МІЖРЕГІОНАЛЬНА
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ

МАУП

С. С. Шкільняк

МАТЕМАТИЧНА ЛОГІКА

ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ

Навчальний посібник

Київ
ДП «Видавничий дім «Персонал»
2009

ББК 22.12я73
Ш66

Рецензенти: *А. Ю. Дорошенко*, д-р фіз.-мат. наук, проф.
П. П. Кулябко, канд. фіз.-мат. наук, доц.
І. В. Бейко, д-р фіз.-мат. наук, проф.
Ю. А. Бєлов, д-р фіз.-мат. наук, проф.
П. П. Кулябко, канд. фіз.-мат. наук, доц.

Схвалено Вченою радою Міжрегіональної Академії управління персоналом (протокол № 5 від 30.05.07; протокол № 10 від 26.12.07)

Шкільняк, С. С.

Ш66 Математична логіка; Основи теорії алгоритмів : навч. посіб. / С. С. Шкільняк. — К.: ДП «Вид. дім «Персонал», 2009. — 280 с. — Бібліогр. в кінці частин.

ISBN 978-966-608-979-6

У посібнику викладено поняття і результати, які належать до основ математичної логіки і основ теорії алгоритмів. Вивчаються пропозиційна логіка, класична логіка першого порядку, а також інтуїціоністська та модальні логіки, формальні моделі алгоритмів і обчислюваних функцій, питання розв'язності та нерозв'язності, відносна обчислюваність, звідності. Наприкінці розділів наведено питання для самоконтролю, задачі та вправи для самостійного виконання.

Для студентів спеціальностей "Інформатика" та "Прикладна математика".

ББК 22.12я73

ISBN 978-966-608-979-6

Історія розвитку математичної логіки

Логіка – наука про закони мислення. Математична логіка є наукою про закони математичного мислення.

Предметом математичної логіки є математичні теорії в цілому, які вивчають за допомогою логіко-математичних мов. При цьому в першу чергу цікавляться питаннями несуперечливості математичних теорій та їх розв'язності.

Логіка як теоретична наука зародилася в Давній Греції у VI ст. до н. е. В ті часи виникли грецькі демократичні держави-поліси, які надавали своїм громадянам (звичайно, не рабам) найсприятливіші умови для вільного розвитку. Ставало вже замало перевіряти ті чи інші міркування на їх відповідність поглядам вождів чи авторитетів, тому виникла потреба вміти переконувати рівних собі громадян. У цих умовах з'явилося багато пройдисвітів, які дурили людей нібито правильними, але насправді некоректними міркуваннями, здебільшого використовуючи приховану тонку підміну понять. Такі міркування, які містять навмисну, але замасковану помилку, з тих пір називають *софізмами.*

Найвідомішими є античні софізми "сидячий", "вкритий", "рогатий".

Софізм "сидячий": *Сидячий встав. Хто встав, той стоїть. Отже, сидячий стоїть.*

Софізм "вкритий": *Чи знаєш ти цього вкритого чоловіка, що спить під стіною? Не знаю. Але це ж твій батько! Отже, ти не знаєш свого батька.*

Софізм "рогатий": *Те що ти не втратив, ти маєш. Ти не втратив роги. Отже, ти рогатий.*

Звичайно, і тоді і нині не бракує некоректних міркувань із ненавмисним порушенням логічних законів. Такі некоректні міркування називають *паралогізмами.* Типовим прикладом паралогізму є таке міркування: *Якщо через дріт пропускають електричний струм, то дріт нагрівається. Дріт нагрівається. Отже, через нього пропускають електричний струм.*

Зауважимо, що чітку межу між софізмами і паралогізмами провести важко. Наприклад, до паралогізмів можна віднести таке міркування: *Усі студенти складають іспити. Микола складає іспити. Отже, Микола – студент.* Водночас видається ближчим до софізмів таке міркування: *Ссавці населяють сушу й океани. Миші – ссавці. Отже, миші населяють сушу й океани.*

Поширення софізмів вимагало належної протидії, точного формулювання принципів і правил логічних міркувань. Це зумовило зародження логіки як науки про закони мислення. Перші кроки логіки та всієї теоретичної науки пов'язані з діяльністю філософів іонійської школи, основоположником якої був *Фалес Мілетський* (625–547 рр. до н. е.). Він вперше довів деякі математичні твердження: діаметр ділить коло навпіл, кути при основі рівнобедреного трикутника рівні. Звичайно, самі твердження дуже прості та видаються очевидними, але принципово новим було те, що для них пропонувалось суто логічне доведення. Потужний крок у розвитку теоретичної науки і математики зробили добре нам відомий *Піфагор* (570–500 рр. до н. е.) та його учні, які вивели низку досить складних математичних тверджень суто логічним шляхом. Досить згадати теорему Піфагора, доведення існування ірраціональних чисел. Останній факт *в принципі* не можна встановити емпірично.

Загальні принципи логічних міркувань далі розвинув знаменитий філософ *Платон* (427–347 рр. до н. е.). Основоположником логіки як цілісної науки, батьком логіки, є славетний *Арістотель* (384–322 рр. до н. е.). Саме він чітко сформулював три основні закони традиційної логіки (тотожності, несуперечливості, виключеного третього), розробив закони логічного виведення, запропонував аксіоматичний метод. Арістотель створив першу формально-аксіоматичну систему логіки – силогістику, заклав основи модальної логіки.

Згідно з аксіоматичним методом твердження стосовно деякої предметної області поділяються на основні, або початкові, які приймаються без доведення (Арістотель назвав їх аксіомами), та вивідні, що отримуються із початкових чисто логічним шляхом (Арістотель назвав їх теоремами). При цьому правила логічного виведення повинні гарантувати збереження істинності (із істинних тверджень виводиться істинне твердження).

Арістотелева логіка є наукою про закони вивідного знання, таку логіку називають формальною. *Формальна логіка* вивчає акти мислення (поняття, судження, умовиводи, доведення) тільки з огляду їх форми, логічної структури, абстрагуючись від конкретного змісту.

Після Арістотеля майже 2 тис. років не вдавалось зробити принципові кроки в розвитку логіки. Звичайно, наука не стояла на місці. Згадаймо хоча б таких відомих мислителів середньовіччя, як *Аніцій Боецій* (475–525, робив спроби математизації логіки), *П'єр Абеляр* (1079–1142, розвинув номіналістичний напрям), *Томас Аквінський* (1225–1274, обґрунтовував християнську догматику вченням Арістотеля), *Рамон Луллій* (1235–1315, займався проблемами логічного наслідку), *Френсіс Бекон* (1561–1626, розвинув індуктивний метод), *Рене Декарт* (1596–1650, розвинув дедуктивний метод).

Розвиток логіки як науки значною мірою стимулювався проблемою пошуку універсального методу доведень. Визначним кроком у розвитку логіки була ідея створення універсального логічного числення, яку запропонував видатний німецький математик і філософ *Готфрід Лейбніц* (1646–1716). Ця ідея далеко випередила свій час і фактично привела до виникнення математичної логіки. Лейбніц також сформулював четвертий основний закон традиційної логіки – закон достатньої підстави.

Подальші успіхи логіки пов'язані з іменами філософів, логіків і математиків XVIII–XIX ст. У першу чергу варто згадати таких визначних філософів, як *І. Кант* (1724–1804) та *Г. Гегель* (1770–1831), які дуже багато зробили для розвитку традиційної логіки.

Середина XIX ст. ознаменувалась виникненням першої завершеної системи *математичної* логіки, яку запропонував *Дж. Буль* (1815–1864). Така система – алгебра логіки – базувалась на строгій логіко-математичній мові. Математична логіка по суті є формальною логікою, що використовує математичні методи.

У XIX ст. математична логіка розвивалась, зокрема, зусиллями *А. де Моргана* (1806–1873), *Е. Шредера* (1845–1902), *П. Порецького* (1846–1907), *Ч. Пірса* (1839–1914). Логіко-математичні мови і теорія їх змісту розвинені в роботах *Г. Фреге* (1848–1925), який ввів поняття предикату і кванторів. Це дало змогу застосовувати логічні засоби для дослідження основ математики. Цілі розділи математики на мові математичної логіки та аксіоматизація арифметики було викладено

Дж. Пеано (1858–1932). Спроба Г. Фреге і *Б. Рассела* (1872–1970) зведення всієї математики до логіки не досягла основної мети, але зумовила створення багатого логічного апарату, без якого оформлення математичної логіки як повноцінного розділу математики було б неможливе.

Величезний вплив на розвиток математичної логіки справило відкриття на межі XIX–XX ст. парадоксів, пов'язаних з основними поняттями теорії множин.

Парадокс – логічно правильне міркування, яке призводить до взаємовиключних висновків. На відміну від софізмів, парадокси не порушують законів формальної логіки, тому їх ще називають логічними катастрофами.

Парадокси відомі з античних часів. Найвідомішим є *парадокс брехуна*. В стислій формі він має такий вигляд: *Я брешу.* Якщо той, хто це проголошує, не бреше, то висловлене ним є істина, тобто він бреше. Якщо ж той, хто це проголошує, бреше, то висловлене ним є фальш, тобто він не бреше. Отже, він бреше і не бреше одночасно!

Іншою формою парадоксу брехуна є *парадокс критянина*, або парадокс Епіменіда. Цей парадокс згадується в "Посланні до Тита св. Апостола Павла". Парадокс критянина можна сформулювати так: *Критянин Епіменід сказав: усі критяни брехуни.*

Парадокс брехуна довгий час сприймався як певний курйоз, пов'язаний з неоднозначністю та недовизначеністю речень природної мови, тому не викликав особливих турбот. Проте ситуація стала загрозливою, коли парадокси були відкриті в основі самої математики – теорії множин. Першим із них був парадокс Буралі-Форті (1897), пов'язаний з ординалами (порядковими числами). Парадокс Кантора (1899) пов'язаний з потужністю множини усіх множин (див. [6, 9]). Найрезонанснішим став блискучий за формою парадокс Рассела (1906).

Парадокс Рассела. Назвемо множину нормальною, якщо вона не є елементом самої себе. Наприклад, множина стільців не є стільцем, множина котів не є котом і т. п. Переважна більшість множин нормальні. Проте можна вказати ненормальні множини. Множина ненормальна, якщо вона є елементом самої себе. Наприклад, множина усіх абстрактних понять є абстрактним поняттям. Ненормальною є множина усіх множин. Зрозуміло, що кожна множина або нормальна, або ненормальна. Поставимо питання: *множина усіх нормальних множин нормальна чи ненормальна?*

Нехай S – множина усіх нормальних множин. Тоді $\boldsymbol{S} = \{A \mid A \notin A\}$. Маємо $A \in S \Leftrightarrow A \notin A$. Звідси $S \in S \Leftrightarrow S \notin S$! Отримали суперечність.

Б. Расселу належить така популярна форма його парадоксу – *парадокс цирульника*. В одному містечку цирульник мусить голити всіх тих і тільки тих чоловіків містечка, хто не голиться сам. Чи мусить цирульник голити себе?

Наведемо ще приклади парадоксів.

Парадокс Беррі. Деякі речення є визначеннями натуральних чисел. Наприклад, "парне просте число", "сто в двадцятій степені" тощо. Український алфавіт скінченний, тому лише скінченна кількість натуральних чисел може бути визначена реченнями української мови, що мають менше ста букв. Множина натуральних чисел нескінченна, тому є натуральні числа (причому їх множина нескінченна), які не можуть бути визначені реченнями української мови, що мають менше ста букв. Серед таких чисел є найменше. Його можна визначити реченням "*Найменше натуральне число, яке не можна визначити реченням української мови, що має менше ста букв"*. Але таке речення має менше ста букв! (98 букв, якщо враховувати пробіли й кому).

Парадокс Греллінга. Деякі прикметники мають ту саму властивість, яку називають. Наприклад, прикметники "багатоскладовий", "український". Назвемо такі прикметники автологічними. Проте більшість прикметників не мають тієї властивості, яку називають. Наприклад, прикметники "колючий", "солоний", "зелений". Назвемо їх гетерологічними. Яким є прикметник "гетерологічний", він автологічний чи гетерологічний?

Відкриття парадоксів дуже збентежило математиків і змусило їх шукати вихід із кризи. Голландський математик *Л. Брауер* (1881–1966) висунув *інтуїціоністську програму*, в якій запропонував відмовитися від актуальної нескінченності та логічного закону виключеного третього, вважаючи допустимими в математиці тільки конструктивні доведення.

Інший шлях запропонував славетний німецький математик *Д. Гільберт* (1862–1943), який виступив з програмою обґрунтування математики на базі математичної логіки. Програма Гільберта передбачала побудову формально-аксіоматичних моделей (формальних систем) основних розділів математики та подальше доведення їх несуперечливості надійними, інтуїтивно переконливими засобами, які Гільберт назвав фінітними. Несуперечливість означає неможливість одночасного виведення деякого твердження та його заперечення.

Таким чином, математична теорія, несуперечливість якої хочемо довести, стає предметом вивчення певної математичної науки, яку Гільберт назвав метаматематикою, або теорією доведень. Саме з розроблення Д. Гільбертом та його учнями теорії доведень на базі розвинутої в роботах Г. Фреге та Б. Рассела логічної мови починається становлення математичної логіки як самостійної математичної дисципліни.

На жаль, програма Гільберта не може бути реалізована сповна. Це випливає із знаменитих теорем *К. Гьоделя* (1906–1978) про неповноту, які засвідчують принципову обмеженість формально-аксіоматичного методу. Як показав Гьодель, кожна достатньо багата несуперечлива формальна теорія неповна, тобто існують записані мовою теорії твердження, які не можна ні довести, ні спростувати. При цьому несуперечливість такої теорії не може бути доведена засобами самої теорії. До таких теорій, як це не прикро, належить теорія натуральних чисел – формальна арифметика. Зазначимо, що існують досить переконливі доведення несуперечливості багатьох теорій, зокрема, формальної арифметики, хоча такі доведення із необхідністю здійснюються засобами, що не можуть бути формалізовані в межах самих теорій.

Одним із центральних у математичній логіці є поняття числення. Воно відображає і узагальнює інтуїтивну уяву про індуктивне породження об'єктів, яке поширене в математиці. Зокрема, до числень відносять формально-аксіоматичні системи того чи іншого розділу математики.

Під *численням* розуміють скінченну множину точно визначених породжувальних правил, які дають змогу із певних заданих об'єктів отримувати інші об'єкти. Породжувальні правила називають також *правилами виведення* (ПВ). Об'єкти, до яких застосовуються правила виведення, називають *засновками*. Отриманий із засновків об'єкт називають *висновком*.

Множину породжених численням об'єктів задають індуктивно. На першому кроці процесу породження (виведення) початкові об'єкти задаються ПВ із порожньою множиною засновків. Об'єкт вважається породженим на певному кроці, якщо він отримується за допомогою певного ПВ із об'єктів, породжених на попередніх кроках.

Якщо на першому кроці процесу породження дозволити брати початкові об'єкти із певної множини A, дістанемо числення з входом. Таке числення $\mathfrak{I}$ перетворює множину A на множину об'єктів B, породжених із об'єктів множини A за допомогою числення $\mathfrak{I}$.

Найвизначнішими досягненнями математичної логіки є розроблення і дослідження формальних моделей алгоритму та алгоритмічно обчислюваної функції.

Поняття алгоритму належить до первісних понять математики, таких як множина чи натуральне число. Обчислювальні процеси алгоритмічного характеру відомі людству з глибокої давнини. Проте в явному вигляді поняття алгоритму сформувалося лише на початку XX ст.

Під *алгоритмом* розуміють скінченну множину точно визначених правил для суто механічного розв'язання задач певного класу. Таке формулювання можна розглядати тільки як пояснення, а не визначення, тому що поняття алгоритму в силу його первісності не можна виразити через інші поняття математики. Для подальшого уточнення цього поняття зазвичай вказують [21, 27, 29] такі його характерні властивості: фінітність, масовість, дискретність, елементарність, детермінованість, результативність.

Зауважимо, що правила алгоритму наказові, вони однозначно *визначають* перехід від одних об'єктів до інших, тоді як правила числення *дозволяють* робити такі переходи.

Для опису алгоритму потрібно вказати множину його *початкових*, або *вхідних* даних, та множину даних, до яких належать результати роботи алгоритму, або *вихідних* даних.

За допомогою алгоритму кожний конкретний результат отримується за скінченну кількість кроків із скінченної множини вхідних даних. У цьому випадку кажуть, що до таких даних алгоритм застосовний. Проте в деяких ситуаціях процес виконання алгоритму для певних вхідних даних продовжується необмежено. Тоді вважають, що до таких даних алгоритм незастосовний.

Той факт, що алгоритм $\aleph$ видає результат b при роботі над вхідним даним d, позначимо $b = \aleph(d)$.

Кожний алгоритм $\aleph$ із множиною вхідних даних X та множиною вихідних даних Y визначає функцію $f: X \to Y$, взагалі кажучи, часткову.

Якщо алгоритм $\aleph$ застосовний до d, то значення $f(d)$ дорівнює $\aleph(d)$. Якщо алгоритм $\aleph$ незастосовний до d, то $f(d)$ невизначене. В цьому випадку кажуть, що алгоритм $\aleph$ обчислює функцію f.

Функція *алгоритмічно обчислювана* (АОФ), якщо існує алгоритм, який її обчислює.

Множина L *алгоритмічно перелічна*, якщо L є множиною значень деякої АОФ, тобто існує алгоритм, який перелічує елементи множини L і тільки їх.

Множина L *алгоритмічно розв'язна* відносно множини U, якщо існує алгоритм, який дозволяє для кожного $x \in U$ визначати, $x \in L$ чи $x \notin L$.

Узагальненням поняття алгоритму є поняття *відносного алгоритму*, або *алгоритму з оракулом*. На деяких етапах такий алгоритм може звертатися до певного зовнішнього щодо алгоритму об'єкта – оракула. Видані оракулом відповіді трактуються як дані, вироблені на таких кроках звертання.

Функція *алгоритмічно обчислювана відносно оракула* $\wp$, якщо існує алгоритм з оракулом $\wp$, який її обчислює.

Кожний алгоритм можна трактувати як числення із входом, яке має такі ПВ, що виконання кожного із них відповідає виконанню одного кроку алгоритму. Отже, з одного боку, поняття алгоритму можна звести до поняття числення в розумінні зведення алгоритмічного процесу до процесу породження. З іншого боку, поняття числення можна звести до поняття алгоритму в розумінні зведення розгалуженого процесу породження до послідовного процесу переліку так, щоб алгоритм переліку відтворив усі породжені численням об'єкти і тільки їх.

Теорія алгоритмів як окремий розділ математики, що вивчає загальні властивості алгоритмів, виникла в межах математичної логіки в 30-х роках XX ст. Необхідність уточнення поняття алгоритму стала неминучою після усвідомлення неможливості існування алгоритмів розв'язання низки масових проблем, в першу чергу, пов'язаних з арифметикою та математичною логікою.

Для доведення неіснування алгоритму потрібно мати його точне математичне визначення, тому після сформування поняття алгоритму як нової та окремої сутності першочерговою стала проблема знаходження адекватних формальних моделей алгоритму та дослідження їх властивостей. Такі моделі були запропоновані як для первісного поняття алгоритму (машини Тьюрінга, регістрові машини, нормальні алгоритми Маркова та ін.), так і для похідного поняття АОФ (λ-означувані функції, частково рекурсивні функції та ін.). Доведено, що кожна з цих моделей задає (з точністю до кодування) один і той самий клас функцій. Тому є всі підстави вважати, що кожна з таких моделей дає строге математичне уточнення інтуїтивного поняття АОФ. Таке твердження стосовно АОФ та точно визначеного

класу частково рекурсивних функцій вперше сформулював у 1936 р. А. Чорч (теза Чорча).

Запропоновану Г. Лейбніцем ідею створення універсального логічного числення було втілено в теорії доведень. *Теорія доведень* – розділ математичної логіки, який вивчає поняття доведення в математиці та його застосування. Величезний внесок в створення і розвиток теорії доведень зробили Д. Гільберт та учні його школи – Г. Генцен, В. Аккерман, П. Бернайс та ін. Вже в середині 30-х років XX ст. засобами щойно створеної теорії алгоритмів було доведено алгоритмічну нерозв'язність проблеми всюди істинності формул логіки предикатів 1-го порядку (А. Чорч, А. Тьюрінг). Це робить в принципі неможливим існування універсальної процедури пошуку доведень. Хоча універсального алгоритму пошуку доведень немає і бути не може, існують алгоритми, які дають змогу знайти доведення формули, якщо ця формула всюди істинна. Якщо ж формула не всюди істинна, такі алгоритми можуть роботу не завершувати. Враховуючи результати А. Чорча та А. Тьюрінга про алгоритмічну нерозв'язність проблеми всюди істинності, сподіватись на краще немає підстав.

Автоматизація пошуку доведень теорем, без сумніву, належить до найважливіших застосувань математичної логіки. Ефективне знаходження доведень конче необхідне для успішного розв'язання низки задач, що виникають у сучасних інтелектуальних інформаційних системах. Такими є, зокрема, задачі подання знань і роботи з ними в базах даних і базах знань, задачі логічного програмування і дедуктивних баз даних.

Найважливіші методи пошуку доведень: метод семантичних таблиць, формальною основою якого є створене Г. Генценом секвенційне числення; запропонований Г. Генценом метод натурального виведення (системи натурального виведення дуже подібні до секвенційних систем, але "крупноблочніші") та запропонований у 1969 р. Дж. А. Робінсоном метод резолюцій.

Секвенційними численнями називають формально-аксіоматичні системи, які формалізують відношення логічного наслідку між двома множинами формул. Вони були запропоновані видатним німецьким логіком Г. Генценом (1909–1945) в 1934 р. Виведення в секвенційних численнях має вигляд дерева, вершинами якого є секвенції.

Поняття і методи математичної логіки засвідчують високу ефективність при моделюванні різноманітних предметних областей і програмних систем. При цьому традиційно використовується класична

логіка предикатів. Така логіка детально досліджена і має широке застосування. Класична логіка є основою низки спеціальних логік (модальних, темпоральних, епістемічних, релевантних тощо), для неї збудовано багато систем автоматизованого доведення. Але, незважаючи на всі її позитивні якості, вона не дає змоги адекватно виразити потреби моделювання та програмування.

Класична логіка має низку обмежень, які ускладнюють її застосування в моделюванні та програмуванні. В першу чергу, це превалювання синтаксичних, формальних аспектів, тоді як для моделювання та програмування набагато важливішими є семантичні, смислові аспекти. В класичній логіці функції та предикати трактуються як тотальні скінченно-арні відображення, а в програмуванні та при моделюванні використовуються набагато потужніші класи часткових функцій та предикатів над іменними (номінативними) даними. Класична логіка недостатньо враховує структурованість, неповноту, частковість інформації про предметну область, її динаміку.

Необхідність посилення можливостей класичної логіки для розв'язання задач моделювання та програмування стала передумовою виникнення композиційно-номінативних логік (КНЛ). Зазначені логіки базуються на загальних класах часткових відображень, заданих на довільних наборах іменованих значень. Такі відображення названі *квазіарними*.

Розбудову КНЛ природно здійснювати на основі єдиного для логіки та програмування композиційно-номінативного підходу [10]. Цей підхід базується на принципах композиційності та номінативності, він спирається на фундаментальний загальнометодологічний принцип розвитку як сходження від абстрактного до конкретного. Принцип композиційності трактує засоби побудови функцій та предикатів як алгебраїчні операції. Для логіки це означає зведення логічних зв'язок та кванторів до композицій предикатів. Принцип номінативності свідчить про необхідність використання відношень іменування для побудови семантичних моделей та опису предикатів.

Застосування композиційно-номінативного підходу дає змогу побудувати низку логічних моделей різноманітних предметних областей, що перебувають на різних рівнях абстрактності та загальності.

Серед КНЛ виділяються логіки, які є найближчими до класичних логік і зберігають їх основні властивості за істотного розширення класу семантичних моделей. Такими є логіки еквітонних квазіарних предикатів, які природно назвати неокласичними. *Еквітонність* оз-

начає, що значення відображення не змінюється при розширенні даних. Неформально це означає незмінність вже встановленого знання. Неокласичні логіки, з одного боку, дають змогу використовувати як теоретичні результати, так і багатий досвід застосування класичної логіки, а з іншого – вони більше адаптовані до потреб моделювання і програмування.

Традиційні логіки описують конкретний стан тверджень про світ, вони не зовсім адекватні світові, що змінюється та розвивається. Тому для адекватнішого опису такого світу видається доцільним використовувати модальні логіки. Виняткова гнучкість цих логік дає можливість застосувати їх для аналізу та моделювання найрізноманітніших апектів діяльності людини. В першу чергу, це стосується створення інтелектуальних інформаційних систем, зокрема систем знань, експертних систем, задач опису та моделювання складних динамічних систем.

Спектр модальних логік дуже широкий. Передусім серед них вирізняються традиційні, або алетичні, модальні логіки, а також темпоральні (часові), деонтичні, епістемічні. При поєднанні можливостей неокласичних та модальних логік дістаємо композиційно-номінативні модальні логіки, зокрема транзиційні та темпоральні композиційно-номінативні модальні логіки.

Розвиток інформаційних технологій та програмування зумовлює невпинне розширення сфери застосування математичної логіки. Стосовно загальносвітоглядного аспекту поняття і методи математичної логіки дають обґрунтування правильності тих чи інших способів отримання істинного знання. Щодо прагматичного аспекту апарат математичної логіки належить до основних засобів моделювання різноманітних предметних областей, він є основою, ядром сучасних інформаційних систем. З кожним роком зростає глибоке проникнення ідей та методів математичної логіки в інформатику, обчислювальну математику, лінгвістику, філософію.

1. ОСНОВНІ ПОНЯТТЯ ЛОГІКИ

1.1. Основні закони традиційної логіки

Класична математична логіка характеризується мінімальністю використовуваних понять та поширенням формалізації на найзагальнішу область застосовності. Вона спирається на чотири основні закони традиційної логіки. Три із них відомі ще з часів Арістотеля. Це закон тотожності, закон несуперечливості та закон виключеного третього. Четвертий закон – закон достатньої підстави – сформулював Г. Лейбніц.

Закон тотожності в загальному вигляді можна сформулювати так: *"У процесі одного і того ж міркування використовувані поняття не повинні змінюватись*".

У спрощеному вигляді цей закон формулюється таким чином: *А суть А*.

Звідси отримуємо дуже просте математичне формулювання закону тотожності у вигляді формули $A \leftrightarrow A$.

Звичайно, формула $A \leftrightarrow A$ створює ілюзію тривіальності закону тотожності, але цього ніяк не скажеш про наведене вище його загальне формулювання, яке засвідчує фундаментальну роль закону тотожності для організації мислення людини. Саме підміна значень слів найчастіше використовується для побудови софізмів.

Закон несуперечливості: *"Обидва твердження А та $\neg A$ не можуть виконуватися одночасно"*.

В такому вигляді це певне спрощення оригінального формулювання Арістотеля. Інколи закон несуперечливості називають законом суперечливості (*lex contradictionis*).

Математичне формулювання закону несуперечливості має вигляд формули $\neg(A \,\&\, \neg A)$.

Зрозуміло, що в цьому законі, як підкреслював ще Арістотель, обидва твердження A та $\neg A$ повинні розглядатися одночасно в одному і тому ж контексті. Справді, один і той же об'єкт в один і той же час в одному і тому ж місці не може мати взаємовиключних властивостей. Наприклад, конкретна особа в даний час не може мати більше 20 років і менше 20 років.

Закон виключеного третього у формулюванні Арістотеля має такий вигляд: "*Обидва твердження A та $\neg A$ не можуть заперечуватися одночасно*".

Арістотель також зауважував, що далеко не до всіх висловлень цей закон застосовний. Як приклад він наводив твердження "*Завтра буде морський бій*". Що можна сказати про його істинність сьогодні? Зрозуміло, що жодне з цих тверджень сьогодні не є ані істинним, ані хибним.

Дослідженням подібних тверджень, що не можна трактувати тільки як істинні чи хибні, в яких міститься певна оцінка міри їх істинності, займаються *модальні* логіки. Основи модальних логік були закладені ще античними філософами (в першу чергу Арістотелем, Діодором Кроносом).

Сильною формою закону виключеного третього (яку наводив ще Арістотель) є така: "*Одне з тверджень A чи $\neg A$ істинне*". Цю форму називають *tertium non datur*. Звідси отримуємо математичне формулювання закону виключеного третього у вигляді формули $A \vee \neg A$.

Закон достатньої підстави "молодший" на 2 тис. років за перші три закони традиційної логіки. Його сформулював видатний німецький філософ і математик Г. Лейбніц: "*Жодне твердження не може прийматися без достатніх підстав*".

Це означає, що жодне твердження не може визнаватися істинним, якщо воно не є наслідком раніше прийнятих тверджень або чітко встановлених фактів. Прийняття цього закону відокремлює логіку точних наук від змістовної, житейської логіки. На відміну від перших трьох логічних законів закон достатньої підстави не має математичного формулювання у вигляді формули.

1.2. Основні визначення та позначення

Вважатимемо відомими базові математичні поняття множини, відношення, відображення, декартового добутку та ін. Введемо необхідні поняття та позначення.

Множини *натуральних чисел*, *цілих чисел*, *раціональних чисел* та *дійсних чисел* будемо позначати відповідно N, Z, Q та R.

Множину всіх підмножин довільної множини A позначимо 2^A.

Нехай A та R – довільні множини, f – довільна однозначна часткова функція вигляду f: $A \to R$.

Якщо значення $f(a)$ визначене, то пишемо $f(a)\downarrow$. Якщо $f(a)$ невизначене, то пишемо $f(a)\uparrow$. Якщо $f(a)$ визначене та дорівнює b, то пишемо $f(a)\downarrow = b$ або $f(a)\downarrow b$.

Для довільних функцій f та g пишемо $f(a) \cong g(b)$, якщо з $f(a)\downarrow$ та $g(b)\downarrow$ випливає $f(a) = g(b)$.

Функція f: $A \to R$ *є тотальною,* якщо $f(a)\downarrow$ для всіх $a \in A$.

З кожною f: $A \to R$ зв'яжемо множини D_f та E_f, де $D_f \subseteq A$ та $E_f \subseteq R$. Вважатимемо, що для кожного $a \in D_f$ виконується умова $f(a)\downarrow$, для кожного $a \in A \setminus D_f$ виконується умова $f(a)\uparrow$, та що умова $f(a)\downarrow$ означає $f(a) \in E_f$.

Множини D_f та E_f називатимемо відповідно *множиною визначення* та *множиною значень* функції f.

Пару (D_f, E_f) назвемо *типом* функції.

Множину $\{(a, f(a)) \mid a \in D_f\}$ назвемо *графіком* функції f.

Основним поняттям логіки із семантичного погляду є поняття предикату. З цим поняттям тісно пов'язане поняття висловлення.

Під *висловленням* розуміють речення, яке можна розглядати з точки зору його істинності чи хибності. *Суб'єкт* (*суб'єкти*) – те, про кого або про що йдеться у висловленні. *Предикат* виражає властивості суб'єкта (суб'єктів) та *відношення* між суб'єктами.

Висловлення може набувати одне з двох істиннісних значень – істина або фальш, тому із семантичного погляду предикат можна уточнити як функцію, що конкретним іменованим суб'єктам ставить у відповідність значення T або F.

Предикат стає висловленням, якщо вказати значення імен його суб'єктів. Наприклад, предикат "x є простим числом" стає висловленням "5 є простим числом", яке істинне, якщо значенням імені x є число 5. Той самий предикат стає висловленням "4 є простим числом", яке хибне, якщо значенням імені x є число 4.

Називаючи *даним* множину пар імен суб'єктів та їх значень, дістаємо, що *висловлення є значенням предиката на конкретному даному.* Предикат стає висловленням при фіксуванні конкретних значень імен даного, до якого предикат застосовується.

Наприклад, застосувавши предикат "якщо $x>y$ та $y>z$, то $x>z$" до даного $[x \mapsto 17, y \mapsto 11, z \mapsto 3]$, дістанемо висловлення "якщо 17>11 та 11>3, то 17>3".

Поняття предикату та висловлення як математичні поняття можна розглядати тоді, коли виділено спеціальні істиннісні значення T та F, що інтерпретуються як "істина" та "фальш".

У загальному випадку під *предикатом* на множині A розуміють довільну часткову функцію вигляду P: $A \to Bool$, де $Bool = \{T, F\}$. Іншими словами, предикати на множині A – це функції типу $(A, Bool)$.

Для довільних предикатів P та Q введемо такі позначення.

$P(a) \Rightarrow Q(b)$, якщо з $P(a)\downarrow = T$ та $Q(b)\downarrow$ випливає $Q(b) = T$.

$P \Rightarrow Q$, якщо $P(d) \Rightarrow Q(d)$ для довільних $d \in A$

Предикат P на множині A назвемо (частково) *істинним,* або *неспростовним*, якщо для довільних $a \in A$ із умови $P(a)\downarrow$ випливає $P(a) = T$.

Предикат P на множині A назвемо *строго істинним,* якщо $P(a) = T$ для довільних $a \in A$.

Для тотальних предикатів поняття істинності та неспростовності збігаються.

Предикат P на множині A назвемо *виконуваним*, якщо існує $a \in A$ таке, що $P(a)\downarrow = T$.

Областю істинності та *областю хибності* довільного предикату P на множині A назвемо множини $\boldsymbol{I_P} = \{d \in A \mid P(d) = T\}$ та $\boldsymbol{F_P} = \{d \in A \mid P(d) = F\}$. Якщо P тотальний, то $\boldsymbol{I_P} \cup \boldsymbol{F_P} = A$.

Нехай Fn – деяка множина функцій. Довільну функцію вигляду $Fn^n \to Fn$ назвемо *n-арною композицією*, або *n-арною операцією* на множині функцій Fn.

Під *формальною системою* (ФС) будемо розуміти трійку (L, A, P), де L – мова, A – множина *аксіом*, P – множина *правил виведення*.

Мова задається *алфавітом* та правилами побудови слів мови, які називаються *формулами*. Кожна аксіома є формулою.

Правила виведення (ПВ) діють на множині формул. Записуємо правила виведення у вигляді $P_1, P_2, ..., P_n \mid\!- P$, де $P_1, P_2, ..., P_n$ – засновки, P – висновок.

Формулу, отриману із аксіом за допомогою ПВ, назвемо *теоремою*.

Виведенням називають скінченну послідовність формул $\Phi_1, ..., \Phi_m$, в якій кожна із формул або є аксіомою, або отримана із попередніх формул за допомогою деякого правила виведення.

Питання для самоконтролю

1. Що вивчає математична логіка?
2. Що таке софізм? Наведіть приклади софізмів.
3. У чому сутність аксіоматичного методу?
4. Що таке парадокс? Наведіть приклади парадоксів.
5. Що таке числення? Наведіть приклади числень.
6. Що таке алгоритм? Вкажіть характерні властивості алгоритмів.
7. Що таке алгоритмічно обчислювана функція?

8. Що таке алгоритмічно перелічна множина? Алгоритмічно розв'язна множина?
9. Сформулюйте основні закони традиційної логіки.
10. Наведіть математичні формулювання перших трьох основних законів традиційної логіки.
11. Дайте визначення функції.
12. Дайте визначення тотальної функції.
13. Що таке множина визначення та множина значень функції?
14. Що таке графік функції?
15. Що таке висловлення? Наведіть приклади висловлень.
16. Що таке предикат? Наведіть приклади предикатів.
17. Який зв'язок висловлень та предикатів?
18. Дайте визначення предиката як математичного поняття.
19. Дайте визначення істинного предиката.
20. Дайте визначення виконуваного предиката.
21. Чи може бути (частково) істинний предикат невиконуваним?
22. Що таке область істинності та область хибності предиката?
23. Що таке n-арна композиція? Наведіть приклади композицій.
24. Що таке формальна система?
25. Що таке правило виведення? Як записуємо правила виведення?
26. Що таке теорема?
27. Що таке виведення?

Вправи

1. Проаналізуйте наступні міркування. Якщо вони некоректні, то які закони логіки порушують?

1) Якщо ти їси менше, то голоднішаєш. Якщо ти голоднішаєш, то їси більше. Отже, якщо ти їси менше, то їси більше [32].
2) Той, хто хоче щось вивчити, цього не знає. Незнаючий – невіглас. Отже, тільки невігласи хочуть вчитись [32].
3) Студенти, які хочуть вчитися, вчаться і без заохочення. Студентів, які не хочуть вчитися, заохочувати марно. Отже, студентів заохочувати не потрібно [32].
4) Сіль та цукор білі. Ніщо не може бути одночасно цукром і сіллю. Отже, ніщо не може бути білим [6].
5) Злодій не хоче брати щось погане. Надбання доброго – добра справа. Отже, злодій робить добру справу (*античний софізм*).

6) Вчитель завжди хоче, щоб його учень став мудрим і перестав бути невігласом. Таким чином, він хоче, щоб його учень став тим, ким він не є, та перестав бути тим, ким він є. Отже, він хоче перевести його із буття в небуття, тобто знищити (*античний софізм*).

7) Якщо тобі холодно, ти тепло вдягаєшся. Якщо тепло вдягнутися, то стане гаряче. Якщо тобі гаряче, ти роздягаєшся. Отже, якщо тобі холодно, ти роздягаєшся.

2. Покажіть, що кожна алгоритмічно розв'язна множина є алгоритмічно перелічною.

3. Доведіть, що формула Ф є теоремою $\Leftrightarrow$ Ф має виведення.

2. ПРОПОЗИЦІЙНА ЛОГІКА

На *пропозиційному* рівні розгляду ми не проникаємо у внутрішню, суб'єктно-предикатну структуру об'єктів дослідження (висловлень чи предикатів). Предикати розглядаються як функції вигляду $P: A \to \{T, F\}$, де A – абстрактна множина, тобто її елементи неструктуровані. На такому рівні засобом утворення складніших висловів чи предикатів із простіших є *логічні операції* (композиції), які не враховують стуктурованості даних – *пропозиційні композиції*, або *логічні зв'язки*.

Основними логічними зв'язками є заперечення $\neg$, диз'юнкція $\vee$, кон'юнкція **&**, імплікація $\to$, еквіваленція $\leftrightarrow$, роздільна диз'юнкція $\oplus$. Такі зв'язки відповідають певним зворотам природної мови:

- *заперечення* відповідає зворотам "не …", "невірно, що …";
- *диз'юнкція* – звороту "… або …";
- *кон'юнкція* – зворотам "… та …", "… і …";
- *імплікація* – зворотам "якщо … то …", "із ... випливає ...";
- *еквіваленція* – зворотам "... тоді і тільки тоді, коли...", "… рівносильне …", "… еквівалентне …";
- *роздільна диз'юнкція* – звороту "або …, або …".

Але природна мова неоднозначна. Наприклад, слово "коса" має принаймі три різних значення. Те саме можна сказати про речення "Микола зустрів Галю на лузі з квітами". Не кращі справи із зворотом "… або …".

Наприклад, розглянемо такі речення:

"Чергові збори відбудуться завтра або післязавтра."

"Трикутник ABC прямокутний або рівнобедрений."

"AB = BC, або трикутник ABC рівнобедрений".

У першому реченні "або" трактується як роздільна диз'юнкція, в другому – як нероздільна диз'юнкція, в третьому – еквіваленція. Така неоднозначність недоречна при побудові математичної теорії. Тому для глибокого і точного дослідження предикатів і висловлень потрібно ввести строгу логіко-математичну мову.

Для дослідження логічної структури предикатів і висловлень на пропозиційному рівні вводимо *мову логіки висловлень*, або *мову пропозиційної логіки* (мова ПЛ). Множину базових пропозиційних компо-

зицій можна задавати різними способами, головне, щоб така множина визначала повний клас усіх пропозиційних композицій (див. теорему Поста про функціональну повноту [5]).

У цьому посібнику виберемо множину базових композицій (логічних зв'язок) $\{\neg, \vee\}$. Поширені також варіанти мови ПЛ із множинами базових композицій $\{\neg, \to\}$, $\{\neg, \mathbf{\&}\}$, $\{\neg, \vee, \mathbf{\&}\}$ та $\{\neg, \vee, \mathbf{\&}, \to\}$.

У разі зафіксованої множини базових композицій мови ПЛ можуть різнитися способами запису основних об'єктів мови – пропозиційних формул. Будемо тут використовувати *префіксну* форму запису пропозиційних формул, коли символ логічної операції (композиції) передує аргументам. Таку форму запису формул запропонував видатний польський логік Я. Лукасєвіч (1878–1956), тому її також називають *польською* формою.

2.1. Композиції пропозиційного рівня

Як зазначалося вище, традиційними логічними зв'язками є такі: 1-арна композиція *заперечення* $\neg$ та бінарні композиції *диз'юнкція* $\vee$, *кон'юнкція* $\mathbf{\&}$, *імплікація* $\to$, *еквіваленція* $\leftrightarrow$, *роздільна диз'юнкція* $\oplus$.

При визначенні логічних зв'язок (композицій) $\neg$, $\vee$, $\to$, $\mathbf{\&}$, $\leftrightarrow$, $\oplus$ в загальному випадку варто враховувати *частковість* предикатів. У цьому плані наші визначення узагальнюють визначення класичних логічних зв'язок для тотальних предикатів та висловлень.

Предикати $\neg(P)$, $\vee(P, Q)$ $\to(P, Q)$, $\mathbf{\&}(P, Q)$, $\leftrightarrow(P, Q)$, $\oplus(P, Q)$ звичайно позначатимемо $\neg P$, $P \vee Q$, $P \to Q$, $P \mathbf{\&} Q$, $P \leftrightarrow Q$, $P \oplus Q$.

