приводу універсальна 10.5
  - “– МС ПР повна 10.5
  - “– оптимальної складності 6.1
  - “– подання знань типова 1.8
  - “– правил класичного числення
  - предикатів 2.1
  - “–“– продукцій 1.8, 1.11, 2.2,
  - 8.1, 8.2, 8.3
  - “– розпізнавання абстрактна 5.3
  - “– сітки Петрі 1.8, 1.1, 2.7, 7.1, 7.1, 8.4
  - “–“– нейронної 1.8, 1.11, 2.6
  - “–“– семантичної 1.8, 1.11, 2.3, 9.1, 8.4
  - “– сіткова алгоритмів керування
  - “–“– петлеподібної АРТС 11.4
  - “–“– структури алгоритму організації
  - керування ІВК 8.4
  - “– теоретико-множинна 5.3
  - “– фреймова 1.8, 1.11, 2.4, 9.2, 9.4
  - Моделювання автоматизоване 6.4
  - “– аналітичне 6.1, 6.5, 10.2, 11.3
  - “– дискретно-подійне 6.5
  - “– електронне 6.1
  - “– імітаційне 6.1, 6.2, 6.5, 10.1, 10.2
  - “–“– в АТСС 10.4
  - “– композиційне ієрархічне 11.1
  - “– математичне 6.1, 10.1
  - “– статистичне 6.1, 10.1
  - “– фізичне 6.1
  - Модифікування динамічного навантаження СП 7.3
  - Модуль взаємодії сітковий 11.2
  - “– з функціями метакерування 11.2
  - “– гнучкий виробничий тестовий 7.1
  - “– логічного виведення 1.4
  - “– планування експериментів 6.4, 8.3
  - “– пошуковий 6.4, 8.9, 8.3, 10.6
  - “– програмний 10.4
  - Момент зчленування 10.5
  - “– номінальний 10.5
- Н**
- Навантаження СП модифіковане 7.2
  - “– СП-моделі динамічне 7.2
  - “–“– ГКІС 7.2
  - “– ієрархічне 7.2
  - Навчання системи 5.1
  - Належність нечітка 3.3
  - Невизначеність 6.1
  - “– випадкова 3.5
  - “– “ворожа” 6.1
  - “– лінгвістична
  - “– “погана” 6.1
  - “– стохастична 6.1
  - Нейрон 1.8, 2.6
  - “– фаззифікуючий 4.3
  - Неокогнітрон 2.6
  - Норма трикутна 3.3
  - Носій нечіткої множини 3.3
- О**
- Об’єкт 5.1
  - “– дослідження 10.5
  - “– елементарний 5.5
  - “– керування 8.4
  - “– моделювання 10.1
  - “–“– імітаційного 10.4
  - “– СС 2.3
  - Область задавання СНСП-моделі 7.5
  - Ознака логічна 5.6
  - “– первинна 5.3
  - “– структурна 5.5, 5.6
  - “– топологічна 5.5, 5.6
  - Оператор 1.7



- “– взаємодії (композиції) 2.8
- “– –“– –“– послідовної 2.8
- “– вибору 2.8
- “– дія-префікс 2.8
- “– пасивності 2.8
- “– переривання 2.8
- “– –“– катастрофічного 2.8
- “– процесної алгебри 2.8
- Операція над протоколами 2.8
- “– критична 10.4, 10.6
- “– min-max 3.3
- Основи автоматизованого моделювання  
формальні 6.4, 10.6
- Оцінювання експертне 8.4
- “– рейтингове 10.1
  