Зазначені предикати задамо так:

$$(\neg P)(d) = \begin{cases} T, \text{ якщо } P(d)\downarrow = F, \\ F, \text{ якщо } P(d)\downarrow = T, \\ \text{невизначене, якщо } P(d)\uparrow. \end{cases}$$

$$(P \vee Q)(d) = \begin{cases} T, \text{ якщо } P(d) = T \text{ або } Q(d) = T, \\ F, \text{ якщо } P(d) = F \text{ та } Q(d) = F, \\ \text{невизначене в усіх інших випадках.} \end{cases}$$

$$(P\to Q)(d)=\begin{cases}T, \text{якщо } P(d)=F \text{ або } Q(d)=T,\\ F, \text{якщо } P(d)=T \text{ та } Q(d)=F,\\ \text{невизначене в усіх інших випадках.}\end{cases}$$

$$(P\&Q)(d)=\begin{cases}T, \text{якщо } P(d)=T \text{ та } Q(d)=T,\\ F, \text{якщо } P(d)=F \text{ або } Q(d)=F,\\ \text{невизначене в усіх інших випадках.}\end{cases}$$

$$(P\leftrightarrow Q)(d)=\begin{cases}T, \text{якщо } P(d)\downarrow,\ Q(d)\downarrow \text{ та } P(d)=Q(d),\\ F, \text{якщо } P(d)\downarrow,\ Q(d)\downarrow \text{ та } P(d)\neq Q(d),\\ \text{невизначене в усіх інших випадках.}\end{cases}$$

$$(P\oplus Q)(d)=\begin{cases}T, \text{якщо } P(d)\downarrow,\ Q(d)\downarrow \text{ та } P(d)\neq Q(d),\\ F, \text{якщо } P(d)\downarrow,\ Q(d)\downarrow \text{ та } P(d)=Q(d),\\ \text{невизначене в усіх інших випадках.}\end{cases}$$

Визначені нами логічні зв'язки називають *Клінієвими*, оскільки їх увів видатний американський логік С. Кліні [22]. Вперше логічні зв'язки, що враховують частковість предикатів, при створенні тризначної логіки запропонував Я. Лукасєвіч. Проте в Лукасєвіча дещо інакше визначені імплікація та еквіваленція, що робить їх нееквітонними.

Розглянемо основні властивості логічних зв'язок.

1) комутативність $\vee$ та **&**:

$$P\vee Q = Q\vee P,$$

$$P\&Q = Q\&P;$$

2) асоціативність $\vee$ та **&**:

$$(P\vee Q)\vee R = P\vee(Q\vee R),$$

$$(P\&Q)\&R = P\&(Q\&R);$$

3) дистрибутивність $\vee$ відносно **&** та **&** відносно $\vee$:

$$(P\vee Q)\&R = (P\&R)\vee(Q\&R),$$

$$(P\&Q)\vee R = (P\vee R)\&(Q\vee R);$$

4) ідемпотентність $\vee$ та **&**:

$$P \vee P = P,$$

$$P \,\&\, P = P;$$

5) закони поглинання:

$$P \,\&\, (P \vee Q) = P,$$

$$P \vee (P \,\&\, Q) = P;$$

6) зняття подвійного заперечення

$$\neg\neg P = P;$$

7) закони де Моргана:

$$\neg(P \vee Q) = (\neg P) \,\&\, (\neg Q),$$

$$\neg(P \,\&\, Q) = (\neg P) \vee (\neg Q);$$

8) закон контрапозиції:

$$(P \to Q) = ((\neg Q) \to (\neg P));$$

9) закон (правило) modus ponens:

$$P \,\&\, (P \to Q) \Rightarrow Q;$$

10) зведення $\to$, $\leftrightarrow$ та $\oplus$ до $\neg$, $\vee$ та **&**:

$$P \to Q = (\neg P) \vee Q,$$

$$P \leftrightarrow Q = (P \to Q) \,\&\, (Q \to P) = (P \,\&\, Q) \vee ((\neg P) \,\&\, (\neg Q)),$$

$$P \oplus Q = \neg(P \leftrightarrow Q) = (\neg(P \to Q)) \vee (\neg(Q \to P)) =$$

$$= (P \,\&\, (\neg Q)) \vee ((\neg P) \,\&\, Q);$$

11) комутативність $\leftrightarrow$ та $\oplus$:

$$P \leftrightarrow Q = Q \leftrightarrow P,$$

$$P \oplus Q = Q \oplus P;$$

12) асоціативність $\leftrightarrow$ та $\oplus$:

$$(P \leftrightarrow Q) \leftrightarrow R = P \leftrightarrow (Q \leftrightarrow R),$$

$$(P \oplus Q) \oplus R = P \oplus (Q \oplus R).$$

Композиції $\neg$ та $\vee$ назвемо *базовими пропозиційними композиціями.*

Композиції $\to$, **&**, $\leftrightarrow$ та $\oplus$ є *похідними*, вони виражаються, згідно з властивостями 7) і 9), через базові пропозиційні композиції $\neg$ та $\vee$.

2.2. Мова пропозиційної логіки

Мови ПЛ відрізняються наборами символів базових пропозиційних композицій (логічних зв'язок) і способами запису пропозиційних формул. Ми вибрали множину базових композицій $\{\neg, \vee\}$ та префіксну форму запису.

Алфавіт мови ПЛ складається із символів логічних зв'язок $\neg$ і $\vee$ та множини *Ps* пропозиційних символів. Множина *Ps*, як правило, нескінченна.

Правильно побудовані вирази мови ПЛ називають *пропозиційними формулами* (ПФ).

Дамо індуктивне визначення пропозиційної формули.

1) кожний $A \in Ps$ є ПФ; такі ПФ назвемо *атомарними*;

2) якщо Φ та Ψ є ПФ, то $\neg\Phi$ та $\vee\Phi\Psi$ є ПФ.

Множину всіх ПФ позначимо *Fp*.

Для бінарних операцій звичніше користуватися *інфіксною* формою, коли символ операції записується між аргументами, але тоді потрібні додаткові символи – дужки "(" і ")". Формули в інфіксній формі вважаємо скороченнями ПФ. Наприклад, формула $\neg(\Phi\vee\Psi)$ – скорочення "справжньої" ПФ $\neg\vee\Phi\Psi$.

Пропозиційні композиції $\&$, $\rightarrow$, $\leftrightarrow$ та $\oplus$ можна виразити через пропозиційні композиції $\neg$ та $\vee$. Тому вирази $\Phi\&\Psi$, $\Phi\rightarrow\Psi$, $\Phi\leftrightarrow\Psi$ та $\Phi\oplus\Psi$ вважатимемо скороченнями ПФ $\neg\vee\neg\Phi\neg\Psi$, $\vee\neg\Phi\Psi$, $\neg\vee\neg\vee\neg\Phi\Psi\neg\vee\neg\Psi\Phi$ та $\vee\neg\vee\neg\Phi\Psi\neg\vee\neg\Psi\Phi$.

Для зменшення кількості дужок задамо в порядку спадання такий пріоритет символів логічних зв'язок: $\neg$, $\&$, $\oplus$, $\vee$, $\rightarrow$, $\leftrightarrow$. Введемо також правило розставлення дужок справа наліво. Наприклад, $A\rightarrow B\rightarrow C\rightarrow\neg A\vee D$ означає $A\rightarrow(B\rightarrow(C\rightarrow((\neg A)\vee D)))$.

Скорочення ПФ надалі також називатимемо ПФ.

Розглянемо традиційну інтерпретацію ПФ на множині висловів за допомогою істиннісних оцінок.

Істиннісною оцінкою мови ПЛ назвемо довільний вираз $\tau : Ps \rightarrow \{T, F\}$. Істиннісна оцінка задає значення атомарних ПФ.

Для визначення значення неатомарних ПФ вираз $\tau : Ps \rightarrow \{T, F\}$ продовжимо до $\tau : Fp \rightarrow \{T, F\}$. Для цього покладемо

$$\tau(\neg\Phi) = T \Leftrightarrow \tau(\Phi) = F;$$

$$\tau(\vee\Phi\Psi) = T \Leftrightarrow \tau(\Phi) = T \textit{ або } \tau(\Psi) = T.$$

З одного боку, кожна ПФ з множиною пропозиційних імен *X* задає певну *X*-арну функцію на 2-елементній множині $\{T, F\}$, тобто *булеву*

функцію. З іншого боку, кожна X-арна булева функція задається деякою ПФ, причому для фіксованої булевої функції множина таких ПФ нескінченна. Зауважимо, що для булевих функцій замість T та F звичайно пишуть 1 та 0.

ПФ називають *тавтологією*, якщо вона має істиннісне значення T при кожній істиннісній оцінці мови ПЛ. Отже, ПФ є тавтологією, якщо вона істинна на кожному наборі значень її пропозиційних імен.

Кожна тавтологія задає тотальну булеву функцію, що набуває тільки значення 1, тобто константу 1.

Суперечністю називають ПФ, якщо вона має істиннісне значення F при кожній істиннісній оцінці мови ПЛ. Іншими словами, ПФ – суперечність, якщо вона хибна на кожному наборі значень її пропозиційних імен.

Кожна суперечність задає тотальну булеву функцію, що набуває тільки значення 0, тобто константу 0.

Зрозуміло, що Ф тавтологія $\Leftrightarrow$ Ф суперечність.

Тавтології називають також *законами пропозиційної логіки.*

Деякі тавтології мають власні назви.

Основоположні закони традиційної логіки виражають такі тавтології:

- закон тотожності: $P \leftrightarrow P$;
- закон виключеного третього: $(\neg P) \vee P$;
- закон несуперечливості: $\neg(P \& (\neg P))$.

До найважливіших законів пропозиційної логіки належать, зокрема, записані мовою ПЛ властивості пропозиційних композицій:

1) закони комутативності для $\vee$ та $\&$:

$$P \vee Q \leftrightarrow Q \vee P,$$

$$P \& Q \leftrightarrow Q \& P;$$

2) закони асоціативності $\vee$ та $\&$:

$$(P \vee Q) \vee R \leftrightarrow P \vee (Q \vee R),$$

$$(P \& Q) \& R \leftrightarrow P \& (Q \& R),$$

$$((P \leftrightarrow Q) \leftrightarrow R) \leftrightarrow (P \leftrightarrow (Q \leftrightarrow R));$$

3) закони дистрибутивності для $\vee$ та $\&$:

$$(P \vee Q) \& R \leftrightarrow (P \& R) \vee (Q \& R),$$

$$(P \& Q) \vee R \leftrightarrow (P \vee R) \& (Q \vee R);$$

4) закони ідемпотентності для $\vee$ та $\&$:

$$P \leftrightarrow P \vee P,$$
$$P \leftrightarrow P \& P;$$

5) закони поглинання:

$$P \& (P \vee Q) \leftrightarrow P,$$
$$P \vee (P \& Q) \leftrightarrow P;$$

6) закон зняття подвійного заперечення:

$$\neg\neg P \leftrightarrow P;$$

7) закони де Моргана:

$$\neg(P \vee Q) \leftrightarrow (\neg P) \& (\neg Q),$$
$$\neg(P \& Q) \leftrightarrow (\neg P) \vee (\neg Q);$$

8) закон контрапозиції:

$$(P \to Q) \leftrightarrow (\neg Q \to \neg P);$$

9) закони комутативності $\leftrightarrow$ та $\oplus$:

$$(P \leftrightarrow Q) \leftrightarrow (Q \leftrightarrow P),$$
$$(P \oplus Q) \leftrightarrow (Q \oplus P);$$

10) закони асоціативності $\leftrightarrow$ та $\oplus$:

$$(P \leftrightarrow Q) \leftrightarrow R = P \leftrightarrow (Q \leftrightarrow R),$$
$$(P \oplus Q) \oplus R = P \oplus (Q \oplus R).$$

Розглянемо властивості та відношення для множин пропозиційних формул.

Множина ПФ $\{\Phi_1, ..., \Phi_n\}$ *суперечлива*, якщо $\Phi_1 \& \ldots \& \Phi_n$ суперечність.

На множині ПФ введемо такі відношення:

- логічного (тавтологічного) наслідку $\models$;
- логічної (тавтологічної) еквівалентності $\sim_т$.

Формула Ψ є *логічним* (*тавтологічним*) *наслідком* формули Φ, що позначаємо $\Phi \models \Psi$, якщо формула $\Phi \to \Psi$ – тавтологія.

Пропозиційні формули Φ та Ψ *логічно* (*тавтологічно*) *еквівалентні,* що позначаємо $\Phi \sim_т \Psi$, якщо $\Phi \models \Psi$ та $\Psi \models \Phi$.

ПФ Ψ є *логічним* (*тавтологічним*) *наслідком множини ПФ* $\{\Phi_1, ..., \Phi_n\}$, що позначаємо $\{\Phi_1, ..., \Phi_n\} \models \Psi$, якщо $\Phi_1 \& \ldots \& \Phi_n \models \Psi$.

Замість $\varnothing \models \Phi$ писатимемо $\models \Phi$.

Основні властивості відношень $\models$ та $\sim_т$:

1) відношення $\models$ рефлексивне і транзитивне;

2) відношення $\sim_т$ рефлексивне, транзитивне і симетричне.

3) Φ тавтологія $\Leftrightarrow \models \Phi$;

4) $\Phi \sim_т \Psi \Leftrightarrow \models \Phi \leftrightarrow \Psi \Leftrightarrow \Phi \leftrightarrow \Psi$ тавтологія.

Поняття логічного наслідку поширимо на довільні множини формул.

Нехай $\Gamma \subseteq Fp$ та $\Delta \subseteq Fp$ – деякі множини формул. Надалі для спрощення запису замість $\{\Phi\} \cup \Gamma$ будемо звичайно писати Φ, Γ або Γ, Φ.

Говоримо, що *із* Γ *випливає* Δ, або Δ є *логічним наслідком* Γ, якщо для кожної істиннісної оцінки $\tau: Fp \to \{T, F\}$ із того, що $\tau(\Phi) = T$ для всіх $\Phi \in \Gamma$, випливає, що $\tau(\Psi) = T$ для деякої $\Psi \in \Delta$.

Те, що Δ є логічним наслідком Γ, будемо позначати $\Gamma \models \Delta$.

Отже, $\Gamma \not\models \Delta \Leftrightarrow$ існує істиннісна оцінка $\tau: Fp \to \{T, F\}$ така, що для всіх $\Phi \in \Gamma$ маємо $\tau(\Phi) = T$ та для всіх $\Psi \in \Delta$ маємо $\tau(\Psi) = F$.

Відношення логічного наслідку для множин формул рефлексивне, але нетранзитивне. Справді, очевидно $\Delta \models \Delta$, але із $\Gamma \models \Delta$ та $\Delta \models \Sigma$ не мусить випливати $\Gamma \models \Sigma$. Останнє засвідчує наступний приклад.

Приклад 2.2.1. Маємо $\{\Phi \vee \Psi \vee \neg\Phi\} \models \{\Phi \vee \Psi, \Psi \vee \neg\Phi\}$ та $\{\Phi \vee \Psi, \Psi \vee \neg\Phi\} \models \Psi$, але $\{\Phi \vee \Psi \vee \neg\Phi\} \not\models \Psi$.

Розглянемо властивості відношення $\models$ логічного наслідку для множин формул на пропозиційному рівні.

Особливо важливими є такі властивості:

G1) Якщо $\Gamma \cap \Delta \neq \varnothing$, то $\Gamma \models \Delta$.

G2) Нехай $\Gamma \models \Delta$ та $\Delta \subseteq \Sigma$. Тоді $\Gamma \models \Sigma$.

Інші властивості відношення $\models$ на пропозиційному рівні:

П1) $\neg\Phi, \Gamma \models \Delta \Leftrightarrow \Gamma \models \Delta, \Phi$.

П2) $\Gamma \models \Delta, \neg\Phi \Leftrightarrow \Phi, \Gamma \models \Delta$.

П3) $\Phi \vee \Psi, \Gamma \models \Delta \Leftrightarrow \Phi, \Gamma \models \Delta$ та $\Psi, \Gamma \models \Delta$.

П4) $\Gamma \models \Delta, \Phi \vee \Psi \Leftrightarrow \Gamma \models \Delta, \Phi, \Psi$.

П5) $\Phi \& \Psi, \Gamma \models \Delta \Leftrightarrow \Phi, \Psi, \Gamma \models \Delta$.

П6) $\Gamma \models \Delta, \Phi \& \Psi \Leftrightarrow \Gamma \models \Delta, \Phi$ та $\Gamma \models \Delta, \Psi$.

П7) $\Phi \to \Psi, \Gamma \models \Delta \Leftrightarrow \Gamma \models \Delta, \Phi$ та $\Psi, \Gamma \models \Delta$.

П8) $\Gamma \models \Delta, \Phi \to \Psi \Leftrightarrow \Phi, \Gamma \models \Delta, \Psi$.

П9) $\Phi \leftrightarrow \Psi, \Gamma \models \Delta \Leftrightarrow \Phi, \Psi, \Gamma \models \Delta$ та $\Gamma \models \Delta, \Phi, \Psi$.

П10) $\Gamma \models \Delta, \Phi \leftrightarrow \Psi \Leftrightarrow \Phi, \Gamma \models \Delta, \Psi$ та $\Psi, \Gamma \models \Delta, \Phi$.

Для доведення цих властивостей використовуються визначення пропозиційних композицій та визначення відношення $\models$.

Властивості П1–П4, в яких фігурують тільки базові композиції, назвемо базовими властивостями відношення $\models$ на пропозиційному рівні.

Неважко переконатись, що для пропозиційної логіки справджуються важливі теореми еквівалентності.

Теорема 2.2.1 (*семантичної еквівалентності*). *Нехай* Φ' *отримана із формули* Φ *заміною деяких входжень формул* $\Phi_1, ..., \Phi_n$ *на* $\Psi_1, ..., \Psi_n$ *відповідно. Якщо* $\Phi_1 \sim_т \Psi_1, ..., \Phi_n \sim_т \Psi_n$, *то* $\Phi \sim_т \Phi'$.

Теорема доводиться індукцією за побудовою формули.

Теорема 2.2.2 (*заміни еквівалентних*). *Нехай* $\Phi \sim \Psi$. *Тоді*
$\Phi, \Gamma \models \Delta \Leftrightarrow \Psi, \Gamma \models \Delta$ *та* $\Gamma \models \Delta, \Phi \Leftrightarrow \Gamma \models \Delta, \Psi$.

2.3. Пропозиційне числення

Розглянемо аксіоматичні системи для пропозиційної логіки. Системи, які розглядатимемо в цьому розділі, відносять до систем гільбертівського типу, оскільки вони базуються на запропонованій Д. Гільбертом понятті формальної системи. Такі аксіоматичні системи називаються пропозиційними численнями, або численнями висловлень.

Під *пропозиційним численням* (ПЧ) розуміємо формальну систему (L, A, P), де L – мова ПЛ, A – множина аксіом ПЧ, P – множина правил виведення ПЧ.

Множина A задається єдиною схемою аксіом $\neg\Phi\vee\Phi$, тобто складається із всіх ПФ (пропозиційних формул) вигляду $\neg\Phi\vee\Phi$, які називатимемо *пропозиційними аксіомами*.

Множина P складається з таких правил виведення:

П1) $\Phi \vdash \Psi\vee\Phi$ – правило розширення;
П2) $\Phi\vee\Phi \vdash \Phi$ – правило скорочення;
П3) $\Phi\vee(\Psi\vee\Xi) \vdash (\Phi\vee\Psi)\vee\Xi$ – правило асоціативності;
П4) $\Phi\vee\Psi, \neg\Phi\vee\Xi \vdash \Psi\vee\Xi$ – правило перетину.

Теоремою ПЧ називають ПФ, яка виводиться із пропозиційних аксіом за допомогою скінченної кількості застосувань правил виведення П1–П4.

Те, що ПФ Φ – теорема, позначатимемо $\vdash\Phi$.

Різні варіанти ПЧ можуть різнитися мовами та наборами аксіом і правил виведення.

Наведемо приклади виведень в ПЧ.

Теорема 2.3.1. *Якщо* $\vdash A \vee B$, *то* $\vdash B \vee A$.

$\vdash$A$\vee$B за припущенням, $\vdash \neg\neg A \vee A$ як аксіома, тому $\vdash B \vee A$ за П4.

Теорема 2.3.2. *Якщо* $\vdash A$ *та* $\vdash A \to B$, *то* $\vdash B$.

$\vdash$A за припущенням, $\vdash B \vee A$ за П1, звідки $\vdash A \vee B$. За припущенням $\vdash A \to B$, тобто $\vdash \neg\neg A \vee B$, тому $\vdash B \vee B$ за П4, звідки $\vdash B$ за П2.

Теорема 2.3.3. *Якщо* $\vdash A_1$, ..., $\vdash A_n$ *та* $\vdash A_1 \to ... \to A_n \to B$, *то* $\vdash B$.

Справді, застосуємо n раз твердження теореми 2.3.2.

Твердження теорем 2.3.1 та 2.3.2 можна інтерпретувати як нові, похідні правила виведення:

$A \vee B \vdash B \vee A$ – *правило комутативності* (*ПК*);

$A, A \to B \vdash B$ – *правило modus ponens* (*MP*).

Теорема 2.3.4. *Якщо* $\vdash (A \vee B) \vee C$, *то* $\vdash A \vee (B \vee C)$.

Із $\vdash (A \vee B) \vee C$ за ПК $\vdash C \vee (A \vee B)$, звідки $\vdash (C \vee A) \vee B$ за П3. Далі $\vdash B \vee (C \vee A)$ за ПК та $\vdash (B \vee C) \vee A$ за П3. Тепер за ПК $\vdash A \vee (B \vee C)$.

Твердження теореми 2.3.4 та П3 в сукупності можна інтерпретувати як нове, узагальнене правило виведення:

$\vdash (A \vee B) \vee C \Leftrightarrow \vdash A \vee (B \vee C)$ – *повна асоціативність* (*АС*).

Теорема 2.3.5. *Якщо* $\vdash A \vee B$, *то* $\vdash \neg\neg A \vee B$.

Маємо $\vdash \neg\neg A \vee \neg A$ (аксіома), тому $\vdash \neg A \vee \neg\neg$ A за ПК. Звідси та з умови $\vdash A \vee B$ маємо $\vdash B \vee \neg\neg A$ за П4, тому $\vdash \neg\neg A \vee B$ за ПК.

Висновки правил П1–П4 є тавтологічними наслідками засновків. Пропозиційна аксіома – тавтологія, тому кожна теорема ПЧ є тавтологією. Отже, має місце *теорема коректності* пропозиційної логіки.

Теорема 2.3.6. *Кожна теорема ПЧ є тавтологія.*

Зворотне твердження називають *теоремою тавтології.* Воно дає повноту пропозиційної логіки.

Теорема 2.3.7 (теорема тавтології). *Кожна тавтологія є теоремою.*

В пропозиційній логіці теорему тавтології, як правило, об'єднують з теоремою коректності. В такому розширеному вигляді теорема тавтології виглядає так:

Теорема тавтології для ПЧ. *Множина теорем ПЧ збігається з множиною тавтологій.*

Наслідок. *Якщо* $\{\Phi_1, ..., \Phi_n\} \models \Phi$ *та* $\vdash \Phi_1, ..., \vdash \Phi_n$, то $\vdash \Phi$.

Якщо $\{\Phi_1, ..., \Phi_n\} \models \Phi$, то $\Phi_1 \to ... \to \Phi_n \to \Phi$ – тавтологія, звідки $\vdash \Phi_1 \to ... \to \Phi_n \to \Phi$ за теоремою тавтології. Враховуючи $\vdash \Phi_1, ..., \vdash \Phi_n$, маємо $\vdash \Phi$ за теоремою 3.2.3.

Із наслідку ТТ, зокрема, випливають такі окремі наслідки:

1) $\vdash A \& B \Leftrightarrow \vdash A$ та $\vdash B$;

2) $\vdash A \to B \Leftrightarrow \vdash \neg B \to \neg A$;

3) якщо $\vdash A \to B$ та $\vdash B \to C$, то $\vdash A \to C$;

4) $\vdash A \leftrightarrow B \Leftrightarrow \vdash A \to B$ та $\vdash B \to A$;

5) Якщо $\vdash A \leftrightarrow B$, то ($\vdash A \Leftrightarrow \vdash B$).

Теорему тавтології скорочено позначатимемо ТТ. Розглянемо приклад використання ТТ.

Приклад 2.3.1. Якщо $\vdash A \to B$ та $\vdash C \to D$, то $\vdash A \vee C \to B \vee D$. Чи вірно: якщо $\vdash A \vee C \to B \vee D$, то $\vdash A \to B$ та $\vdash C \to D$?

Неважко переконатись, що $\{A \to B, C \to D\} \models A \vee C \to B \vee D$. Звідси і з умови $\vdash A \to B$ та $\vdash C \to D$ дістаємо $\vdash A \vee C \to B \vee D$ за ТТ.

Нехай із $\vdash A \vee C \to B \vee D$ випливає $\vdash A \to B$ та $\vdash C \to D$. Звідси необхідно $A \vee C \to B \vee D \models (A \to B) \& (C \to D)$. При $\tau(A) = \tau(C) = \tau(D) = T$ та $\tau(B) = F$ маємо $\tau(A \vee C \to B \vee D) = T$ та $\tau((A \to B) \& (C \to D)) = F$.

Отже, твердження невірне.

Властивість ПФ бути тавтологією трактується як її семантична істинність, властивість бути теоремою – як синтаксична істинність. Отже, теорема тавтології засвідчує адекватність семантичної та синтаксичної істинності, тобто повноту логічних засобів ПЧ.

Із ТТ безпосередньо випливає *несуперечливість* ПЧ: не існує ПФ А такої, що $\vdash$А та $\vdash \neg$А.

Справді, якщо |–А та |–¬А, то А та ¬А – одночасно тавтології.
На основі ТТ отримуємо також *розв'язність ПЧ.*

Теорема 2.3.8. *Множина теорем ПЧ алгоритмічно розв'язна стосовно множини всіх ПФ.*

Вкажемо алгоритм розв'язання множини теорем ПЧ стосовно множини всіх пропозиційних формул. За даною ПФ А із n пропозиційними іменами побудуємо задану нею булеву функцію f. Тоді $\models A \Leftrightarrow f$ є константа 1. За теоремою тавтології $\models A \Leftrightarrow |–A$, тому для перевірки умови |–А треба перевірити, чи функція f є константа 1. Для цього обчислюємо значення f на кожному із 2^n наборів значень її імен.

2.4. Секвенції. Секвенційні форми, секвенційні дерева

Властивості відношення |= дають змогу звести логічний наслідок складної формули (разом з множиною інших формул) до логічних наслідків простіших формул, що утворюють складнішу. Отже, питання про логічний наслідок складної формули (разом з множиною інших формул) зводиться до питання про логічний наслідок з множини формул, які містять вже не саму складну формулу, а її компоненти.

Таким чином, питання про відношення логічного наслідку між двома множинами формул, в одну з яких входить складна формула, зводиться до питання про відношення логічного наслідку між двома множинами формул, в які вже входять компоненти складної формули. Це дуже важлива властивість *підформульності*.

Формально-аксіоматичні системи, які формалізують відношення логічного наслідку між двома множинами формул, називають *секвенційними численнями*. Основними об'єктами таких систем є секвенції, роль правил виведення відіграють секвенційні форми.

У класичному генценівському варіанті [6] *секвенціями* називають об'єкти вигляду Г→Δ, де Г та Δ – скінченні множини формул, → – новий символ, що не входить до алфавіту мови логіки.

Секвенційне числення будується так, що секвенція Г→Δ *вивідна* (має виведення) тоді і тільки тоді, коли Г |= Δ.

Отже, семантичним властивостям П1–П4 зіставимо їх синтаксичні аналоги – *секвенційні форми*. Секвенційні форми є правилами виведення секвенційних числень.

Секвенції Г→Δ такі, що $Г \cap Δ \neq \varnothing$, називають *замкненими*.

Замкнені секвенції відіграють роль аксіом. Справді, з умови $\Gamma \cap \Delta \neq \varnothing$ випливає $\Gamma = \Delta$.

Таким чином, справджується наведена нижче теорема.

Теорема 2.4.1. *Нехай секвенція* $\Gamma \to \Delta$ *замкнена. Тоді* $\Gamma \models \Delta$.

Секвенційні форми записують у вигляді

$$\frac{\Sigma}{\Omega} \text{ або } \frac{\Sigma \quad \Lambda}{\Omega},$$

де Σ, Λ, Ω – секвенції.

Секвенції над рискою називають засновками, секвенції під рискою – висновками. В нашому випадку засновки – це секвенції, зіставлені правим частинам відповідних семантичних властивостей, висновки – це секвенції, зіставлені їх лівим частинам.

Визначивши секвенції як множини формул, природно вважати, що секвенційні форми застосовуються до формул секвенції в довільному порядку (можливо, лише до певної скінченної підмножини доступних на даному етапі формул секвенції, див. нижче).

Якщо в секвенції $\Gamma \to \Delta$ трактувати Γ та Δ як послідовності формул, з формальної точки зору можна додатково ввести секвенційні форми для перестановки формул вигляду

$$\frac{\Gamma, A, B, \Delta \to \mathrm{K}}{\Gamma, B, A, \Delta \to \mathrm{K}} \text{ та } \frac{\Gamma \to \Delta, A, B, \mathrm{K}}{\Gamma \to \Delta, B, A, \mathrm{K}},$$

де A, B – формули; Γ, Δ, K – послідовності формул.

Секвенційні форми подібного вигляду, в яких не здійснюється розщеплення складнішої формули на простіші, називають [6] *структурними*. Трактуючи секвенції як множини формул, розглядаємо тільки секвенційні форми, в яких таке розщеплення здійснюється – *логічні* форми.

Враховуючи, що на пропозиційному рівні базовими є композиції $\vee$ та $\neg$, згідно з властивостями П1–П4 можна ввести такі базові секвенційні форми:

$$\frac{\Gamma \to \Delta, A}{\neg A, \Gamma \to \Delta}; \qquad \frac{A, \Gamma \to \Delta}{\Gamma \to \Delta, \neg A};$$

$$\frac{A, \Gamma \to \Delta \qquad B, \Gamma \to \Delta}{A \vee B, \Gamma \to \Delta}; \qquad \frac{\Gamma \to \Delta, A, B}{\Gamma \to \Delta, A \vee B}.$$

Форма запису секвенцій з використанням символу $\to$ є традиційною, вона цілком відповідає записам логічного наслідку для множин формул. Проте зручнішою є дещо модифікована форма запису секвенцій, подібна до форми запису семантичних таблиць Бета [6, 32].

Кожну формулу секвенції відмітимо (специфікуємо) зліва одним з двох символів – $\vdash$ чи $\dashv$. Якщо формула зліва від $\to$, відмічаємо її символом $\vdash$; якщо справа – символом $\dashv$. Тепер кожна формула секвенції набуває вигляду $\vdash\Phi$ або $\dashv\Phi$, причому відмітка однозначно вказує на місце формули в секвенції – зліва чи справа від $\to$. Тому в записі секвенції символ $\to$ можна випустити.

Секвенцію, утворену із секвенції $\Gamma\to\Delta$ наведеною вище відміткою формул, позначимо $\vdash\Gamma\dashv\Delta$. Не деталізуючи, секвенції відмічених формул також позначатимемо Σ.

Секвенція Σ *замкнена*, якщо існує формула Φ така, що $\vdash\Phi\in\Sigma$ та $\dashv\Phi\in\Sigma$.

Використовуючи форму запису секвенцій з відміченими формулами, зазначені вище базові секвенційні форми набувають такого вигляду (зліва будемо записувати назву форми):

$$\vdash\neg\ \frac{\dashv A,\Sigma}{\vdash\neg A,\Sigma};\qquad \dashv\neg\ \frac{\vdash A,\Sigma}{\dashv\neg A,\Sigma};$$

$$\vdash\vee\ \frac{\vdash A,\Sigma\qquad \vdash B,\Sigma}{\vdash A\vee B,\Sigma};\qquad \dashv\vee\ \frac{\dashv A,\dashv B,\Sigma}{\dashv A\vee B,\Sigma}.$$

Виділену формулу висновку, яка розбивається при виконанні секвенційної форми, називають основною.

Виділені формули в засновках секвенційних форм називають предками основної (виділеної) формули висновку. Проте, якщо розглядати процес побудови секвенційного дерева, коли секвенційні форми застосовуються знизу догори, предки можна трактувати як потомки.

Враховуючи наявність похідних композицій & та $\to$, згідно з властивостями П5–П8 введемо такі похідні секвенційні форми:

$$\vdash\to\ \frac{\dashv A,\Sigma\qquad \vdash B,\Sigma}{\vdash A\to B,\Sigma};\qquad \dashv\to\ \frac{\vdash A,\dashv B,\Sigma}{\dashv A\to B,\Sigma};$$

$$\vdash\& \frac{\vdash A, \vdash B, \Sigma}{\vdash A \,\&\, B, \Sigma}; \qquad \dashv\& \frac{\dashv A, \Sigma \quad \dashv B, \Sigma}{\dashv A \,\&\, B, \Sigma}.$$

Беручи до уваги властивості відношення $\models$ на пропозиційному рівні, дістаємо теорему.

Теорема 2.4.2. *Нехай* $\frac{\Sigma}{\Omega}$ *та* $\frac{\Sigma \quad \Lambda}{\Omega}$ *– секвенційні форми, причому* $\Sigma = \vdash \Lambda \dashv \mathrm{K}, \mathrm{Y} = \vdash \mathrm{X} \dashv \mathrm{Z}$ *та* $\Omega = \vdash \Gamma \dashv \Delta$. *Тоді*:

1) *якщо* $\Lambda \models \mathrm{K}$, *то* $\Gamma \models \Delta$;

2) *якщо* $\Lambda \models \mathrm{K}$ *та* $\mathrm{X} \models \mathrm{Z}$, *то* $\Gamma \models \Delta$.

Зауважимо, що запис секвенційних форм в традиційному вигляді та з використанням секвенцій з відміченими формулами – це лише різні способи запису одних і тих самих об'єктів – правил виведення секвенційного числення.

Виведення в секвенційних численнях має вигляд дерева, вершинами якого є секвенції. Такі дерева називають *секвенційними*.

Дамо індуктивне визначення секвенційного дерева:

1) секвенція Σ утворює *тривіальне* секвенційне дерево з єдиною вершиною Σ, яка є коренем дерева;

2) нехай α – секвенційне дерево з коренем Σ, β – секвенційне дерево з коренем Y, нехай $\frac{\Sigma}{\Omega}$ та $\frac{\Sigma \quad \Lambda}{\Omega}$ – секвенційні форми. Тоді $\begin{matrix}\alpha \\ | \\ \Omega\end{matrix}$ – секвенційне дерево з коренем Ω, $\begin{matrix}\alpha \quad \beta \\ \backslash\, / \\ \Omega\end{matrix}$ – секвенційне дерево з коренем Ω.

Тривіальне секвенційне дерево *замкнене*, якщо це замкнена секвенція. Нетривіальне секвенційне дерево *замкнене*, якщо кожний його лист (кінцева вершина, відмінна від кореня) – замкнена секвенція.

Секвенційне дерево з коренем Σ називають також секвенційним деревом секвенції Σ.

Секвенція Σ *вивідна*, або *має виведення*, якщо існує замкнене секвенційне дерево з коренем Σ. Таке дерево назвемо *виведенням* секвенції Σ.

Секвенція X – *наступник* секвенції Y в секвенційному дереві δ з коренем Σ, якщо в δ існує шлях $\Sigma = \Sigma_1, \ldots, \Sigma_n, \ldots, \Sigma_m, \ldots$ такий, що $\mathrm{X} = \Sigma_n$ та $\mathrm{Y} = \Sigma_m$.

2.5. Коректність та повнота секвенційних числень

Сформулюємо теорему коректності для пропозиційних секвенційних числень.

Теорема 2.5.1. *Нехай секвенція* $\vdash\Gamma\dashv\Delta$ *вивідна. Тоді* $\Gamma \models \Delta$.

Доводимо індукцією за побудовою замкненого секвенційного дерева для секвенції $\vdash\Gamma\dashv\Delta$. Нехай для $\vdash\Gamma\dashv\Delta$ маємо замкнене тривіальне дерево, тоді воно складається з єдиної вершини $\vdash\Gamma\dashv\Delta$, яка є замкненою секвенцією. За теоремою 2.4.1 маємо $\Gamma \models \Delta$.

Нехай для $\vdash\Gamma\dashv\Delta$ маємо замкнене секвенційне дерево γ, причому на останньому кроці виведення було застосовано секвенційну форму $\frac{\Sigma}{\Omega}$, де $\Sigma = \vdash\Lambda\dashv K$ та $\Omega = \vdash\Gamma\dashv\Delta$. У цьому випадку γ має вигляд $\begin{array}{c}\alpha\\ |\\ \Omega\end{array}$, де α – замкнене дерево з коренем Σ. За припущенням індукції маємо $\Lambda \models K$. За теоремою 2.4.2 $\Gamma \models \Delta$.

Нехай для $\vdash\Gamma\dashv\Delta$ маємо замкнене секвенційне дерево γ, на останньому кроці виведення було застосовано секвенційну форму $\frac{\Sigma \quad Y}{\Omega}$, де $\Sigma = \vdash\Lambda\dashv K$, $Y = \vdash X\dashv Z$ та $\Omega = \vdash\Gamma\dashv\Delta$.

У цьому випадку γ має вигляд $\begin{array}{c}\alpha\quad\beta\\ \backslash\,/\\ \Omega\end{array}$, де α та β – замкнені дерева з коренями Σ та Y відповідно. За припущенням індукції $\Lambda \models K$ та $X \models Z$. За теоремою 2.4.2 $\Gamma \models \Delta$**.**

Для доведення повноти пропозиційних секвенційних числень використаємо метод модельних (хінтіківських) множин.

Множина $\boldsymbol{H}$ відмічених пропозиційних формул *модельна*, якщо виконуються такі умови:

НА) для кожної атомарної формули Φ лише одна з формул $\vdash\Phi$ чи $\dashv\Phi$ може належати до $\boldsymbol{H}$;

Н$\neg$) якщо $\vdash\neg\Phi\in\boldsymbol{H}$, то $\dashv\Phi\in\boldsymbol{H}$;

якщо $\dashv\neg\Phi\in\boldsymbol{H}$, то $\vdash\Phi\in\boldsymbol{H}$.

Н$\vee$) якщо $\vdash\Phi\vee\Psi\in\boldsymbol{H}$, то $\vdash\Phi\in\boldsymbol{H}$ або $\vdash\Psi\in\boldsymbol{H}$;

якщо $\dashv\Phi\vee\Psi\in\boldsymbol{H}$, то $\dashv\Phi\in\boldsymbol{H}$ та $\dashv\Psi\in\boldsymbol{H}$.

Розглянемо процедуру побудови дерева для секвенції **Σ.** Така побудова починається з кореня дерева, яке "росте" вгору. В процесі побудови ми від складніших формул переходимо до простіших, тому секвенційні форми застосовуються знизу вгору.

Процедуру побудови секвенційного дерева розіб'ємо на етапи. Кожна секвенційна форма застосовується до скінченної множини доступних на даний момент формул.

На початку кожного етапу виконується *крок доступу*: до списку доступних формул додаємо по одній формулі зі списку $\vdash$-формул та списку $\dashv$-формул. Якщо недоступних $\vdash$-формул чи $\dashv$-формул немає (відповідний список вичерпаний), то на подальших кроках доступу додаємо по одній формулі невичерпаного списку. На початку побудови дерева доступна лише пара перших формул списків (або єдина $\vdash$-формула чи $\dashv$-формула, якщо один із списків порожній).

Нехай виконано k етапів процедури. На етапі k+1 перевіряємо, чи буде кожен з листів дерева замкненою секвенцією. Якщо всі листи замкнені, то процедура завершена позитивно, ми отримали замкнене секвенційне дерево. Якщо ні, то для кожного незамкненого листа ξ робимо наступний крок доступу, після чого добудуємо скінченне піддерево з вершиною ξ.

Активізуємо всі доступні неатомарні формули ξ. По черзі до кожної активної формули застосовуємо відповідну секвенційну форму. Після виконання секвенційної форми формула пасивна, до пасивних та утворених на даному етапі формул секвенційні форми не застосовуються. Повтори формул усуваються.

При побудові секвенційного дерева можливі такі випадки:

1. Процедура завершена позитивно, маємо замкнене дерево.

2. Процедура завершена негативно, маємо скінченне незамкнене дерево. Тоді в секвенційному дереві існує скінченний шлях $\Sigma = \Sigma_1, \Sigma_2, \ldots, \Sigma_n$, всі вершини якого – незамкнені секвенції. Такий шлях $\wp$ назвемо незамкненим. До формул останньої секвенції Σ_n шляху $\wp$ жодна секвенційна форма вже незастосовна, тому всі формули секвенції Σ_n атомарні.

3. Процедура не завершується, маємо нескінченне незамкнене дерево (це можливо для випадку нескінченних секвенцій). За лемою Кеніга [6] нескінченне дерево із скінченним розгалуженням має хоча б один нескінченний шлях. Вершини цього шляху не можуть бути замкненими секвенціями, оскільки в разі появи замкненої секвенції до неї вже не застосовується жодна секвенційна форма, і

процес побудови для цього шляху обривається. Отже, в дереві існує нескінченний шлях $\Sigma = \Sigma_1, \Sigma_2, ..., \Sigma_n, ...$ всі вершини якого – незамкнені секвенції. Такий шлях $\wp$ також назвемо незамкненим. Кожна із формул секвенції Σ зустрінеться на цьому шляху і стане доступною, тому кожна з пропозиційних змінних, що входять до складу формул секвенції Σ, зустрінеться на шляху $\wp$ як відмічена атомарна формула.

Теорема 2.5.2. *Нехай $\wp$ – незамкнений шлях в секвенційному дереві, $\boldsymbol{H}$ – множина всіх відмічених формул секвенцій цього шляху. Тоді $\boldsymbol{H}$ – модельна множина.*

Для переходу від нижчої вершини шляху до вищої використовується одна з базових секвенційних форм. Переходи, згідно з такими формами, точно відповідають пунктам Н$\neg$ та Н$\vee$ визначення модельної множини. Кожна неатомарна формула, що зустрічається на шляху $\wp$, рано чи пізно буде розкладена згідно з відповідною секвенційною формою. Всі секвенції шляху $\wp$ незамкнені, тому виконується пункт НА визначення модельної множини.

Теорема 2.5.3. *Нехай $\boldsymbol{H}$ – модельна множина. Тоді існує істинніснa оцінка $\tau: Fp \to \{T, F\}$ така, що:*

1) *з умови $\vdash \Phi \in \boldsymbol{H}$ випливає $\tau(\Phi) = T$;*

2) *з умови $\dashv \Phi \in \boldsymbol{H}$ випливає $\tau(\Phi) = F$.*

Доведення проведемо індукцією за складністю формули згідно із побудовою модельної множини.

Для атомарних формул (пропозиційних змінних) покладемо:

- якщо $\vdash A \in \boldsymbol{H}$, то $\tau(A) = T$;
- якщо $\dashv A \in \boldsymbol{H}$, то $\tau(A) = F$.

Для всіх інших пропозиційних змінних τ можна задати довільно, бо такі змінні не входять до складу формул $\boldsymbol{H}$.

Нехай $\vdash \neg\Phi \in \boldsymbol{H}$. За визначенням $\boldsymbol{H}$ маємо $\dashv \Phi \in \boldsymbol{H}$. За припущенням індукції $\tau(\Phi) = F$, звідки $\tau(\neg\Phi) = T$.

Нехай $\dashv \neg\Phi \in \boldsymbol{H}$. За визначенням $\boldsymbol{H}$ маємо $\vdash \Phi \in \boldsymbol{H}$. За припущенням індукції $\tau(\Phi) = T$, звідки $\tau(\neg\Phi) = F$.

Нехай $\vdash \Phi \vee \Psi \in \boldsymbol{H}$. За визначенням $\boldsymbol{H}$ маємо $\vdash \Phi \in \boldsymbol{H}$ або $\vdash \Psi \in \boldsymbol{H}$. За припущенням індукції $\tau(\Phi) = T$ або $\tau(\Psi) = T$, звідки $\tau(\Phi \vee \Psi) = T$.

Нехай $\dashv \Phi \vee \Psi \in \boldsymbol{H}$. За визначенням $\boldsymbol{H}$ маємо $\dashv \Phi \in \boldsymbol{H}$ та $\dashv \Psi \in \boldsymbol{H}$. За припущенням індукції $\tau(\Phi) = F$ та $\tau(\Psi) = F$, звідки $\tau(\Phi \vee \Psi) = F$.

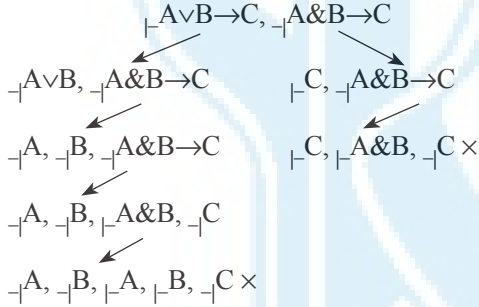
Звідси отримуємо теорему повноти пропозиційних секвенційних числень.

Теорема 2.5.4 (повноти пропозиційних секвенційних числень). *Нехай* $\Gamma \models \Delta$. *Тоді секвенція* $\vdash \Gamma \dashv \Delta$ *вивідна.*

Припустимо супротивне: $\Gamma \models \Delta$ та $\vdash \Gamma \dashv \Delta$ невивідна. Якщо $\Sigma = \vdash \Gamma \dashv \Delta$ невивідна, то в секвенційному дереві для Σ існує незамкнений шлях. Згідно з теоремою 2.5.2 множина $\boldsymbol{H}$ всіх відмічених формул секвенцій цього шляху – модельна множина.

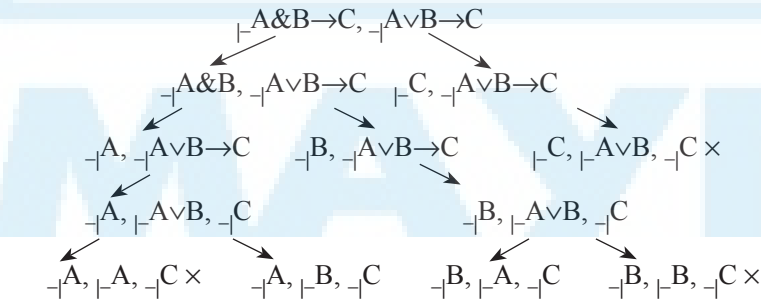
Згідно з теоремою 2.5.3 існує істиннісна оцінка τ така: з умови $\vdash \Phi \in \boldsymbol{H}$ випливає $\tau(\Phi) = T$, а з умови $\dashv \Phi \in \boldsymbol{H}$ випливає $\tau(\Phi) = F$. Згідно з $\Sigma \subseteq \boldsymbol{H}$ це справджується для формул секвенції Σ. Тому для всіх $\Phi \in \Gamma$ маємо $\tau(\Phi) = T$, для всіх $\Psi \in \Delta$ маємо $\tau(\Psi) = F$. Це заперечує $\Gamma \models \Delta$.

Приклад 2.5.1. Для встановлення вірності $A \vee B \to C \models A \& B \to C$ будуємо виведення секвенції $\vdash A \vee B \to C, \dashv A \& B \to C$.



Збудували замкнене секвенційне дерево, тому справджується $A \vee B \to C \models A \& B \to C$.

Приклад 2.5.2. Для встановлення вірності $A \& B \to C \models A \vee B \to C$ будуємо виведення секвенції $\vdash A \& B \to C, \dashv A \vee B \to C$.



Отримали незамкнене дерево, тому невірно, що $A\&B\to C \models A\vee B\to C$.

При цьому по двох незамкнених листах можна прочитати такі два контрприклади:

– $\tau(A) = F$, $\tau(B) = T$, $\tau(C) = F$;

– $\tau(A) = T$, $\tau(B) = F$, $\tau(C) = F$.

Важливим наслідком теореми повноти є теорема про елімінацію перетинів.

Теорема 2.5.5 (про елімінацію перетинів). *Нехай секвенції* $\vdash \Phi, \Sigma$ *та* $\dashv \Phi, \Sigma$ *вивідні. Тоді секвенція* $\vdash \Sigma$ *вивідна.*

Нехай $\Sigma = \vdash \Gamma \dashv \Delta$. З умови $\vdash \Phi, \Sigma$ вивідна та $\dashv \Phi, \Sigma$ вивідна за теоремою коректності маємо $\Phi, \Gamma \models \Delta$ та $\Gamma \models \Delta, \Phi$. Звідси негайно $\Gamma \models \Delta$, тому за теоремою повноти $\Sigma = \vdash \Gamma \dashv \Delta$ вивідна.

Звичайно, таке доведення цілком неконструктивне і не дає жодного способу перетворення замкнених секвенційних дерев для $\vdash \Phi, \Sigma$ та $\dashv \Phi, \Sigma$ в замкнене секвенційне дерево для Σ. Цього недоліку позбавлене набагато складніше доведення Г. Генцена, здійснене ним для логіки предикатів 1-го порядку.

Система Генцена містила секвенційну форму, названу *правилом перетину*. В спрощеному вигляді у наших позначеннях це правило можна записати так: $\dfrac{\vdash A, \Sigma \qquad \dashv A, \Sigma}{\Sigma}$.

У своїй основній теоремі про нормальну форму (hauptsatz, див. [6]) Генцен показав, як виведення із застосуванням правила перетину можна перебудувати у виведення без застосування цього правила. Теорему про нормальну форму називають також теоремою про елімінацію перетинів.

Теорема Генцена посідає центральне місце в теорії доведень і є основою багатьох сучасних методів автоматизованого пошуку виведень.

2.6. Метод резолюцій

Метод резолюцій є найпоширенішим методом пошуку доведень. На момент виникнення він дуже добре підійшов для реалізації на тогочасних комп'ютерах, тому і до цього часу найчастіше використовується в системах пошуку доведень. Семантичні таблиці (секвенції)

та натуральний вивід були незручні для реалізації на відносно примітивній обчислювальній техніці 60-х років XX ст., адже вони вимагають використання складних структур даних та розвинутих методологій програмування, хоча б рівня об'єктно-орієнтованого програмування, чого в ті часи ще не було. Відомо [32], що натуральний вивід може бути наскільки завгодно коротшим за метод резолюцій, але метод резолюцій і нині посідає панівне становище серед методів пошуку доведень. Це дуже нагадує ситуацію з програмним забезпеченням фірми Microsoft, яка займає практично монопольне становище. Незважаючи на низку недоліків, витіснити її продукти навіть на порядок кращими дуже важко, адже користувачі у своїй масі настільки звикли до цих систем, що не бажають знати нічого нового. Проте недоліки методу резолюцій жодною мірою не применшують його позитивні якості та практичне значення.

Метод резолюцій для логік 1-го порядку базується на зведенні формул до певної стандартної форми. Ідейною основою методу для логіки предикатів є теорема Ербрана. У процесі розвитку було розроблено низку модифікацій та стратегій застосування методу резолюцій, що підвищують його ефективність. Це, зокрема, лінійна резолюція, лок-резолюція, семантична резолюція, стратегія поглинання тощо.

На основі методу резолюцій у 70-х роках XX ст. було створено мову логічного програмування Пролог. Досвід використання Прологу свідчить, що він дуже ефективний для написання відносно невеликих програм із складною логікою, але для обчислювальних фрагментів програми існують набагато прийнятніші для цього мови, наприклад Pascal.

У нашому посібнику обмежимось розглядом методу резолюцій пропозиційної логіки. Спершу введемо деякі стандартні визначення.

Пропозиційну формулу вигляду A або $\neg A$, де $A \in Ps$, назвемо *літерою*.

У загальному випадку літери – це атомарні формули (пропозиційні символи для пропозиційної логіки) або заперечення атомарних формул.

Пропозиційну формулу, яка має вигляд диз'юнкції кількох літер, назвемо *диз'юнктом*.

Однолітерний диз'юнкт назвемо *одиничним*.

Символ 0 трактуватимемо як диз'юнкт, що не має жодної літери. Назвемо його *порожнім диз'юнктом*. Порожній диз'юнкт можна інтерпретувати як суперечність – логічну константу F.

Літери A та $\neg A$ назвемо *контрарними*.

Нехай D та D' – диз'юнкти такі, що існують літера L в D та контрарна їй літера L' в D'.

Резольвентою диз'юнктів D та D' назвемо диз'юнкт R, утворений диз'юнкцією всіх літер D, відмінних від L, та всіх літер D', відмінних від L'.

У цьому випадку кажуть, що резольвента R побудована за диз'юнктами D та D' згідно з *правилом резолюцій*.

Отже, правило резолюцій (ПР) має вигляд $D, D' \vdash R$, де D та D' – диз'юнкти, R – їх резольвента.

Зокрема, якщо D та D' – контрарні літери, то R – порожній диз'юнкт, тобто R суть 0.

Теорема 2.6.1. *Нехай R – резольвента диз'юнктів D та D'. Тоді* $\{D, D'\} \models R$.

Нехай диз'юнкти D та D' мають контрарні літери L та L'. Переставимо їх в диз'юнктах D та D' наперед. Отримаємо диз'юнкти $L \vee C$ та $L' \vee C'$ такі, що $D \sim_т L \vee C$ та $D' \sim_т L' \vee C'$. Тоді $R \sim_т C \vee C'$.

Справді, нехай для визначеності L' суть $\neg L$. Тоді $D' \sim_т \neg L \vee C'$. Але $\{L \vee C, \neg L \vee C'\} \models C \vee C'$, звідки $\{D, D'\} \models R$.

Таким чином, правило резолюцій коректне.

Нехай $\boldsymbol{S}$ – множина диз'юнктів. Послідовність диз'юнктів $D_1, D_2, ..., D_n$ – *резолютивне виведення* диз'юнкту D_n із $\boldsymbol{S}$, якщо кожний D_i або належить до $\boldsymbol{S}$, або отриманий із попередніх диз'юнктів цієї послідовності за допомогою ПР.

Диз'юнкт D *виводиться* із $\boldsymbol{S}$, якщо існує резолютивне виведення D із $\boldsymbol{S}$.

Резолютивне виведення 0 із $\boldsymbol{S}$ назвемо *спростуванням* $\boldsymbol{S}$, або *доведенням суперечливості* множини диз'юнктів $\boldsymbol{S}$.

Справді, множина $\boldsymbol{S}$ суперечлива $\Leftrightarrow$ існує резолютивне виведення 0 із $\boldsymbol{S}$.

Позаяк $\{D_1, ..., D_m\} \models D \Leftrightarrow \{D_1, ..., D_m, \neg D\}$ суперечлива, для доведення $\{D_1, ..., D_m\} \models D$ достатньо вказати резолютивне виведення 0 із $\{D_1, ..., D_m, \neg D\}$.

Без обмежень загальності можна вважати, що жоден з диз'юнктів множини $\boldsymbol{S}$ не містить пари контрарних літер, тобто не є тавтологією. Справді, якщо D – тавтологія, то $\{D_1, ..., D_m, D\}$ суперечлива $\Leftrightarrow \{D_1, ..., D_m\}$ суперечлива.

Розглянемо приклади резолютивних виведень.

Приклад 2.6.1. Чи вірно $\{P \vee Q, \neg P \vee R, \neg Q \vee S, \neg R \vee S \vee P, \neg S \vee P\} \models P$?

Виведемо 0 із множини $\{P \vee Q, \neg P \vee R, \neg Q \vee S, \neg R \vee S \vee P, \neg S \vee P, \neg P\}$. Із $P \vee Q$ та $\neg P \vee R$ за ПР маємо $Q \vee R$. Із $Q \vee R$ та $\neg Q \vee S$ за ПР маємо $R \vee S$. Із $R \vee S$ та $\neg R \vee S \vee P$ за ПР маємо $S \vee P$. Із $S \vee P$ та $\neg S \vee P$ за ПР маємо P. Із P та $\neg P$ за ПР маємо 0.

Ми довели, що множина $\{P \vee Q, \neg P \vee R, \neg Q \vee S, \neg R \vee S \vee P, \neg S \vee P, \neg P\}$ суперечлива, тому твердження про наявність $\models$ вірне.

Приклад 2.6.2. Чи суперечлива $\{P \vee \neg Q, \neg Q \vee S \vee \neg P, \neg S, Q, R \vee S\}$?

Виведемо 0 із зазначеної множини. Із $P \vee \neg Q$ та $\neg Q \vee S \vee \neg P$ за ПР маємо $\neg Q \vee S$. Із $\neg Q \vee S$ та $\neg S$ за ПР маємо $\neg Q$. Із $\neg Q$ та Q за ПР маємо 0.

Отже, множина $\{P \vee \neg Q, \neg Q \vee S \vee \neg P, \neg S, Q, R \vee S\}$ суперечлива. Зауважимо, що для виведення 0 не використовувались всі диз'юнкти множини, а саме, диз'юнкт $R \vee S$.

Приклад 2.6.3. Чи суперечлива $\{P \vee Q, Q \vee S \vee \neg P, \neg Q\}$?

Із $P \vee Q$ та $\neg Q$ за ПР маємо P. Із P та $Q \vee S \vee \neg P$ за ПР маємо $Q \vee S$.

Із $Q \vee S$ та $\neg Q$ за ПР маємо S. Із $\neg Q$ та $Q \vee S \vee \neg P$ за ПР маємо $S \vee \neg P$. Із $S \vee \neg P$ та $P \vee Q$ знову маємо $Q \vee S$.

Переконаємось, що 0 вивести неможливо. Для пропозиційних символів, що входять до складу диз'юнктів нашої множини, ми вивели літери P та S, маємо в множині літеру $\neg Q$ як одиничний диз'юнкт. Тому розглянемо таку оцінку: $\tau(P) = T$, $\tau(S) = T$, $\tau(Q) = F$. За такої оцінки всі диз'юнкти множини набувають значення T, звідки множина несуперечлива.

Питання для самоконтролю

1. Як розглядаються предикати на пропозиційному рівні?
2. Що таке логічні зв'язки (пропозиційні композиції)?
3. Які ви знаєте логічні зв'язки?
4. Наведіть приклади неоднозначності природної мови.
5. Наведіть приклади множин базових пропозиційних композицій (логічних зв'язок).
6. Дайте визначення традиційних логічних зв'язок (заперечення, диз'юнкція, кон'юнкція, імплікація, еквіваленція, роздільна диз'юнкція).
7. Які пропозиційні композиції ми вибрали як базові?
8. Що таке префіксна (польська) форма запису?
9. Що таке інфіксна форма запису?

10. Задайте алфавіт мови пропозиційної логіки.
11. Дайте індуктивне визначення пропозиційної формули.
12. Що таке атомарна пропозиційна формула?
13. Що таке скорочення пропозиційної формули? Наведіть приклади ПФ та їх скорочень.
14. Як виражаються $\&$, $\rightarrow$, $\leftrightarrow$ та $\oplus$ через $\neg$ та $\vee$?
15. Що таке істиннісна оцінка мови ПЛ?
16. Як задається значення атомарних та неатомарних ПФ?
17. Що таке булева функція?
18. Дайте визначення тавтології. Наведіть приклади тавтологій.
19. Що таке суперечність? Наведіть приклади суперечностей.
20. Вкажіть тавтології, які виражають основні закони традиційної логіки.
21. Запишіть основні властивості пропозиційних композицій.
22. Дайте визначення логічного (тавтологічного) наслідку для ПФ. Наведіть приклади.
23. Дайте визначення логічної (тавтологічної) еквівалентності ПФ. Наведіть приклади.
24. Що таке логічний (тавтологічний) наслідок скінченної множини ПФ?
25. Вкажіть основні властивості відношень $\models$ та $\sim_{т}$.
26. Що таке суперечлива множина ПФ?
27. Дайте визначення відношення логічного наслідку для множин ПФ. Наведіть приклади.
28. Чи вірно, що відношення логічного наслідку для множин ПФ рефлексивне? Транзитивне? Відповідь аргументуйте.
29. Вкажіть основні властивості відношення логічного наслідку для множин ПФ.
30. Чи вірно: якщо диз'юнкція двох формул є тавтологією, то кожна з цих формул є тавтологією? Відповідь аргументуйте.
31. Чи вірно: якщо кон'юнкція двох формул – тавтологія, то кожна з цих формул є тавтологією? Відповідь аргументуйте.
32. Чи вірно: якщо імплікація двох формул – тавтологія, то кожна з цих формул є тавтологією? Відповідь аргументуйте.
33. Чи вірно: якщо еквіваленція двох формул – тавтологія, то кожна з цих формул є тавтологією? Відповідь аргументуйте.
34. Сформулюйте теорему семантичної еквівалентності.
35. Сформулюйте теорему заміни еквівалентних.
36. Що таке аксіоматичні системи гільбертівського типу?

37. Що таке пропозиційне числення?
38. Що таке пропозиційна аксіома?
39. Наведіть правила виведення пропозиційного числення.
40. Що таке теорема пропозиційного числення?
41. Сформулюйте правило комутативності (ПК).
42. Сформулюйте правило modus ponens (МР).
43. Сформулюйте правило повної асоціативності (АС).
44. Сформулюйте теорему коректності для ПЧ.
45. Сформулюйте теорему тавтології для ПЧ.
46. Які ви знаєте наслідки теореми тавтології?
47. Чому теорему тавтології називають теоремою повноти ПЧ?
48. Чому пропозиційне числення несуперечливе?
49. Вкажіть алгоритм розв'язання ПЧ.
50. У чому полягає властивість підформульності?
51. Що таке секвенційні числення?
52. Дайте визначення секвенції в Генценівському варіанті.
53. Що таке секвенційна форма?
54. Що таке замкнена секвенція?
55. Що таке засновки та висновки секвенційних форм?
56. Що таке логічні та структурні секвенційні форми?
57. Вкажіть базові секвенційні форми.
58. Що таке секвенція відмічених (специфікованих) формул?
59. Дайте визначення замкненої секвенції специфікованих формул.
60. Вкажіть базові секвенційні форми для випадку секвенцій специфікованих формул.
61. Вкажіть похідні секвенційні форми.
62. Як задаються виведення в секвенційних численнях?
63. Що таке секвенційне дерево? Дайте визначення.
64. Дайте визначення замкненого секвенційного дерева.
65. Дайте визначення вивідної секвенції.
66. Сформулюйте теорему коректності для секвенційних числень.
67. Дайте визначення модельної множини пропозиційних формул.
68. Сформулюйте теорему повноти для секвенційних числень.
69. Сформулюйте правило перетину.
70. Сформулюйте теорему Генцена про елімінацію перетинів.
71. Що таке літера?
72. Що таке диз'юнкт?
73. Що таке контрарні літери?

74. Як інтерпретуємо порожній диз'юнкт?
75. Що таке резольвента диз'юнктів?
76. Як формулюється правило резолюцій?
77. У чому полягає коректність правила резолюцій?
78. Що таке резолютивне виведення?
79. Що таке спростування множини диз'юнктів?

Вправи

1. Доведіть, що наведені у 2.1 закони ПЛ є тавтологіями.
2. Знайдіть ПФ Ψ таку, щоб наступна ПФ була тавтологією:
   1) $(\Psi\&A\to\neg B)\to((B\to\neg A)\to\Psi)$;
   2) $((C\to\neg A\&B)\to\Psi)\to\Psi\&C\&(B\to A)$.
3. Доведіть наведені в 2.1 властивості відношень $\models$ та $\sim_т$.
4. Доведіть: $\{\Phi_1, ..., \Phi_m\}\models\Psi \Leftrightarrow \{\Phi_1, ..., \Phi_m, \neg\Psi\}$ є суперечливою.
5. Визначте, чи вірно:
   1) $\{A\vee B, \neg A\vee C\}\models B\vee C$;
   2) $\{A\to B, C\to D\}\models A\vee C\to B\vee D$;
   3) $\{A\to B, C\to D\}\models A\&C\to B\&D$;
   4) $\{A\to B, C\to D\}\models A\vee C\to B\&D$;
   5) $\{A\to B, C\to D\}\models A\&C\to B\vee D$;
   6) $A\&D\to B\vee C\models(A\to B)\vee(A\to C)\vee(D\to B)$;
   7) $\{A\to B, C\to D, A\vee C\}\models B\vee D$;
   8) $\{A\to B, C\to D, \neg B\vee\neg D\}\models\neg A\vee\neg C$.
6. Визначте, чи вірно:
   1) $\{A, A\to B\}\models B$;
   2) $\{B, A\to B\}\models A$;
   3) $\{\neg A, A\to B\}\models\neg B$;
   4) $\{\neg B, A\to B\}\models\neg A$;
   5) $\{A\to B, C\to D, A\vee D\}\models B\vee C$;
   6) $\{A\vee B, B\to C, \neg A\vee\neg C\}\models\neg A\vee D$;
   7) $\{A\vee B\to C\to D, \neg A\&\neg B\}\models C\&\neg D$;
   8) $\{A\to C, B\to D, A\vee B\}\models D\&C$.
7. Визначте, в якому відношенні щодо $\models$ перебувають ПФ A&B→C&D, A∨B→C&D, A&B→C∨D та A∨B→C∨D.
8. Доведіть у пропозиційному численні без використання ТТ:
   1) $\vdash A\to B\to A$;
   2) $\vdash A\to\neg A\to B$;
   3) $\vdash B\to A\vee B$;

4) $\vdash(\neg A\to B)\vee\neg A$;
5) $\vdash A\vee A\to A$;
6) $\vdash(A\to B)\to(\neg B\to\neg A)$;
7) якщо $\vdash A\to C$ та $\vdash B\to C$, то $\vdash A\vee B\to C$;
8) якщо $\vdash A\to B\to C$ та $\vdash B$, то $\vdash A\to C$;
9) якщо $\vdash A\vee(B\to C)$ та $\vdash B$, то $\vdash A\vee C$;
10) якщо $\vdash A\to B\to C$ та $\vdash A\to B$, то $\vdash A\to C$;
11) якщо $\vdash A\vee B\vee C$, то $\vdash C\vee A\vee B$ та $\vdash A\vee C\vee B$;
12) якщо $\vdash A\vee B\vee C$, то $\vdash C\vee B\vee A$, $\vdash B\vee C\vee A$ та $\vdash B\vee A\vee C$;
13) якщо $\vdash A\vee B$, то $\vdash\neg A\to C\vee B$;
14) якщо $\vdash A\to B$, то $\vdash A\vee C\to B\vee C$;
15) якщо $\vdash A\vee A\vee B$, то $\vdash A\vee B$;
16) якщо $\vdash A\vee B\vee A\vee C$, то $\vdash B\vee A\vee C$.

9. Чи вірно:
1) якщо $\vdash A\&C\to B\&D$, то $\vdash A\to B$ та $\vdash C\to D$;
2) якщо $\vdash A\vee C\to B\&D$, то $\vdash A\to B$ та $\vdash C\to D$;
3) якщо $\vdash A\&C\to B\vee D$, то $\vdash A\to B$ та $\vdash C\to D$;
4) якщо $\vdash A\to B$ та $\vdash C\to D$, то $\vdash A\&C\to B\&D$;
5) якщо $\vdash A\to B$ та $\vdash C\to D$, то $\vdash A\&C\to B\vee D$;
6) якщо $\vdash A\to B$ та $\vdash C\to D$, то $\vdash A\vee C\to B\&D$.

10. Чи вірно:
1) а) якщо $\vdash(A\to B)\to C$, то $\vdash A\to C$;
б) якщо $\vdash A\to C$, то $\vdash(A\to B)\to C$;
2) а) якщо $\vdash A\to B\&C$, то $\vdash A\to B$;
б) якщо $\vdash A\to B$, то $\vdash A\to B\&C$;
3) а) якщо $\vdash A\&B\to C$, то $\vdash A\to C$;
б) якщо $\vdash A\to C$, то $\vdash A\&B\to C$;
4) а) якщо $\vdash A\vee C\to B$, то $\vdash A\to B$;
б) якщо $\vdash A\to B$, то $\vdash A\vee C\to B$.

11. Чи вірно: якщо ($\vdash A\Leftrightarrow\vdash B$), то $\vdash A\leftrightarrow B$.

12. Використовуючи властивості П5–П10 відношення $\vdash$, введіть похідні секвенційні форми $\vdash\!\!\!_{\to}$, $\dashv\vdash\!\!\!_{\to}$, $\vdash\!\!\!_{\&}$, $\dashv\vdash\!\!\!_{\&}$, $\vdash\!\!\!_{\leftrightarrow}$, $\dashv\vdash\!\!\!_{\leftrightarrow}$.

13. Побудуйте в пропозиційному секвенційному численні виведення чи доведіть його відсутність, вказавши контрприклад, для вказаних нижче формул (побудова в секвенційному численні виведення формули Ф означає побудову виведення секвенції $\dashv\vdash\!\!\!$Ф):
1) $(A\vee B\to C)\to(A\to C)$;
2) $(A\to B\&C)\to(A\to C)$;

3) $(A\to C)\to(B\to C)\to(A\to B\vee C)$;
4) $(A\to C)\to(B\to C)\to(C\to A\vee B)$;
5) $(A\to B)\to(A\to C)\to(A\&B\to C)$;
6) $(A\to B)\to(A\to C)\to(B\&C\to A)$;
7) $((A\to B)\to C)\to(A\vee C)$;
8) $(A\to B)\&\ (B\to C)\to(\neg A\to C)$;
9) $((A\to C)\to D)\&\neg D\to A\&\neg C$;
10) $(A\to B)\&(A\to C)\&(A\to D)\to\neg A$;
11) $(A\&B\to C)\to(A\to C)\vee(B\to C)$;
12) $((A\to B)\to(C\to D))\to(A\to C)\&(D\to B)$.

14. Використовуючи пропозиційне секвенційне числення, встановіть, чи вірно:

1) $\{A\vee B, \neg A\vee C\}\models B\vee C$;
2) $\{A\to B, C\to D\}\models A\vee C\to B\vee D$;
3) $\{A\to B, C\to D\}\models A\&C\to B\&D$;
4) $\{A\to B, C\to D\}\models A\vee C\to B\&D$;
5) $\{A\to B, C\to D\}\models A\&C\to B\vee D$;
6) $\{A\to B, C\to D, A\vee D\}\models B\vee C$;
7) $\{A\vee B, B\to C, \neg A\vee\neg C\}\models\neg A\vee D$;
8) $\{A\vee B\to C\to D, \neg A\&\neg B\}\models C\&\neg D$;
9) $\{A\to C, B\to D, A\vee B\}\models D\&C$;
10) $\{A\to B, C\to D, \neg B\vee\neg D\}\models\neg A\vee\neg C$.

15. Доведіть чи спростуйте методом резолюцій пропозиційної логіки:

1) $\{A\to B, C\to D, A\vee C\}\models B\vee D$;
2) $\{B\to A, C\to D, B\&\neg D\}\models A\&\neg C$;
3) $\{\neg A\to B, Q\to D, \neg B\vee\neg D\}\models A\vee\neg Q$;
4) $\{A\to C, \neg D\to B, \neg A\to\neg B\}\models\neg C\to D$;
5) $\{A\to B, C\to D, \neg B\vee D\&C\}\models\neg A\vee\neg C$;
6) $\{A\vee B\vee C, \neg A\vee C, \neg B\ \}\models\neg C$;
7) $A\&D\to B\vee C\ \models(A\to B)\vee(A\to C)\vee(D\to B)$.

16. Використовуючи метод резолюцій пропозиційної логіки, визначте, в якому відношенні щодо $\models$ перебувають ПФ $A\&B\to C\&D$, $A\vee B\to C\&D$, $A\&B\to C\vee D$ та $A\vee B\to C\vee D$.

3. ЛОГІКИ ПЕРШОГО ПОРЯДКУ

На рівні логік предикатів 1-го порядку функції та предикати в загальному випадку розглядаються як скінченноарні (фінарні). Логічними композиціями таких логік є успадковані з пропозиційного рівня логічні зв'язки $\neg$, $\vee$, $\rightarrow$, &, $\leftrightarrow$ та операції квантифікації (квантори) $\exists x$, $\forall x$. У логіках 1-го порядку використовуються також композиції суперпозиції (підстановки), але для випадку класичних логік вони в явному вигляді не визначаються.

Назва "логіка 1-го порядку" пов'язана з тим, що квантори застосовуються тільки до імен компонентів даних (предметних імен). Моделями такої логіки є класичні алгебраїчні системи фінарних функцій та тотальних фінарних предикатів.

У логіках 1-го порядку функції та предикати застостовуються не до абстрактних, а до структурованих даних. Такі дані є множинами пар, першою компонентою яких є ім'я, а другою компонентою – значення цього імені. При цьому одне ім'я не може іменувати два різних значення. Дані такого вигляду називаються (однозначними) *іменними множинами*. Кожну множину пар можна трактувати як функцію, звідки отримуємо наведене нижче визначення.

Нехай A та V – довільні множини. *V-іменною множиною* (скорочено V-ІМ) над A назвемо довільну однозначну функцію δ: $V \rightarrow A$.

Кожна V-ІМ як функція задається своїм графіком, тому V-ІМ будемо звичайно подавати у вигляді $[v_1 \mapsto a_1, ..., v_n \mapsto a_n, ...]$. Тут $v_i \in V$, $a_i \in A$, причому $v_i \neq v_j$ при $i \neq j$.

Множину всіх V-ІМ над A позначатимемо ${}^V A$.

Для V-ІМ вводимо теоретико-множинні операції $\cap$ та \.

Вводимо також функцію *im*: ${}^V A \rightarrow 2^V$:

$$im(\delta) = \{v \in V \mid v \mapsto a \in \delta \text{ для деякого } a \in A\}.$$

Множину всіх V-ІМ $\delta \in {}^V A$ таких, що $im(\delta) = X$, де $X \subseteq V$, позначатимемо A^X. Такі V-ІМ є тотальними однозначними функціями із X в A. V-ІМ δ скінченна, або фінітна, якщо $im(\delta)$ скінченна.

Множину всіх фінітних V-ІМ над A позначатимемо ${}^V A_F$.

Введемо параметричну операцію $\|X$ звуження V-ІМ за множиною $X \subseteq V$:

$$\delta \| X = \{v \mapsto a \in \delta \mid v \in X\}.$$

Введемо операцію ∇ *накладки* V-ІМ δ_2 на V-ІМ δ_1:

$$\delta_1 \nabla \delta_2 = \delta_2 \cup (\delta_1 \| (V \backslash im(\delta_2))).$$

Безпосередньо із визначення операції ∇ випливає:

Якщо $\alpha \subseteq \beta$, то $\alpha \nabla \delta \subseteq \beta \nabla \delta$.

Довільну функцію вигляду $f: {}^{V}A \to R$ назвемо *V-квазіарною.*

Функцію вигляду $f: {}^{V}A \to A$ назвемо *V-квазіарною функцією на A.*

Функцію вигляду $p: {}^{V}A \to \{T, F\}$ назвемо *V-квазіарним предикатом на A.*

Множину V-квазіарних функцій на A позначимо Fn^A.

Множину V-квазіарних предикатів на A позначимо Pr^A.

Довільну функцію вигляду $f: {}^{V}A_F \to R$ назвемо *V-фінарною.*

Якщо множина імен V мається на увазі, то V-квазіарні та V-фінарні функції назвемо просто квазіарними та фінарними.

Фінарні функції називають також скінченноарними.

Довільну функцію вигляду $f: A^X \to R$ назвемо *X-арною* функцією. Традиційні n-арні функції, тобто функції вигляду $f: A^n \to R$, можуть трактуватися як $\{1, ..., n\}$-арні функції. Тому $\{1, ..., n\}$-арні функції назвемо *n-арними* функціями.

Дуже важливою властивістю квазіарних функцій є еквітонність.

V-квазіарну функцію $f: {}^{V}A \to R$ назвемо *еквітонною,* якщо для довільних $d, d' \in {}^{V}A$ із $f(d)\downarrow$ та $d' \sqsupseteq d$ випливає $f(d')\downarrow = f(d)$.

Еквітонність означає, що значення відображення не змінюється при розширенні даних. Таке обмеження справджується для функцій та предикатів класичної логіки, воно притаманне практично всім функціям, які розглядаються в математиці.

Предметне ім'я x *неістотне* для V-квазіарної функції f, якщо для довільних $d \in {}^{V}A$ та $a, b \in A$ маємо $f(d\nabla x \mapsto a) \cong f(d\nabla x \mapsto b)$.

Неістотність (фіктивність) предметного імені (змінної) для функції означає, що значення функції не залежить від значення цього імені. Наприклад, y неістотне для функції $f(x, y) = x + 0 \cdot y$.

У разі еквітонних функцій ім'я x неістотне для $f \Leftrightarrow$ для довільних $d \in {}^{V}A$ та $a \in A$ маємо $f(d) \cong f(d\nabla x \mapsto a)$.

При визначенні 1-арних композицій квантифікації $\exists \boldsymbol{x}$ та $\forall \boldsymbol{x}$ в загальному випадку будемо враховувати частковість предикатів. Предикати $\exists \boldsymbol{x}(P)$ та $\forall \boldsymbol{x}(P)$ звичайно позначатимемо $\exists \boldsymbol{x}P$ та $\forall \boldsymbol{x}P$.

Зазначені предикати задамо так:

$$(\exists \boldsymbol{x}P)(d) = \begin{cases} T, \text{ якщо існує } b \in A\colon\ P(d\nabla x \mapsto b) = T, \\ F, \text{ якщо } P(d\nabla x \mapsto a)\downarrow = F \text{ для усіх } a \in A, \\ \text{невизначене в усіх інших випадках.} \end{cases}$$

$$(\forall \boldsymbol{x}P)(d) = \begin{cases} F, \text{ якщо існує } b \in A\colon\ P(d\nabla x \mapsto b) = F, \\ T, \text{ якщо } P(d\nabla x \mapsto a)\downarrow = T \text{ для усіх } a \in A, \\ \text{невизначене в усіх інших випадках.} \end{cases}$$

Властивості композицій $\exists \boldsymbol{x}$ та $\forall \boldsymbol{x}$ цілком аналогічні властивостям відповідних класичних логічних операцій квантифікації, визначених для тотальних предикатів.

Наведемо основні властивості композицій $\exists \boldsymbol{x}$ та $\forall \boldsymbol{x}$:

1) комутативність однотипних кванторів:
$\exists \boldsymbol{x}\exists \boldsymbol{y}P = \exists \boldsymbol{y}\exists \boldsymbol{x}P$,
$\forall \boldsymbol{x}\forall \boldsymbol{y}P = \forall \boldsymbol{y}\forall \boldsymbol{x}P$;
2) закони де Моргана для кванторів:
$\neg\exists \boldsymbol{x}P = \forall \boldsymbol{x}\neg P$,
$\neg\forall \boldsymbol{x}P = \exists \boldsymbol{x}\neg P$;
3) неістотність квантифікованих предметних імен:
$\exists \boldsymbol{x}\exists \boldsymbol{x}P = \exists \boldsymbol{x}P$,
$\exists \boldsymbol{x}\forall \boldsymbol{x}P = \forall \boldsymbol{x}P$,
$\forall \boldsymbol{x}\exists \boldsymbol{x}P = \exists \boldsymbol{x}P$,
$\forall \boldsymbol{x}\forall \boldsymbol{x}P = \forall \boldsymbol{x}P$;
4) закони $P \Rightarrow \exists \boldsymbol{x}P$ та $\forall \boldsymbol{x}P \Rightarrow P$;
5) закон $\exists \boldsymbol{y}\forall \boldsymbol{x}P \Rightarrow \forall \boldsymbol{x}\exists \boldsymbol{y}P$;
6) закони дистрибутивності кванторів щодо $\vee$ та $\boldsymbol{\&}$:
$\exists \boldsymbol{x}P \vee \exists \boldsymbol{x}Q = \exists \boldsymbol{x}(P \vee Q)$,
$\forall \boldsymbol{x}P \,\boldsymbol{\&}\, \forall \boldsymbol{x}Q = \forall \boldsymbol{x}(P \,\boldsymbol{\&}\, Q)$,
$\exists \boldsymbol{x}(P \,\boldsymbol{\&}\, Q) \Rightarrow \exists \boldsymbol{x}P \,\boldsymbol{\&}\, \exists \boldsymbol{x}Q$,
$\forall \boldsymbol{x}P \vee \forall \boldsymbol{x}Q \Rightarrow \forall \boldsymbol{x}(P \vee Q)$.

Неважко переконатись, що композиції $\neg$, $\vee$, $\exists \boldsymbol{x}$ та $\forall \boldsymbol{x}$ зберігають еквітонність V-квазіарних предикатів.

3.1. Алгебраїчні системи

Семантичними моделями першопорядкових логік є агебраїчні системи (АС) із тотальними фінарними функціями та предикатами. Такі АС – це пари вигляду $(A, Fn^A \cup Pr^A)$. Множину A називають *носієм*,

або *основою*, алгебраїчної системи $\boldsymbol{A} = (A, Fn^A \cup Pr^A)$. При цьому вважають $A \neq \varnothing$.

Нехай σ – довільна множина така, що існує тотальне однозначне відображення $\boldsymbol{I}: \sigma \to Fn^A \cup Pr^A$. Елементи множини σ трактуємо як *імена* деяких функцій та предикатів із $Fn^A \cup Pr^A$. Такі імена називають *функціональними символами* (ФС) та *предикатними символами* (ПС), іменовані ними функції та предикати – *базовими*. Множину σ функціональних та предикатних символів називають *сигнатурою*.

Нехай Fs – множина ФС, Ps – множина ПС. Тоді сигнатура $\sigma = Fs \cup Ps$, множина $\boldsymbol{I}$ задаватиме тотальні однозначні відображення $Fs \to Fn^A$ та $Ps \to Pr^A$.

Пару $((A, Fn^A \cup Pr^A), \boldsymbol{I})$ назвемо АС *з доданою сигнатурою* σ, або σ-АС. Такі АС позначаємо у вигляді $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, або $\boldsymbol{A} = (A, \sigma)$, якщо $\boldsymbol{I}$ мається на увазі.

Для кожного $g \in Fs$ функцію $G \in Fn^A$ таку, що $\boldsymbol{I}(g) = G$, назвемо *значенням* ФС g при інтерпретації $\boldsymbol{I}$ на АС $\boldsymbol{A} = ((A, Fn^A \cup Pr^A), \boldsymbol{I})$. Позначимо таку функцію g_A.

Предикат $P \in Pr^A$ такий, що $\boldsymbol{I}(p) = P$, назвемо *значенням* ПС p при інтерпретації $\boldsymbol{I}$ на АС $\boldsymbol{A}$. Такий предикат позначимо p_A.

В множині Fs може виділятися підмножина *константних* символів $Cn \subseteq Fs$. Значеннями константних символів завжди будуть функції-константи на A.

Для класичної логіки базові функції та предикати n-арні. Тоді з кожним ФС та ПС пов'язане натуральне число – *арність* такого символа.

Для кожного $h \in \sigma$ арність h_A дорівнює арності символу h. При обмеженні розляду тотальними функціями функції-константи трактуються як виділені елементи A.

АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$ називають *підсистемою* АС $\boldsymbol{B} = (B, \boldsymbol{I}, \sigma)$, та відповідно $\boldsymbol{B}$ називають *надсистемою* АС $\boldsymbol{A}$, якщо $A \subseteq B$ і для всіх $h \in \sigma$ $h_A \subseteq h_B$ (тобто для всіх $a \in A$ $h_B(a) = h_A(a)$). Цей факт позначатимемо $\boldsymbol{A} \subseteq \boldsymbol{B}$. У цьому разі АС $\boldsymbol{B}$ називають *розширенням* АС $\boldsymbol{A}$, а АС $\boldsymbol{A}$ – *звуженням* АС $\boldsymbol{B}$.

Множина $C \subseteq A$ утворює *підсистему* $\boldsymbol{C} = (C, \sigma)$ алгебраїчної системи $\boldsymbol{A} = (A, \sigma)$, якщо C замкнена стосовно всіх f_A, де $f \in \sigma$.

Не для кожної $C \subseteq A$ можна говорити про підсистему (C, σ).

Приклад 3.1.1. Для АС (N, σ), де $\sigma = \{+, =\}$, а символи + та = інтерпретуються природним чином, множина непарних чисел $N_н \subseteq N$

незамкнена стосовно +, тому $N_н$ не утворює підсистеми. Водночас множина парних чисел $N_n \subseteq N$ утворює власну підсистему (N_n, σ) системи (N, σ).

Нехай множини $A_1 \subseteq A$ та $A_2 \subseteq A$ замкнені стосовно всіх базових функцій АС (A, σ). Тоді $A_1 \cap A_2$ теж замкнена стосовно всіх базових функцій АС (A, σ), якщо тільки $A_1 \cap A_2 \neq \varnothing$.

Отже, якщо (A_1, σ) та (A_2, σ) – підсистеми АС (A, σ), то або $(A_1 \cap A_2, \sigma)$ – підсистема АС (A, σ), або $A_1 \cap A_2 = \varnothing$.