- П**
- Пара атрибутивна 2.3
- Параметр двигуна 10.5
- “– динамічний 10.5
- “– кінематичний 10.5
- Передумова 8.4
- Перетворення інтегральне Карунена-  
Лоева 5.4
- “– ортогональне 5.4
- Перехід 2.7, 7.1
- “– активний 7.6
- “– вхідний 7.1
- “– дублер 8.4
- “– збуджений 7.5, 7.6
- “– незбуджений 7.5
- Підготовка виробництва  
конструкторська 8.4
- “– технологічна 8.4
- Підмножина операцій ініціююча 9.5
- Підсистема управління преселективною  
обробкою 9.5
- Підставлення 2.1, 6.4
- “– сіткового блоку 8.4
- Підстановка 1.7
- Підхід індуктивний 6.2
- “– логістичний 10.1
- “– проєкційний 5.5
- “– комбінований 10.2
- “– конструктивний 5.5
- Планування модельне 7.5
- “– в просторі станів 8.1
- Подія 2.3, 2.7
- “– примітивна 7.5
- Позиція 2.7
- “– активна 7.5
- “– безпечна 8.4
- “– блоку вхідна 7.1
- “– джерело 7.1
- “– вхідна 7.1
- “– вихідна 7.1
- “– обмежена 7.1, 8.4
- “– СП 2.7, 7.1
- Помилковість 1.7
- Поняття 2.3
- Послідовність розв’язків оптимальна 8.1
- Постумова 8.4
- Пошук евристичний 8.1
- “– у глибину 8.1
- “– –“– ієрархії просторів 8.3
- “– –“– просторі станів 8.3
- “– –“– у ширину 8.1
- “– найменших завершень 8.3
- Правило 1.1
- “– виведення 1.7, 2.1
- “– зв’язування квантором існування 2.1
- “– нечіткого умовного логічного  
виведення 3.3
- “– modus ponens 2.1, 3.6
- “– переписування 1.7
- “– продукційне 1.7, 1.11
- “– резолюції 2.1
- “– –“– узагальнене 2.1
- “– спеціалізації 2.1
- “– умовчання 3.5
- “– універсальної конкретизації 2.1
- Предикат 2.1
- Призначення пріоритетів динамічне 9.1
- Прийняття рішень модельне 7.5
- Примітив 8.2
- Принцип зворотного зв’язку 9.5
- “– інтелектуального моделювання 9.5
- “– найменших завершень 8.3
- “– низхідного уточнення 8.3
- “– максимального розпаралелювання 9.5
- “– оптимізації керуючих операцій 9.5
- “– розподіленого світлофорного  
керування 11.4
- “– віртуальних асинхронних транспорт-  
них світлофорів 10.1
- Прогнозування подій 10.4
- Програмування динамічне 8.1
- Продукція 1.7
- Продуктивність моделі ПР 7.4
- Проектування паралельне 5.3

–“– центральне 5.3  
 Простір агрегатних станів 9.5  
 –“– дискретний 9.4  
 –“– ознаковий 5.2, 5.4  
 –“– стану 8.1  
 Протокол асинхронний 10.1  
 –“– розподілення потужності 11.3  
 Процедура дефазифікації 4.2  
 –“– композиції 4.2  
 –“– логічного розв’язку 4.2  
 –“– побудови формули 1.7, 2.1  
 –“– приєднана 2.4  
 –“– фазифікації 4.2  
 Процес буфер 2.8  
 –“– детермінований 2.8  
 –“– марковський 6.5, 10.1  
 –“– недетермінований 2.8  
 –“– оптимальний 6.1  
 –“– проектування 8.2  
 –“– селективного підбору 9.5  
 –“– стохастичний 6.1  
 –“– транспортер 2.8  
 –“– тренування нечіткої НС 4.5

**Р**

Реалізація гіперпросторових моделей  
 процедурна 10.1  
 Режим аварійний 9.5  
 –“– нештатний 9.5  
 –“– “– внутрішній 9.5  
 –“– “– глобальний 9.5  
 –“– “– інформаційний 9.5  
 –“– “– локалізований 9.5  
 –“– “– програмний 9.5  
 –“– реального часу 9.2  
 –“– функціонування циклічний  
 усталений 10.1  
 –“– штатний 9.5  
 –“– “– цільовий без аномалій 9.5  
 –“– “– “– з аномаліями 9.5  
 –“– “– “– що не забезпечує цільову  
 функцію 9.5  
 Резолюція (резольвента) 2.1  
 Ресурс критичний 10.4  
 Ризик розв’язання умовний 7.5  
 Рівняння Лагранжа 6.5  
 –“– різницеве 6.5  
 Робот інтелектуальний 9.4  
 –“– промисловий 1.2  
 Розв’язання логічне 4.4

–“– “– в нейро-фаззі-системі 4.4  
 Розпізнавання 5.1, 5.6  
 –“– дискримінантне 5.2, 5.3  
 Розподіленість просторово-агрегатна 9.5,  
 10.5  
 –“– “– ієрархічна 9.5, 10.5, 10.6  
 –“– “– подійна 9.5, 10.5  
 Розробка моделюючого алгоритму 10.4

**С**

Самокерування 9.4  
 Селекція масова 6.2  
 Семантика потоків ініціалізації сіткова 11.2  
 Середовище семантично-узгоджене 11.1  
 –“– “– “– гіперпросторового  
 моделювання 11.1  
 –“– “– “– комплексно-інтегроване 11.1  
 Сила номінальна 10.5  
 Силогізм 3.5  
 –“– гіпотетичний 3.5  
 Синтез керуючої структури 11.4  
 Синхронізація мультиподійна 11.4  
 –“– транспортних світлофорів 11.4  
 Система 9.5  
 –“– автоматизації імітаційного  
 моделювання 1.4, 6.4  
 –“– автоматизована транспортна  
 розподілена 11.4  
 –“– “– “– “– петлеподібна 11.4  
 –“– “– транспортно-складська 6.4  
 –“– агрегатована 9.5  
 –“– адаптивна з нейро-фаззі-виведенням  
 4.5  
 –“– БД 1.2  
 –“– БЗ 1.2  
 –“– гнучка комп’ютерно-інтегрована 1.5,  
 5.6, 6.5, 7.1, 10.1, 11.1  
 –“– “– “– “– розподілена 11.1  
 –“– дедуктивного виведення 1.3  
 –“– дискретно-подійна 6.5  
 –“– експертна 1.4, 6.3, 9.2  
 –“– з агрегатними станами 9.5  
 –“– знань 1.7  
 –“– “– “– про зовнішнє середовище 9.4  
 –“– “– “– “– операції робота 9.4  
 –“– “– “– формальна 1.7  
 –“– інтелектуальна 9.1  
 –“– “– “– гібридна 6.3  
 –“– “– “– моделювання 6.4  
 –“– “– “– реального часу 9.1, 9.2

- “– комбіновано-гібридна 6.5
  - “– масового обслуговування 6.5
  - “– нейро-фаззі 4.3, 9.5
  - “– –“– інтегрована 4.5
  - “– –“– реального часу 9.1, 9.3
  - “– оптимальна 6.1
  - “– пошуку дедуктивна 2.1
  - “– управління БД 1.2
  - “– що ґрунтується на знаннях 1.2
  - Ситуація нештатна 9.5
  - Сітка нейронна 1.8, 2.6, 4.2, 4.3, 4.4, 10.5, 9.6
  - “– –“– з оберненим зв'язком 2.6
  - “– –“– інтелектуалізована 4.3
  - “– –“– нечітка 4.1, 4.3
  - “– –“– прямоспрямована 2.6
  - “– –“– штучна 2.6, 4.2, 9.5
  - “– Петрі 1.8, 2.7, 6.5, 7.1, 10.1
  - “– –“– гнучка 7.6
  - “– –“– –“– синхронізована 7.6
  - “– –“– –“– динамічна 7.4
  - “– –“– –“– живуча 11.4
  - “– –“– –“– класична 2.7
  - “– –“– –“– логічна 7.1
  - “– –“– –“– нечітка 7.1
  - “– –“– –“– проста 7.1
  - “– –“– –“– навантажена 7.2
  - “– –“– –“– обмежена 11.4
  - “– –“– –“– розфарбована ієрархічна 9.4
  - “– –“– –“– часова 7.2, 7.5
  - “– –“– –“– радіально-базисних функцій 2.6
  - “– семантична 1.8, 1.11, 2.3
  - “– Хопфілда 2.6
  - Склад реєстрованих параметрів і задач 10.4
  - Слово вихідне 6.4
  - “– вхідне 6.4
  - Слот 1.8, 2.4
  - “– термінальний 2.4
  - Список динамічних атрибутів 10.4
  - СП-модель автоматизована 7.3
  - “– –“– навантажена 7.2
  - “– –“– –“– динамічно 7.2
  - “– –“– –“– самонавантажена 7.3
  - Стадія життєвого циклу виробництва експлуатаційна 8.4
  - “– –“– –“– –“– виробу 8.4
  - “– –“– –“– –“– підготовча 8.4
  - Стан агрегатний 9.5
  - “– безпечний 11.3
  - “– блокування процесів 11.3
  - “– –“– нештатний 9.5
  - “– –“– –“– правильний 9.5
  - “– –“– –“– проміжний 9.5
  - “– –“– –“– штатний 9.5
  - Стратегія розподіленого керування 11.3
  - “– розподілу заявок 11.3
  - “– синхронізуючих зон 11.3
  - Стрілка Пірса 2.1, 3.6
  - Структура ЛСП блочна 7.1
  - “– –“– умовно живуча 7.1
  - “– імітаційної моделі 10.4
  - “– нечіткої керуючої компоненти модульна 11.2
  - Сукупність перетинних циклів 7.1
- Т**
- Твердження 1.7
  - “– аксіоматичне 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8
  - “– істинності 1.7
  - “– помилковості 1.7
  - Теорема 1.7, 2.1
  - “– дедукції 2.1
  - “– де Моргана 3.6
  - “– про суперечливість 2.1
  - Теорія агрегативних систем 9.5
  - “– ігор 6.1
  - “– масового обслуговування 10.1
  - “– можливостей 3.5
  - “– розкладів 10.1
  - Терм 2.1
  - “– –“– множина розширена лінгвістичної змінної 3.3
  - Топологія НС 2.6
  - Транзакція 2.8, 10.1
  - Траєкторія базова виходу у множині режимів 9.5
  - “– керуючого сигналу 10.5
- У**
- Умова безпеки СП 8.4
  - “– безтупиковості 10.1
  - “– живучості СП 7.1, 7.3, 8.4
  - “– завершення спрацьовування переходу 8.4
  - “– збудження переходу 8.4
  - “– спрацьовування переходу 8.4
  - “– ініціююча 9.5
  - “– невизначеності 6.1
  - “– узгодження даних ОМ та ІМ 6.4