Підсистему $(A_1 \cap A_2, \sigma)$ назвемо *перетином* підсистем (A_1, σ) та (A_2, σ).

Теорема 3.1.1. *Перетин M носіїв всіх підсистем алгебраїчної системи* (A, σ) *або утворює підсистему* (M, σ)*, або є* $\varnothing$.

Нехай $\{A_\alpha\}_{\alpha \in J}$ – множина носіїв всіх підсистем системи $\boldsymbol{A} = (A, \sigma)$, тобто множина всіх підмножин A, замкнених відносно базових функцій АС (A, σ). Покажемо, що $M = \bigcap_{\alpha \in J} A_\alpha \neq \varnothing$ замкнена відносно всіх базових функцій $f_A, f \in \sigma$.

Нехай $a_1, ..., a_n \in M$. Тоді $a_1, ..., a_n \in A_\alpha$ для всіх $\alpha \in J$. За замкненістю A_α відносно f_A маємо $f_A(a_1, ..., a_n) \in A_\alpha$ для всіх $\alpha \in J$. Звідси отримуємо $f_A(a_1, ..., a_n) \in \bigcap_{\alpha \in I} A_\alpha = M$. Таку АС (M, σ) назвемо *найменшою підсистемою* АС (A, σ). Зрозуміло, що якщо сигнатура σ містить константні символи, то АС (A, σ) має найменшу підсистему.

Нехай $\{A_\alpha\}_{\alpha \in J}$ – множина носіїв всіх підсистем системи $\boldsymbol{A} = (A, \sigma)$. Для довільної $B \subseteq A$ множина $C = \bigcap_{\alpha \in I} A_\alpha$, де $I = \{\alpha \in J \mid B \subseteq A_\alpha\}$, є найменшою множиною, замкненою відносно всіх базових функцій системи $\boldsymbol{A} = (A, \sigma)$. Така C визначає АС (C, σ), яку називають *підсистемою системи* (A, σ), *породженою множиною B*. Якщо при цьому $C = A$, то АС (A, σ) *породжується підмножиною* $B \subseteq A$.

Різні підмножини можуть породжувати одну і ту саму підсистему.

Приклад 3.1.2. Система $(N, \{+, =\})$ породжується множиною $\{0, 1\}$.

Приклад 3.1.3. Система $\boldsymbol{N} = (N, \{0, 1, +, \times, =\})$ породжується множиною $\{0, 1\}$. Наявність константних символів 0 та 1 призводить до

того, що в кожній підсистемі $\boldsymbol{N}$ носій містить 0 та 1, але тоді підсистема збігається з усією системою. Отже, $\boldsymbol{N}$ власних підсистем не має.

Приклад 3.1.4. Система $(Z^+, \{+, =\})$ має підсистеми вигляду $(kZ^+, \{+, =\})$, де $kZ^+ = \{kx \mid x \in Z^+\}$, для довільних $k \in Z^+$.

Для формального опису та дослідження семантичних моделей класичних логік 1-го порядку – алгебраїчних систем – використовуються мови класичних логік 1-го порядку, або просто *мови 1-го порядку*.

3.2. Мови першого порядку

Алфавіт класичної мови 1-го порядку складається із таких символів:

- предметні імена (змінні) $x, y, z, \ldots$;
- функціональні символи (ФС) $f_0, f_1, f_2, \ldots$ заданої арності;
- предикатні символи (ПС) $p_0, p_1, p_2, \ldots$ заданої арності;
- символи логічних операцій (композицій) $\neg$, $\vee$ та $\exists x$.

У множині Fs може виділятися підмножина константних символів $Cn \subseteq Fs$.

Спеціальний предикатний символ рівності =, якщо $= \in Ps$, завжди інтерпретуємо як предикат рівності, причому рівність трактуємо як тотожність.

Символи $\neg$, $\vee$, $\exists$, = та предметні імена назвемо *логічними* символами. Функціональні та предикатні символи, окрім =, назвемо *нелогічними* символами. Множина $\sigma = Fs \cup Ps$ функціональних та предикатних символів — це *сигнатура* мови 1-го порядку.

Основними конструкціями мови 1-го порядку є терми та формули.

Терми використовують для позначення, назви суб'єктів, *формули* – для запису тверджень про суб'єкти.

Індуктивне визначення терма:

1) кожне предметне ім'я та кожна константа є термом; такі терми назвемо *атомарними*;

2) якщо $t_1, \ldots, t_n$ – терми, f – n-арний функціональний символ, то $ft_1 \ldots t_n$ – терм.

Атомарною формулою називається вираз вигляду $pt_1 \ldots t_n$, де p – n-арний предикатний символ; $t_1, \ldots, t_n$ – терми.

Індуктивне визначення *формули*:

1) кожна атомарна формула є формулою;

2) якщо Φ та Ψ – формули, то $\neg\Phi$ та $\vee\Phi\Psi$ – формули;

3) якщо Φ – формула, x – предметне ім'я, то $\exists x\Phi$ – формула.

Аналогічно до мови ПЛ вирази $\Phi\&\Psi$, $\Phi\to\Psi$ та $\Phi\leftrightarrow\Psi$ вважаємо скороченнями формул $\neg\vee\neg\Phi\neg\Psi$, $\vee\neg\Phi\Psi$ та $\neg\vee\neg\vee\neg\Phi\Psi\neg\vee\neg\Psi\Phi$. Користуємося також символом $\forall x$, вважаючи вираз $\forall x\Phi$ скороченням формули $\neg\exists x\neg\Phi$. Для бінарних ФС та ПС і символів $\vee$, &, $\to$ та $\leftrightarrow$ звичайно застосовуємо інфіксну форму запису.

Пріоритет символів логічних композицій вважаємо нижчим за пріоритет ПС, а пріоритет ПС – нижчим за пріоритет ФС.

Для символів логічних композицій встановимо такий пріоритет: $\exists x$, $\neg$, &, $\vee$, $\to$, $\leftrightarrow$.

Використовуючи додаткові символи – кому "," і дужки "(" та ")", для термів вигляду $ft_1\ldots t_n$ вживатимемо скорочення $f(t_1\ldots t_n)$, або t_1ft_2, якщо символ f бінарний. Те саме для атомарних формул. Для атомарних формул вигляду $\neg = t_1\ t_2$ вживатимемо скорочення $t_1\neq t_2$. Скорочення термів та формул теж називатимемо термами та формулами.

Множини термів та формул мови 1-го порядку позначатимемо відповідно *Tr* та *Fr*.

Формули мови 1-го порядку сигнатури σ назвемо формулами 1-го порядку сигнатури σ.

Можна вказати два рівні відмінності мов 1-го порядку:

1) варіанти мови однієї сигнатури, що відрізняються наборами символів логічних операцій та способами запису термів і формул;

2) істотно різні мови, що відрізняються сигнатурами.

Мова 1-го порядку L' сигнатури σ′ називається *розширенням* мови 1-го порядку L сигнатури σ, якщо $\sigma'\supseteq\sigma$. У цьому разі мова L *є звуженням* мови L'.

У формулі вигляду $\exists x\Phi$ або $\forall x\Phi$ формулу Φ називають *областю дії* квантора по x. Вираз вигляду $\exists x$ або $\forall x$ називають *кванторним префіксом*.

Входження імені (змінної) x у формулу Φ *зв'язане*, якщо воно перебуває в області дії деякого квантора по x, інакше таке входження x в Φ *вільне*.

Якщо існує вільне входження імені x у формулу Φ, то x – *вільне ім'я* (*вільна змінна*) формули Φ.

Формулу Φ із вільними іменами $x_1,..,x_n$ позначаємо $\Phi(x_1,..,x_n)$.

Наприклад, у формулі $\exists z(x+z=y)\vee z=0$ входження змінних x та y вільні, перше входження z зв'язане, друге – вільне. Тому ця формула має вільні змінні x, y, z.

Терм, який не містить входжень предметних імен, називається *замкненим термом*. Зокрема, таким є кожний константний символ.

Ще одним прикладом є замкнений терм $(1+0)\times 1$.

Формула *замкнена*, якщо вона не має вільних імен. Наприклад, $1+0=0+1$ та $\forall x\exists y(y>x)$ – замкнені формули.

Наведемо кілька прикладів мов 1-го порядку.

Приклад 3.2.1. *Мова арифметики* L_{ar} визначається сигнатурою $\sigma_{ar}=\{0, 1, +, \times, =\}$, де 0 та 1 – константні символи, + та $\times$ – бінарні функціональні символи, = – бінарний предикатний символ.

Терм мови арифметики назвемо *арифметичним термом*.

Формулу мови арифметики – *арифметичною формулою.*

Наприклад, $1+1$ – замкнений арифметичний терм; $x\times(y+z)$ – арифметичний терм; $\exists z(x+z=y)$ – арифметична формула.

Приклад 3.2.2. Мова теорії множин L_{set} визначається сигнатурою $\sigma_{set}=\{\in, =\}$, де $\in$ та = – бінарні предикатні символи.

Наприклад, $z\in x$ – атомарна формула, $\forall z(z\in x\to z\in y)$ – формула, $\exists x\neg\exists y(y\in x)$ – замкнена формула мови L_{set}. Зауважимо, що останні дві формули відповідно означають "$x\subseteq y$" та "існує порожня множина $\varnothing$".

Приклад 3.2.3. Мова теорії впорядкованих множин L_{ord} визначається сигнатурою $\sigma_{ord}=\{<, =\}$, де < та = – бінарні предикатні символи.

Наприклад, $x<y$ – атомарна формула, $z<x\to x<y\to z\in y$ – формула, $\forall x\exists y(y<x)$ – замкнена формула мови L_{ord}.

Зв'язані імена у формулах можна замінювати іншими предметними іменами, але при цьому може виникнути *колізія* – ситуація, коли вільні імена стали зв'язаними. Наприклад, із формули $\exists z(x+z=y)$ можна отримати формулу $\exists t(x+t=y)$, коли колізії немає, та формулу $\exists x(x+x=y)$, коли колізія змінила зміст формули.

Вільні входження предметних імен у формулу або терм можна замінювати термами.

Позначимо $\Phi_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$ формулу, отриману із формули Φ заміною всіх вільних входжень імен $x_1, \ldots, x_n$ на терми $t_1, \ldots, t_n$ відповідно. Для термів аналогічно вводимо позначення $t_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$.

У загальному випадку формули $\Phi_{x,y}[a, b]$ та $(\Phi_x[a])_y[b]$ різні. Наприклад, якщо Φ – це формула $x \in y$, то $\Phi_{x,y}[y, z]$ – формула $y \in z$, $(\Phi_x[y])_y[z]$ – формула $z \in z$.

У разі заміни вільних входжень предметних імен термами можливі *колізії*, коли вільне ім'я стає зв'язаним. Наприклад, нехай Φ – це формула $\exists z(x + z = y)$. Тоді $\Phi_x[u]$ – формула $\exists z(u + z = y)$, $\Phi_x[z]$ – формула $\exists z(z + z = y)$; маємо колізію.

Звідси випливає таке визначення: терм *t допустимий для заміни вільного імені x у формулі* Φ, якщо *x* не перебуває в області дії ніякого квантора по деякому імені, яке входить до складу *t*.

Задамо семантику мови 1-го порядку, інтерпретуючи мову на алгебраїчній системі тієї самої сигнатури.

Інтерпретацією, або *моделлю,* мови *L* сигнатури σ називатимемо АС з доданою сигнатурою вигляду $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$.

Множину *A* називають *областю інтерпретації*.

Значення символів та виразів мови *L* задамо на $\boldsymbol{A}$ таким чином:

- предметні імена інтерпретуємо як імена елементів (змінні) на носії *A*;
- символи логічних операцій інтерпретуємо як відповідні логічні операції;
- константні символи інтерпретуємо як конкретні елементи множини *A*, тобто як функції-константи на *A*;
- предикатні та функціональні символи інтерпретуємо як предикати та функції відповідної арності, визначені на *A*, причому бінарний предикатний символ "=" завжди інтерпретуватимемо як предикат рівності на *A*.

Таким чином, конкретна інтерпретація мови *L* на АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$ визначається відображенням $\boldsymbol{I}$: $\sigma \to Fn^A \cup Pr^A$.

Значення символів c, f, p позначаємо відповідно c_A, f_A, p_A: $\boldsymbol{I}(c) = c_A$, $\boldsymbol{I}(f) = f_A$, $\boldsymbol{I}(p) = p_A$.

Для інтерпретації термів і формул мови *L* задамо відображення $\boldsymbol{J}$: $Tr \cup Fr \to Fn^A \cup Pr^A$, яке індуктивно визначається за допомогою $\boldsymbol{I}$.

Для термів маємо:

– $\boldsymbol{J}(x) = \boldsymbol{'x}$;

– $\boldsymbol{J}(ft_1 \ldots t_n) = \boldsymbol{I}(f)(\boldsymbol{J}(t_1), \ldots, \boldsymbol{J}(t_n)) = f_A(\boldsymbol{J}(t_1), \ldots, \boldsymbol{J}(t_n))$.

Для атомарних формул маємо:

– $\boldsymbol{J}(pt_1 \ldots t_n) = \boldsymbol{I}(p)(\boldsymbol{J}(t_1), \ldots, \boldsymbol{J}(t_n)) = p_A(\boldsymbol{J}(t_1), \ldots, \boldsymbol{J}(t_n))$.

Для формул маємо:

- нехай $\boldsymbol{J}(\Phi) = P$, тоді $\boldsymbol{J}(\neg\Phi) = \neg P$, $\boldsymbol{J}(\exists x\Phi) = \exists \boldsymbol{x} P$.
- нехай $\boldsymbol{J}(\Phi) = P$ та $\boldsymbol{J}(\Psi) = Q$, тоді $\boldsymbol{J}(\vee\Phi\Psi) = P \vee Q$.

Кожний терм з вільними іменами $v_1, ..., v_n$ інтерпретується як $\{v_1, ..., v_n\}$-арна функція на A, кожна формула з вільними іменами $v_1, ..., v_n$ інтерпретується як $\{v_1, ..., v_n\}$-арна функція на A. Зокрема, кожний замкнений терм інтерпретується як функція-константа на A, кожна замкнена формула – як предикат-константа на A.

Функцію, що є значенням терма t на АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, позначаємо t_A; предикат, що є значенням формули Φ на АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, позначаємо Φ_A. Це означає, що $\boldsymbol{J}(t) = t_A$, $\boldsymbol{J}(\Phi) = \Phi_A$.

Формулу Φ назвемо *істинною при інтерпретації* $\boldsymbol{A}$, або *істинною на* $\boldsymbol{A}$, або $\boldsymbol{A}$*-істинною*, якщо предикат Φ_A є істинним.

Це означає: X-арний предикат Φ_A такий, що для всіх $d \in A^X$ маємо $\Phi_A(d) = T$.

Те, що формула Φ істинна на АС $\boldsymbol{A}$, позначаємо $\boldsymbol{A} \models \Phi$.

Формула Φ називається *всюди істинною*, якщо вона істинна за кожної інтерпретації.

Те, що Φ всюди істинна, позначимо $\models \Phi$.

Формулу Φ назвемо *виконуваною при інтерпретації* $\boldsymbol{A}$, або *виконуваною на АС* $\boldsymbol{A}$, або $\boldsymbol{A}$*-виконуваною*, якщо предикат Φ_A є виконуваним. Це означає: X-арний предикат Φ_A такий, що для деякого $d \in A^X$ маємо $\Phi_A(d) = T$.

Формула Φ називається *виконуваною*, якщо вона виконувана за деякої інтерпретації.

Приклад 3.2.4. Формула $x = x$ всюди істинна.

Приклад 3.2.5. Формула $\forall x \forall y(x = y)$ істинна на всіх 1-елементних АС і тільки на них; формула $\neg\forall x \forall y(x = y)$ істинна на всіх k-елементних АС, де $k>1$, і тільки на них.

Замиканням формули Φ з вільними іменами $x_1, ..., x_n$ назвемо замкнену формулу $\forall x_1 ... \forall x_n \Phi$, яку позначатимемо $\bar{\Phi}$.

Із визначень випливає *семантична теорема замикання:*

Теорема 3.2.1. *Для кожних* $\boldsymbol{A}$ *та* Φ маємо $\boldsymbol{A} \models \Phi \Leftrightarrow \boldsymbol{A} \models \bar{\Phi}$.

Окремим випадком всюди істинних формул є *тавтології*, тобто маємо формули, які мають структуру тавтологій мови ПЛ.

Формулу назвемо *пропозиційно нерозкладною,* якщо вона атомарна або має вигляд $\exists x\Phi$.

Нехай Fr_0 – множина всіх пропозиційно нерозкладних формул мови L; Fr – множина всіх формул мови L.

Істиннісною оцінкою мови L назвемо довільне відображення $\tau : Fr_0 \to \{T, F\}$.

Продовжимо τ до відображення $\tau : Fr \to \{T, F\}$ таким чином:

– $\tau(\neg\Phi) = T \Leftrightarrow \tau(\Phi) = F$;

– $\tau(\vee\Phi\Psi) = T \Leftrightarrow \tau(\Phi) = T$ або $\tau(\Psi) = T$.

Формула Φ мови L є *тавтологією*, якщо для кожної істиннісної оцінки τ мови L маємо $\tau(\Phi) = T$.

Зрозуміло, що кожна тавтологія є всюди істинною формулою, але зворотне твердження невірне. Наприклад, всюди істинна формула вигляду $x = x$ не є тавтологією.

На множині формул введемо відношення тавтологічного наслідку $\models$, тавтологічної еквівалентності $\sim_т$, логічного наслідку $|=$, слабкого логічного наслідку $\|=$ та логічної еквівалентності $\sim$.

Формула Ψ є *тавтологічним наслідком* формули Φ, що позначатимемо $\Phi \models \Psi$, якщо формула $\Phi \to \Psi$ – тавтологія.

Формули Φ та Ψ *тавтологічно еквівалентні,* що позначатимемо $\Phi \sim_т \Psi$, якщо $\Phi \models \Psi$ та $\Psi \models \Phi$.

Формула Ψ є *логічним наслідком* формули Φ, що позначатимемо $\Phi | = \Psi$, якщо формула $\Phi \to \Psi$ всюди істинна.

Формули Φ та Ψ *логічно еквівалентні,* що позначатимемо $\Phi \sim \Psi$, якщо $\Phi | = \Psi$ та $\Psi | = \Phi$.

Зрозуміло, що $\Phi \sim \Psi \Leftrightarrow$ формули $\Phi \to \Psi$ та $\Psi \to \Phi$ всюди істинні.

Формула Ψ є *слабким логічним наслідком* формули Φ, що позначатимемо $\Phi \| = \Psi$, якщо для кожної інтерпретації $\boldsymbol{A}$ із умови $\boldsymbol{A} | = \Phi$ випливає $\boldsymbol{A} | = \Psi$.

Формула Ψ є *логічним наслідком множини формул* $\{\Phi_1, ..., \Phi_n\}$, що позначатимемо $\{\Phi_1, ..., \Phi_n\} | = \Psi$, якщо $\Phi_1 \& \ldots \& \Phi_n | = \Psi$.

Аналогічно визначаємо $\{\Phi_1, ..., \Phi_n\} \models \Psi$ та $\{\Phi_1, ..., \Phi_n\} \| = \Psi$.

Замість $\varnothing \models \Phi$, $\varnothing | = \Phi$ та $\varnothing \| = \Phi$ писатимемо $\models \Phi$, $| = \Phi$ та $\| = \Phi$.

Основні властивості відношень $\models$, $|=$, $\|=$ та $\sim$:

1) Φ тавтологія $\Leftrightarrow \models \Phi$;

2) Φ всюди істинна $\Leftrightarrow | = \Phi \Leftrightarrow \| = \Phi$;

3) якщо $\Phi \models \Psi$, то $\Phi | = \Psi$; але не завжди із $\Phi | = \Psi$ випливає $\Phi \models \Psi$;

4) якщо $\Phi | = \Psi$, то $\Phi \| = \Psi$; але не завжди із $\Phi \| = \Phi$ випливає $\Phi | = \Psi$;

5) $\Phi \sim_{\tau} \Psi \Leftrightarrow \Phi \leftrightarrow \Psi$ тавтологія

6) $\Phi \sim \Psi \Leftrightarrow \models \Phi \leftrightarrow \Psi$;

7) відношення $\models$, $\models$ та $\|\!=$ рефлексивні та транзитивні;

8) відношення ~ рефлексивне, транзитивне і симетричне.

Для 3) та 4) маємо такі *контрприклади*:

Приклад 3.2.6. $\exists x \exists y (x = y) \models \exists y \exists x (x = y)$, але невірно $\exists x \exists y (x = y) \not\models \exists y \exists x (x = y)$.

Приклад 3.2.7. $(x = 0) \| \!= \forall x (x = 0)$ але невірно $(x = 0) \models \forall x (x = 0)$.

За теоремою замикання $(x = 0) \| \!= \forall x (x = 0)$. Але $(x = 0)_N (0) = T$ та $(\forall x (x = 0))_N = F$, тому маємо $(x = 0 \to \forall x (x = 0))_N (0) = F$, звідки $(x = 0) \not\models \forall x (x = 0)$.

Теорема 3.2.2. *Якщо x не вільне в* Ψ, *то* $\Phi \to \Psi \| \!= \exists x \Phi \to \Psi$.

Нехай X – множина вільних імен формули $\Phi \to \Psi$. Припустимо супротивне: існує $\boldsymbol{A} = (A, \sigma)$ така, що $\boldsymbol{A} \models \Phi \to \Psi$ та $\boldsymbol{A} \not\models \exists x \Phi \to \Psi$. Тоді існує $d \in A^X$ таке, що $(\exists x \Phi \to \Psi)_A (d) = F$, звідки $(\exists x \Phi)_A (d) = T$ та $\Psi_A (d) = F$. Згідно з $(\exists x \Phi)_A (d) = T$ маємо $\Phi_A (d \nabla x \mapsto b) = T$ для деякого $b \in A$. Але x не вільне в Ψ, тому $\Psi_A (d \nabla x \mapsto b) = \Psi_A (d) = F$. Звідси $(\Phi \to \Psi)_A (d \nabla x \mapsto b) = F$, що суперечить $\boldsymbol{A} \models \Phi \to \Psi$.

Визначення логічного наслідку для множин формул мови 1-го порядку подібне до відповідного визначення для множин пропозиційних формул.

Нехай Γ та Δ – множини формул мови певної сигнатури, $\boldsymbol{A} = (A, \boldsymbol{I})$ – АС тієї самої сигнатури.

Δ є логічним наслідком Γ в АС $\boldsymbol{A}$, якщо для всіх $d \in A^X$ із того, що $\Phi_A (d) = T$ для всіх $\Phi \in \Gamma$, випливає, що $\Psi_A (d) = T$ для деякої $\Psi \in \Delta$.

Те, що Δ є логічним наслідком Γ в АС $\boldsymbol{A}$, позначаємо $\Gamma \models_A \Delta$.

Δ є логічним наслідком Γ, якщо $\Gamma \models_A \Delta$ для всіх АС $\boldsymbol{A} = (A, \boldsymbol{I})$ тієї самої сигнатури.

Те, що Δ є логічним наслідком Γ, позначаємо $\Gamma \models \Delta$.

Отже, $\Gamma \not\models \Delta \Leftrightarrow$ існують АС $\boldsymbol{A} = (A, \boldsymbol{I})$ та $d \in {}^V A$ такі: для всіх $\Phi \in \Gamma$ маємо $\Phi_A (d) = T$ та для всіх $\Psi \in \Delta$ маємо $\Psi_A (d) = F$.

Відношення логічного наслідку для множин формул мови 1-го порядку рефлексивне, але нетранзитивне.

Теорема заміни еквівалентних справджується для логік 1-го порядку:

Теорема 3.2.3. *Нехай* $\Phi \sim \Psi$. *Тоді маємо* $\Phi, \Gamma \models \Delta \Leftrightarrow \Psi, \Gamma \models \Delta$ та $\Gamma \models \Delta, \Phi \Leftrightarrow \Gamma \models \Delta, \Psi$.

Розглянуті вище для відношення $\models$ властивості пропозиційного рівня успадковуються на рівні логік 1-го порядку.

Вкажемо тепер властивості відношення $\models$, пов'язані з елімінацією кванторів.

$\exists_{\models}$) $\Gamma \models \Delta, \Phi_x[y_1], \ldots, \Phi_x[y_n], \exists x\Phi \Leftrightarrow \Gamma \models \Delta, \exists x\Phi$;

$\forall_{\models}$) $\Phi_x[y_1], \ldots, \Phi_x[y_n], \forall x\Phi, \Gamma \models \Delta \Leftrightarrow \forall x\Phi, \Gamma \models \Delta$;

$\exists_{\not\models}$) $\exists x\Phi, \Gamma \not\models \Delta \Leftrightarrow \Phi_x[y], \Gamma \not\models \Delta$ за умови: вільна змінна $y \notin \Gamma \cup \Delta \cup \{\exists x\Phi\}$;

$\forall_{\not\models}$) $\Gamma \models \Delta, \forall x\Phi \Leftrightarrow \Gamma \models \Delta, \Phi_x[y]$ за умови: вільна змінна $y \notin \Gamma \cup \Delta \cup \{\exists x\Phi\}$.

Формула Φ мови L *є k-істинною*, якщо $\boldsymbol{A} \models \Phi$ для кожної k-елементної інтерпретації $\boldsymbol{A}$ мови L.

Формула Φ *є скінченно-істинною*, якщо Φ є k-істинною для кожного $k>0$. Отже, скінченно-істинна формула є істинною за кожної скінченної інтерпретації.

Приклад 3.2.8. Формула $\exists x_1 \ldots \exists x_k((x_1 \neq x_2) \& \ldots \& (x_1 \neq x_k) \& \ldots \& (x_{k-1} \neq x_k))$, яку позначимо E_k, стверджує: існує $\geq k$ різних елементів області інтерпретації. Отже, E_k є n-істинною для всіх $n \geq k$.

Приклад 3.2.9. Формула $\exists x_1 \exists x_2 \ldots \exists x_k \forall y((y=x_1) \vee \ldots \vee (y=x_k))$, яку позначимо G_k, стверджує, що існує $\leq k$ різних елементів області інтерпретації. Отже, G_k є n-істинною для всіх $1 \leq n \leq k$.

Звідси отримуємо: формула $E_k \& G_k$ є k-істинною, причому така $E_k \& G_k$ не є n-істинною для кожного $n \neq k$.

Теорема 3.2.4. *Проблема k-істинності алгоритмічно розв'язна.*

Нехай Φ – довільна формула. Тоді Φ містить скінченну кількість функціональних та предикатних символів. Тому існує скінченна кількість інтерпретацій з носієм потужності k, на яких можна по-різному інтерпретувати функціональні та предикатні символи формули Φ. Якщо Φ істинна на кожній з таких інтерпретацій, то вона є k-істинною.

Теорема 3.2.5. *Існує скінченно-істинна, але не всюди істинна формула 1-го порядку.*

Розглянемо формулу S вигляду $S_1\&S_2\&S_3$, де S_1 має вигляд $\forall x\neg\Phi(x,x)$; S_2 має вигляд $\forall x\forall y\forall z(\Phi(x,y)\&\Phi(y,z)\to\Phi(x,z))$; S_3 має вигляд $\forall x\exists y\Phi(x,y)$. Нехай арифметична формула $\Phi(x,y)$ має вигляд $\exists z(x+z=y\&x\neq y)$, тобто $\Phi(x,y)$ виражає предикат "$x<y$" на $\boldsymbol{N}$. Тоді $\boldsymbol{N}\models$S, звідки замкнена формула $\neg S$ не всюди істинна.

Припустимо, що $\neg S$ не є скінченно-істинна. Тоді для деякої скінченної $\boldsymbol{A}=(A,\sigma)$ маємо $\boldsymbol{A}\not\models\neg S$, звідки $\boldsymbol{A}\models S$ за замкненістю формули S. Тому $\boldsymbol{A}\models S_1$, $\boldsymbol{A}\models S_2$ та $\boldsymbol{A}\models S_3$. Формули S_1 і S_3 задають умови рефлексивності та транзитивності для Φ_A, тому Φ можна інтерпретувати як предикат "<" на A. Тоді формула S_3 гарантує існування нескінченного ланцюга $a_0<a_1<...<a_n<...$ елементів A, починаючи з довільного $a_0\in A$. Отже, множина A нескінченна. Маємо суперечність. Тому формула $\neg S$ є скінченно-істинною.

3.3. Еквівалентні перетворення формул

Теорема 3.3.1 (семантична теорема еквівалентності). *Нехай* Φ' *отримана із формули* Φ *заміною деяких входжень формул* $\Phi_1, ..., \Phi_n$ *на* $\Psi_1, ..., \Psi_n$ *відповідно. Якщо* $\Phi_1\sim\Psi_1, ..., \Phi_n\sim\Psi_n$, *то* $\Phi\sim\Phi'$.

Доводимо індукцією за побудовою формули.

1. Нехай формула Φ атомарна. Довільним входженням формули в Φ може бути тільки сама Φ. Тому або ніякої заміни немає, або замінюємо всю Φ. У першому випадку Φ' збігається з Φ, тому $\Phi\sim\Phi'$. У другому випадку Φ суть Φ_i для деякого $i\in\{1, ..., n\}$. Тоді Φ' – суть Ψ_i. За умовою $\Phi_i\sim\Psi_i$, тобто $\Phi\sim\Phi'$.

2. Нехай Φ має вигляд $\neg\Psi$. Тоді довільним входженням формули в Φ є або вся Φ, або вона цілком міститься в Ψ. У першому випадку доводимо аналогічно 1). У другому випадку Φ' – суть формули $\neg\Psi'$, де Ψ' отримана із Ψ так, як описано в теоремі. За припущенням індукції $\Psi\sim\Psi'$, звідки $\neg\Psi\sim\neg\Psi'$, тобто $\Phi\sim\Phi'$.

3. Нехай Φ має вигляд $\vee\Psi\Xi$. Довільне входження формули в Φ є або вся Φ, або цілком міститься в Ψ, або цілком міститься в Ξ. В першому випадку доводимо аналогічно 1). У другому випадку Φ' – суть формули $\vee\Psi'\Xi'$, де Ψ' та Ξ' отримані із Ψ та Ξ так, як описано в теоремі. За припущенням індукції $\Psi\sim\Psi'$ та $\Xi\sim\Xi'$, звідки $\vee\Psi\Xi\sim\vee\Psi'\Xi'$, тобто $\Phi\sim\Phi'$.

4. Нехай Φ має вигляд $\exists x\Psi$. Довільним входженням формули в Φ є або вся Φ, або вона цілком міститься в Ψ. В першому випад-

ку доводимо аналогічно 1). У другому випадку Φ' – суть формули $\exists x\Psi'$, де Ψ' отримана із Ψ так, як описано в теоремі. За припущенням індукції $\Psi \sim \Psi'$, звідки (пропонуємо довести) $\exists x\Psi \sim \exists x\Psi'$, тобто $\vdash \Phi \sim \Phi'$.

Теорема 3.3.2 (семантична теорема рівності для термів). *Нехай терм* τ' *отриманий із терма* τ *заміною деяких входжень термів* $t_1, ..., t_n$ *на терми* $s_1, ..., s_n$ *відповідно. Якщо* $\models t_1 = s_1, ..., \models t_n = s_n$, *то* $\models \tau = \tau'$.

Теорема 3.3.3 (семантична теорема рівності для формул). *Нехай* Φ' *отримана із формули* Φ *заміною деяких входжень термів* $t_1, ..., t_n$ *на терми* $s_1, ..., s_n$ *відповідно. Якщо* $\models t_1 = s_1, ..., \models t_n = s_n$, *то* $\Phi \sim \Phi'$.

Тоереми 3.3.2 та 3.3.3 є індукцією за побудовою терма та формули відповідно.

Неважко переконатись, що справджується теорема, наведена нижче.

Теорема 3.3.4. 1) $\exists xB \sim \exists yB_x[y]$, *якщо* y *не вільна у* B.

2) $\neg\forall xB \sim \exists x\neg B$ *та* $\neg\exists xB \sim \forall x\neg B$;

3) $\exists xB \vee C \sim \exists x(B \vee C)$ *та* $\forall xB \vee C \sim \forall x(B \vee C)$, *якщо* x *не вільна у* C;

4) $B \vee \exists xC \sim \exists x(B \vee C)$ *та* $B \vee \forall xC \sim \forall x(B \vee C)$, *якщо* x *не вільна у* B.

Формула A' називається *варіантою* формули A, якщо A' можна отримати із A послідовними замінами такого типу: підформулу $\exists xB$ замінюємо на $\exists yB_x[y]$, де y не вільна у B.

Теорема 3.3.5 (семантична теорема про варіанту). *Якщо* A' – *варіанта формули* A, *то* $A \sim A'$.

Для доведення теореми достатньо скористатися теоремою 3.3.1 та пунктом 1) теореми 3.3.4.

Формула A перебуває у *пренексній формі*, якщо вона має вигляд $Qx_1 ... Qx_n B$, де Qx_k – кванторний префікс $\exists x_k$ або $\forall x_k$, B – безкванторна формула, яку називають *матрицею* формули A. Зокрема, якщо формула є безкванторною, то вона вже перебуває у пренексній формі.

Формулу в пренексній формі називають *пренексною формулою*.

У визначенні пренексної форми насправді фігурують формули, для яких така пренексна форма є скороченням. Але, коли йдеться про пренексну форму, $\forall$ не прийнято виражати через $\neg$ та $\exists$.

Введемо *пренексні операції* над формулами, які дають змогу кожну формулу перетворити на еквівалентну їй пренексну формулу. Ці операції грунтуються на теоремах 3.3.1, 3.3.4 та 3.3.5.

Пренексними операціями над формулою А назвемо такі операції:

а) заміна A деякою її варіантою;

б) заміна в А підформул вигляду $\neg\exists xB$ та $\neg\forall xB$ на $\forall x\neg B$ та $\exists x\neg B$ відповідно;

в) заміна в A підформул вигляду $QxB\vee C$ на $Qx(B\vee C)$, якщо x не вільне в C; заміна в A підформул вигляду $B\vee QxC$ на $Qx(B\vee C)$, якщо x не вільне у B.

Пренексною формою формули A називатимемо пренексну формулу A', утворену із A за допомогою пренексних операцій.

Теорема 3.3.6. *Кожна формула має пренексну форму, причому якщо* A′ *– пренексна форма формули* A, *то* A~A′.

Доведення теореми здійснюється індукцією за довжиною формули. Зробіть це як вправу.

Розглянутий метод побудови пренексної форми передбачає роботу в системі логічних операцій $\{\neg, \vee, \exists\boldsymbol{x}, \forall\boldsymbol{x}\}$. Для уникнення елімінації $\&$ та $\rightarrow$ можна ввести додаткові пренексні операції:

г) заміна в A підформул вигляду $QxB\&C$ на $Qx(B\&C)$, якщо x не вільне в C, та підформул вигляду $B\&QxC$ на $Qx(B\&C)$, якщо x не вільне у B;

г) заміна в A підформул вигляду $B\rightarrow QxC$ на $Qx(B\rightarrow C)$, якщо x не вільне у B;

д) заміна в A підформул вигляду $\exists xB\rightarrow C$ на $\forall x(B\rightarrow C)$, та підформул вигляду $\forall xB\rightarrow C$ на $\exists x(B\rightarrow C)$, якщо x не вільне в C.

Зрозуміло, що виконання кожної з операцій типу г)–д) зводиться до виконання певної послідовності операцій б) та в). Але для $\leftrightarrow$ подібних операцій немає.

3.4. Виразність в алгебраїчних системах. Арифметичні предикати, множини, функції

Нехай $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$ – деяка АС.

Предикат P на A виразний в $\mathbf{A} = (A, \boldsymbol{I}, \sigma)$ формулою Φ сигнатури σ, якщо P – суть предикат Φ_A.

Предикат P на A *виразний в* АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, якщо P виразний в $\boldsymbol{A}$ деякою формулою Φ сигнатури σ.

Іншими словами, предикат P на A виразний в АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, якщо існує така формула Φ сигнатури σ, що P – суть предикат Φ_A.

Множина, що є областю істинності предиката, виразного в АС $\boldsymbol{A}$, називається *виразною в АС* $\boldsymbol{A}$ *множиною*.

Функція, графік якої – виразна в АС $\boldsymbol{A}$ множина, називається виразною в АС $\boldsymbol{A}$ функцією.

Приклад 3.4.1. Предикат "$x = 0$" в АС $(N, \{\times, =\})$, $(Q, \{\times, =\})$, $(R, \{\times, =\})$ виражається формулою $\forall y(x \times y = x)$.

Приклад 3.4.2. Предикат "$x = 1$" в АС $(N, \{\times, =\})$, $(Z, \{\times, =\})$, $(R, \{\times, =\})$ виражається формулою $\forall y(x \times y = y)$.

Приклад 3.4.3. Предикат "$x = 0$" в АС $(N, \{+, =\})$, $(Z, \{+, =\})$, $(R, \{+, =\})$ виражається формулою $x+x = x$.

Приклад 3.4.4. Предикат "$x = 1$" в АС $(N, \{+, =\})$ виражається формулою $\forall u \forall v(x = u+v \to u = u+u \vee v = v+v) \,\&\, \neg x = x+x$.

Приклад 3.4.5. Предикат "$|x-y| = 2$" в АС $(Z, \{|x-y|=1, =\})$ виражається формулою $\exists z(|x-z| = 1 \& |z-y| = 1 \& \neg x = y$.

Приклад 3.4.6. Предикат "$|x-y| = 3$" в АС $(Q, \{y = x+3, =\})$ виражається формулою $y = x+3 \vee = y+3$.

Приклад 3.4.7. Предикат "$z = x+1$" виражається в АС $(Z, \{<, =\})$ формулою $(x<z) \& \neg \exists v(x<v \& v<z)$.

Множину натуральних чисел N з виділеними константами 0 та 1, визначеними на N стандартними бінарними операціями (функціями) додавання (+) і множення ($\times$) та стандартним бінарним предикатом рівності, назвемо *стандартною інтерпретацією*, або *стандартною моделлю* мови арифметики. Іншими словами, стандартна інтерпретація L_{ar} – це АС $\boldsymbol{N} = (N, \sigma_{ar})$.

Арифметична формула, яка істинна на $\boldsymbol{N}$, називається *істинною арифметичною формулою* (ІАФ).

Кожна всюди істинна арифметична формула є ІАФ, але не кожна ІАФ всюди істинна. Наприклад, формула $\neg \exists x(x+1 = 0)$ є ІАФ, але вона не істинна на АС $\boldsymbol{Z} = (Z, \sigma_{ar})$ та $\boldsymbol{R} = (R, \sigma_{ar})$.

Предикати, множини та функції, виразні в $\boldsymbol{N} = (N, \sigma_{ar})$, назвемо *арифметичними.*

Отже, функція f є арифметичною, якщо її графік Γ_f є арифметичною множиною.

Звідси маємо: арифметична формула Φ виражає функцію f, якщо Φ виражає предикат "$y = f(x_1, \ldots, x_n)$".

Приклад 3.4.8. Предикати "x є парним числом" та "x ділиться на y" є арифметичними, вони виражаються формулами $\exists y(x = y{+}y)$ та $\exists z(x = y{\times}z)$.

Приклад 3.4.9. Предикат "x є простим числом" арифметичний. Він виражається формулою $\forall y \forall z(x = y{\times}z \to y = 1 \vee z = 1)\&\neg x = 1$.

Приклад 3.4.10. Предикати "$x \le y$" та "$x < y$" арифметичні, вони виражаються арифметичними формулами $\exists z(x{+}z = y)$ та $\exists z(x{+}z = y \& x \ne y)$.

Використовуючи приклад 3.4.10, в записах арифметичних формул надалі вживатимемо скорочення вигляду $x \le y$ та $x < y$.

Приклад 3.4.11. Предикат "$x \le y$" в АС $\boldsymbol{N} = (N, \sigma_{ar})$, $\boldsymbol{R} = (R, \sigma_{ar})$, $\boldsymbol{Z} = (Z, \sigma_{ar})$ виражається різними арифметичними формулами.

Справді, для $\boldsymbol{N}$ маємо $\exists z(x{+}z = y)$; для $\boldsymbol{R}$ маємо $\exists z(x{+}z{\times}z = y)$, для $\boldsymbol{Z}$ маємо $\exists z \exists u \exists v \exists w(x{+}z{\times}z{+}u{\times}u{+}v{\times}v{+}w{\times}w = y)$.

Питання для самоконтролю

1. Які логічні операції використовуються в логіках 1-го порядку?
2. Що є семантичними моделями логік 1-го порядку?
3. Що таке іменна множина? Наведіть приклади іменних множин.
4. Які ви знаєте операції над іменними множинами?
5. Чому для іменних множин операція $\cup$ не завжди визначена? Наведіть відповідні приклади.
6. Дайте визначення операції ∇ накладки іменних множин.
7. Що таке фінітна іменна множина?
8. Що таке квазіарна функція?
9. Дайте визначення V-квазіарної функції на A, V-квазіарного предиката на A.
10. Що таке фінарна (скінченно-арна) функція?

11. Дайте визначення X-арної функції.
12. Дайте визначення n-арної функції.
13. Як ви розумієте властивість еквітонності? Дайте визначення еквітонної функції.
14. Як ви розумієте неістотність предметного імені для функції? Дайте відповідне визначення.
15. Дайте визначення композицій квантифікації $\exists \boldsymbol{x}$ та $\forall \boldsymbol{x}$.
16. Наведіть основні властивості композицій $\exists \boldsymbol{x}$ та $\forall \boldsymbol{x}$.
17. Чи справджується $\exists \boldsymbol{x} P \Rightarrow P$? Наведіть відповідний контрприклад.
18. Чи справджується $P \Rightarrow \forall \boldsymbol{x} P$? Наведіть відповідний контрприклад.
19. Чи справджується $\exists \boldsymbol{x}(P \,\&\, Q) = \exists \boldsymbol{x} P \,\&\, \exists \boldsymbol{x} Q$? Наведіть відповідний контрприклад.
20. Чи справджується $\forall \boldsymbol{x} P \vee \forall \boldsymbol{x} Q = \forall \boldsymbol{x}(P \vee Q)$? Наведіть відповідний контрприклад.
21. Дайте визначення агебраїчної системи. Наведіть приклади АС.
22. Що таке носій агебраїчної системи?
23. Що таке функціональні символи? Предикатні символи? Константні символи?
24. Що таке сигнатура агебраїчної системи?
25. Дайте визначення АС з доданою сигнатурою.
26. Що таке підсистема АС? Надсистема АС?
27. Що таке розширення АС? Звуження АС?
28. За яких умов множина $C \subseteq A$ утворює підсистему $\boldsymbol{C} = (C, \sigma)$ алгебраїчної системи $\boldsymbol{A} = (A, \sigma)$? Наведіть відповідні приклади.
29. Що таке перетин підсистем?
30. Що таке найменша підсистема агебраїчної системи? Чи завжди вона існує? Наведіть відповідні приклади.
31. Дайте визначення підсистеми АС (A, σ), породженої множиною $B \subseteq A$.
32. Дайте визначення: АС (A, σ) породжується підмножиною $B \subseteq A$. Наведіть приклади.
33. Що входить до алфавіту мови 1-го порядку?
34. Опишіть логічні та нелогічні символи мови 1-го порядку.
35. Що таке сигнатура мови 1-го порядку?
36. Що таке терми та формули мови 1-го порядку? Для чого вони використовуються?
37. Дайте визначення терма мови 1-го порядку.

38. Дайте визначення атомарної формули мови 1-го порядку.
39. Дайте визначення формули мови 1-го порядку.
40. Вкажіть пріоритет символів логічних операцій (композицій).
41. У чому полягають відмінності мов 1-го порядку?
42. Що таке розширення мови 1-го порядку? Звуження мови 1-го порядку?
43. Що таке область дії квантора?
44. Що таке кванторний префікс?
45. Дайте визначення зв'язаного та вільного входження змінної у формулу. Наведіть приклади.
46. Дайте визначення вільної змінної формули. Наведіть приклади.
47. Дайте визначення замкненого терма. Наведіть приклади.
48. Дайте визначення замкненої формули. Наведіть приклади.
49. Яка сигнатура мови арифметики?
50. Що таке арифметичний терм? Наведіть приклади.
51. Що таке арифметична формула? Наведіть приклади.
52. Дайте визначення мови теорії множин. Наведіть приклади формул мови теорії множин.
53. Дайте визначення мови теорії впорядкованих множин. Наведіть приклади формул цієї мови.
54. Що таке колізія? Як уникати колізій?
55. За яких умов терм допустимий для заміни вільного імені в формулі?
56. Що таке інтерпретація (модель) мови 1-го порядку?
57. Що таке область інтерпретації?
58. Як інтерпретуються символи мови 1-го порядку?
59. Як визначається відображення інтерпретації термів і мови?
60. Дайте визначення формули, істинної при інтерпретації на $\boldsymbol{A}$. Наведіть приклади.
61. Дайте визначення всюди істинної формули. Наведіть приклади.
62. Дайте визначення формули, виконуваної при інтерпретації на $\boldsymbol{A}$. Наведіть приклади.
63. Дайте визначення виконуваної формули. Наведіть приклади.
64. Що таке замикання формули? Наведіть приклади.
65. Сформулюйте семантичну теорему замикання:
66. Що таке пропозиційно нерозкладна формула?
67. Як визначається істиннісна оцінка мови 1-го порядку?

68. Що таке тавтологія мови 1-го порядку?
69. Як співвідносяться класи тавтологій та всюди істинних формул?
70. Дайте визначення тавтологічного наслідку. Наведіть приклади.
71. Дайте визначення тавтологічної еквівалентності. Наведіть приклади.
72. Дайте визначення логічного наслідку. Наведіть приклади.
73. Дайте визначення логічної еквівалентності. Наведіть приклади.
74. Дайте визначення слабкого логічного наслідку. Наведіть приклади.
75. Що таке логічний наслідок скінченної множини формул?
76. Вкажіть основні властивості відношень $\models$, $|=$, $\|=$ та $\sim$.
77. Дайте визначення відношення логічного наслідку для множин формул. Наведіть приклади.
78. Чи вірно, шо відношення логічного наслідку для множин формул рефлексивне? Транзитивне? Відповідь аргументуйте.
79. Вкажіть основні властивості відношення логічного наслідку для множин формул.
80. Сформулюйте теорему заміни еквівалентних для логік 1-го порядку.
81. Що таке *k*-істинна формула?
82. Що таке скінченно-істинна формула?
83. Наведіть алгоритм розв'язання проблеми *k*-істинності.
84. Як співвідносяться класи скінченно-істинних та всюди істинних формул?
85. Сформулюйте семантичну теорему еквівалентності.
86. Сформулюйте семантичну теорему рівності для термів.
87. Сформулюйте семантичну теорему рівності для формул.
88. Що таке варіанта? Наведіть приклади.
89. Сформулюйте семантичну теорему про варіанту.
90. Що таке пренексна форма?
91. Що таке пренексна формула?
92. Які ви знаєте пренексні операції?
93. Сформулюйте теорему про пренексну форму.
94. Що таке виразний предикат? Наведіть приклади.
95. Що таке виразна множина? Наведіть приклади.
96. Що таке виразна функція? Наведіть приклади.
97. Що таке стандартна інтерпретація (модель) мови арифметики?
98. Що таке істинна арифметична формула? Наведіть приклади.

99. Що таке арифметичний предикат, арифметична множина, арифметична функція? Наведіть приклади.

Вправи

1. Доведіть, що система $(Z, \{+, -, \times, =\})$ має найменшу підсистему $(\{0\}, \{+, -, \times, =\})$.

2. Чи має найменшу підсистему система $(Z^+, \{+, =\})$?

3. Доведіть, що система $(N, \{\times, =\})$ не породжується жодною скінченою підмножиною $B \subseteq N$.

4. Вкажіть формули L_{ar}, що виражають такі предикати:
   1) "існує більше трьох парних чисел";
   2) "не існує простих чисел, кратних 10";
   3) "існує не більше трьох повних кубів";
   4) "існує менше чотирьох повних квадратів";
   5) "існує рівно 3 парних чисел, що є точними кубами";
   6) "існує більше двох простих чисел, не кратних 5";
   7) "невірно, що існує рівно 2 числа, що є сумою 4-х квадратів";
   8) "існує менше чотирьох парних непростих чисел";
   9) "множина парних чисел нескінченна";
   10) "існує єдине парне просте число";
   11) "існує принаймі 4 непарних простих чисел";
   12) "кожне парне число, більше за 2, є сумою двох простих чисел".

5. Вкажіть формули L_{ar}, що виражають такі функції:
   1) функції $x-y$; $x \dot{-} y$, $|x-y|$;
   2) функції $[x/y]$; $\mathrm{mod}(x, y)$;
   3) функції $HCD(x, y)$; $HCK(x, y)$;
   4) функції $[\sqrt{x}\,]$; $[\sqrt[x]{y}\,]$;
   5) функції $\mathrm{mod}(x, [y/z])$;
   6) функції $HCK([\sqrt{x}\,], y)$;
   7) функції $[\sqrt{HCD([x/y], z)}]$.

6. Вкажіть формули L_{set}, які означають:
   1) "$X \subset Y$";
   2) "$X = Y \cup Z$";
   3) "$X = 2^Y$";

4) “$X = (Y \cap Z) \setminus S$”;
5) “$X = Y \setminus (Z \cup S)$”;
6) “$X \cap Y = Z \setminus S$”;
7) “$Z \cup S = 2^{X \setminus Y}$”.

7. Встановіть, чи вірно:
1) $\models \forall x \exists y A \to \exists y \forall x A$;
2) $\models \exists y \forall x A \to \forall x \exists y A$;
3) $\models \exists x A \vee \exists x B \leftrightarrow \exists x (A \vee B)$;
4) $\models \exists x (A \& B) \leftrightarrow \exists x A \& \exists x B$;
5) $\models \forall x A \& \forall x B \leftrightarrow \forall x (A \& B)$;
6) $\models \forall x (A \vee B) \leftrightarrow \forall x A \vee \forall x B$;
7) $\models \exists x A \vee B \to \mathrm{A} \vee \exists x B$;
8) $\models A \& \forall x B \to \forall x \mathrm{A} \& B$;
9) $\models (A \to B) \to (\exists x A \to \exists x B)$;
10) $\models (A \to B) \to (\forall x A \to \forall x B)$.

8. Чи вірно:
1) а) $\models \exists x (P \& Q) \to \exists x P \& Q$;
б) $\models \exists x P \& Q \to \exists x (P \& Q)$;
в) $\forall x P \to \exists x Q \| = \forall x P \to Q$;
2) а) $P \to Q \models \forall x P \to \forall x Q$;
б) $\forall x P \to \forall x Q \models P \to Q$;
в) $\forall x P \to \forall x Q \| = \exists x P \to \exists x Q$;
3) а) $\forall x P \to Q \models \forall x (P \to Q)$;
б) $\exists x (P \to Q) \models \exists x P \to Q$;
в) $\forall x P \to \exists x Q \| = P \to \exists x Q$;
4) а) $\models \forall x (P \vee Q) \to \forall x P \vee Q$;
б) $\models \forall x P \vee Q \to \forall x (P \vee Q)$;
в) $\exists x P \to \exists x Q \| = \forall x P \to \forall x Q$.

9. Визначте, в якому відношенні щодо $\models$ та $\|=$ перебувають формули вигляду:
1) $A \to B$ та $\exists x A \to \exists x B$;
2) $A \to B$ та $\forall x A \to \forall x B$;
3) $\forall x (A \vee B)$ та $\forall x A \vee \forall x B$;
4) $\exists x A \& \exists x B$ та $\exists x (A \& B)$;
5) $\forall x (A \vee B)$ та $\forall x A \vee B$;
6) $\exists x A \vee B$ та $\exists x (A \vee B)$;

7) $\forall x(A\&B)$ та $\forall xA\&B$;
8) $\exists xA\&B$ та $\exists x(A\&B)$;
9) $\exists x(A\to B)$ та $\exists xA\to B$;
10) $\forall x(A\to B)$ та $\forall xA\to B$.

10. Вкажіть пренексну форму для таких формул:
1) $\forall xA(x)\to\forall y(\exists zB(x, y, z)\to\neg\forall xA(x)\&\exists xC(x, y))$;
2) $\forall x\neg\exists yA(x, y)\to\forall xB(x)\to\neg\exists yA(x, y)$;
3) $\neg\forall xA(x)\&\exists xB(x)\vee\forall x(\forall yC(x, y)\to A(y))$;
4) $\forall x\exists yA(x, y, z)\&\neg\exists xB(x, y)\to\forall xC(x, y)$;
5) $\forall xA(x)\to\forall y(\forall zB(x, y, z)\to\neg\forall xA(x))$;
6) $\exists z(x = y+z)\to(x = y)\vee\exists z((x = y+z)\&\neg(z = 0))$.

4. АКСІОМАТИЧНІ СИСТЕМИ ЛОГІК ПЕРШОГО ПОРЯДКУ

Формально-аксіоматичні системи гільбертівського типу класичних логік 1-го порядку називають численнями 1-го порядку, або теоріями 1-го порядку.

Під *теорією* 1-*го порядку* розумітимемо формальну систему $T = (L, A, P)$, де L – мова 1-го порядку, A – множина аксіом, яка розбита на множину *логічних* аксіом та множину *власних* аксіом, P – множина правил виведення.

Розглянемо кілька зауважень щодо термінології. В широкому плані під теорією 1-го порядку розуміють довільну множину замкнених формул деякої мови 1-го порядку. Інколи це поняття розуміють вужче, як множину замкнених формул деякої мови 1-го порядку, які виводяться з певної множини аксіом (що означає замкненість теорії як множини стосовно правил виведення). В цьому випадку кожна теорія визначається відповідним численням предикатів та множиною власних аксіом. Водночас під численням 1-го порядку часто розуміють тільки числення предикатів 1-го порядку. Поширеним є одночасне вживання термінів "числення предикатів" та "прикладне числення предикатів", коли прикладне числення предикатів 1-го порядку означає те, що ми назвали теорією 1- го порядку.

4.1. Теорії першого порядку

Множина логічних аксіом задається такими схемами аксіом:

Ax1) $\neg\Phi\vee\Phi$ – пропозиційні аксіоми;

Ax2) $\Phi_x[t]\to\exists x\Phi$ – аксіоми підстановки;

Ax3) $x = x$ – аксіоми тотожності;

Ax4) $x_1 = y_1\to\ldots\to x_n = y_n\to fx_1\ldots x_n = fy_1\ldots y_n$ та $x_1 = y_1\to\ldots\to x_n = y_n\to$
$\to px_1\ldots x_n\to py_1\ldots y_n$ – аксіоми рівності.

Множина ПВ P складається з таких правил виведення:

П1) $\Phi \vdash \Psi\vee\Phi$ – правило розширення;

П2) $\Phi\vee\Phi \vdash \Phi$ – правило скорочення;

П3) $\Phi\vee(\Psi\vee\Xi) \vdash (\Phi\vee\Psi)\vee\Xi$ – правило асоціативності;

П4) $\Phi \vee \Psi, \neg\Phi \vee \Xi \vdash \Psi \vee \Xi$ – правило перетину;

П5) $\Phi \to \Psi \vdash \exists x\Phi \to \Psi$, якщо x не вільна в Ψ, – правило $\exists$-введення.

Логічні аксіоми наявні у всіх теоріях 1-го порядку, власні аксіоми визначають специфіку певної теорії.

Теоремою теорії 1-го порядку T називають формулу, яка виводиться із аксіом за допомогою скінченної кількості застосувань ПВ.

Множину теорем теорії T позначатимемо $Th(T)$.

Те, що A – теорема, позначаємо $T \vdash A$, або $\vdash A$, якщо T мається на увазі.

Абстрагуючись від наборів символів логічних операцій, способів запису термів та формул, наборів логічних аксіом і правил виведення, можна стверджувати:

Твердження 4.1.1. Теорія 1-го порядку визначається сигнатурою мови та множиною власних аксіом.

Сигнатурою теорії 1-го порядку називають сигнатуру мови цієї теорії.

Формулу мови теорії називатимемо також формулою теорії.

Теорія T' називається *розширенням* теорії T, якщо кожна формула теорії T є формулою теорії T' та $Th(T) \subseteq Th(T')$. У цьому випадку теорію T називають *звуженням* теорії T'.

Розширення (звуження) T' теорії T *просте,* якщо T та T' мають однакові мови.

Теорії 1-го порядку T_1 та T_2 називаються *еквівалентними*, якщо в них однакові мови та множини теорем.

Потужністю теорії T називають потужність множини $Th(T)$.

Зокрема, теорія 1-го порядку із зліченною сигнатурою зліченна, теорія 1-го порядку із сигнатурою потужності α має потужність α.

Розглянемо кілька прикладів теорій 1-го порядку.

Приклад 4.1.1. Теорія 1-го порядку, яка не містить власних аксіом, називається *численням предикатів 1-го порядку* (скорочено ЧП-1).

Приклад 4.1.2. Особливе місце серед формальних теорій посідає теорія натуральних чисел – *формальна арифметика*. Позначимо її *Ar*.

Мовою *Ar* є мова L_{ar}.

Власні аксіоми *Ar* такі:

Ar1) $\neg(x+1 = 0)$;

Ar2) $x+1 = y+1 \to x = y$;

Ar3) $x+0=x$;
Ar4) $x+(y+1)=(x+y)+1$;
Ar5) $x\times 0=0$;
Ar6) $x\times(y+1)=x\times y+x$;
Ar7) $A_x[0]\ \&\forall x(A\to A_x[x+1])\to\forall xA$ – аксіоми індукції.
Кожна власна аксіома формальної арифметики є ІАФ.

Приклад 4.1.3. *Елементарною теорією груп* називається теорія 1-го порядку *Gr* сигнатури $\{\bullet, e, =\}$, де e – константний символ, $\bullet$ – бінарний функціональний символ. Власні аксіоми *Gr* такі:
G1) $x\bullet(y\bullet z)=(x\bullet y)\bullet z$;
G2) $\forall x(e\bullet x=x)$;
G3) $\forall x\exists y(y\bullet x=e)$.

Теорема 4.1.1. 1) *логічні аксіоми – всюди істинні формули;* 2) *висновки правил* П1–П4 – *тавтологічні наслідки засновків;* 3) *висновок правила* П5 – *слабкий логічний наслідок засновку*.

Доведемо 1) для аксіоми *Ax*3, для інших аксіом твердження очевидне.

Нехай X – множина вільних імен формули $\Phi_x[t]\to\exists x\Phi$. Припустимо супротивне: існує $\boldsymbol{A}=(A,\sigma)$ така, що $\boldsymbol{A}\not\models\Phi_x[t]\to\exists x\Phi$. Тоді існує $d\in A^X$ таке, що $(\Phi_x[t]\to\exists x\Phi)_A(d)=F$, звідки $(\Phi_x[t])_A(d)=T$ та $(\exists x\Phi)_A(d)=F$. Нехай $t_A(d)=b\in A$; в силу $(\Phi_x[t])_A(d)=T$ тоді $\Phi_A(d\nabla x\mapsto b)=T$. Але $(\exists x\Phi)_A(d)=F$, тому для всіх $a\in A$ $\Phi_A(d\nabla x\mapsto a)=F$, зокрема $\Phi_A(d\nabla x\mapsto b)=F$. Дістали суперечність.

Твердження 2) очевидне, залишається довести 3). Нехай X – множина вільних імен формули $\Phi\to\Psi$. Припустимо супротивне: існує інтерпретація $\boldsymbol{A}=(A,\sigma)$ така, що $\boldsymbol{A}\models\Phi\to\Psi$ та $\boldsymbol{A}\not\models\exists x\Phi\to\Psi$. Тоді існує $d\in A^X$ таке, що $(\exists x\Phi\to\Psi)_A(d)=F$, звідки $(\exists x\Phi)_A(d)=T$ та $\Psi_A(d)=F$. В силу $(\exists x\Phi)_A(d)=T$ маємо $\Phi_A(d\nabla x\mapsto b)=T$ для деякого $b\in A$. Але ім'я x не вільне в Ψ, тому $\Psi_A(d\nabla x\mapsto b)=\Psi_A(d)=F$. Звідси дістаємо $(\Phi\to\Psi)_A(d\nabla x\mapsto b)=F$, що суперечить $\boldsymbol{A}\models\Phi\to\Psi$.

Умова "x не вільне в Ψ" істотна для правила П5. Справді, маємо $x=0\to x=0\ \|\not\models\exists x(x=0)\to x=0$, оскільки $x=0\to x=0$ всюди істинна, а згідно з $(\exists x(x=0)\to x=0)_N(1)=F$ маємо $\boldsymbol{N}\not\models\exists x(x=0)\to x=0$.

Для правила П5 $\|=$ не можна посилити до $\models$. Справді, маємо $x=0\to 1=0\ \|=\exists x(x=0)\to 1=0$, але $((x=0\to 1=0)\to(\exists x(x=0)\to 1=0))_N(1)=F$, тому $x=0\to 1=0\not\models\exists x(x=0)\to 1=0$.

Наслідок. *Кожна теорема* ЧП-1 *є всюди істинною формулою*.

Справді, логічні аксіоми всюди істинні, правила виведення зберігають властивість всюди істинності.

Моделлю теорії 1-го порядку T називається інтерпретація мови теорії, на якій істинні всі власні аксіоми теорії T.

Приклад 4.1.5. Моделлю ЧП-1 є кожна інтерпретація його мови.

Приклад 4.1.6. Моделлю елементарної теорії груп Gr є кожна група.

Приклад 4.1.7. Моделлю формальної арифметики Ar є $\boldsymbol{N}$ – стандартна інтерпретація L_{ar}. Таку модель називають *стандартною моделлю* формальної арифметики.

Формула Φ називається *істинною* в теорії T, якщо Φ істинна на кожній моделі теорії T.

Теорема 4.1.2 (теорема істинності). *Кожна теорема теорії 1-го порядку T істинна в T.*

Власні аксіоми істинні в T за визначенням, логічні аксіоми істинні в T, оскільки вони всюди істинні. Правила виведення для кожної інтерпретації $\boldsymbol{A}$ зберігають властивість істинності формул на $\boldsymbol{A}$.

Теорема 4.1.3 (теорема тавтології). *Кожна тавтологія є теоремою.*

Наслідок. *Якщо* $\{\Phi_1, ..., \Phi_n\} \models \Phi$ *та* $\vdash\Phi_1, ..., \vdash\Phi_n$, *то* $\vdash\Phi$.

Розглянемо приклади виведень в теоріях 1-го порядку.

Будемо використовувати теорему тавтології (ТТ).

Теорема 4.1.4 $\vdash\forall xA\to A$.

Маємо $\vdash\neg\neg A\to\exists x\neg A$ (аксіома Aх3), звідси за ТТ $\vdash\neg\neg\exists x\neg A\to A$, тобто $\vdash\forall xA\to A$.

Теорема 4.1.5 (правило $\forall$-введення). *Якщо* $\vdash A\to B$ *та x не вільне в A, то* $\vdash A\to\forall xB$.

Якщо $\vdash A\to B$, то $\vdash\neg\neg B\to\neg A$ за ТТ, звідки $\vdash\exists x\neg B\to\neg A$ за П5. Тоді $\vdash\neg\neg\neg A\to\neg\exists x\neg B$ за ТТ, отже $\vdash A\to\forall xB$.

Теорема 4.1.6 (правило дистрибутивності). *Якщо* $\vdash A\to B$, *то* $\vdash\exists xA\to\exists xB$ *та* $\vdash\forall xA\to\forall xB$.

Маємо $\vdash A \to B$ (умова) та $\vdash B \to \exists x B$ (аксіома Ax3), звідки за ТТ $\vdash A \to \exists x B$, тому за П5 дістаємо $\vdash \exists x A \to \exists x B$. З умови маємо $\vdash \neg\neg B \to \neg A$ за ТТ, маємо $\vdash \neg\neg A \to \exists x \neg A$ (аксіома Ax3), звідси за ТТ $\vdash \neg\neg B \to \exists x \neg A$. За П5 $\vdash \exists x \neg B \to \exists x \neg A$, тому $\vdash \neg\neg\exists x \neg A \to \neg\neg\exists x \neg B$ за ТТ, тобто $\vdash \forall x A \to \forall x B$.

Теорема 4.1.7 (правило узагальнення). *Якщо* $\vdash A$, *то* $\vdash \forall x A$.

Якщо $\vdash A$, то за П1 $\vdash \forall x A \vee A$, звідки за ТТ $\vdash \neg\neg A \to \forall x A$. Тоді $\vdash \exists x \neg A \to \forall x A$ за П5, тобто $\vdash \forall x A \vee \forall x A$. Тепер $\vdash \forall x A$ за П2.

Теорема 4.1.8 (правило уособлення). *Якщо* $\vdash \forall x$A, *то* $\vdash$A.

За теоремою 4.1.4 $\vdash \forall x A \to A$. Звідси і з умови $\vdash \forall x A$ за МР маємо $\vdash A$.

Теорема 4.1.9 (теорема замикання). $\vdash A \Leftrightarrow \vdash \overline{A}$, де $\overline{A}$ – замикання формули A.

Негайно випливає з теорем 4.1.7 та 4.1.8.

Теорема 4.1.10 (ПП). *Якщо* $\vdash$A, *то* $\vdash A_{x_1, \ldots, x_n}[t_1, \ldots, t_n]$.

Маємо $\vdash \neg\neg A_x[t] \to \exists x \neg A$ (аксіома $Ax2$), тому $\vdash \forall x A \to A_x[t]$ за ТТ. Із $\vdash A$ за правилом узагальнення $\vdash \forall x A$, тому за МР $\vdash A_x[t]$.

Нехай предметні імена $y_1, \ldots, y_n$ не входять до складу формул A та $A_{x_1, \ldots, x_n}[t_1, \ldots, t_n]$. Позначимо В формулу $A_{x_1, \ldots, x_n}[y_1, \ldots, y_n]$. Згідно $\vdash A_x[t]$ послідовно маємо $\vdash A_{x_1}[y_1]$, $\vdash A_{x_1, x_2}[y_1, y_2]$, ..., $\vdash A_{x_1, \ldots, x_n}[y_1, \ldots, y_n]$, $\vdash B_{y_1}[t_1]$, ..., $\vdash B_{y_1, \ldots, y_n}[t_1, \ldots, t_n]$. Але формула $B_{y_1, \ldots, y_n}[t_1, \ldots, t_n]$ – це і є формула $A_{x_1, \ldots, x_n}[t_1, \ldots, t_n]$.

Теорема 4.1.10 дає змогу ввести похідне правило виведення:

правило підстановки (ПП): $\text{A} \vdash A_{x_1, \ldots, x_n}[t_1, \ldots, t_n]$.

Теорема 4.1.11 (теорема підстановки). $\vdash A_{x_1, \ldots, x_n}[t_1, \ldots, t_n] \to \exists x_1 \ldots \exists x_n A$ *та* $\vdash \forall x_1 \ldots \forall x_n A \to A_{x_1, \ldots, x_n}[t_1, \ldots, t_n]$.

Як $Ax3$ маємо $\vdash A \to \exists x_n$A, ..., $\vdash \exists x_{i+1} \ldots \exists x_n \text{A} \to \exists x_i\, \exists x_{i+1} \ldots \exists x_n$A, ..., $\vdash \exists x_2 \ldots \exists x_n \text{A} \to \exists x_1 \ldots \exists x_n$A, звідки $\vdash \text{A} \to \exists x_1 \ldots \exists x_n$A за ТТ. Тепер за ПП дістаємо $\vdash A_{x_1, \ldots, x_n}[t_1, \ldots, t_n] \to \exists x_1 \ldots \exists x_n A$.

Використовуючи теорему 4.1.4, замість аксіоми $Ax3$, аналогічно попередньому маємо $\vdash \forall x_1 \ldots \forall x_n A \to A$, звідки за ПП отримуємо $\vdash \forall x_1 \ldots \forall x_n A \to A_{x_1, \ldots, x_n}[t_1, \ldots, t_n]$.

Теорема 4.1.12 (правило симетрії). *Для довільних термів a та b маємо* $\vdash a = b \leftrightarrow b = a$.

Маємо $\vdash x = y \to x = x \to x = x \to y = x$ (аксіома рівності для ПС =), звідки $\vdash x = x \to x = x \to y = x \to y = x$ за ТТ. Але $\vdash x = x$ (аксіома тотожності), тому послідовно $\vdash x = x \to y = x \to y = x$ та $\vdash x = y \to y = x$ за МР. Аналогічно $\vdash y = x \to x = y$, тому $\vdash x = y \leftrightarrow y = x$ за ТТ. Звідси за ПП $\vdash a = b \leftrightarrow b = a$.

Теорема 4.1.13 (*правило транзитивності*). *Для довільних термів a, b та c маємо* $\vdash a = b \to b = c \to a = c$.

Маємо $\vdash y = x \to y = z \to y = y \to x = z$ (аксіома рівності для ПС =), звідки $\vdash y = y \to y = x \to y = z \to x = z$ за ТТ. Але $\vdash y = y$ (аксіома тотожності), тому за МР $\vdash y = x \to y = z \to x = z$. Згідно з правилом симетрії $\vdash x = y \to y = x$, тому $\vdash x = y \to y = z \to x = z$ за ТТ. За ПП дістаємо $\vdash a = b \to b = c \to a = c$.

Нехай Г – деяка множина формул. Теорію 1-го порядку, отриману із теорії Т додаванням формул множини Г як власних аксіом, будемо позначати Т[Г]. Якщо $\Gamma = \{\Psi_1, ..., \Psi_n\}$, то замість $T[\{\Psi_1, ..., \Psi_n\}]$ пишемо $T[\Psi_1, ..., \Psi_n]$. При $\Gamma = \{\Psi\}$ пишемо $T[\Psi]$.

Теорема 4.2.14 (теорема дедукції). *Нехай* А – *замкнена формула. Тоді для довільної формули B маємо*: $T \vdash A \to B \Leftrightarrow T[A] \vdash B$.

Доводимо індукцією за довжиною виведення в T формули B із A.

Нехай B – аксіома T. Тоді $T \vdash B$, звідки за П1 $T \vdash A \to B$.

Нехай B є формулою A. Тоді $T \vdash A \to A$ як AxПР, тобто $T \vdash A \to B$.

Нехай B отримана із C за допомогою одного з правил П1, П2, П3. Тоді $T \vdash C \to B$ за ТТ. За припущенням індукції в силу $T[A] \vdash C$ маємо $T \vdash A \to C$. Звідси $T \vdash A \to B$ за ТТ.

Нехай B отримана із C та D за допомогою П4. Тоді за ТТ $T \vdash C \& D \to B$. За припущенням індукції в силу $T[A] \vdash C$ та $T[A] \vdash D$ маємо $T \vdash A \to C$ та $T \vdash A \to D$. Тепер $T \vdash A \to B$ за ТТ.

Нехай на останньому кроці виведення B в $T[A]$ B отримана із $C \to D$ за допомогою П5, тобто B суть формула $\exists x C \to D$. За припущенням індукції $T \vdash A \to C \to D$, звідки $T \vdash C \to A \to D$ за ТТ. Але x не вільне в D, формула A замкнена, тому x не вільне в $A \to D$. За П5 тоді $T \vdash \exists x C \to (A \to D)$. Тепер за ТТ $T \vdash A \to \exists x C \to D$, тобто $T \vdash A \to B$.

Наслідок. *Нехай* $A_1, ..., A_n$ – *замкнені формули. Тоді для довільної формули B маємо* $T \vdash A_1 \to ... \to A_n \to B \Leftrightarrow T[A_1, ..., A_n] \vdash B$.

Якщо не вимагати замкненості формули A, теорема дедукції невірна. Справді, $Ar[x = 0] \vdash x = 0$, тому $Ar[x = 0] \vdash y = 0$ за прави-

лом підстановки. Але формула $x = 0 \to y = 0$ не є ІАФ, тому невірно $Ar \vdash x = 0 \to y = 0$.

Для теорій 1-го порядку вірні синтаксичні варіанти теорем рівності, еквівалентності, теорем про зведення до A' пренексної форми.

Теорема 4.1.15 (теорема еквівалентності). *Нехай A' отримана із формули* A *заміною деяких входжень формул $B_1, ..., B_n$ на $P_1, ..., P_n$ відповідно. Якщо $\vdash B_1 \leftrightarrow P_1, ..., \vdash B_n \leftrightarrow P_n$, то $\vdash A \leftrightarrow A'$.*

Теорема 4.1.16 (теорема рівності для термів). *Нехай терм* τ' *отриманий із терма* τ *заміною деяких входжень термів $t_1, ..., t_n$ на терми $s_1, ..., s_n$ відповідно. Якщо $\vdash t_1 = s_1, ..., \vdash t_n = s_n$,* то $\vdash \tau = \tau'$.

Теорема 4.1.17 (теорема рівності для формул). *Нехай формула* Φ' *отримана із формули* Φ *заміною деяких входжень термів $t_1, ..., t_n$ на терми $s_1, ..., s_n$ відповідно. Якщо $\vdash t_1 = s_1, ..., \vdash t_n = s_n$,* то $\vdash \Phi \leftrightarrow \Phi'$.

Теорема 4.1.18. *Якщо A' – варіанта формули* A, *то $\vdash A \leftrightarrow A'$.*

Теорема 4.1.19. 1) $\vdash \neg\forall xB \leftrightarrow \exists x\neg B$ *та* $\vdash \neg\exists xB \leftrightarrow \forall x\neg B$;
2) $\vdash \exists xB \vee C \leftrightarrow \exists x(B \vee C)$ *та* $\vdash \forall xB \vee C \leftrightarrow \forall x(B \vee C)$, *якщо x не вільна в C*;
3) $\vdash B \vee \exists xC \leftrightarrow \exists x(B \vee C)$ *та* $\vdash B \vee \forall xC \leftrightarrow \forall x(B \vee C)$, *якщо x не вільна у B*.

Теорема 4.1.20. *Якщо A' − пренексна форма формули A, то $\vdash A \leftrightarrow A'$.*

4.2. Несуперечливість, повнота, розв'язність теорій першого порядку

Теорія 1-го порядку T називається *несуперечливою*, якщо не існує формули Φ такої, що $T \vdash \Phi$ та $T \vdash \neg\Phi$.

Несуперечлива теорія 1-го порядку T називається *повною*, якщо для кожної замкненої формули Φ маємо $T \vdash \Phi$ або $T \vdash \neg\Phi$.

Теорема 4.2.1. *Теорія 1-го порядку T суперечлива $\Leftrightarrow T \vdash S$ для кожної формули S мови теорії T.*

Імплікація справа наліво очевидна. Покажемо імплікацію зліва направо. Якщо T суперечлива, то існує замкнена формула Φ така, що $T \vdash \Phi$ та $T \vdash \neg\Phi$. Звідси за П1 маємо відповідно $T \vdash S \vee \Phi$ та $T \vdash S \vee \neg\Phi$,

звідки $T \vdash \Phi \vee S$ та $T \vdash \neg\Phi \vee S$ за ПК. Тепер $T \vdash S \vee S$ за П4, звідки $T \vdash S$ за П2.

***Теорема* 4.2.2.** *Числення предикатів 1-го порядку неповне.*

Позначимо S замкнену формулу $\forall x \forall y(x = y)$, істинну тільки на 1-елементних інтерпретаціях. Тоді $\neg S$ істинна на всіх n-елементних інтерпретаціях, де $n > 1$. Якщо $\vdash S$, то $\models S$, що неможливо; якщо ж $\vdash \neg S$, то $\models \neg S$, що теж неможливо.

Теорема 4.2.3 (теорема несуперечливості). *Нехай T − теорія 1-го порядку, A − замкнена формула така, що* $A \notin \mathrm{Th}(T)$**.** *Тоді теорія $T[\neg A]$ несуперечлива.*

Припустимо супротивне: $T[\neg A]$ суперечлива. Тоді за теоремою 4.2.1 $T[\neg A] \vdash A$, звідки $T \vdash \neg A \to A$ за теоремою дедукції. Тому за ТТ $T \vdash A$, що суперечить $A \notin Th(T)$.

Теорема 4.2.4. *Нехай T – теорія 1-го порядку, A – замкнена формула. Тоді маємо*: $T \vdash A \Leftrightarrow T[\neg A]$ *суперечлива.*

Нехай $T \vdash A$. Тоді $T[\neg A] \vdash A$. Але ж $T[\neg A] \vdash \neg\neg A$, бо $\neg A$ є аксіомою $T[\neg A]$, тому $T[\neg A]$ суперечлива.

Нехай тепер $T[\neg A]$ суперечлива. Тоді $T[\neg A] \vdash A$, звідки $T \vdash \neg\neg A \to A$ за теоремою дедукції. Звідси $T \vdash A$ за ТТ.

Теорема 4.2.5 (Лінденбаум)**.** *Кожна несуперечлива теорія 1-го порядку має несуперечливе просте повне розширення.*

Доводимо для випадку зліченних теорій.

Нехай T – несуперечлива теорія 1-го порядку. Множина всіх формул T зліченна, тому множина всіх замкнених формул T також зліченна. Нехай $B_0, B_1, ..., B_n, ...$ – перелік всіх замкнених формул T.

Задамо послідовність $\Sigma_0, \Sigma_1, ..., \Sigma_n, ...$ теорій таким чином:

$$\Sigma_0 = T;$$

$$\Sigma_{n+1} = \begin{cases} \Sigma_n, \text{ якщо } \neg B_n \in Th(\Sigma_n), \\ \Sigma_n[B_n], \text{ якщо } \neg B_n \notin Th(\Sigma_n). \end{cases}$$

Індукцією по n доведемо, що кожна із теорій Σ_n несуперечлива.

Теорія $\Sigma_0 = T$ несуперечлива за умовою. Нехай Σ_n несуперечлива. Тоді Σ_{n+1} теж несуперечлива. Справді, якщо $\Sigma_{n+1} = \Sigma_n$, то Σ_{n+1} не-

суперечлива за припущенням. Якщо $\Sigma_{n+1} = \Sigma_n[B_n]$, то Σ_{n+1} несуперечлива за теоремою 4.2.3.

Нехай Σ – теорія 1-го порядку, множина аксіом якої є об'єднанням множин аксіом всіх теорій Σ_n. Тоді Σ несуперечлива. Справді, якщо Σ суперечлива, то в Σ існує виведення суперечності – формули вигляду $A\&\neg A$. Таке виведення використовує скінченну кількість аксіом, тому всі ці аксіоми є аксіомами теорії Σ_n для деякого n. Тому $\Sigma_n \vdash A\&\neg A$, що неможливо, оскільки Σ_n несуперечлива.

Теорія Σ повна. Справді, кожна замкнена формула є формулою B_m для деякого m. Якщо $\neg B_m \in Th(\Sigma_m)$, то $\neg B_m \in Th(\Sigma)$, тобто маємо $\Sigma \vdash \neg B_m$. Якщо $\neg B_m \notin Th(\Sigma_m)$, то $\Sigma_{m+1} = \Sigma_m[B_m]$, звідки $\Sigma_{m+1} \vdash B_m$, тому маємо $\Sigma \vdash Bm$. Отже, теорія Σ повна.

Розглянемо важливі поняття розв'язності та перелічності аксіоматичних теорій. Розв'язність теорії означає алгоритмічну розв'язність множини її теорем відносно множини всіх формул мови теорії. Перелічність означає, що множина теорем теорії алгоритмічно перелічна.

Найпростішим прикладом розв'язної теорії є пропозиційне числення.

Теорема 4.2.6. *Нехай T – теорія 1-го порядку із алгоритмічно перелічною множиною аксіом. Тоді T перелічна.*

Нехай $\aleph$ – алгоритм для переліку аксіом теорії T. Тоді алгоритм, який перелічує всі теореми теорії T, по черзі виконує такі дії:

1) видати алгоритмом $\aleph$ чергову аксіому A;

2) до множини вже отриманих теорем одноразово застосувати всіма можливими способами правила П1–П5;

3) отримані нові теореми та нову аксіому A по черзі подати на вихід і поповнити ними множину теорем.

Теорема 4.2.7 (розв'язності). *Нехай T – повна теорія 1-го порядку із алгоритмічно перелічною множиною аксіом. Тоді T розв'язна.*

Нехай $\aleph$ – алгоритм для переліку аксіом теорії T. Вкажемо алгоритм $\Re$, який за кожною формулою S встановлює, $T \vdash S$ чи $T \vdash \neg S$, тобто $S \in Th(T)$ чи $S \notin Th(T)$. Спершу алгоритм $\Re$ за формулою S будує її замикання $\overline{S}$, потім по черзі виконує такі дії:

1) видати алгоритмом $\aleph$ чергову аксіому A;

2) до множини вже отриманих теорем одноразово застосувати всіма можливими способами правила П1–П5; множину отриманих таким чином нових формул позначимо **Fr**;

3) а) якщо $\overline{S} \in \mathbf{Fr} \cup \{A\}$, то видати результат "так" і зупинитись;

б) якщо $\neg \overline{S} \in \mathbf{Fr} \cup \{A\}$, то видати результат "ні" і зупинитись;

в) якщо $\overline{S} \notin \mathbf{Fr} \cup \{A\}$ та $\neg \overline{S} \notin \mathbf{Fr} \cup \{A\}$, то поповнити множину теорем формулами із $\mathbf{Fr} \cup \{A\}$ і перейти до 1).

За теоремою замикання $T \vdash S \Leftrightarrow T \vdash \overline{S}$ та $T \vdash \neg S \Leftrightarrow T \vdash \neg \overline{S}$. При $T \vdash \overline{S}$ маємо $T \vdash S$; при $T \vdash \neg \overline{S}$ за несуперечливістю T неможливе $T \vdash \overline{S}$, тому неможливе $T \vdash S$. Але $T \vdash \overline{S}$ або $T \vdash \neg \overline{S}$ за повнотою T, тому $\Re$ завжди зупиниться і видасть разультат.

Наслідок. *Нехай T – повна перелічна теорія 1-го порядку. Тоді T розв'язна.*

Повертаючись до доведення теореми Лінденбаума, зауважимо, що множина аксіом теорії Σ_n далеко не у всіх випадках алгоритмічно перелічна, оскільки проблема $\neg B_n \in Th(\Sigma_n)$, взагалі кажучи, алгоритмічно нерозв'язна. Тому в загальному випадку теорія Σ – несуперечливе просте повне розширення теорії T – неперелічна.

4.3. Теорема Гьоделя про повноту

Теорема Гьоделя про повноту засвідчує повноту логічних засобів логік 1-го порядку. Вона стверджує, що логічних засобів теорії 1-го порядку, тобто її аксіом та правил виведення, достатньо для виведення кожної істинної в теорії формули. Іншими словами, теорема про повноту засвідчує адекватність семантичної та синтаксичної істинності.

Традиційне доведення теореми про повноту базується на побудові моделі для несуперечливої теорії.

Теорема 4.3.1 (про модель). *Нехай T – несуперечлива теорія 1-го порядку потужності α. Тоді T має модель потужності $\leq \alpha$.*

Теорема 4.3.2 (Гьоделя про повноту, 1-е формулювання). *Нехай T – теорія 1-го порядку. Тоді формула Φ істинна в $T \Leftrightarrow T \vdash \Phi$.*

Теорема 4.3.3 (Гьоделя про повноту, 2-е формулювання). *Теорія 1-го порядку T несуперечлива $\Leftrightarrow$ T має модель.*

Якщо T несуперечлива, то T має модель згідно з теоремою 4.3.1. Якщо T має модель $\boldsymbol{M}$, то кожна формула вигляду $\neg A \& A$ на $\boldsymbol{M}$ хибна, тому $T \vdash \neg A \& A$ неможливо, звідки T несуперечлива.

Теорема 4.3.2 випливає із теореми 4.3.3. Справді, нехай $\bar{\Phi}$ – замикання формули Φ. Тоді маємо: $T \vdash \Phi \Leftrightarrow T \vdash \bar{\Phi}$ (за теоремою замикання) $\Leftrightarrow$ $T[\neg\bar{\Phi}]$ суперечлива (за теоремою 4.2.3) $\Leftrightarrow$ теорія $T[\neg\bar{\Phi}]$ не має моделі (за теоремою 4.3.3) $\Leftrightarrow$ $\bar{\Phi}$ істинна в T (бо кожна модель для $T[\neg\bar{\Phi}]$ – це така модель $\boldsymbol{M}$ теорії T, для якої $\boldsymbol{M} \not\models \bar{\Phi}$, тому якщо теорія $T[\neg\bar{\Phi}]$ не має моделі, то $\boldsymbol{M} \models \bar{\Phi}$ для кожної моделі $\boldsymbol{M}$ теорії T, тобто $\bar{\Phi}$ істинна в T) $\Leftrightarrow$ Φ істинна в теорії T (за теоремою замикання).