–“– функціонування достатня 6.4  
–“– –“– необхідна 6.4  
Умовчання нормальне 3.5  
–“– –“– закрите 3.5  
Управління оптимальне 6.1

**Ф**

Фаззі-контролер 4.2, 9.5  
–“– логіка 3.3  
–“– управління 4.4, 9.5  
Фаззіфікація 3.4, 9.5, 11.2  
Факт 1.1, 2.3  
Фільтрація анізотропна 5.4  
–“– зображень 5.4  
Формування вхідного слова 8.4  
Формула 1.7  
–“– атомарна 2.1  
–“– –“– логіки першого порядку 2.1  
–“– Байеса 3.5  
–“– вивідна 2.1  
–“– правильно побудована 2.1  
–“– –“– –“– логіки предикатів 2.1  
Фрейм 1.8, 2.4  
–“– класу 2.4  
Функціонування підпорядковане 6.3  
–“– рівноправне 6.3

Функція активаційна 2.6, 9.5  
–“– належності 3.1, 9.5  
–“– –“– нечіткої 3.3, 9.5  
–“– розподілу 5.2

**Ц**

Цикл 7.1  
–“– життєвий виробу 8.4  
–“– –“– наскрізний 8.4

**Ч**

Час руху найкоротший 11.4  
–“– спрацьовування переходу СП 7.5  
–“– реальний 9.5  
Частота дискретизації 5.4  
Черга заявок на обслуговування 10.4  
Число нечітке (3.3)

**Ш**

Шаблон 2.4  
Шлях у структурі ЛСП 7.1  
–“– імплікантний 7.1  
Штрих Шеффера 2.1, 3.6

**Щ**

Щільність розподілу 5.2



# ЗМІСТ

<b>ПЕРЕДМОВА ЗАГАЛЬНА</b> .....	3
<b>СПИСОК СКОРОЧЕНЬ</b> .....	9
<b>ПЕРЕДМОВА</b> .....	13
<b>ВСТУП</b> .....	15
<b>Розділ 1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЕКСПЕРТНІ СИСТЕМИ</b> .....	25
1.1. Інформація та знання.....	25
1.2. Означення баз даних і баз знань інтелектуальної системи.....	27
1.3. Зв'язки між складовими знань у базах даних.....	29
1.4. Означення та структура експертної системи.....	31
1.5. Інженерія знань.....	35
1.6. Проблемний характер формалізації та подання прикладних знань.....	36
1.7. Поняття та означення формальної системи.....	37
1.8. Способи подання знань .....	39
1.9. Мовні засоби для подання знань в інтелектуалізованих системах.....	50
1.10. Порівняльна характеристика сучасних способів подання знань.....	53
<i>Контрольні завдання</i> .....	56
<b>Розділ 2. ТИПОВІ МОДЕЛІ ПОДАННЯ ЗНАНЬ У СУЧАСНИХ ЕКСПЕРТНИХ СИСТЕМАХ</b> .....	57
2.1. Логічні моделі подання знань .....	57
2.2. Продукційні системи.....	70
2.3. Моделі подання знань семантичними сітками .....	74
2.4. Подання знань у вигляді фреймів .....	81
2.5. Модель дошки оголошень .....	91
2.6. Моделі штучних нейронних сіток .....	94
2.7. Сітки Петрі як формальний апарат подання і відображення процесів складних систем.....	108
2.8. Апарат процесних алгебр як основа формального визначення поведінки системи у семантичному, модульному та ієрархічному поданні .....	112
<i>Контрольні завдання</i> .....	123
<b>Розділ 3. НЕДОСТОВІРНІ ТА НЕЧІТКІ ЗНАННЯ</b> .....	126
3.1. Загальні положення.....	126
3.2. Поняття про недостовірні та нечіткі знання.....	128
3.3. Чіткі та нечіткі множини .....	129
3.4. Нечітке логічне виведення .....	137
3.5. Ненадійні знання та виведення.....	143
3.6. Формалізація знань інтелектуалізованих роботів.....	151
<i>Контрольні завдання</i> .....	158
<b>Розділ 4. НЕЙРО-ФАЗЗИ-ТЕХНОЛОГІЇ В СИСТЕМАХ ІЗ ШТУЧНИМ ІНТЕЛЕКТОМ</b> .....	160
4.1. Вихідні міркування.....	160
4.2. Принципи побудови і структура нейро-фаззі-систем .....	160
4.3. Особливості формування нейро-фаззі систем.....	162
4.4. Відображення системи фаззі-управління в нейросітках.....	164
4.5. Алгоритми тренування нейро-фаззі-систем .....	168
<i>Контрольні завдання</i> .....	173