Теорема Гьоделя про повноту своїм 1-м формулюванням засвідчує адекватність семантичної та синтаксичної істин, тому в такому формулюванні теорему Гьоделя про повноту називають теоремою адекватності.

Із теореми Гьоделя про повноту випливає наступна теорема.

Теорема 4.3.4 (теорема Льовенгейма – Сколема про спуск). *Нехай теорія 1-го порядку потужності* α *має модель, тоді вона має модель потужності* $\leq \alpha$.

Справді, якщо теорія T має модель, то вона несуперечлива. За теоремою 4.3.1 T має модель потужності $\leq \alpha$.

Наслідок. *Якщо зліченна теорія 1-го порядку має нескінченну модель, то вона має зліченну модель.*

В аксіоматичній теорії множин T_{set} (див. [9]) можна строго довести теорему Кантора про неіснування взаємооднозначного відображення множини A на її булеан (множину всіх підмножин) 2^A. За наслідком теореми 4.3.1 T_{set} має зліченну модель, тобто кількість всіх можливих множин зліченна, звідки 2^N теж зліченна! Ця уявна суперечність має назву "*парадокс Сколема"*. Проте справжнього парадоксу тут немає. Хоча множина 2^N в моделі дійсно зліченна, тобто існує бієктивне відображення 2^N на N, це відображення як множина не належить моделі, тобто не є елементом носія моделі. Отже, множина 2^N в моделі незліченна з внутрішнього погляду T_{set}, хоча вона є зліченною із зовнішнього погляду.

Парадокс Сколема свідчить, що кожна аксіоматизація теорії множин у вигляді теорії 1-го порядку із зліченною множиною аксіом не відображає повністю понять "множина", "булеан множини", "зліченна множина", "взаємно-однозначне відображення" і т. п. Ці поняття в принципі не можуть бути адекватно описані за допомогою таких теорій.

Проблема всюди істинності формул 1-го порядку в загальному випадку алгоритмічно нерозв'язна [9, 13]. Водночас існує теорема щодо її часткової розв'язності.

Теорема 4.3.5. *Проблема всюди істинності формул 1-го порядку зліченної сигнатури частково розв'язна.*

За теоремою Гьоделя про повноту для кожної формули A маємо $\vdash A \Leftrightarrow \models A$. Числення предикатів 1-го порядку зліченної сигнатури має перелічну множину аксіом, тому за теоремою 4.2.6 множина його теорем перелічна, звідки проблема "$\vdash A$" частково розв'язна.

Важливим наслідком теореми про повноту є теорема компактності.

Теорія T *скінченно аксіоматизована,* якщо множина її власних аксіом скінченна.

Скінченно аксіоматизованою частиною (САЧ) теорії T називають просте скінченно аксіоматизоване звуження теорії T.

Теорема 4.3.6 (теорема компактності, 1-е формулювання). *Формула* Ф *істинна в* $T \Leftrightarrow$ Ф *істинна в деякій САЧ* $K \subseteq T$.

Нехай Ф істинна в T. За теоремою 4.3.2, тоді $T \vdash$ Ф. Таке виведення використовує тільки скінченну кількість аксіом, тому може бути здійснене в межах деякої *САЧ* $K \subseteq T$. Отже, $K \vdash$ Ф, звідки Ф істинна в K.

Кожна модель теорії T є моделлю кожного її звуження, тому, якщо Ф істинна в деякій САЧ $K \subseteq T$, то Ф істинна в T.

Теорема 4.3.7 (теорема компактності, 2-е формулювання). *Теорія 1-го порядку* T *має модель* $\Leftrightarrow$ *кожна САЧ теорії* T *має модель.*

Якщо T має модель $\boldsymbol{M}$, то всі аксіоми T істинні на $\boldsymbol{M}$. Отже, кожна аксіома із довільної скінченної підмножини аксіом T істинна на $\boldsymbol{M}$. Тому $\boldsymbol{M}$ є моделлю кожної САЧ теорії T.

Якщо кожна САЧ теорії T має модель, то T несуперечлива, бо виведення кожної суперечності (формули вигляду $\neg A \& A$) використовує скінченну кількість аксіом. Тому за теоремою 4.3.1 теорія T має модель.

Розглянемо приклади використання теореми компактності.

Теорема 4.3.8. *Якщо теорія 1-го порядку* T *має скінченні моделі будь-якої великої потужності, то* T *має нескінченну модель.*

Нехай E_n – формула, яка стверджує, що існує $\geq n$ різних елементів, тобто E_n істинна на $\geq n$-елементних інтерпретаціях і тільки на них (на-

приклад, E_2 суть $\exists x \exists y (x \neq y)$). Розглянемо теорію $T_1 = T[E_2, ..., E_n, ...]$, отриману додаванням до T всіх формул E_n для $n \geq 2$ як нових аксіом.

Нехай K – довільна САЧ теорії T_1, m – найбільше число таке, що E_n є аксіомою теорії K (якщо жодна з формул E_n не є аксіомою теорії K, то $m = 1$). Тоді кожна модель теорії T потужності $\geq m$ є моделлю для K.

Отже, кожна САЧ теорії T_1 має модель, звідки за теоремою 4.4.2 T_1 має модель $\boldsymbol{M}$. На $\boldsymbol{M}$ істинна кожна із формул E_n, тому $\boldsymbol{M}$ нескінченна.

Теорема 4.3.9 (теорема Льовенгейма – Сколема про підйом). *Нехай теорія 1-го порядку T потужності α має нескінченну модель. Тоді T має модель довільної потужності $\beta \geq \alpha$.*

Нехай $\boldsymbol{M} = (M, \sigma)$ – нескінченна модель для T; нехай $\{c_i\}_{i \in \beta}$ – множина нових констант потужності β. Розглянемо теорію $T_1 = T[\wp]$ сигнатури $\sigma' = \sigma \cup \{c_i\}_{i \in \beta}$, де $\wp$ складається з усіх можливих формул вигляду $c_i \neq c_j$ для $i \neq j$, $i, j \in \beta$. Кожна САЧ K теорії T_1 містить скінченну кількість нових аксіом вигляду $c_i \neq c_j$, тому всі нові константи з їх складу можна так інтерпретувати на $\boldsymbol{M}$, щоб ці аксіоми були істинними на (M, σ'), звідки (M, σ') – модель для K.

Отже, кожна САЧ теорії T_1 має модель, звідки за теоремою 4.3.7 T_1 має модель, тому T_1 несуперечлива. За теоремою 4.3.1 теорія T_1 має модель $\boldsymbol{M}_1 = (M_1, \sigma')$ потужності $\leq \beta$. Але всі формули вигляду $c_i \neq c_j$ для $i \neq j$ істинні на $\boldsymbol{M}_1$, тому потужність $\boldsymbol{M}_1 \geq \beta$, звідки потужність $\boldsymbol{M}_1$ рівна β. Звідси (M_1, σ) є моделлю для T.

Наслідок 1. *Нехай зліченна теорія 1-го порядку T має нескінченну модель. Тоді T має модель довільної потужності $\beta \geq \alpha$.*

Наслідок 2. *Ar має нескінченні моделі як завгодно великої потужності.*

Моделі формальної арифметики *Ar*, які неізоморфні її стандартній моделі, називаю *нестандартними*, або сколемівськими. Наслідок 2 засвідчує існування незліченних нестандартних моделей. Проте [6] існують навіть зліченні нестандартні моделі!

Існування нестандартних моделей формальної арифметики свідчить про неадекватність, неповноту опису множини натуральних чисел за допомогою аксіом *Ar*. Водночас, використовуючи принцип математичної індукції, можна неформально показати єдиність з точністю до ізоморфізму моделі для *Ar*. Річ тут у тому, що принцип мате-

матичної індукції виконується для всіх властивостей натуральних чисел, а таких властивостей – континуум. Водночас схема аксіом індукції *Ar*7 забезпечує виконання принципу математичної індукції тільки для зліченної множини властивостей натуральних чисел, які можуть бути виражені мовою арифметики.

Таким чином, принцип математичної індукції формалізується в *Ar* неповністю, що свідчить про принципову відмінність між інтуїтивним та формальним його розумінням.

Проте ситуація із формальною арифметикою не настільки песимістична. Якщо вимагати *обчислюваності* функцій "+" та "–", то існує *єдина* з точністю до ізоморфізму модель *Ar* [2].

Отже, весь ефект нестандартності моделей *Ar* спричинений необчислюваністю функцій "+" та "×". Якщо ці функції вважати алгоритмічно обчислюваними, що є цілком природним, то нестандартні моделі зникають.

4.4. Теореми Гьоделя про неповноту

До найвидатніших досягнень сучасної математики, безперечно, можна віднести відомі теореми Гьоделя про неповноту. Вони засвідчують принципову обмеженість формально-аксіоматичного методу побудови достатньо складних математичних теорій. Перша теорема Гьоделя про неповноту встановлює для широкого класу формальних систем, які містять або в яких можна виразити формальну арифметику (навіть не всю її, а тільки певний фрагмент, в якому виразні всі рекурсивні функції), існування тверджень, нерозв'язних в тому сенсі, що твердження та його заперечення невивідні в системі. Друга теорема про неповноту стверджує, що несуперечливість таких систем не можна встановити внутрішніми засобами самих систем.

Теорема 4.4.1 (перша теорема Гьоделя про неповноту). *Якщо формальна арифметика Ar несуперечлива, то Ar неповна.*

Таким чином, навіть ті властивості натуральних чисел, які виражаються мовою арифметики, не можуть адекватно описуватися теорією 1-го порядку з алгоритмічно перелічною множиною аксіом. Справді, за теоремою 4.2.6 множина теорем такої теорії алгоритмічно перелічна, тоді як множина ІАФ неперелічна. Тому кожне перелічне розши-

рення формальної арифметики з необхідністю неповне і має неізоморфні моделі.

Для доведення теореми про неповноту Гьодель використав розроблений ним метод нумерацій. Він побудував конкретну арифметичну формулу S, яка виражає *власну невигідність* (формальна аналогія відомого парадоксу брехуна), звідки отримав $S \not\in Th(Ar)$ та $\neg S \not\in Th(Ar)$.

Першу теорему Гьоделя можна довести для кожної формальної системи, в якій моделюється Ar. Таким чином, кожна достатньо багата несуперечлива аксіоматична система *необхідно* неповна. Більш того, така неповнота має *принциповий* характер.

Розглянемо тепер питання про розв'язність Ar.

Можна встановити нерозв'язність Ar та її несуперечливих розширень, звідки випливає узагальнення першої теореми Гьоделя про неповноту.

Встановлення А. Чорчем нерозв'язності формальної арифметики Ar було одним із перших успіхів теорії алгоритмів. Згодом було доведено, що така нерозв'язність справджується також для всіх несуперечливих розширень Ar.

Теорема 4.4.2. *Нехай T – довільне несуперечливе розширення Ar. Тоді T нерозв'язна.*

Звідси, як наслідок, дістаємо узагальнення першої теореми Гьоделя про неповноту.

Теорема 4.4.3. *Нехай T – довільне несуперечливе розширення Ar, причому T перелічна. Тоді теорія T неповна.*

Якщо T повна, то за теоремою 4.2.8 теорія T розв'язна. Але це суперечить теоремі 4.4.4.

Гьодель побудував арифметичну формулу Con, яка виражає несуперечливість Ar, і довів другу теорему.

Теорема 4.4.5 (друга теорема Гьоделя про неповноту). *Якщо формальна арифметика Ar несуперечлива, то Con не є теоремою Ar.*

Друга теорема Гьоделя про неповноту, як і його перша теорема, справджується для кожної формальної системи, в якій моделюється Ar. Тому для кожної достатньо багатої системи доведення її несуперечливості з необхідністю вимагає засобів, які перебувають за межами самої системи.

Теореми Гьоделя про неповноту свідчать про *принципову обмеженість* формально-аксіоматичного методу.

Таким чином, кожна спроба "втиснути" достатньо багату математичну теорію в межі певної формальної системи неминуче призводить до тверджень, які неможливо ні довести, ні спростувати в межах цієї системи. Для доведення несуперечливості такої системи внутрішніх її засобів недостатньо, ми неминуче мусимо використовувати якісь сильніші, зовнішні щодо системи засоби. Це, звичайно, не означає, що неможливо надійно довести несуперечливість формальної арифметики чи подібних систем. Просто методи доведення несуперечливості ***Ar*** вже не можуть бути фінітними, конструктивними. Нині відомо багато таких доведень різноманітними способами, так що несуперечливість арифметики можна вважати надійно обґрунтованою.

4.5. Секвенційні числення логік першого порядку

Розглянемо секвенційні числення класичних логік кванторного рівня – чистих логік предикатів 1-го порядку.

Враховуючи, що для класичних логік на кванторному рівні базовими є композиції $\vee$, $\neg$ та $\exists \boldsymbol{x}$, для секвенцій наведених формул можна ввести такі базові секвенційні форми (зліва – назва форми):

$$\vdash\neg \ \frac{\dashv A, \Sigma}{\vdash \neg A, \Sigma}; \qquad \dashv\neg \ \frac{\vdash A, \Sigma}{\dashv \neg A, \Sigma};$$

$$\vdash\vee \ \frac{\vdash A, \Sigma \quad \vdash B, \Sigma}{\vdash A \vee B, \Sigma}; \qquad \dashv\vee \ \frac{\dashv A, \dashv B, \Sigma}{\dashv A \vee B, \Sigma};$$

$$\vdash\exists \ \frac{\vdash A_x[y], \Sigma}{\vdash \exists x A, \Sigma}$$ за умови, що вільна змінна $y \notin \Sigma \cup \{\exists x A\}$;

$$\dashv\exists \ \frac{\dashv A_x[z_1], \ldots, \dashv A_x[z_m], \Sigma, \dashv \exists x A}{\dashv \exists x A, \Sigma}.$$

При застосуванні $\dashv\exists$ $\{z_1, \ldots, z_m\}$ – множина вільних імен множини доступних формул секвенції $\dashv\exists x A, \Sigma$ та її наступників. Виділена формула $\exists x A$ висновку продубльована у засновку.

Враховуючи властивості П1–П4 та $\exists_{\dashv}$, $\exists_{\vdash}$, отримуємо теорему.

Теорема 4.5.1. *Нехай* $\Sigma = {}_{|-}\Lambda_{-|}\, K$, $Y = {}_{|-}X_{-|}\, Z$, $\Omega = {}_{|-}\Gamma_{-|}\, \Delta$, *нехай* $\frac{\Sigma}{\Omega}$ *та* $\frac{\Sigma \quad Y}{\Omega}$ *– секвенційні форми. Тоді:*

1) *якщо* $\Lambda \models K$, *то* $\Gamma \models \Delta$;

2) *якщо* $\Lambda \models K$ *та* $X \models Z$, *то* $\Gamma \models \Delta$.

Індукцією за побудовою замкненого секвенційного дерева для секвенції ${}_{|-}\Gamma_{-|}\, \Delta$ доводиться теорема коректності для секвенційних числень класичних логік кванторного рівня – чистих логік предикатів 1-го порядку.

Теорема 4.5.2. *Нехай секвенція* ${}_{|-}\Gamma_{-|}\Delta$ *вивідна. Тоді* $\Gamma \models \Delta$.

Процедура побудови секвенційного дерева для секвенції Σ поділяється на етапи. При цьому кожне застосування секвенційної форми здійснюється лише до скінченної множини *доступних* формул.

На початку кожного етапу виконується *крок доступу*. Це означає, що до списку доступних формул додається по одній формулі з списків ${}_{|-}$-формул та ${}_{-|}$-формул. Якщо в секвенції немає недоступних ${}_{|-}$-формул чи ${}_{-|}$-формул (відповідний список вичерпаний), то на подальших кроках доступу додаємо по одній формулі невичерпаного списку.

Вважатимемо, що на початку побудови секвенційного дерева для секвенції Σ зафіксований деякий список *TN* (взагалі кажучи, нескінченний) предметних імен, що не зустрічаються в формулах секвенції Σ (список "нових" імен).

На початку побудови секвенційного дерева доступна лише пара перших формул списків (або єдина ${}_{|-}$-формула чи ${}_{-|}$-формула, якщо один із списків порожній).

Нехай виконано k етапів процедури. На етапі k+1 перевіряємо, чи буде кожен з листів дерева замкненою секвенцією (беремо до уваги тільки доступні формули секвенцій, хоча ця умова неістотна). Якщо *всі* листи дерева замкнені, то процедура завершена позитивно, ми отримали замкнене секвенційне дерево.

Нехай існують незамкнені листи секвенційного дерева. Для кожного такого листа ξ робимо наступний крок доступу, після чого виконуємо наступні дії, добудовуючи скінченне піддерево з вершиною ξ:

1) активізуємо всі доступні непримітивні формули ξ;

2) по черзі до кожної активної формули застосовуємо відповідну секвенційну форму.

Спершу виконуємо всі ${}_{|-}\exists$-форми. При застосуванні ${}_{|-}\exists$ беремо y як перше незадіяне на шляху від кореня до даного листа ім'я списку *TN*.

Після застосування $\vdash\exists$ до кожної з решти активних формул застосовуємо відповідну їй форму $\vdash\neg$, $\dashv\neg$, $\vdash\vee$, $\dashv\vee$, $\dashv\exists$.

При застосуванні $\dashv\exists$ множина $\{z_1,\ldots,z_m\}$ складається з усіх вільних імен доступних формул листа та його наступників; якщо ж таких імен немає, беремо перше незадіяне ім'я списку *TN*.

При побудові секвенційного дерева можливі такі випадки:

1) процедура завершена позитивно, маємо замкнене дерево;

2) процедура завершена негативно або не завершується, маємо незамкнене дерево. Тоді в дереві існує незамкнений шлях $\wp$. Кожна з формул секвенції Σ зустрінеться на цьому шляху і стане доступною.

Нехай $\boldsymbol{H}$ – множина всіх формул секвенцій шляху $\wp$, W – множина всіх вільних імен формул із $\boldsymbol{H}$.

Далі доводимо, що $\boldsymbol{H}$ – модельна множина [10].

Якщо $\boldsymbol{H}$ – модельна множина, то існують АС $\boldsymbol{A} = (A, \boldsymbol{I})$ з $|A| = |W|$ та ін'єктивна δ: $W \to A$ такі:

1) якщо $\vdash\Phi\in\boldsymbol{H}$, то $\Phi_A(\delta) = T$;

2) якщо $\dashv\Phi\in\boldsymbol{H}$, то $\Phi_A(\delta) = F$.

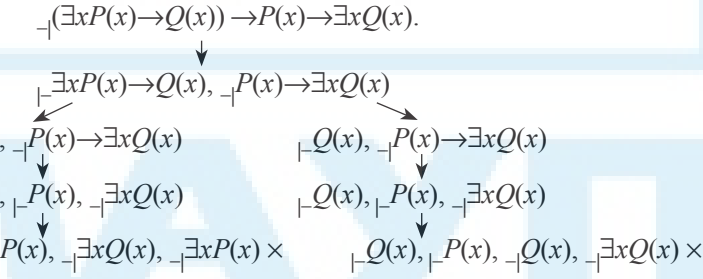
Згідно з $\Sigma\subseteq\boldsymbol{H}$ така оцінка формул заперечує $\Gamma \models \Delta$.

Звідси отримуємо теорему повноти для секвенційних числень чистих логік предикатів 1-го порядку.

Теорема 4.5.3. *Нехай* $\Gamma \models \Delta$. *Тоді секвенція* $\vdash\Gamma\dashv\Delta$ *вивідна.*

Нехай $\Gamma \models \Delta$, але $\vdash\Gamma\dashv\Delta$ невивідна. Тоді в секвенційному дереві для $\Sigma = \vdash\Gamma\dashv\Delta$ існує незамкнений шлях. Як зазначено вище, тоді існує така оцінка формул секвенції Σ, що заперечує $\Gamma \models \Delta$.

Приклад 4.5.1. Для встановлення $\models(\exists xP(x)\to Q(x))\to P(x)\to\exists xQ(x)$ побудуємо виведення секвенції $\dashv(\exists xP(x)\to Q(x))\to P(x)\to\exists xQ(x)$.



Отримали замкнене секвенційне дерево, тому справджується $\models(\exists xP(x)\to Q(x))\to P(x)\to\exists xQ(x)$.

Приклад 4.5.2. Для встановлення вірності $P(x)\to\exists xQ(x) \models \exists xP(x)\to Q(x)$ побудуємо виведення секвенції $\vdash P(x)\to\exists xQ(x), \dashv\exists xP(x)\to Q(x)$.

$$\vdash \exists x\forall yP(x,y), \dashv\forall y\exists xP(x,y)$$
$$\downarrow$$
$$\vdash \forall yP(u,y), \dashv\forall y\exists xP(x,y)$$
$$\downarrow$$
$$\vdash \forall yP(u,y), \dashv\exists xP(x,v)$$
$$\downarrow$$
$$\vdash P(u,u), \vdash P(u,v), \vdash \forall yP(u,y), \dashv\exists xP(x,v)$$
$$\downarrow$$
$$\vdash P(u,u), \vdash P(u,v), \vdash \forall yP(u,y), \dashv P(u,v), \dashv P(v,v) \dashv\exists xP(x,v) \times$$

Отримали незамкнене секвенційне дерево, тому невірно, що $P(x)\to \exists xQ(x) \models \exists xP(x)\to Q(x)$.

Для лівого незамкненого шляху отримуємо контрмодель $\boldsymbol{A}$, для якої $\delta = [x\mapsto a, u\mapsto b]$, $P_A(a) = F$, $\mathrm{P}_A(b) = T$, $\mathrm{Q}_A(a) = F$.

Для правого незамкненого шляху маємо контрмодель $\boldsymbol{B}$, для якої $\delta = [x\mapsto a, u\mapsto b, v\mapsto c]$, $Q_B(b) = T$, $P_B(c) = T$, $Q_B(a) = F$.

Приклад 4.5.3. Для встановлення вірності логічного наслідку $\exists x\forall yP(x,y) \models \forall y\exists xP(x,y)$ побудуємо виведення секвенції $\vdash \exists x\forall yP(x,y)$, $\dashv\forall y\exists xP(x,y)$.

$$\vdash \exists x\forall yP(x,y), \dashv\forall y\exists xP(x,y)$$
$$\downarrow$$
$$\vdash \forall yP(u,y), \dashv\forall y\exists xP(x,y)$$
$$\downarrow$$
$$\vdash \forall yP(u,y), \dashv\exists xP(x,v)$$
$$\downarrow$$
$$\vdash P(u,u), \vdash P(u,v), \vdash \forall yP(u,y), \dashv\exists xP(x,v)$$
$$\downarrow$$
$$\vdash P(u,u), \vdash P(u,v), \vdash \forall yP(u,y), \dashv P(u,v), \dashv P(v,v) \dashv\exists xP(x,v) \times$$

Отримали замкнене секвенційне дерево, тому справджується $\exists x\forall y\mathrm{P}(x,y) \models \forall y\exists x\mathrm{P}(x,y)$.

Питання для самоконтролю

1. Які логічні операції використовуються в логіках 1-го порядку?
2. Що таке теорія 1-го порядку?
3. Чому множина аксіом теорії 1-го порядку розподіляється на множину логічних аксіом та множину власних аксіом?
4. Вкажіть множину логічних аксіом теорій 1-го порядку.
5. Що визначає множина власних аксіом?
6. Вкажіть множину правил виведення теорій 1-го порядку.
7. Що таке теорема теорії 1-го порядку?
8. Чим визначається теорія 1-го порядку?
9. Що таке розширення теорії 1-го порядку?
10. Що таке звуження теорії 1-го порядку?
11. Яке розширення теорії 1-го порядку називають простим?
12. Дайте визначення еквівалентних теорій 1-го порядку.
13. Що таке потужність теорії 1-го порядку?
14. Що таке зліченна теорія 1-го порядку?
15. Що таке числення предикатів 1-го порядку?
16. Вкажіть множину власних аксіом формальної арифметики.
17. Наведіть множину власних аксіом елементарної теорії груп.
18. Чому висновок правила П5 не є логічним наслідком засновку.
19. Як співвідносяться класи теорем числення предикатів 1-го порядку та всюди істинних формул?
20. Що таке модель теорії 1-го порядку?
21. Наведіть приклади теорії 1-го порядку та їх моделей.
22. Опишіть моделі числення предикатів 1-го порядку.
23. Що таке стандартна модель формальної арифметики?
24. Що таке істинна в теорії формула?
25. Сформулюйте теорему істинності.
26. Сформулюйте теорему тавтології та її наслідок.
27. Сформулюйте правило $\forall$-введення.
28. Сформулюйте правило $\exists$-дистрибутивності та правило $\forall$-дистрибутивності.
29. Як формулюються правило узагальнення та правило уособлення?
30. Сформулюйте синтаксичну теорему замикання.
31. Наведіть правило підстановки та теорему підстановки.
32. У чому полягає правило симетрії?
33. Сформулюйте правило транзитивності.
34. Як формулюється теорема дедукції?

35. Сформулюйте наслідок теореми дедукції.
36. Чому важлива умова замкненості формули A у формулюванні теореми дедукції?
37. Наведіть синтаксичні варіанти теорем рівності та еквівалентності.
38. Сформулюйте синтаксичні варіанти теорем про зведення до пренексної форми.
39. Дайте визначення несуперечливої теорії 1-го порядку.
40. Дайте визначення повної теорії 1-го порядку.
41. Чому суперечливі теорії 1-го порядку тривіальні?
42. Чому числення предикатів 1-го порядку нерівне?
43. Сформулюйте теорему несуперечливості.
44. Сформулюйте теорему суперечливості.
45. Наведіть теорему Лінденбаума.
46. Чи конструктивне доведення теореми Лінденбаума?
47. Що означає розв'язність теорії 1-го порядку?
48. Що означає перелічність теорії 1-го порядку?
49. Чи існують неперелічні теорії 1-го порядку із алгоритмічно перелічною множиною аксіом?
50. Сформулюйте теорему розв'язності та її наслідок.
51. Сформулюйте теорему про модель.
52. Наведіть перше та друге формулювання теореми Гьоделя про повноту.
53. Чому перше формулювання теореми Гьоделя про повноту називають теоремою адекватності?
54. Сформулюйте теорему Льовенгейма – Сколема про спуск.
55. Який наслідок для зліченних теорій 1-го порядку має теорема Льовенгейма – Сколема про спуск?
56. У чому полягає парадокс Сколема для аксіоматичної теорії множин? Чи є тут справжній парадокс?
57. Що засвідчує парадокс Сколема?
58. Чи вірно, що проблема всюди істинності формул 1-го порядку злі-ченної сигнатури алгоритмічно розв'язна? Частково розв'язна?
59. Що таке скінченно аксіоматизована теорія 1-го порядку?
60. Що таке скінченно аксіоматизована частина теорії 1-го порядку?
61. Сформулюйте теорему компактності (наведіть перше та друге формулювання).
62. Наведіть приклади використання теореми компактності?
63. Сформулюйте теорему Льовенгейма – Сколема про підйом.

64. Що засвідчує існування нестандартних моделей формальної арифметики?
65. Чому принцип математичної індукції формалізується в *Ar* неповністю?
66. Що засвідчують теореми Гьоделя про неповноту?
67. Сформулюйте першу теорему Гьоделя про неповноту.
68. Сформулюйте другу теорему Гьоделя про неповноту.
69. Чи вірно, що кожне несуперечливе розширення *Ar* нерозв'язне?
70. Чи вірно, що кожне несуперечливе розширення *Ar* неповне?
71. У чому полягає значення теорем Гьоделя про неповноту?
72. Наведіть базові секвенційні форми секвенційних числень чистих логік предикатів 1-го порядку
73. Сформулюйте теорему коректності для секвенційних числень логік 1-го порядку
74. Опишіть процедуру побудови секвенційного дерева для випадку секвенційних числень логік 1-го порядку
75. Сформулюйте теорему повноти для секвенційних числень логік 1-го порядку.

Вправи

1. Доведіть незалежність кожної схеми логічних аксіом та кожного правила виведення від інших схем аксіом та правил виведення числення предикатів.

2. Вкажіть виведення в численні предикатів таких формул:

1) $\vdash \exists x A \& B \leftrightarrow \exists x(A \& B)$, якщо x не вільна у B;
2) $\vdash \forall x A \& B \leftrightarrow \forall x(A \& B)$, якщо x не вільна у B;
3) $\vdash \exists x \exists y A \leftrightarrow \exists y \exists x A$;
4) $\vdash \forall x \forall y A \leftrightarrow \forall y \forall x A$;
5) $\vdash \exists x(A \vee B) \leftrightarrow \exists x A \vee \exists x B$;
6) $\vdash \forall x A \& \forall x B \leftrightarrow \forall x(A \& B)$;
7) $\vdash \forall x(A \& B) \rightarrow \forall x A \& B$;
8) $\vdash (A \rightarrow \forall x B) \rightarrow (\forall x A \rightarrow B)$;
9) $\vdash (\exists x A \rightarrow \forall x B) \rightarrow \forall x(A \rightarrow B)$;
10) $\vdash (\forall x A \rightarrow \exists x B) \rightarrow \exists x(A \rightarrow B)$;
11) $\vdash \exists x(A \rightarrow B) \rightarrow (\forall x A \rightarrow \exists x B)$;
12) $\vdash \forall x(A \rightarrow B) \rightarrow (\forall x A \rightarrow \forall x B)$;
13) $\vdash \forall x(A \rightarrow B) \rightarrow (\exists x A \rightarrow \exists x B)$;
14) $\vdash (\forall x A \rightarrow \forall x B) \rightarrow \exists x(A \rightarrow B)$;
15) $\vdash (\exists x A \rightarrow \exists x B) \rightarrow \exists x(A \rightarrow B)$.

3. Встановіть, чи вірно:

1) $\vdash \exists x \forall y A \to \forall y \exists x A$;
2) $\vdash \forall y \exists x A \to \exists x \forall y A$;
3) $\vdash \exists x A \& \exists x B \to \exists x (A \& B)$;
4) $\vdash \exists x (A \& B) \to \exists x A \& \exists x B$;
5) $\vdash \forall x A \vee \forall x B \to \forall x (A \vee B)$;
6) $\vdash \forall x (A \vee B) \to \forall x A \vee \forall x B$;
7) $\vdash \exists x A \& B \to \exists x (A \& B)$;
8) $\vdash \exists x (A \& B) \to \exists x A \& B$;
9) якщо $\vdash \exists x A \& B$, то $\vdash \exists x (A \& B)$.

4. Доведіть синтаксичні варіанти теорем еквівалентності та рівності (теореми 4.1.15 – 4.1.17).

5. Доведіть синтаксичні варіанти теорем про варіанту, пренексні операції та пренексну форму (теореми 4.1.18 – 4.1.20).

6. Чи можна послабити умову замкненості формули A у формулюванні теореми дедукції, щоб ця теорема залишалась вірною?

7. Доведіть, що чисте числення 1-арних предикатів розв'язне (його сигнатура містить тільки 1-арні ПС і не містить ФС та символу =).

8. Видалимо з теорії Ar функціональний символ $\times$ і аксіоми Ar5 та Ar6. Доведіть, що отримана таким чином теорія повна і розв'язна.

9. Побудуйте у секвенційному численні кванторного рівня виведення чи доведіть його відстутність для таких формул:

1) $\forall x (A \to B) \to (\exists x A \to \exists x B)$;
2) $\forall x (A \to B) \to (\forall x A \to \forall x B)$;
3) $\forall x (A \to B) \to (\exists x A \to \forall x B)$;
4) $\exists x (A \to B) \to (\forall x A \to \exists x B)$;
5) $\exists x (A \to B) \to (\forall x A \to \forall x B)$;
6) $\exists x (A \to B) \to (\exists x A \to \exists x B)$;
7) $(\forall x A \to \forall x B) \to \forall x (A \to B)$;
8) $(\forall x A \to \forall x B) \to \exists x (A \to B)$;
9) $(\exists x A \to \exists x B) \to \forall x (A \to B)$;
10) $(\exists x A \to \exists x B) \to \exists x (A \to B)$;
11) $(\exists x A \to \forall x B) \to \forall x (A \to B)$;
12) $\exists x (A \vee B) \leftrightarrow \exists x A \vee \exists x B$;
13) $\forall x (A \& B) \leftrightarrow \forall x A \& \forall x B$;
14) $\exists x (A \& B) \to \exists x A \& \exists x B$;
15) $\exists y \forall x A \to \forall x \exists y A$;
16) $\forall x \exists y A \to \exists y \forall x A$;
17) $\exists x (A \vee \exists x B) \to \exists x \neg A$.

5. ІНТУЇЦІОНІСТСЬКА ЛОГІКА

Криза засад математики на межі XIX–XX ст., зумовлена відкриттям парадоксів теорії множин, спонукала вчених шукати шляхи виходу із цього стану. Один із таких шляхів запропонував Д. Гільберт, висунувши програму порятунку класичної математики.

Гільберт виходив з того, що математика має справу переважно з ідеальними об'єктами. Такі об'єкти використовують актуальну (завершену) нескінченність, вони далеко виходять за межі безпосереднього осмислення та обґрунтування на інтуїтивній основі. Як зазначав Гільберт, у принципі ідеальні об'єкти та твердження потрібні лише як проміжні ланки для отримання реальних результатів і в цьому розумінні не є необхідними. Проте математика не може існувати без ідеальних об'єктів, вони необхідні для ефективності нашого мислення, без них не можна обійтися в одержанні *реальних* результатів. Наприклад, аналітична теорія чисел використовує для доведень тверджень про цілі числа засоби теорії дійсних чисел та теорії комплексних чисел, причому для багатьох теорем про цілі числа неаналітичні доведення невідомі. Тому потрібно обґрунтувати *принципову можливість* видалення ідеальних об'єктів та тверджень із виведень реальних тверджень. Можливість такої перебудови виведень необхідно доводити максимально надійними, інтуїтивно переконливими засобами, які не викликають сумнівів. Такі засоби Гільберт назвав *фінітними*, оскільки вони повинні уникати використання актуальної нескінченності. Для фінітного доведення теорем про перебудову виведень потрібно дати точне математичне уточнення мови та логічного виведення. Це означає побудову *формальної системи* для відповідного розділу математики. Після формалізації необхідно довести суто фінітними методами несуперечливість та повноту отриманої формальної системи.

Як ми вже знаємо, повна реалізація програми Гільберта неможлива. Це стало зрозумілим після отримання результатів К. Гьоделя. Проте ще задовго до цього сумніви в можливості повного обґрунтування математики на основі програми Гільберта висловив голландський математик Л. Брауер. Високо оцінюючи програму Гільберта в цілому, він заявляв, що навіть якби Гільберт довів несуперечливість класичної математики, це не зробило б класичну математику коректною. Брауер

писав: “Неправильна теорія, яка не наштовхнулась на суперечність, не стає від цього правильнішою, подібно до того, як злочинна поведінка, не зупинена правосуддям, не стає від цього менш злочинною” [6]. Ще в 1908 р. Брауер стверджував, що закони математики не мають ні абсолютного, ні апріорного характеру. Вони є узагальненням роботи із скінченними множинами стійких у часі об’єктів, тому поширення таких законів на нескінченні множини об’єктів неадекватне. Отже, необхідно або цілком відмовитися від нескінченних множин, що не зовсім розумно, або перейти до нової логіки, інтуїтивно зрозумілої. Така логіка повинна описувати математичні твердження не як абстрактні істину чи фальш, а як твердження про можливість виконання деякої побудови. Математичне доведення мусить давати побудову та її обґрунтування. Такі методи, що дають побудову, Брауер назвав *ефективними*, пропоновану ним логіку і математику – *інтуїціоністською*.

Потужним імпульсом розвитку інтуїціоністської математики і логіки стало виникнення теорії алгоритмів. Нині існує багато різновидностей інтуїціоністської логіки. Весь напрям в математиці та логіці, для якого основоположними є поняття задачі та побудови, а не істини та обґрунтування, називають *конструктивізмом*. У цьому плані *інтуїціонізмом* називають напрям, який безпосередньо базується на брауерових постулатах.

Після появи інтуїціоністської логіки постало питання про її формалізацію. Цікавим є те, що сам Брауер стверджував, що, на відміну від класичної математики, інтуїціоністська математика в принципі не може бути адекватно формалізована, і водночас він запропонував своєму учневі А. Гейтінгу створити формальні моделі інтуїціоністської логіки, що й було успішно зроблено. Згодом з’явились семантичні моделі (інтерпретації) інтуїціоністської логіки. Дуже цікаву інтерпретацію, яка базується на брауеровому розумінні формул як задач, запропонував А. Колмогоров, а потім розвинув А. Гейтінг. Таку інтерпретацію називають інтерпретацією Колмогорова, а також ВКН-інтерпретацією [32]. В цій інтерпретації поняттю *істинності* формули класичної логіки відповідає поняття *реалізовності* формули як задачі.

5.1. Мова інтуїціоністської логіки

Для інтуїціоністської логіки не діють закон виключеного третього та пов’язані з ним закони де Моргана, закон зняття подвійного заперечення. Пропозиційні зв’язки $\neg$, $\vee$, **&**, $\rightarrow$ тепер незалежні, еквіваленція подається через імплікацію та кон’юнкцію: $P \leftrightarrow Q = (P \rightarrow Q)\mathbf{\&}(Q \rightarrow P)$. Операції квантифікації $\exists \boldsymbol{x}$ та $\forall \boldsymbol{x}$ теж незалежні.

Розглянемо мову інтуїціоністської пропозиційної логіки (ІПЛ).

Алфавіт мови ІПЛ складається із символів логічних зв'язок $\neg$, $\vee$, $\&$, $\rightarrow$ та множини *Ps* пропозиційних символів.

Визначення формули мови ІПЛ індуктивне:

1) кожний $A \in Ps$ є формулою;

2) якщо Φ та Ψ – формули, то $\neg\Phi$, $\vee\Phi\Psi$, $\&\Phi\Psi$, $\rightarrow\Phi\Psi$ – формули.

Множину формул мови ІПЛ позначимо *Fp*.

Розглянемо мову інтуїціоністської логіки предикатів 1-го порядку (ІЛП). Обмежимося випадком чистої логіки, або логіки кванторного рівня (це означає, що не виділено спеціальний предикат рівності та в сигнатурі мови немає функціональних символів).

Алфавіт мови ІЛП складається із таких символів:

- предметні імена (змінні) x, y, z, ...;
- предикатні символи (ПС) p_0, p_1, p_2, ... заданої арності;
- символи логічних операцій $\neg$, $\vee$, $\&$, $\rightarrow$ та $\exists x$, $\forall x$.

Множину *Ps* предикатних символів назвемо *сигнатурою* мови ІЛП.

Атомарною формулою мови ІЛП називається вираз вигляду $px_1 \ldots x_n$, де p – n-арний ПС, x_1, ..., x_n – предметні змінні.

Індуктивне визначення формули мови ІЛП таке:

1) кожна атомарна формула є формулою;

2) якщо Φ та Ψ – формули, то $\neg\Phi$, $\vee\Phi\Psi$, $\&\Phi\Psi$, $\rightarrow\Phi\Psi$ – формули;

3) якщо Φ – формула, x – предметне ім'я, то $\exists x\Phi$ та $\forall x\Phi$ – формули.

Множину формул мови ІЛП позначимо *Fr*.

5.2. Реляційна семантика інтуїціоністської логіки

На відміну від класичної логіки, яка є логікою конкретного знання, інтуїціоністська логіка передбачає накопичення знань. На цій ідеї Брауера базуються найпопулярніші семантичні моделі інтуїціоністської логіки – *моделі можливих світів*, або *реляційні моделі*.

Моделі можливих світів започатковані Л. Брауером і А. Гейтінгом, далі розвинуті С. Кріпке та Я. Хінтіккою. Такі моделі успішно використовуються також для опису семантики модальних логік. Про інші підходи до семантики інтуїціоністської логіки див., наприклад, [32, 36].

Моделлю можливих світів інтуїціоністської логіки, або *реляційною інтуїціоністською моделлю,* назвемо трійку $\boldsymbol{M} = (\boldsymbol{S}, \triangleright, \boldsymbol{I})$. Тут $\boldsymbol{S}$ – множина світів; $\triangleright$ – бінарне відношення на $\boldsymbol{S}$; $\boldsymbol{I}$ – відображення інтерпретації. Відношення $\triangleright$ є відношенням часткового порядку на $\boldsymbol{S}$.

Для випадку інтуїціоністської пропозиційної логіки відображення інтерпретації уточнимо так:

$\boldsymbol{I}$: $Ps \times \boldsymbol{S} \to \{T, F\}$.

Світи узгоджуються із відношенням $\triangleright$ таким чином: якщо $\alpha \triangleright \beta$ та $\boldsymbol{I}(A, \alpha) = T$, то $\boldsymbol{I}(A, \beta) = T$. Це означає, що при підйомі по світах істинність атомарних формул не може перейти у фальш.

Відображення інтерпретації $\boldsymbol{I}$: $Ps \times \boldsymbol{S} \to \{T, F\}$ індуктивно продовжимо до відображення $\boldsymbol{J}$: $Fp \times \boldsymbol{S} \to \{T, F\}$:

1) $\boldsymbol{J}(\mathrm{A}, \alpha) = \boldsymbol{I}(\mathrm{A}, \alpha)$ для всіх $\mathrm{A} \in Ps$;

2) $\boldsymbol{J}(\Phi \vee \Psi, \alpha) = T \Leftrightarrow \boldsymbol{J}(\Phi, \alpha) = T$ або $\boldsymbol{J}(\Psi, \alpha) = T$;

3) $\boldsymbol{J}(\Phi \& \Psi, \alpha) = T \Leftrightarrow \boldsymbol{J}(\Phi, \alpha) = T$ та $\boldsymbol{J}(\Psi, \alpha) = T$;

4) $\boldsymbol{J}(\neg\Phi, \alpha) = T \Leftrightarrow$ для всіх β таких, що $\alpha \triangleright \beta$, маємо $\boldsymbol{J}(\Phi, \beta) = F$.