<b>Розділ 5. РОЗПІЗНАВАННЯ ОБРАЗІВ</b> .....	174
5.1. Основні принципи розпізнавання.....	174
5.2. Розпізнавання в просторі ознак .....	180
5.3. Теоретико-множинна модель системи розпізнавання об'єктів роботизації.....	184
5.4. Основні методи попередньої обробки сигналів і зображень.....	187
5.5. Формалізоване подання об'єктів ГКІС.....	193
5.6. Узагальнена концептуальна модель гнучкого виробництва.....	199
5.7. Формалізований опис ознак об'єктів та оперативний аналіз сенсорної інформації.....	211
5.8. Методи і засоби розпізнавання об'єктів роботизації .....	215
<i>Контрольні завдання</i> .....	227
<b>Розділ 6. КОМБІНОВАНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ</b> .....	229
6.1. Методи математичного моделювання та управління виробничими процесами.....	229
6.2. Деякі підходи до моделювання в комбінованих інтелектуальних системах .....	237
6.3. Гібридні інтелектуальні системи .....	245
6.4. Інтелектуальні системи моделювання .....	251
6.5. Загальні основи моделювання складних комп'ютерно-інтегрованих систем .....	261
<i>Контрольні завдання</i> .....	271
<b>Розділ 7. СІТКИ ПЕТРІ ЯК БАЗОВИЙ АПАРАТ МОДЕЛЮВАННЯ І ПРОЕКТУВАННЯ ПРОЦЕСІВ ГКІС</b> .....	273
7.1. Класична модель сіток Петрі .....	273
7.2. Об'єктно-орієнтоване на подання складних ОМ розширення базового апарату СП.....	283
7.3. Загальні підходи до структурного перетворення СП.....	284
7.4. Типові сіткові блоки як основа структурного перетворення СП (I модифікація) .....	287
7.5. Сіткові (блочні) моделі процесів ГКІС (II модифікація – перший етап проектування СК ГКІС) .....	293
7.6. Аналіз умов організації виробничого процесу ГКІС за його сітковою моделлю (другий етап проектування СК ГКІС) .....	299
7.7. Синтез алгоритмів керування виробничими процесами ГКІС (третій етап проектування СК ГКІС) .....	301
<i>Контрольні завдання</i> .....	307
<b>Розділ 8. ІНТЕЛЕКТУАЛІЗАЦІЯ МОДЕЛЮВАННЯ СКЛАДНИХ ОБ'ЄКТІВ НА ОСНОВІ РОЗШИРЕНЬ АПАРАТУ СП</b> .....	309
8.1. Логічні сітки Петрі.....	309
8.2. Орієнтоване на інтелектуалізоване моделювання розширення апарату СП .....	319
8.3. Організація динамічного навантаження при плануванні й проведенні експериментів на ІНСП-моделі.....	322
8.4. Алгоритмічне забезпечення модуля планування експериментів.....	326
8.5. Детермінований та ймовірнісний підходи до моделювання ієрархічних структур ГКІС .....	332

8.6. Обґрунтування нових модифікацій СП для динамічного моделювання ГКІС .....	341
Контрольні завдання.....	349