5) $\boldsymbol{J}(\Phi \to \Psi, \alpha) = T \Leftrightarrow$ для всіх β таких, що $\alpha \triangleright \beta$, маємо: якщо $\boldsymbol{J}(\Phi, \beta) = T$, то $\boldsymbol{J}(\Psi, \beta) = T$.

Те, що $\boldsymbol{J}(\Phi, \alpha) = T$, тобто істинність формули Φ у світі α, позначаємо $\alpha \models \Phi$.

Формула Φ *істинна в реляційній моделі* $\boldsymbol{M}$, що позначаємо $\boldsymbol{M} \models \Phi$, якщо для всіх $\alpha \in \boldsymbol{S}$ маємо $\alpha \models \Phi$.

Формула Φ *інтуїціоністськи істинна*, що позначаємо $\models \Phi$, якщо для кожної реляційної моделі $\boldsymbol{M}$ маємо $\boldsymbol{M} \models \Phi$.

Для випадку інтуїціоністської логіки предикатів світами є алгебраїчні системи заданої сигнатури σ, яка визначає мову ІЛП.

Відображення інтерпретації атомарних формул на світах задаємо так:

$\boldsymbol{I}$: $\bigcup\limits_{\alpha \in S} (Ps \times \boldsymbol{S} \to Pr^{\alpha})$.

Світи узгоджуються із відношенням $\triangleright$ таким чином:

- нехай $\alpha = (A, \sigma)$, $\beta = (B, \sigma)$ та $\alpha \triangleright \beta$. Тоді $A \subseteq B$;
- нехай $p \in Ps$. Якщо $\alpha \triangleright \beta$ та $p_{\alpha}(a_1, ..., a_n) = T$, то $p_{\beta}(a_1, ..., a_n) = T$.

Отже, при підйомі по світах їх носії можуть тільки розширюватися, при цьому істинність атомарних формул не може перейти у фальш.

Значення формули у світі α визначаємо індуктивно:

1) для атомарних формул $p_{\alpha}(d) = T$ означає $\boldsymbol{I}(p, \alpha)(d) = T$;

2) $(\Phi \vee \Psi)_{\alpha}(d) = T \Leftrightarrow \Phi_{\alpha}(d) = T$ або $\Psi_{\alpha}(d) = T$;

3) $(\Phi \& \Psi)_{\alpha}(d) = T \Leftrightarrow \Phi_{\alpha}(d) = T$ та $\Psi_{\alpha}(d) = T$;

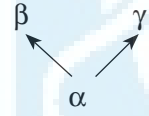
4) $(\neg\Phi)_\alpha(d) = T \Leftrightarrow$ для всіх β таких, що $\alpha \triangleright \beta$, маємо $\Phi_\beta(d) = F$;
5) $(\Phi\to\Psi)_\alpha(d) = T \Leftrightarrow$ для всіх β таких, що $\alpha \triangleright \beta$, маємо: якщо $\Phi_\beta(d) = T$, то $\Psi_\beta(d) = T$;
6) $(\exists x\Phi)_\alpha(d) = T \Leftrightarrow$ для деякого $a \in A$ маємо $\Phi_\alpha(d\nabla x\mapsto a) = T$;
7) $(\forall x\Phi)_\alpha(d) = T \Leftrightarrow$ для всіх β таких, що $\alpha \triangleright \beta$, для всіх $a \in B$ маємо $\Phi_\beta(d\nabla x\mapsto a) = T$.

Істинність формули Φ у світі α позначаємо $\alpha \models \Phi$.

Формула Φ *істинна в реляційній моделі* $\boldsymbol{M}$, що позначаємо $\boldsymbol{M} \models \Phi$, якщо для всіх $\alpha \in \boldsymbol{S}$ маємо $\alpha \models \Phi$.

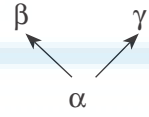
Формула Φ мови сигнатури σ *інтуїціоністськи істинна*, що позначаємо ${}_{\boldsymbol{I}} \models \Phi$, якщо для кожної реляційної моделі $\boldsymbol{M}$ із світами сигнатури σ маємо $\boldsymbol{M} \models \Phi$.

Приклад 5.2.1. Покажемо, що формула $A\vee\neg A$ не є інтуїціоністськи істинною. Для цього вкажемо для неї контрмодель – реляційну модель $\boldsymbol{M}$ таку, що $\boldsymbol{M} \not\models A\vee\neg A$.



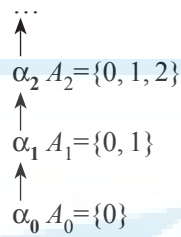
Задамо $\boldsymbol{I}(A, \alpha) = F$, $\boldsymbol{I}(A, \beta) = T$, $\boldsymbol{I}(A, \gamma) = F$. Зрозуміло, що невірно $\alpha \models A$. Для $\alpha \models \neg A$ необхідно $\alpha \not\models A$, $\gamma \not\models A$, $\beta \not\models A$. Але $\boldsymbol{I}(A, \beta) = T$, тому $\beta \models A$. Отже, невірно $\alpha \models \neg A$, звідки $\alpha \not\models A\vee\neg A$, тому $\boldsymbol{M} \not\models A\vee\neg A$.

Приклад 5.2.2. Покажемо, що формула $(A\to B)\vee(B\to A)$ не є інтуїціоністськи істинною. Для цього вкажемо для неї контрмодель $\boldsymbol{M}$ таку, що $\boldsymbol{M} \not\models (A\to B)\vee(B\to A)$.



Задамо $\boldsymbol{I}(A, \alpha) = F$, $\boldsymbol{I}(B, \alpha) = F$, $\boldsymbol{I}(A, \beta) = T$, $\boldsymbol{I}(B, \beta) = F$, $\boldsymbol{I}(A, \gamma) = F$, $\boldsymbol{I}(B, \gamma) = T$. Тоді $\beta \models A$ та $\beta \not\models B$, звідки, враховуючи $\alpha \triangleright \beta$, невірно $\alpha \models A\to B$. Але $\gamma \models B$ та $\gamma \not\models A$, тому, враховуючи $\alpha \triangleright \gamma$, невірно $\alpha \models B\to A$. Отже, невірно $\alpha \models (A\to B)\vee(B\to A)$, тому $\boldsymbol{M} \not\models (A\to B)\vee(B\to A)$.

Приклад 5.2.3. Вкажемо модель $\boldsymbol{M}$ таку, що $\boldsymbol{M} \models \neg\forall x(P(x)\vee\neg P(x))$.



Для кожного світу α_n його носій – це $A_n = \{0, 1, \ldots, n\}$. Задамо $P_{\alpha_n}(k) = T$ для всіх $k<n$ та $P_{\alpha_n}(n) = F$. Маємо $P_{\alpha_0}(0) = F$; але $\neg P_{\alpha_0}(0) = T$ означає, що $P_{\alpha_n}(0) = F$ для всіх n, що невірно, тому $\neg P_{\alpha_0}(0) = F$. Отже, $(P \vee \neg P)_{\alpha_0}(0) = F$. Маємо $P_{\alpha_1}(1) = F$; але $\neg P_{\alpha_1}(1) = T$ означає, що $P_{\alpha_1}(1) = F$ для всіх $n\geq 1$, що невірно, тому $\neg P_{\alpha_1}(1) = F$. Отже, $(P \vee \neg P)_{\alpha_1}(1) = F$. Продовжуючи, отримуємо $(P \vee \neg P)_{\alpha_2}(2) = F$ і т. д. Отже, для кожного α_n $(P \vee \neg P)_{\alpha_n}(n) = F$, тому для кожного α_n $(\forall x(P(x) \vee \neg P(x)))_{\alpha_n} = F$. Звідси $\alpha_n \models \neg\forall x(P(x)\vee\neg P(x))$ для кожного α_n, тому $\boldsymbol{M} \models \neg\forall x(P(x)\vee\neg P(x))$.

Зауважимо, що в класичній логіці $\models \forall x(\mathrm{A}(x)\vee\neg\mathrm{A}(x))$. Отже, в інтуїціоністській логіці є формули, які суперечать формулам класичної логіки.

5.3. Формально-аксіоматичні системи інтуїціоністської логіки

Розглянемо формально-аксіоматичні системи інтуїціоністської логіки пропозиційного рівня. Аксіоматичні системи гільбертівського типу для інтуїціоністської ПЛ називаються інтуїціоністськими пропозиційними численнями (ІПЧ).

Множина правил виведення ІПЧ складається з єдиного правила: МР) A, $A\to B \vdash B$ – *modus ponens*.

Множина аксіом ІПЧ визначається такими схемами аксіом:

$A1)$ $A\to(B\to A)$;

$A2)$ $(A\to(B\to C))\to((A\to B)\to(A\to C))$;

$A3)$ $A\&B\to A$;

$A4)$ $A\&B\to B$;

$A5)$ $A\to(B\to A\&B)$;

$A6)$ $A\to A\vee B$;

$A7$) $B \to A \vee B$;

$A8$) $(A \to C) \to (B \to C)) \to (A \vee B \to C)$;

$A9$) $(A \to B) \to ((A \to \neg B) \to \neg A)$;

AI) $\neg A \to (A \to B)$.

Зауважимо, що при заміні схеми аксіом AI схемою аксіом $\neg\neg A \to A$, отримаємо пропозиційне числення класичного типу, еквівалентне пропозиційному численню, розглянутому в 2.3.

Кожна теорема ІПЧ є теоремою класичного ПЧ, але зворотне невірне. Зокрема, в ІПЧ не можна вивести формули $\neg\neg A \to A$, $A \vee \neg A$, проте можна довести $A \to \neg\neg A$.

Аксіоматичні системи генценівського типу для інтуїціоністської ПЛ називаються інтуїціоністськими секвенційними пропозиційними численнями (ІСПЧ). Зауважимо, що Г. Генцен одночасно побудував секвенційні числення для класичної та інтуїціоністської логіки.

Різні варіанти інтуїціоністських секвенційних числень наведені в [11, 22, 32].

Побудуємо варіант ІСПЧ, тісно пов'язаний з реляційною семантикою інтуїціоністської логіки. Такий підхід до побудови секвенційних числень та семантичних таблиць інтуїціоністської логіки розглянутий в [32].

Інтуїціоністською специфікацією назвемо слово вигляду $\alpha|-$ чи $\alpha-|$, де α – інтуїціоністський префікс, що є іменем світу, в якому специфікована формула має відповідне значення. *Інтуїціоністський префікс* – це слово, символами якого є імена натуральних чисел.

Усі формули початкової секвенції мають порожній інтуїціоністський префікс.

Для префіксів пишемо $\alpha \le \beta$, якщо β має вигляд $\alpha\gamma$. Якщо $\alpha \le \beta$ та $\alpha \ne \beta$, то пишемо $\alpha < \beta$. Для префіксів $\alpha \le \beta$ означає, що $\alpha \triangleright \beta$, тобто світ β є наступником світу α.

Світ вигляду αn назвемо безпосереднім наступником світу α.

Вважатимемо, що замкненість секвенції (суперечність) дає пара специфікованих формул вигляду $_{\alpha|-}\Phi$ та $_{\alpha\beta-|}\Phi$. Це відповідає умові, що формула, істинна у світі α, зберігає істинність в усіх його наступниках.

Крім того, введемо такі додаткові умови замкненості секвенції:

- поява в секвенції пари формул $_{\alpha|-}\neg\Phi$ та $_{\alpha\beta|-}\Phi$;
- поява в секвенції пари формул $_{\alpha|-}\Phi$ та $_{\alpha\beta|-}\neg\Phi$.

Зауважимо, що пара формул $_{\alpha-|}\Phi$ та $_{\alpha\beta|-}\Phi$, де $\beta \ne \varepsilon$, не дає замкненості секвенції.

Секвенційні форми для випадку ІСПЧ модифікуються так.

Форми $\vdash\vee$, $\dashv\vee$, $\vdash\&$, $\dashv\&$ аналогічні відповідним формам секвенційних числень класичної логіки. Вони не змінюють інтуїціоністський префікс нових формул – предків основної формули.

$$\vdash\vee \frac{\alpha\vdash A,\Sigma \qquad \alpha\vdash B,\Sigma}{\alpha\vdash A\vee B,\Sigma};\qquad \dashv\vee \frac{\alpha\dashv A,\ \alpha\dashv B,\Sigma}{\dashv A\vee B,\Sigma};$$

$$\vdash\& \frac{\alpha\vdash A,\ \alpha\vdash B,\Sigma}{\alpha\vdash A\,\&\,B,\Sigma};\qquad \dashv\& \frac{\alpha\dashv A,\Sigma \qquad \alpha\dashv B,\Sigma}{\alpha\dashv A\,\&\,B,\Sigma}.$$

Для форм $\dashv\neg$ та $\dashv\to$ нові формули стверджуються чи заперечуються не у світі α основної формули висновку, а у світі-наступнику αn. При цьому кожний раз таке n вибирається новим (на шляху від початкової секвенції).

$$\dashv\neg \frac{\alpha n\vdash A,\Sigma}{\alpha\dashv\neg A,\Sigma};\qquad \dashv\to \frac{\alpha n\vdash A,\ \alpha n\dashv B,\Sigma}{\alpha\dashv A\to B,\Sigma}.$$

Тут n – нове, відмінне від усіх імен натуральних чисел, що фігурують у префіксах формул світів секвенції-висновку.

Секвенційні форми $\vdash\neg$ та $\vdash\to$ вимагають багаторазового розбиття основної формули, адже у разі появи нових світів – наступників світу α тут виникають спростовувані формули, які не можуть автоматично переноситися на світи-наступники.

$$\vdash\neg \frac{\alpha\vdash\neg A,\ \beta_1\dashv A,\ldots,\ \beta_m\dashv A,\ \Sigma}{\alpha\vdash\neg A,\Sigma}.$$

Тут $\beta_1,\ldots,\beta_m$ – імена усіх світів-наступників світу α, які фігурують у секвенції-висновку.

Виконання форми $\vdash\to$ означає побудову піддерева, коренем якого є секвенція-висновок.

Нехай така секвенція-висновок Ξ має вигляд $\alpha\vdash A\to B,\Sigma$. Нехай $\beta_1,\ldots,\beta_m$ – імена усіх світів-наступників світу α, які фігурують в Ξ (це означає $\alpha\le\beta_i$ для всіх $i\in\{1,\ldots,m\}$, при цьому вважаємо $\alpha=\beta_1$). Позаяк із $\alpha\vdash A\to B$ випливає $\beta_i\vdash A\to B$ для всіх $i\in\{1,\ldots,m\}$, то виконання форми $\vdash\to$ зводиться до побудови піддерева θ з коренем Ξ, листами якого можуть бути усі можливі секвенції вигляду $\Phi_1,\ldots,\Phi_m,\Xi$, де кожна формула Φ_i має вигляд $\beta_i\vdash A$ чи $\beta_i\vdash B$. Якщо для деякої вер-

шини будованого піддерева маємо $_{\beta_i}|\!-B$, то, враховуючи збереження істинності при русі світами згідно з $\triangleright$, подальше розбиття на шляхах з цієї вершини формули $_{|\!-}A\to B$ по світах β_j таких, що $\beta_i \le \beta_j$, надлишкове. При отриманні замкненої секвенції подальше розбиття теж не виконується.

Таким чином, секвенційна форма $_{|\!-}\to$ має вигляд

$$|\!-\to \frac{\theta}{\alpha|\!- A\to B,\Sigma}.$$

Приклад 5.3.1. Побудуємо виведення секвенції $_{-|}\neg A\vee B\to(A\to B)$.

$$\begin{array}{l}
{}_{-|}\neg A\vee B\to(A\to B)\\
\downarrow\\
{}_{1|\!-}\neg A\vee B,\ {}_{1-|}A\to B\\
\downarrow\\
{}_{1|\!-}\neg A\vee B,\ {}_{12|\!-}A,\ {}_{12-|}B\\
\swarrow\qquad\qquad\searrow\\
{}_{1|\!-}\neg A,\ {}_{12|\!-}A,\ {}_{12-|}B\ \times \qquad {}_{1|\!-}B,\ {}_{12|\!-}A,\ {}_{12-|}B\ \times
\end{array}$$

Отримали замкнене секвенційне дерево, тому формула $\neg A\vee B\to(A\to B)$ інтуїціоністськи істинна.

Приклад 5.3.2. Побудуємо виведення секвенції $_{-|}(\mathrm{A}\to\mathrm{B})\to\neg\mathrm{A}\vee\mathrm{B}$.

$$\begin{array}{l}
{}_{-|}(A\to B)\to\neg A\vee B\\
\downarrow\\
{}_{1|\!-}A\to B,\ {}_{1-|}\neg A\vee B\\
\downarrow\\
{}_{1|\!-}A\to B,\ {}_{1-|}\neg A,\ {}_{1-|}B\\
\downarrow\\
{}_{1|\!-}A\to B,\ {}_{12|\!-}A,\ {}_{1-|}B\\
\swarrow\qquad\qquad\searrow\\
{}_{1|\!-}A\to B,\ {}_{1-|}A,\ {}_{12|\!-}A,\ {}_{1-|}B \qquad {}_{1|\!-}A\to B,\ {}_{1|\!-}B,\ {}_{12|\!-}A,\ {}_{1-|}B\ \times\\
\swarrow\qquad\qquad\searrow\\
{}_{1|\!-}A\to B,\ {}_{1-|}A,\ {}_{12-|}A,\ {}_{12|\!-}A,\ {}_{1-|}B\ \times\ {}_{1|\!-}A\to B,\ {}_{1-|}A,\ {}_{12|\!-}B,\ {}_{12|\!-}A,\ {}_{1-|}B
\end{array}$$

Отримали незамкнене секвенційне дерево, яке дає змогу вказати контрмодель $\boldsymbol{M}$ таку, що $\boldsymbol{M}\mid\!\neq(A\to B)\to\neg A\vee B$:

Згідно з незамкненим листом секвенційного дерева задаємо $\boldsymbol{I}(A, 1) = F$, $\boldsymbol{I}(B, 1) = F$, $\boldsymbol{I}(A, 12) = T$, $\boldsymbol{I}(B, 12) = T$.

Розглянемо тепер формально-аксіоматичні системи інтуїціоністської логіки предикатів 1-го порядку.

Аксіоматичні системи гільбертівського типу для ІЛП називаються інтуїціоністськими численнями предикатів (ІЧП). Логічні аксіоми ІЧП задаються схемами аксіом ІПЧ, до яких додаємо кванторні схеми:

AQ1) $A \to \exists x A$;

AQ2) $\forall x A \to A$.

Множина правил виведення ІЧП складається з трьох правил:

МР) A, $A \to B \mid\!- B$ – *modus ponens*;

П$\exists$) $A \to B \mid\!- \exists x A \to B$, якщо x не вільна в B, – правило $\exists$-введення;

П$\forall$) $A \to B \mid\!- A \to \forall x B$, якщо x не вільна в A, – правило $\forall$-введення.

Аксіоматичні системи генценівського типу для інтуїціоністської логіки 1-го порядку називаються інтуїціоністськими секвенційними численнями предикатів (ІСЧП).

Базовими секвенційними формами ІСЧП є базові секвенційні форми ІСПЧ, до яких додаються кванторні секвенційні форми:

$$\vdash^{\exists} \frac{\alpha \mid\!- A_x[y], \Sigma}{\alpha \mid\!- \exists x A, \Sigma} \text{ за умови, що вільна змінна } y \notin \Sigma \cup \{\exists x A\}.$$

$$\dashv^{\exists} \frac{\alpha \dashv\mid A_x[z_1], \ldots, \alpha \dashv\mid A_x[z_m],\ \Sigma,\ \alpha \dashv\mid \exists x A}{\alpha \dashv\mid \exists x A, \Sigma}.$$

$$\vdash^{\forall} \frac{\alpha \mid\!- A_x[z_1], \ldots, \alpha \mid\!- A_x[z_m],\ \Sigma,\ \alpha \mid\!- \forall x A}{\alpha \mid\!- \forall x A, \Sigma}.$$

При застосуванні $\dashv^{\exists}$ та $\vdash^{\forall}$ $\{z_1, \ldots, z_m\}$ – множина вільних імен множини доступних формул секвенції-висновку та її наступників.

$$\dashv^{\forall} \frac{\alpha n \dashv\mid A_x[y],\ \Sigma}{\alpha \dashv\mid \forall x A,\ \Sigma} \text{ за умови, що вільна змінна } y \notin \Gamma \cup \{\forall x A\}.$$

Тут n – нове, відмінне від усіх імен натуральних чисел, що фігурують у префіксах формул світів секвенції-висновку.

Основні результати теорії доведень класичної логіки переносяться на випадок інтуїціоністської логіки. Для інтуїціоністських числень справджуються відповідні теореми коректності та повноти. Г. Генцен довів свою теорему про нормальну форму (елімінацію перетинів) одночасно для класичної логіки та інтуїціоністської логіки. В певному розумінні інтуїціоністська логіка слабша за класичну, адже не кожна теорема інтуїціоністського числення є теоремою класичного числення. Водночас класична логіка ізоморфно занурюється в інтуїціоністську логіку (теорема Глівенка), тобто класичну логіку можна трактувати як підсистему інтуїціоністської логіки.

Питання для самоконтролю

1. У чому полягає сутність програми Гільберта?
2. Поясніть сутність підходу Брауера до побудови інтуїціоністської логіки.
3. Які закони класичної логіки не діють для інтуїціоністської логіки?
4. Вкажіть алфавіт мови інтуїціоністської пропозиційної логіки.
5. Наведіть визначення формули мови інтуїціоністської пропозиційної логіки.
6. Вкажіть алфавіт мови інтуїціоністської логіки предикатів 1-го порядку.
7. Наведіть визначення формули мови інтуїціоністської логіки предикатів 1-го порядку.
8. Що таке модель можливих світів (реляційна модель) інтуїціоністської логіки?
9. Як задається відображення інтерпретації для випадку інтуїціоністської пропозиційної логіки?
10. Що означає узгодженість світів із відношенням $\triangleright$ для випадку інтуїціоністської пропозиційної логіки?
11. Як задається значення формули у світі α для випадку інтуїціоністської пропозиційної логіки?
12. Як визначається істинність формули в реляційній моделі інтуїціоністської пропозиційної логіки?
13. Дайте визначення інтуїціоністськи істинної формули інтуїціоністської пропозиційної логіки.
14. Що таке контрмодель?
15. Що є світами реляційної моделі для інтуїціоністської логіки предикатів?
16. Як задається відображення інтерпретації для інтуїціоністської логіки предикатів?

17. Що означає узгодженість світів із відношенням $\triangleright$ для інтуїціоністської логіки предикатів?
18. Як задається значення формули у світі α для інтуїціоністської логіки предикатів?
19. Як визначається істинність формули в реляційній моделі інтуїціоністської логіки предикатів?
20. Дайте визначення інтуїціоністськи істинної формули інтуїціоністської логіки предикатів.
21. Наведіть приклади тавтологій, які не є інтуїціоністськи істинними.
22. Вкажіть приклади всюди істинних формул, які є інтуїціоністськи істинними.
23. Наведіть приклади всюди істинних формул, які не є інтуїціоністськи істинними.
24. Що таке інтуїціоністське пропозиційне числення?
25. Наведіть аксіоми інтуїціоністського пропозиційного числення.
26. Як співвідносяться множини теорем класичного та інтуїціоністського пропозиційних числень?
27. Що таке інтуїціоністське секвенційне пропозиційне числення?
28. Що таке інтуїціоністська специфікація?
29. Що таке інтуїціоністський префікс?
30. Які умови замкненості секвенції в інтуїціоністському секвенційному численні?
31. Наведіть базові секвенційні форми інтуїціоністських секвенційних пропозиційних числень.
32. Вкажіть особливості секвенційних форм $\dashv\neg$ та $\dashv\rightarrow$.
33. Які особливості секвенційних форм $\vdash\neg$ та $\vdash\rightarrow$?
34. Що таке інтуїціоністське числення предикатів?
35. Вкажіть логічні аксіоми та правила виведення інтуїціоністського числення предикатів.
36. Що таке інтуїціоністське секвенційне числення предикатів?
37. Наведіть базові секвенційні форми інтуїціоністських секвенційних числень предикатів.
38. Вкажіть особливості секвенційної форми $\dashv\forall$.
39. Яке співвідношення між інтуїціоністською та класичною логіками?

Вправи

1. Побудуйте реляційну модель $\boldsymbol{M}$ інтуїціоністської логіки таку, що $\boldsymbol{M} \not\models (A\leftrightarrow B)\vee(B\leftrightarrow C)\vee(A\leftrightarrow C)$. Зауважимо, що $(A\leftrightarrow B)\vee(B\leftrightarrow C)\vee(A\leftrightarrow C)$ – тавтологія класичної логіки.

2. Узагальнюючи задачу 1, побудуйте таку реляційну модель $\boldsymbol{M}$ інтуїціоністської логіки:

$\boldsymbol{M} \models\!\!\!\neq (A_1 \leftrightarrow A_2) \vee (A_1 \leftrightarrow A_3) \vee \ldots \vee (A_1 \leftrightarrow A_n) \vee \ldots \vee (A_{n-1} \leftrightarrow A_n)$.

Це свідчить, що інтуїціоністська логіка не може задаватися жодною скінченною множиною істиннісних значень.

3. Побудуйте реляційну модель $\boldsymbol{M}$ інтуїціоністської логіки таку, що $\boldsymbol{M} \models \neg\forall x P(x)$ та $\boldsymbol{M} \models\!\!\!\neq \exists x P(x)$.

4. Побудуйте виведення в ІПЧ та в ІСПЧ для таких формул:

1) $A \to \neg\neg A$;
2) $A \to \neg A \to B$;
3) $\neg\neg(\neg\neg A \to A)$;
4) $\neg\neg\neg A \to \neg A$;
5) $A \to B \to A \& B$;
6) $\neg\neg(A \to B) \to (A \to \neg\neg B)$;
7) $(A \to B \to C) \to (B \to A \to C)$;
8) $(A \to B \to C) \to ((A \to B) \to (A \to C))$.

5. Доведіть в ІПЧ та в ІСПЧ:

1) якщо $\vdash A \to B$ та $\vdash B \to C$, то $\vdash A \to C$;
2) якщо $\vdash\!\!\!\neg\neg\neg(A \to B)$ та $\vdash\!\!\!\neg\neg\neg A$, то $\vdash\!\!\!\neg\neg\neg B$.

6. Побудуйте у відповідних інтуїціоністських численнях виведення чи доведіть його відсутність, створивши контрмодель, для таких формул:

1) $\neg\neg(A \vee \neg A)$;
2) $A \to B \to A$;
3) $\neg(A \& B) \to \neg A \vee \neg B$;
4) $\neg(A \vee B) \to \neg A \& \neg B$;
5) $\neg A \vee \neg B \to \neg(A \& B)$;
6) $\neg A \& \neg B \to \neg(A \vee B)$;
7) $((A \to B) \to A) \to A$;
8) $(A \to B) \to (\neg B \to \neg A)$;
9) $(\neg B \to \neg A) \to (A \to B)$;
10) $(A \to \neg B) \to (B \to \neg A)$;
11) $(\neg B \to A) \to (\neg A \to B)$;
12) $(\neg B \to \neg A) \to (A \to B)$;
13) $(A \to B \vee C) \to (A \to B) \vee (A \to C)$;
14) $\neg\neg\forall x(P(x) \vee \neg P(x))$.

6. МОДАЛЬНІ ЛОГІКИ

Твердження, які не можна трактувати тільки як істинні або хибні, а можна охарактеризувати певною мірою істинності, розглядались ще в античні часи. Значна частина логіки Арістотеля присвячена дослідженню тверджень вигляду "кожне *S* необхідно *P*","кожне *S* можливо *P*" і т. п. Властивості тверджень, які в певному аспекті характеризують ступінь їх істинності чи наше ставлення до них, називають *модальностями*.

Модальності "необхідно" та "можливо" називають загальними, або алетичними. Такі модальності позначають $\Box$ та $\diamond$.

Модальності, які мають часовий зміст, називають *часовими*, або *темпоральними*. Такі модальності теж відомі з античності (Арістотель, Діодор Кронос). До часових, як правило, належать модальності "завжди було", "колись було", "завжди буде", "колись буде". Їх називають основними часовими модальностями. Міркування з твердженнями, шо містять часові модальності, вивчає часова, або темпоральна, логіка.

Модальності, які характеризують ступінь обґрунтованості знання, називають *епістемічними*. Такими є модальності "достовірно", "доведено", "підтверджено", "обґрунтовано", "вірогідно", "спростовно". Міркування з твердженнями, що містять епістемічні модальності, вивчає епістемічна логіка, або логіка знання.

Модальності "обов'язково", "дозволено", "заборонено" характеризують норми та нормативні поняття. Такі модальності називаються *деонтичними*. Логіка, в якій вивчаються міркування з деонтичними модальностями, називається деонтичною, або прескриптивною. Вживана також назва "логіка норм".

Виникнення сучасної модальної логіки започатковане дослідженнями К. Льюїса та Я. Лукасєвича. Особливе значення мала праця "Symbolic logic", написана К. Льюїсом та К. Ленгфордом у 1932 р. В роботах цього періоду формалізована система класичних модальностей і розпочато дослідження різних систем модальної логіки (класичні системи $S1$–$S5$). Модальність трактується як спеціальний оператор (композиція), який застосовується до предикатів. Уточнен-

ням і формалізацією модальних логік займалися К. Гьодель, Я. Лукасєвич, Д. Гільберт, Г. Генцен, А. Тарський, Р. Карнап та ін.

Спершу дослідження модальних логік здійснювалося в синтаксичному стилі, для них не було чітко сформульованих семантик. Вперше алгебраїчна семантика для системи *S*4 була побудована в 1948 р. (А. Тарський, Дж. Мак-Кінсі). Інтенсивне розроблення семантик модальних логік розпочалося на рубежі 50–60 років XX ст. Важливим етапом у розвитку модальних логік було розроблення семантик можливих світів, або реляційних семантик (С. Кангер, С. Кріпке, Я. Хінтікка). Концепція можливих світів природно пов'язана з модальною логікою. Фундаментальним є поняття можливості одного світу стосовно іншого. Семантичні моделі модальної логіки на основі зазначеної концепції дають змогу природним чином трактувати як загальні модальності "необхідно" та "можливо", так і спеціальні модальності (часові, епістемічні, деонтичні). Семантики можливих світів виявились плодотворними для дослідження інших типів логік, зокрема інтуїціоністської логіки. Згодом були запропоновані узагальнення реляційної семантики (Д. Скотт, Р. Монтегю).

Останні роки характеризуються бурхливим розвитком модальної логіки. В наш час така логіка привертає значну увагу з боку як спеціалістів-логіків, так і прикладних математиків, філософів, лінгвістів, юристів. Модальна логіка дуже гнучка, вона може ефективно використовуватись для аналізу та моделювання найрізноманітніших предметних сфер та аспектів діяльності людини. Особливого значення модальна логіка набуває у зв'язку зі створенням сучасних інтелектуальних інформаційних систем, зокрема систем та баз знань, експертних систем. Надзвичайно важливим є використання апарату модальної та темпоральної логіки для адекватного опису і моделювання складних динамічних систем.

6.1. Алетичні модальні логіки

Основними модальностями загальної, або алетичної, модальної логіки є "необхідно" та "можливо". Відповідно до цих модальностей введемо модальні композиції (оператори) алетичної модальної логіки $\Box$ (необхідно) та $\Diamond$ (можливо). Такі оператори пов'язані співвідношеннями:

$$\neg\Diamond P = \Box\neg P;\ \neg\Box P = \Diamond\neg P.$$

На початкових етапах дослідження модальної логіки велика увага приділялась вивченню суперпозицій модальностей, таких, наприклад, як “необхідно, що необхідно”, “необхідно, що можливо”, “можливо, що необхідно, що можливо” і т. п. Такі суперпозиції модальностей з інтуїтивної точки зору дещо дивні, тому були запропоновані аксіоми, що дають змогу зводити складні модальності до модальностей певної форми.

Найбільше така звідність реалізована в системі $S5$, де кожна суперпозиція модальностей еквівалентна модальності без суперпозицій. Це означає, що для предиката P маємо шість таких предикатів: P, $\Box P$, $\diamond P$, $\neg P$, $\Box\neg P$, $\diamond\neg P$.

Можна використати слабші редукції модальностей. Найприроднiша з них зводить повторення однакових модальних операторів $\Box$ чи $\diamond$ до єдиного такого оператора. У відповідній системі $S4$ існує 14 різних модальностей: P, $\Box P$, $\diamond P$, $\Box\diamond P$, $\diamond\Box P$, $\Box\diamond\Box P$, $\diamond\Box\diamond P$ та сім дуальних модальностей, коли замість P беремо $\neg P$.

Для ще слабшої системи $S3$ існує 42 різні модальності, а для систем $S2$ та $S1$ кількість різних модальностей нескінченна.

Розглянемо детальніше найпопулярніші аксіоматичні системи алетичної модальної логіки на пропозиційному рівні.

Мова таких систем – розширення мови пропозиційної логіки. Алфавіт мови складається з множини Ps предикатних символів, символів пропозиційних композицій $\neg$, $\vee$ та символу $\Box$ модальної композиції (модального оператора) “необхідно”.

Множина формул Fm визначається індуктивно:

1) кожний $p\in Ps$ є формулою, такі формули атомарні;

2) нехай Φ та Ψ – формули, тоді $\neg\Phi$, $\vee\Phi\Psi$, $\Box\Phi$ – формули.

Символ $\diamond$ трактується як скорочення для $\neg\Box\neg$.

Опис аксіом для систем $S1$–$S5$, запропонований К. Льюїсом, наведено в [11]. Розглянемо тут дещо іншу аксіоматику для систем $S4$ та $S5$, але почнемо розгляд із слабших систем K та T.

Система *K*. Множина аксіом системи складається з аксіом пропозиційної логіки та аксіом, що задаються такою схемою: *AxNr*) $\Box(\Phi\to\Psi)\to(\Box\Phi\to\Box\Psi)$.

Правила виведення складаються з правил виведення пропозиційної логіки, до яких додається *правило модалізації*: ПМ) $\Phi\ \vdash\Box\Phi$.

Системи модальної логіки, які містять схему аксіом *AxNr* та правило ПМ, називаються нормальними.

Зауважимо, що системи *S*1, *S*2 та *S*3 не є нормальними.

Система *T*. Множина аксіом складається з аксіом системи *K* та аксіом, що задаються схемою *Ax*□) $\Box\Phi\to\Phi$. Правила виведення ті самі, що й для системи *K*.

Система *B* (Брауерова система). Множина аксіом складається з аксіом системи *T* та аксіом, що задаються схемою АхВ) $\Phi\to\Box\lozenge\Phi$. Правила виведення ті самі, що й для системи *K*.

Система *S*4. Множина аксіом складається з аксіом системи *T* та аксіом, що задаються схемою *AxS*4) $\Box\Phi\to\Box\Box\Phi$. Правила виведення ті самі, що й для системи *K*.

Система *S*5. Множина аксіом складається з аксіом системи *T* та аксіом, що задаються схемою *AxS*5) $\lozenge\Phi\to\Box\lozenge\Phi$.

Правила виведення ті самі, що й для системи *K*.

Той факт, що формула Φ є теоремою системи *K*, системи *T*, системи *B*, системи *S*4, системи *S*5, позначаємо відповідно $_K|\!-$, $_T|\!-$, $_B|\!-$, $_{S4}|\!-$, $_{S5}|\!-$.

Приклад 6.1.1. Повний опис пропозиційної системи *S*4.

Аксіоми:

*Ax*ПР) $\neg\Phi\vee\Phi$ (пропозиційні аксіоми).

AxNr) $\Box(\Phi\to\Psi)\to(\Box\Phi\to\Box\Psi)$.

Ax□) $\Box\Phi\to\Phi$.

*AxS*4) $\Box\Phi\to\Box\Box\Phi$.

Правила виведення:

П1) $\Phi\ |\!-\Psi\vee\Phi$ – правило розширення;

П2) $\Phi\vee\Phi\ |\!-\Phi$ – правило скорочення;

П3) $\Phi\vee(\Psi\vee\Xi)\ |\!-(\Phi\vee\Psi)\vee\Xi$ – правило асоціативності;

П4) $\Phi\vee\Psi,\ \neg\Phi\vee\Xi\ |\!-\Psi\vee\Xi$ – правило перетину;

ПМ) $\Phi\ |\!-\Box\Phi$ – правило модалізації.

Семантику наведених аксіоматичних систем модальної логіки можна задати різними способами. Спершу було запропоновано алгебраїчні семантики. Але такі семантики виявились не зовсім прийнятними з інтуїтивної точки зору, що змусило шукати іншу, змістов-

нішу інтерпретацію модальних систем. Такими є реляційні семантики, або семантики можливих світів.

Моделлю можливих світів, або *реляційною моделлю*, назвемо трійку $M = (S, \triangleright, I)$, де S – множина світів; $\triangleright$ – бінарне відношення на $\boldsymbol{S}$; $\boldsymbol{I}$ – відображення $\boldsymbol{I}$: $Ps \times \boldsymbol{S} \to \{T, F\}$ інтерпретації атомарних формул на світах.

Нехай $\alpha, \beta \in \boldsymbol{S}$. Традиційно $\alpha \triangleright \beta$ трактуємо так: світ β *можливий* відносно світу α, або світ β *досяжний* із світу α. Це означає, що будь-яке твердження, істинне у світі β, можливе у світі α. Тому відношення $\triangleright$ називають відношенням досяжності.

За такого розуміння кожне твердження, істинне в α, можливе в α, звідки $\alpha \triangleright \alpha$. Це означає, що відношення $\triangleright$ рефлексивне.

Відображення $\boldsymbol{I}$: $Ps \times \boldsymbol{S} \to \{T, F\}$ природним чином індуктивно продовжується до відображення $\boldsymbol{J}$**:** $Fm \times \boldsymbol{S} \to \{T, F\}$:

1) $\boldsymbol{J}(A, \alpha) = \boldsymbol{I}(A, \alpha)$ для всіх $A \in Ps$;

2) $\boldsymbol{J}(\neg\Phi, \alpha) = \neg(\boldsymbol{J}(\Phi, \alpha))$;

3) $\boldsymbol{J}(\vee\Phi\Psi, \alpha) = \vee(\boldsymbol{J}(\Phi, \alpha), \boldsymbol{J}(\Phi, \alpha))$;

4) $\boldsymbol{J}(\Box\Phi, \alpha) = T \Leftrightarrow \boldsymbol{J}(\Phi, \beta) = T$ для всіх $\beta \in \boldsymbol{S}$ таких, що $\alpha \triangleright \beta$, інакше $\boldsymbol{J}(\Box\Phi, \alpha) = F$.

Останнє означає: $\Box\Phi$ істинна у світі α, якщо Φ істинна в усіх світах, досяжних із α.

Формула Φ *істинна в моделі* $\boldsymbol{M}$, що позначаємо $\boldsymbol{M} \models \Phi$, якщо для всіх $\alpha \in \boldsymbol{S}$ маємо $\boldsymbol{J}(\Phi, \alpha) = T$.

Модель можливих світів $\boldsymbol{M}$ називають:

1) T-моделлю, якщо відношення $\triangleright$ рефлексивне;

2) B-моделлю, якщо відношення $\triangleright$ рефлексивне і симетричне;

3) $S4$-моделлю, якщо відношення $\triangleright$ рефлексивне і транзитивне;

4) $S5$-моделлю, якщо відношення $\triangleright$ рефлексивне, транзитивне і симетричне.

Формула Φ *T-істинна* (*B-істинна*, *S4-істинна*, *S5-істинна*), що позначаємо $\models_T \Phi$ (відповідно $\models_B \Phi$, $\models_{S4} \Phi$, $\models_{S5} \Phi$), якщо Φ істинна на кожній T-моделі (B-моделі, $S4$-моделі, $S5$-моделі).

Між аксіомами AxB, $AxS4$, $AxS5$ та характерними властивостями B-моделей, $S4$-моделей, $S5$-моделей існує [11] безпосередній зв'язок. А саме:

- аксіома AxB дає умову симетричності відношення $\triangleright$;
- аксіома $AxS4$ дає умову транзитивності відношення $\triangleright$;
- аксіома $AxS5$ дає умову транзитивності та симетричності відношення $\triangleright$.

Використовуючи аксіоми з модальностями, можна описати інші властивості відношення $\triangleright$, наприклад щільність, зв'язність, функціональність та ін. [5]. Зокрема, щільність описується схемою аксіом $\Box\Box\Phi\rightarrow\Box\Phi$.

Багато важливих властивостей відношення досяжності описуються відповідними схемами аксіом. Тому реляційна семантика має великий успіх. Проте [5] деякі властивості відношення $\triangleright$, зокрема іррефлексивність, асиметричність, антисиметричність, описати аксіомами такого типу неможливо.

Для розглянутих нами модальних систем справджуються теореми коректності та повноти [11].

Теорема 6.1.1. *Для кожної формули* Φ :

1) $_{T}|\!-\Phi \Leftrightarrow {}_{T}|\!=\Phi$;

2) $_{B}|\!-\Phi \Leftrightarrow {}_{B}|\!=\Phi$;

3) $_{S4}|\!-\Phi \Leftrightarrow {}_{S4}|\!=\Phi$;

4) $_{S5}|\!-\Phi \Leftrightarrow {}_{S5}|\!=\Phi$.

В загальнішому вигляді семантика можливих світів стосовно композиційно-номінативних модальних логік розглянута в [10].

6.2. Темпоральні логіки

Основними модальностями часової, або темпоральної, логіки є відомі ще з античних часів модальності "завжди було", "колись було", "завжди буде", "колись буде".

Відповідно до таких модальностей введемо модальні композиції (оператори) темпоральної логіки: $\Box_{\uparrow}$ (завжди буде), $\Box_{\downarrow}$ (завжди було), $\Diamond_{\uparrow}$ (колись буде) та $\Diamond_{\downarrow}$ (колись було).

Модальні оператори $\Box_{\uparrow}$, $\Box_{\downarrow}$, $\Diamond_{\uparrow}$, $\Diamond_{\downarrow}$ називають базовими часовими (темпоральними) операторами. Вони пов'язані такими співвідношеннями:

$\neg\Diamond_{\uparrow}P = \Box_{\uparrow}\neg P$;

$\neg\Box_{\uparrow}P = \Diamond_{\uparrow}\neg P$;

$\neg\Diamond_{\downarrow}P = \Box_{\downarrow}\neg P$;

$\neg\Box_{\downarrow}P = \Diamond_{\downarrow}\neg P$.