**Розділ 9. ШТУЧНИЙ ІНТЕЛЕКТ У СИСТЕМАХ ПРОЕКТУВАННЯ ТА ПЛАНУВАННЯ..... 351**

9.1. Методи та засоби відображення розв'язків оптимізаційних задач.....	351
9.2. Експертні системи в задачах проектування і планування .....	356
9.3. Методи пошуку розв'язків на прикладах задач планування.....	361
9.4. Експертні системи в плануванні та моделюванні процесів підготовчої стадії життєвого циклу виробів .....	370
Контрольні завдання.....	391

**Розділ 10. ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ РЕАЛЬНОГО ЧАСУ..... 393**

10.1. Загальні відомості .....	393
10.2. Вимоги до інтелектуальних систем реального часу та шляхи підвищення їх швидкодії.....	394
10.3. Побудова та особливості застосування інтелектуальних систем реального часу.....	396
10.4. Інтелектуальна підтримка функціонування промислових роботів у реальному часі .....	400
10.5. Складні технічні системи з агрегатними станами як об'єкти керування в реальному часі .....	411
10.6. Підходи до розв'язання невизначеності в умовах функціонування складних об'єктів керування в реальному часі .....	423
Контрольні завдання.....	431

**Розділ 11. ІНТЕЛЕКТУАЛІЗОВАНЕ ДИСКРЕТНО-ПОДІЙНЕ МОДЕЛЮВАННЯ ГНУЧКИХ ВИРОБНИЧИХ СИСТЕМ..... 432**

11.1. Особливості математичного, імітаційного та статистичного моделювання процесів АТСС .....	432
11.2. Загальна постановка задачі.....	443
11.3. Аналітичне моделювання процесів в АТСС.....	446
11.4. Імітаційне моделювання процесів в АТСС.....	452
11.5. Інтелектуалізація моделювання ГКІС з урахуванням динамічних властивостей її складових .....	460
11.6. Структура динамічного навантаження моделі ГКІС.....	473
Контрольні завдання.....	485

**Розділ 12. НОВІ ПІДХОДИ ДО МОДЕЛЮВАННЯ І УПРАВЛІННЯ В ГНУЧКИХ КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ СИСТЕМАХ..... 486**

12.1. Семантично-узгоджене середовище гіперпросторового моделювання ГКІС .....	486
12.2. Нечітке метакерування у ГКІС .....	490
12.3. Розподілене керування «критичними» ресурсами у циклічних ГКІС.....	497
12.4. Розподілене керування із самосинхронізацією потоків ГКІС.....	504
Контрольні завдання.....	519

**ПІСЛЯМОВА..... 520**

**СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ..... 523**

**ПРЕДМЕТНИЙ ПОКАЖЧИК..... 533**

Підручник є продовженням циклу розробленої авторами вперше в Україні навчальної літератури нового покоління з широким впровадженням сучасних інформаційних технологій в електронній версії, виконаній в PDF-форматі, призначеній для Internet-розповсюдження, оснащеної тестуючими та пошуковими системами і лабораторними практикумами, і супроводжується мультимедійною анімацією складних для сприйняття процесів функціонування технологічних та організаційно-технічних систем.

Для студентів, викладачів, аспірантів і науковців, які займаються проблемою штучного інтелекту.

**Ямпольський Леонід Стефанович**  
**Ткач Борис Петрович**  
**Лісовиченко Олег Іванович**

# **СИСТЕМИ ШТУЧНОГО ІНТЕЛЕКТУ В ПЛАНУВАННІ, МОДЕЛЮВАННІ ТА УПРАВЛІННІ**

*Підручник*

Редактор *М. В. Дроздецька*  
Коректор *С. М. Толкачова*  
Комп'ютерне верстання *А. П. Нечипорук*  
Оформлення обкладинки *О. О. Стеценко*

Підп. до друку 25.10.11. Формат 70×100/16. Папір офсетний. Друк офсетний.  
Ум. друк. арк. 43,86. Обл.-вид. арк. 31,65. Наклад 1000 пр.

Міжрегіональна Академія управління персоналом (МАУП)  
03039 Київ-39, вул. Фрометівська, 2, МАУП

ДП «Видавничий дім «Персонал»  
03039, Київ-39, просп. Червонозоряний, 119, літ. XX

*Свідоцтво про внесення до Державного реєстру суб'єктів  
видавничої справи ДК № 3262 від 26.08.2008 р.*

Надруковано в друкарні ДП «Видавничий дім «Персонал»