Для темпоральних модальних систем можна вважати базовими оператори $\Box_{\uparrow}$ та $\Box_{\downarrow}$. Тоді оператори $\Diamond_{\uparrow}$ та $\Diamond_{\downarrow}$ є похідними часовими операторами і визначатимуться так:

$\Diamond_{\uparrow}P$ означає $\neg\Box_{\uparrow}\neg P$;

$\Diamond_{\downarrow}P$ означає $\neg\Box_{\downarrow}\neg P$.

Розглянемо аксіоматичні системи темпоральної модальної логіки на пропозиційному рівні.

Мова таких систем є розширенням мови пропозиційної логіки. Алфавіт мови складається з множини *Ps* предикатних символів, символів пропозиційних композицій $\neg$, $\vee$ та символів $\Box_{\uparrow}$, $\Box_{\downarrow}$ відповідних часових операторів.

Множина формул *Fm* визначається індуктивно:

1) кожний $P \in Ps$ є формулою, такі формули атомарні;

2) нехай Φ та Ψ – формули, тоді $\neg\Phi$, $\vee\Phi\Psi$, $\Box_{\uparrow}\Phi$ $\Box_{\downarrow}\Phi$ – формули.

Символи $\Diamond_{\uparrow}$ та $\Diamond_{\downarrow}$ – це скорочення для $\neg\Box_{\uparrow}\neg$ та $\neg\Box_{\downarrow}\neg$.

Мінімальне темпоральне числення є аналогом системи *K*. Таке числення позначають K_t. Аксіомами системи K_t є аксіоми пропозиційної логіки і аксіоми, що задаються такими схемами:

$AxNr_{\uparrow}$) $\Box_{\uparrow}(\Phi\to\Psi)\to(\Box_{\uparrow}\Phi\to\Box_{\uparrow}\Psi)$;

$AxNr_{\downarrow}$) $\Box_{\downarrow}(\Phi\to\Psi)\to(\Box_{\downarrow}\Phi\to\Box_{\downarrow}\Psi)$;

$AxT_{\uparrow}$) $\Phi\to\Box_{\uparrow}\Diamond_{\downarrow}\Phi$;

$AxT_{\downarrow}$) $\Phi\to\Box_{\downarrow}\Diamond_{\uparrow}\Phi$.

Дві перші аксіоми є стандартними модальними аксіомами для $\Box_{\uparrow}$ та $\Box_{\downarrow}$. Аксіоми $AxT_{\uparrow}$ та $AxT_{\downarrow}$ відображають принципи змішування часів.

Правила виведення темпорального числення складаються з правил виведення пропозиційної логіки, до яких додаються правила модалізації для $\Box_{\uparrow}$ та $\Box_{\downarrow}$:

ПМ$_{\uparrow}$) $\Phi \vdash \Box_{\uparrow}\Phi$;

ПМ$_{\downarrow}$) $\Phi \vdash \Box_{\downarrow}\Phi$.

Темпоральне числення T_t є аналогом системи *T*. Для такого числення множина аксіом складається з аксіом числення K_t та аксіом, що задаються такими схемами:

$Ax\Box_{\uparrow}$) $\Box_{\uparrow}\Phi\to\Phi$;

$Ax\Box_{\downarrow}$) $\Box_{\downarrow}\Phi\to\Phi$.

Темпоральне числення B_t є аналогом системи B. Множина аксіом числення B_t складається з аксіом числення T_t та аксіом, що задаються такими схемами:

$Ax\mathrm{B}_{\uparrow}$) $\Phi \to \Box_{\uparrow} \diamond_{\uparrow} \Phi$;

$Ax\mathrm{B}_{\downarrow}$) $\Phi \to \Box_{\downarrow} \diamond_{\downarrow} \Phi$.

Темпоральне числення $S4_t$ є аналогом системи $S4$. Множина аксіом числення $S4_t$ складається з аксіом числення T_t та аксіом, що задаються такими схемами:

$AxS4_{\uparrow}$) $\Box_{\uparrow}\Phi \to \Box_{\uparrow}\Box_{\uparrow}\Phi$;

$AxS4_{\downarrow}$) $\Box_{\downarrow}\Phi \to \Box_{\downarrow}\Box_{\downarrow}\Phi$.

Аналогом системи $S5$ є темпоральне числення $S5_t$. Множина аксіом числення $S5_t$ складається з аксіом числення T_t та аксіом, що задаються такими схемами:

$AxS5_{\uparrow}$) $\diamond_{\uparrow}\Phi \to \Box_{\uparrow}\diamond_{\uparrow}\Phi$;

$AxS5_{\downarrow}$) $\diamond_{\downarrow}\Phi \to \Box_{\downarrow}\diamond_{\downarrow}\Phi$.

Правила виведення темпоральних числень T_t, B_t, $S4_t$, $S5_t$ такі самі, що й для K_t.

Реляційна семантика темпоральної логіки задається подібно до реляційної семантики алетичної модальної логіки. Аналогічно поняттю моделі можливих світів вводимо поняття темпоральної моделі, але тепер $\triangleright$ фактично вказує напрям часу – від минулого до майбутнього.

Визначення відображення інтерпретації $\boldsymbol{J}$: $Fm \times \boldsymbol{S} \to \{T, F\}$ для темпоральних логік відрізняється від відповідного визначення для алетичної модальної логіки тільки тим, що замість пункту 4) маємо два пункти $4_{\uparrow}$) та $4_{\downarrow}$) для формул вигляду $\Box_{\uparrow}\Phi$ та $\Box_{\downarrow}\Phi$:

$4_{\uparrow}$) $\boldsymbol{J}(\Box_{\uparrow}\Phi, \alpha) = T \Leftrightarrow \boldsymbol{J}(\Phi, \beta) = T$ для всіх $\beta \in \boldsymbol{S}$ таких, що $\alpha \triangleright \beta$, інакше $\boldsymbol{J}(\Box_{\uparrow}\Phi, \alpha) = F$;

$4_{\downarrow}$) $\boldsymbol{J}(\Box_{\downarrow}\Phi, \alpha) = T \Leftrightarrow \boldsymbol{J}(\Phi, \beta) = T$ для всіх $\beta \in \boldsymbol{S}$ таких, що $\beta \triangleright \alpha$, інакше $\boldsymbol{J}(\Box_{\downarrow}\Phi, \alpha) = F$.

Інший варіант пропозиційної темпоральної логіки – лінійна темпоральна логіка – розглянуто в [5]. Модальними операторами такої логіки є $\Box$, $\diamond$ та **0.** Тут $\mathbf{0}P$ означає "в наступний момент P".

Великий інтерес становить розгляд таких розширень мінімальної темпоральної логіки, моделі яких мають властивості фізичного часу (лінійність, дискретність чи континуальність, щільність, скінченність

чи нескінченність, циклічність і т. п.). Відповідні аксіоматичні системи розглядались в багатьох роботах (див., наприклад, [4, 5]).

Популярними також є дослідження, присвячені метричним темпоральним логікам. Мінімальне метричне темпоральне числення побудоване А. Прайором. Детальніше про таке числення див. [4].

Основними часовими операторами є $\uparrow \boldsymbol{n}$ – "буде через n одиниць часу" та $\downarrow \boldsymbol{n}$ – "було n одиниць часу тому". Неметричні часові оператори $\Box_{\uparrow}$, $\Box_{\downarrow}$, $\diamond_{\uparrow}$, $\diamond_{\downarrow}$ виражаються через $\uparrow \boldsymbol{n}$ та $\downarrow \boldsymbol{n}$ так:

$\Box_{\uparrow} P = \forall \boldsymbol{n} (\uparrow \boldsymbol{n} P)$;

$\diamond_{\uparrow} P = \exists \boldsymbol{n} (\uparrow \boldsymbol{n} P)$;

$\Box_{\downarrow} P = \forall \boldsymbol{n} (\downarrow \boldsymbol{n} P)$;

$\diamond_{\downarrow} P = \exists \boldsymbol{n} (\downarrow \boldsymbol{n} P)$.

Метричну темпоральну логіку узагальнив Д. Кліффорд [5]. Він увів "потоки часу" та відповідні часові оператори $\tau\uparrow \boldsymbol{n}$ і $\tau\downarrow \boldsymbol{n}$, які означають "в потоці τ через n одиниць часу буде істинним" і "в потоці τ n одиниць часу тому було істинним". Кліффорд також описав відповідні темпоральні числення.

6.3. Деонтичні логіки

Деонтична логіка – розділ модальної логіки, в якому вивчаються міркування з термінами "обов'язково", "дозволено", "заборонено". Деонтичну логіку називають також логікою норм.

Предметом деонтичної логіки є нормативні міркування. Термін "деонтична" походить від давньогрецького *deontis*, що означає "так, як має бути", "належним чином".

Творцем сучасної деонтичної логіки є Г. фон Врігт, який у 1951 р. сформулював проблеми деонтичної логіки та навів їх розв'язки. Подальшого розвитку деонтична логіка набула в роботах С. Кангера, А. Андерсона, А. Прайора та ін.

Зауважимо, що сама можливість побудови деонтичної логіки сумнівна з точки зору багатьох логіків та філософів. На їхню думку, логіка має справу з реченнями, які є істинними або хибними. Але деонтичні (нормативні) речення не є ані істинними, ані хибними, оскільки вони належать не до реального світу, а до ідеального "світу належного".

Деонтичну логіку природно будувати на базі семантики можливих світів. Поняття "деонтичний світ" введено Е. Кантом, він запропонував класичну інтерпретацію можливого світу як деонтичного. Кант

визначив моральний світ як такий, що відповідає всім модальним законам, яким він може бути згідно з волею розумних істот і яким він має бути згідно із законами моральності. На думку Канта, моральний світ – це ідеальний світ, один з варіантів можливого світу, де все ідеально діє і взаємодіє. Тому, узагальнюючи, деонтичний світ – це світ, який ми хотіли б мати.

Можна розглядати конкретні випадки деонтичного світу. Наприклад, конституція – це опис ідеальної держави, якою вона має бути, опис належних дій та вчинків президента, міністрів та ін. Кант постулював існування єдиного "модального" світу, але деонтична логіка допускає існування багатьох світів. Наприклад, декілька деонтичних світів можна трактувати як різні проекти конституції, що розглядаються Верховною Радою.

Концепцію семантики можливих світів для деонтичної логіки можна конкретизувати у двох аспектах:

1) можливий світ – це деонтичний світ;

2) відношення досяжності – це відношення між світом, в якому будується деонтичний світ, та цим деонтичним світом.

У мінімальному (стандартному) варіанті деонтичної логіки маємо базовий деонтичний оператор **О** – "обов'язково". Похідні деонтичні оператори визначаються так (Q – предикат):

1) оператор **Р** – "дозволено": **Р**Q означає $\neg$**О**$\neg Q$;

2) оператор **F** – "заборонено": **F**Q означає **О**$\neg Q$;

3) оператор **І** – "байдуже": **І**Q означає **Р**Q **&** **Р**$\neg Q$.

Стандартним чином вводимо мову деонтичної логіки та аксіоматичні системи. На пропозиційному рівні отримуємо деонтичну систему *DS*. Множина аксіом і правил виведення системи *DS* складається з аксіом і правил виведення пропозиційної логіки, до яких додаються аксіоми, що є аналогами аксіом *AxNr* і *Ax*$\Box$, та правило виведення П**О**, що є аналогом правила модалізації. Такі аксіоми задаються схемами:

AxDNr) **О**(Ф$\to$Ψ)$\to$(**О**Ф$\to$**О**Ψ);

*Ax*D$\Box$) **О**Ф$\to$**Р**Ф.

Правило виведення П**О** має вигляд

П**О**) Ф $\vdash$ **О**Ф.

Властивості оператора **О** подібні до властивостей оператора $\Box$. Але в деонтичній логіці формула **О**Ф$\to$Ф в загальному випадку не є істинною. Вона означає: "якщо має бути Ф, то Ф".

Реляційна семантика деонтичної логіки задається подібно до реляційної семантики алетичної модальної логіки. На відміну від алетичної модальної логіки, в деонтичній логіці відношення досяжності $\triangleright$ мусить бути:

1) іррефлексивним, тобто невірно $\alpha \triangleright \alpha$;

2) мати властивість незавершеності, тобто для кожного світу α існує світ β такий, що $\alpha \triangleright \beta$.

Ці властивості можна трактувати як незбіг належного і реального та як існування для кожного світу свого деонтичного світу.

У деонтичній логіці існують твердження, інтуїтивно неприйнятні, але які можна формально довести в описаній системі. Це засвідчує не зовсім адекватну формалізацію деонтичної логіки. Зокрема, А. Прайор показав, що формула $\mathbf{O}\neg\Phi\rightarrow\mathbf{O}(\Phi\rightarrow\Psi)$ є теоремою, але вона інтуїтивно неприйнятна. Наприклад, її можна інтерпретувати так: в корпусі факультету заборонено палити, отже, якщо ви палите, можна вбивати. Це дає підстави сумніватися в трактуванні Г. Фон Врігтом формули $\mathbf{O}(\Phi\rightarrow\Psi)$ як твердження про похідний обов'язок: зробити Ψ у разі виконання Φ.

Прайор запропонував визначити похідний обов'язок формулою $\Phi\rightarrow\mathbf{O}\Psi$, але тоді теоремою буде $\neg\Phi\rightarrow(\Phi\rightarrow\mathbf{O}\Psi)$, яка парадоксальна, бо означає: те, що не існує, зобов'язує робити все, що завгодно. Парадоксальною є також формула $\mathbf{F}\Phi\rightarrow\mathbf{F}(\Phi\,\&\,\Psi)$. Ця формула означає: із заборони дії випливає заборона поєднувати її з іншою дією. Це вірно, зокрема, і для дії, що компенсує порушення заборонної дії. Наприклад, якщо заборонено лаятися, то заборонено лаятися і вибачатися.

Побудова кванторних деонтичних логік стикається з багатьма принциповими труднощами, які до цих пір ще достатньо не прояснені. Зокрема, як інтерпретувати предметні змінні, сполучення кванторів та деонтичних операторів, як уникати парадоксів на кванторному рівні. Наприклад, формули $\mathbf{O}\exists x\neg\Phi(x)$ та $\neg\mathbf{P}\forall x\Phi(x)$ еквівалентні, але за деонтичною сутністю різні. Перша формула стверджує, що обов'язково існує індивід, який виконує $\neg\Phi$; друга – що не дозволено кожному виконувати Φ. Парадоксальною є також формула $\exists x\mathbf{O}\neg\Phi\rightarrow\mathbf{O}\exists x\neg\Phi$, яка стверджує: "Якщо існує індивід, що обов'язково порушує закони, то обов'язково існує індивід, що порушує закони". Зауважимо, що вже саме обов'язкове існування індивіда виглядає дещо парадоксально і вимагає уточнення.

Варіантом деонтичної логіки є запропонована А. Андерсоном *логіка санкцій*. Андерсон вирішує проблему побудови логіки норм, що відповідає реальності, таким чином. Він виражає деонтичний оператор **O** за допомогою оператора □ та константного предиката **S**, який означає санкцію (штраф).

Андерсон визначає **O***Q* як □(¬*Q*→**S**), тоді **F***Q* означає □(*Q*→**S**), **I***Q* означає □(*Q*&¬**S**). Відповідне числення *Sd* логіки санкцій містить аксіому ¬□**S**, яка означає, що кари можна уникнути.

Різновидністю логіки санкцій є деонтична логіка з предикатною константою **N** (нормативний кодекс). Тоді **O***Q* задається як □(**N**→*Q*), **F***Q* задається як □(**N**→¬*Q*), **P***Q* – як **N**&*Q*. Відповідне числення містить аксіому ✧**N** , яка означає можливість дотримання нормативного кодексу.

6.4. Епістемічні логіки

Епістемічна логіка – це розділ модальної логіки, в якому досліджуються міркування з такими модальностями, як "відомо", "вірно", "доведено", "спростовано". Засновником епістемічної логіки як науки є відомий фінський логік Я. Хінтікка. В 1962 р. у своїй роботі "Знання і опінія" він вирішив застосувати апарат модальної логіки для порівняльної логічної характеристики того, що є змістом знання і що є змістом опінії, віри. Для епістемічної логіки він запропонував семантику можливих світів, назвавши такі світи епістемічними.

Постулатами епістемічної логіки прийнято вважати [4]:

1) знання пов'язане з реальністю;

2) вірування пов'язані з внутрішнім світом опіній;

3) опінія – це уявлення суб'єкта про свою мислиму діяльність;

4) сумнів – це невпевненість у знанні;

5) незнання – це особливий стан (Сократ казав: "Я знаю тільки те, що я нічого не знаю");

6) знання – це обґрунтована віра.

Основними модальностями епістемічної логіки є **К** та **В**, які відповідають знанню та опінії, вірі. **К***Q* можна трактувати так: "відомо, що *Q"*; **В***Q* – "вірую, що *Q"*.

У загальнішому вигляді вводять параметричні модальні оператори **К***x* та **В***x*. Найчастіше розглядають скінченні множини таких операторів, що відповідає наявності скінченної множини суб'єктів знання – експертів.

У цьому випадку **К***x*Q трактується так: “експерт *x* знає, що *Q”*.

В*x*Q трактується так: “експерт *x* вірить, що *Q”*.

Між знанням і вірою маємо такі співвідношення:

К*x*Q → **В***x*Q; **В***x*Q → **К***x***В***x*Q; **К***x*Q → **В***x***К***x*Q.

Для логіки знання і віри маємо також:

В*x*¬Q → ¬**В***x*Q;

В*x*¬**В***x*Q → ¬**В***x*Q;

В*y***К***x*Q ↔ **В***y*(**В***x*Q &Q).

У деяких версіях епістемічної логіки додатково вводять модальні оператори сумніву **С** та спростування **Z**. **С***x*Q трактується так: “експерт *x* сумнівається, що *Q”*; **Z***x*Q – “експерт *x* спростовує *Q”*.

Розглянемо епістемічну логіку знання пропозиційного рівня. У цьому випадку вводимо тільки модальний оператор знання. В найпростішому варіанті маємо один такий оператор **К**, що відповідає наявності єдиного суб’єкта знання (експерта).

Алфавіт мови пропозиційної епістемічної логіки з одним експертом складається з множини *Ps* предикатних символів, символів пропозиційних композицій ¬, ∨ та символу **К** модального оператора знання.

Множина формул *Fe* визначається індуктивно:

1) кожний *P*∈ *Ps* є формулою, такі формули атомарні.

2) нехай **Ф** та **Ψ** – формули, тоді **¬Ф**, **∨ФΨ**, **КФ** – формули.

Множина аксіом пропозиційної епістемічної логіки складається з аксіом пропозиційної логіки та аксіом, що задаються такими схемами:

AxEN) **К**(**Ф**→**Ψ**)→(**КФ**→**КΨ**);

AxRe) **КФ**→**Ф**;

AxPR) **КФ**→**ККФ**;

AxNR) ¬**КФ**→**К**¬**КФ**.

Аксіома *AxRe* називається *аксіомою реальності знання.* Вона є аналогом аксіоми *Ax*□ алетичної модальної логіки.

Аксіома *AxPR* називається *аксіомою позитивної рефлексії* (якщо я знаю, то я знаю, що знаю). Вона є аналогом аксіоми *AxS*4 алетичної модальної логіки.

Аксіома *AxNR* називається *аксіомою негативної рефлексії* (якщо я не знаю, то я знаю, що не знаю). Вона є аналогом аксіоми *AxS*5 алетичної модальної логіки.

Правила виведення складаються з правил виведення пропозиційної логіки, до яких додається *правило знання*: ПК) **Ф |–КФ.**

Система епістемічної модальної логіки, яка містить схеми аксіом *AxEN* та *AxRe*, є аналогом системи *T* алетичної модальної логіки. Таку систему називають $T_{(1)}$.

Система епістемічної модальної логіки, яка містить схеми аксіом *AxEN*, *AxRe* та *AxPR*, є аналогом системи *S*4 алетичної модальної логіки. Таку систему називають $S4_{(1)}$.

Система епістемічної модальної логіки, яка містить схеми аксіом *AxEN*, *AxRe* та *AxNR*, є аналогом системи *S*5 алетичної модальної логіки. Таку систему називають $S5_{(1)}$.

Реляційна семантика (семантика можливих світів) для систем пропозиційної епістемічної логіки задається подібно до реляційної семантики алетичної модальної логіки.

Відношення досяжності $\triangleright$ трактується таким чином: $\alpha \triangleright \beta$ означає, що експерт в ситуації α розглядає ситуацію β як можливу. При цьому аксіома *AxRe* дає умову рефлексивності відношення досяжності $\triangleright$. Аксіома *AxPR* дає умову транзитивності $\triangleright$. Аксіома *AxNR* дає умову транзитивності та симетричності відношення $\triangleright$.

Узагальнюючи системи епістемічної логіки з одним експертом, вводимо скінченну множину операторів знання $\mathbf{К_1}, \ldots, \mathbf{К_n}$. Відповідно визначається мова епістемічної логіки, аксіоматичні системи замість однієї схеми аксіом AxEN містять *n* схем аксіом такого ж типу для $\mathbf{К_1}, \ldots, \mathbf{К_n}$. Те саме стосується схем аксіом *AxRe*, *AxPR*, *AxNR*.

Системи пропозиційної епістемічної логіки з *n* експертами, відповідні системам $T_{(1)}$, $S4_{(1)}$, $S5_{(1)}$, називають $T_{(n)}$, $S4_{(n)}$, $S5_{(n)}$.

При визначенні реляційної семантики таких систем замість єдиного відношення $\triangleright$ маємо *n* таких відношень $\triangleright_1, \ldots, \triangleright_n$. При цьому $\alpha \triangleright_k \beta$ означає, що *k*-й експерт в ситуації α розглядає ситуацію β як можливу.

Подальший розвиток епістемічної логіки зумовив виникнення систем загального знання, систем внутрішнього (розподіленого) знання [15]. Варто згадати також епістемічну логіку Левеска, семантика якої містить поняття явних і неявних опінійй, та узагальнюючу її логіку обізнаності [15]. На базі епістемічної логіки Левеска створено мову *KL* опису систем знань з неповною інформацією, яка дає змогу не тільки здійснювати запити бази знань, а й отримувати інформацію про повноту чи неповноту отриманих знань. Детальніше про це див. [15].

Питання для самоконтролю

1. Що таке модальність?
2. Які модальності називають алетичними? Наведіть приклади.
3. Які модальності називають темпоральними? Наведіть приклади.
4. Які модальності називають епістемічними? Наведіть приклади.
5. Які модальності називають деонтичними? Наведіть приклади.
6. Де використовуються модальні логіки?
7. Які співвідношення пов'язують модальні оператори $\Box$ та $\Diamond$?
8. Опишіть алфавіт мови систем алетичної модальної логіки.
9. Дайте визначення формули мови систем алетичної модальної логіки.
10. Опишіть систему *K* алетичної модальної логіки.
11. Який вигляд має правило модалізації?
12. Опишіть систему *T* алетичної модальної логіки.
13. Опишіть систему *B* алетичної модальної логіки.
14. Опишіть систему *S*4 алетичної модальної логіки.
15. Опишіть систему *S*5 алетичної модальної логіки.
16. Що таке модель можливих світів (реляційна модель) алетичної модальної логіки?
17. Чому відношення на світах $\triangleright$ називають відношенням досяжності?
18. Як задається значення формули у світі α ?
19. Як визначається істинність формули у реляційній моделі?
20. Дайте визначення реляційної:
    - *T*-моделі;
    - *B*-моделі;
    - *S*4-моделі;
    - *S*5-моделі.
21. Дайте визначення:
    - *T*-істинної формули;
    - *B*-істинної формули;
    - *S*4-істинної формули;
    - *S*5-істинної формули.
22. Які співвідношення пов'язують аксіоми *AxB*, *AxS*4, *AxS*5 та властивості відношення досяжності $\triangleright$?
23. Сформулюйте теореми коректності та повноти для систем алетичної модальної логіки.
24. Назвіть базові часові (темпоральні) оператори.

25. Які співвідношення пов'язують часові оператори?
26. Опишіть алфавіт мови систем темпоральної модальної логіки.
27. Дайте визначення формули мови систем темпоральної модальної логіки.
28. Опишіть мінімальне темпоральне числення.
29. Опишіть темпоральне числення T_t.
30. Опишіть темпоральне числення B_t.
31. Опишіть темпоральне числення $S4_t$.
32. Опишіть темпоральне числення $S5_t$.
33. Опишіть реляційну семантику темпоральної логіки.
34. Як ми розуміємо відношення $\triangleright$ для випадку реляційної моделі темпоральної логіки?
35. Як задається значення формули у світі α для випадку темпоральної логіки?
36. Які ви знаєте оператори метричної темпоральної логіки?
37. Як виражаються оператори $\Box_{\uparrow}$, $\Box_{\downarrow}$, $\Diamond_{\uparrow}$, $\Diamond_{\downarrow}$ через метричні часові оператори?
38. Що вивчає деонтична логіка?
39. Що таке деонтичний світ?
40. Як конкретизувати концепцію семантики можливих світів для деонтичної логіки?
41. Які ви знаєте деонтичні оператори?
42. Опишіть мову деонтичної логіки.
43. Опишіть деонтичну систему *DS*.
44. Які властивості відношення досяжності $\triangleright$ для випадку реляційної семантики деонтичної логіки?
45. Чи можна вважати цілком адекватною формалізацію деонтичної логіки?
46. Які особливості має логіка санкцій Андерсена?
47. Що вивчає епістемічна логіка?
48. Вкажіть основні постулати епістемічної логіки.
49. Наведіть основні модальності епістемічної логіки.
50. Які співвідношення між операторами знання і віри?
51. Які особливості має епістемічна логіка знання?
52. Опишіть мову епістемічної логіки знання з одним експертом.
53. Назвіть аксіоми:
    - реальності знання;
    - позитивної рефлексії;
    - негативної рефлексії.

54. Аналогом яких систем алетичної модальної логіки є системи епістемічної логіки $T_{(1)}$, $S4_{(1)}$, $S5_{(1)}$?
55. Як визначається реляційна семантика епістемічної логіки знання з одним експертом?
56. Як трактується відношення досяжності $\triangleright$ для випадку реляційної семантики епістемічної логіки?
57. Які співвідношення пов'язують аксіоми *AxRe*, *AxPR*, *AxNR* та властивості відношення досяжності $\triangleright$?
58. Опишіть мову епістемічної логіки знання з n експертами.
59. Назвіть системи епістемічної логіки знання з n експертами, відповідні системам $T_{(1)}$, $S4_{(1)}$, $S5_{(1)}$.
60. Які особливості реляційної семантики епістемічної логіки знання з n експертами?

Вправи

1. В якому відношенні щодо логічного наслідку перебувають предикати P, $\Box P$ та $\diamond P$?

2. Дайте повні описи B-числення та $S5$-числення.

3. Поясніть зв'язок аксіоми $Ax\Box$ із рефлексивністю відношення досяжності $\triangleright$.

4. Поясніть зв'язок аксіом AxB, $AxS4$, $AxS5$ із відповідними властивостями відношення $\triangleright$ для B-систем, $S4$-систем, $S5$-систем.

5. Покажіть, що формули вигляду $\Box A \to A$, $A \to \Box A$, $\Box A \to \Box\Box A$, $\Box(A \vee B) \to (\Box A \vee \Box B)$ істинні не на всіх реляційних моделях. Наведіть відповідні приклади таких моделей.

6. Які властивості відношення $\triangleright$ описують наведені схеми аксіом:
   1) $\Box\Phi \to \diamond\Phi$;
   2) $\diamond\Phi \to \Box\Phi$;
   3) $\diamond\Phi \leftrightarrow \Box\Phi$;
   4) $\diamond\Box\Phi \to \Box\diamond\Phi$.

7. Доведіть в T-численні:
   1) якщо $\vdash A \to B$, то $\vdash \Box A \to \Box B$ та $\vdash \diamond A \to \diamond B$;
   2) якщо $\vdash A \leftrightarrow B$, то $\vdash \Box A \leftrightarrow \Box B$ та $\vdash \diamond A \leftrightarrow \diamond B$;
   3) $\vdash \diamond(A \to B) \to (\diamond A \to \diamond B)$;
   4) $\vdash \diamond(A \vee B) \leftrightarrow (\diamond A \vee \diamond B)$;
   5) $\vdash \Box(A \& B) \leftrightarrow (\Box A \& \Box B)$;
   6) $\vdash (\Box A \vee \Box B) \to \Box(A \vee B)$;

7) $\vdash \Diamond(A\&B)\rightarrow(\Diamond A\&\Diamond B)$;

8) $\vdash \Box(A\rightarrow\Diamond(B\rightarrow C))\rightarrow\Diamond(B\rightarrow\Box A\rightarrow\Diamond B)$.

8. Вкажіть схеми аксіом та правила виведення:

1) пропозиційної темпоральної $S4_t$-системи;

2) пропозиційної темпоральної $S5_t$-системи;

3) пропозиційної темпоральної B_t-системи.

9. Вкажіть схеми аксіом та правила виведення:

1) пропозиційної епістемічної $S4_{(2)}$-системи;

2) пропозиційної епістемічної $S4_{(3)}$-системи;

3) пропозиційної епістемічної $S5_{(2)}$-системи;

4) пропозиційної епістемічної $S5_{(3)}$-системи.

10. Дайте повний опис пропозиційних епістемічних систем $T_{(1)}$, $S4_{(1)}$, $S5_{(1)}$ та $T_{(n)}$, $S4_{(n)}$, $S5_{(n)}$. Визначіть реляційну семантику таких систем.

11. Поясніть парадоксальність деонтичної формули $x = y \rightarrow \mathbf{O}x = y$.

12. Дайте повний опис пропозиційного числення *Sd* логіки санкцій Андерсона.

13. Покажіть, що в численні *Sd*:

1) $\vdash \mathbf{O}A\rightarrow\mathbf{P}A$;

2) $\vdash \mathbf{O}(A\rightarrow B)\rightarrow(\mathbf{O}A\rightarrow\mathbf{O}B)$.

СПИСОК ЛІТЕРАТУРИ

Основна

1. *Андерсон Д. А.* Дискретная математика и комбинаторика. — М.: Вильямс, 2003. — 960 с.
2. *Булос Дж., Джеффри Р.* Вычислимость и логика. — М.: Мир, 1994. — 396 с.
3. *Ершов Ю. Л., Палютин Е. А.* Математическая логика. — М.: Наука, 1979. — 320 с.
4. *Ішмуратов А. Т.* Вступ до філософської логіки. — К.: Абрис, 1997. — 360 с.
5. *Капітонова Ю. В., Кривий С. Л., Летичевський О. А. та ін*. Основи дискретної математики. — К.: Наук. думка, 2002. — 579 с.
6. *Клини С.* Математическая логика. — М.: Мир, 1973. — 480 с.
7. *Кондаков Н. И.* Введение в логику. — М.: Наука, 1967. — 466 с.
8. *Лавров И. А., Максимова Л. Л.* Задачи по теории множеств, математической логике и теории алгоритмов. — М.: Наука, 1975. — 240 с.
9. *Мендельсон Э.* Введение в математическую логику. — М.: Наука, 1976. — 320 с.
10. *Нікітченко М. С., Шкільняк С. С.* Основи математичної логіки. — К.: Київ. ун-т, 2006. — 246 с.
11. *Фейс Р.* Модальная логика. — М.: Наука, 1974. — 520 с.
12. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. — М.: Наука, 1983. — 256 с.
13. *Шенфилд Дж.* Математическая логика. — М.: Наука, 1975. — 528 с.
14. *Шкільняк С. С.* Математична логіка: приклади і задачі. — К.: Київ. ун-т, 2002. — 56 с.

Додаткова

15. *Андон Ф. И., Яшунин А. Е., Резниченко В. А*. Логические модели интеллектуальных информационных систем. — К.: Наук. думка, 1999. — 396 с.
16. *Басараб И. А., Никитченко Н. С., Редько В. Н.* Композиционные базы данных. — К.: Либідь, 1992. — 192 с.
17. *Гильберт Д., Бернайс П.* Основания математики. — М.: Наука, 1982. — Т. 1, 2.
18. *Гиндикин С. Г.* Алгебра логики в задачах. — М.: Наука, 1972. — 288 с.
19. *Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л.* Алгебра, языки, программирование. — К.: Наук. думка, 1974. — 328 с.
20. *Карри Х.* Основания математической логики. — М.: Мир, 1969. — 568 с.
21. *Катленд Н.* Вычислимость. Введение в теорию рекурсивных функций. — М.: Мир, 1983. — 256 с.

22. *Клини С.* Введение в метаматематику. — М.: Иностр. лит., 1957. — 526 с.
23. *Колмогоров А. Н., Драгалин А. Г.* Введение в математическую логику. — М.: МГУ, 1982. — 120 с.
24. *Колмогоров А. Н., Драгалин А. Г.* Математическая логика. Дополнительные главы. — М.: МГУ, 1984. — 120 с.
25. *Костюк В. Н.* Элементы модальной логики. — К.: Наук. думка, 1978. — 179 с.
26. *Лисовик Л. П., Редько В. Н.* Алгоритмы и формальные системы. — К.: КГУ, 1981. — 112 с.
27. *Лісовик Л. П., Шкільняк С. С.* Теорія алгоритмів. — К.: Київ. ун-т, 2003. — 164 с.
28. *Мальцев А. И.* Алгебраические системы. — М.: Наука, 1970. — 392 с.
29. *Мальцев А. И.* Алгоритмы и рекурсивные функции. — М.: Наука, 1965. — 392 с.
30. *Манин Ю. И.* Доказуемое и недоказуемое. — М.: Сов. радио, 1979. — 168 с.
31. *Манин Ю. И.* Вычислимое и невычислимое. — М.: Сов. радио, 1980. — 128 с.
32. *Непейвода Н. Н.* Прикладная логика. — Новосибирск: НГУ, 2000. — 521 с.
33. *Нікітченко М. С., Шкільняк С. С.* Математична логіка. — К.: Київ. ун-т. — 2003. — 120 с.
34. *Нікітченко М. С., Шкільняк С. С.* Математична логіка. Додаткові розділи. — К.: Київ. ун-т. — 2004. — 77 с.
35. *Новиков П. С.* Элементы математической логики. — М.: Наука, 1973. — 400 с.
36. *Расева Е., Сикорский Р.* Математика метаматематики. — М.: Наука, 1972. — 592 с.
37. *Роджерс Х.* Теория рекурсивных функций и эффективная вычислимость. — М.: Мир, 1972. — 624 с.
38. *Семантика* модальных и интенсиональных логик. — М.: Прогресс, 1981. — 494 с.
39. *Смирнова Е. Д.* Логика и философия. — М.: РОССПЕН, 1996. — 304 с.
40. *Справочная* книга по математической логике / Под ред. Дж. Барвайса. — М.: Наука, 1982–1983. — Ч. 1–4 .
41. *Столл Р.* Множества. Логика. Аксиоматические теории. — М.: Просвещение, 1968. — 232 с.
42. *Такеути Г.* Теория доказательств. — М.: Мир, 1978. — 412 с.
43. *Успенский В. А., Семенов А. Л.* Теория алгоритмов: основные открытия и приложения. — М.: Наука, 1987. — 288 с.
44. *Хромой Я. В.* Математична логіка.– К.: Вища шк., 1983. — 208 с.
45. *Черч А.* Введение в математическую логику. — М.: Иностр. лит., 1960. — 486 с.

ПОКАЖЧИК ТЕРМІНІВ

ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ

ВСТУП

Поняття алгоритму належить до первісних понять математики, таких як поняття натурального числа, множини, функції. Загальні властивості алгоритмів вивчає розділ математики, який називається *теорією алгоритмів*.

Обчислювальні процеси алгоритмічного характеру відомі людству з глибокої давнини. Такими є арифметичні дії над цілими числами, знаходження найбільшого спільного дільника двох чисел, розкладання натурального числа на прості множники, процес розв'язування системи лінійних рівнянь методом послідовного виключення невідомих тощо. В явному вигляді поняття алгоритму сформувалося на початку XX століття.

Під *алгоритмом* звичайно розуміють скінченну множину точно визначених правил для чисто механічного розв'язування задач певного класу.

Така множина правил задає обчислювальний процес, названий *алгоритмічним*, що починається з довільного початкового даного (вибраного з деякої фіксованої для даного алгоритму множини початкових даних) і спрямований на отримання повністю визначеного цим початковим даним результату.

Таке формулювання можна розглядати лише як пояснення, а не як визначення, тому що поняття алгоритму внаслідок його первісності не можна виразити через інші поняття математики.

Наведене поняття можна далі уточнити, явно виокремивши характерні властивості алгоритму, до яких звичайно відносять:

1) *фінітність* — алгоритм є скінченним об'єктом, що є необхідною умовою його механічної реалізовності;

2) *масовість* — початкові дані для алгоритму можна вибирати з певної множини даних (можливо, нескінченної); це означає, що алгоритм призначений не для однієї конкретної задачі, а для класу однотипних задач;

3) *дискретність* — розчленованість процесу виконання алгоритму на окремі кроки; це означає, що алгоритмічний процес здійснюється в дискретному часі;

4) *елементарність* — кожен крок алгоритму має бути простим, елементарним, можливість виконання якого людиною або машиною не викликає сумнівів;

5) *результативність* — алгоритм має засоби, які дають змогу відбирати із даних, отриманих на певному кроці виконання, результативні дані, після чого алгоритм зупиниться.

До характерних властивостей алгоритму часто відносять *детермінованість* — однозначність процесу виконання алгоритму. Це означає, що при заданих початкових даних кожне дане, отримане на певному (не початковому) кроці, однозначно визначається даними, отриманими на попередніх кроках. Проте варто зазначити, що розроблено досить багато формальних моделей алгоритму, які не мають властивості детермінованості.

За допомогою алгоритму кожний конкретний результат отримується за скінченну кількість кроків із скінченної множини даних. Кажуть, що до таких початкових даних алгоритм *застосовний*.

Проте в деяких ситуаціях процес виконання алгоритму для певних початкових даних триває необмежено. У цьому разі кажуть, що до таких початкових даних алгоритм *незастосовний*.

Для опису алгоритму необхідно визначити множину його початкових даних і множину даних, до яких належать результати. Ці множини називають також множиною вхідних даних і множиною вихідних даних алгоритму.

Алгоритм із множиною вхідних даних X і множиною вихідних даних Y називають *X-Y-алгоритмом.*

Нехай алгоритм $\aleph$ має множину вхідних даних X.

Областю застосування алгоритму $\aleph$ називають підмножину $D \subseteq X$ таку, що до кожного $d \in D$ алгоритм $\aleph$ застосовний.

Якщо $\aleph$ видає результат b при роботі над вхідним даним d, це позначаємо $b = \aleph(d)$.

Кожний X-Y-алгоритм $\aleph$ визначає функцію $f: X \to Y$, взагалі кажучи часткову, задану таким чином.

Для кожного елемента $d \in X$ маємо $f(d) = \aleph(d)$, якщо $d \in D$, тобто $\aleph$ до d застосовний. Якщо $d \notin D$, то $f(d)$ невизначене, тобто алгоритм $\aleph$ до d незастосовний.

Кажуть, що такий алгоритм $\aleph$ *обчислює* функцію f.

Функція *алгоритмічно обчислювана* (АОФ), якщо існує алгоритм, який її обчислює.

Множина L *алгоритмічно перелічна,* якщо вона є областю значень деякої алгоритмічно обчислюваної функції, тобто існує алгоритм, який перелічує елементи множини L і тільки їх.

Множина L *алгоритмічно розв'язна* відносно множини U, якщо існує алгоритм, який дозволяє для кожного $x \in U$ визначати, $x \in L$ чи $x \notin L$.

Кожна алгоритмічно розв'язна непорожня множина є алгоритмічно перелічною.

Справді, нехай L розв'язна відносно U за допомогою алгоритму $\aleph$. Вкажемо алгоритм $\Re$, який перелічує L.

Зафіксуємо довільний елемент $b \in L$. На вхід $\Re$ подаємо довільний $x \in U$ і запускаємо $\aleph$ над x. Якщо $x \in L$, то $\Re$ видає x як результат; якщо $x \notin L$, то $\Re$ видає b як результат.

Узагальненням поняття алгоритму є поняття *відносного алгоритму*, або *алгоритму з оракулом*. На деяких кроках такий алгоритм може звертатися до певного зовнішнього відносно алгоритму об'єкта — оракула. Видані оракулом відповіді трактуються як дані, вироблені на таких кроках звертання.

Кроки звертання до оракула, узагалі кажучи, неелементарні. Водночас поняття елементарності кроків відносне. Можна вважати, що звичайний алгоритм весь час звертається до деякого оракула, але цей оракул відповідає на запитання настільки прості, що сам по собі він непомітний і вважається внутрішньою частиною обчислювальних засобів алгоритму, а не виступає як зовнішній щодо алгоритму об'єкт.

Отже, є всі підстави вважати, що первісним є поняття саме відносного алгоритму.

Узагальненням поняття алгоритмічно обчислюваної функції є поняття *відносно обчислюваної функції*, або функції, алгоритмічно обчислюваної відносно оракула.

Функція називається *алгоритмічно обчислюваною відносно оракула* $\wp$, якщо існує алгоритм з оракулом $\wp$, який її обчислює.

Із поняттям алгоритму тісно пов'язане поняття числення, яке є настільки ж фундаментальним, як і поняття алгоритму. Поняття числення відбиває та узагальнює інтуїтивну уяву про індуктивне породження об'єктів, яке поширене в математиці. До числень належать формально-аксіоматичні системи того чи іншого розділу математики.

Під *численням* розуміють скінченну множину точно визначених породжувальних правил, які дають змогу із певних заданих об'єктів отримувати інші об'єкти.

Породжувальні правила називають також *правилами виведення*.

Об'єкти, до яких застосовуються правила виведення, називають *засновками*. Отриманий із засновків об'єкт називають *висновком*.

Множину породжених численням об'єктів задають індуктивно. На першому кроці процесу породження (виведення) початкові об'єкти задаються породжувальними правилами із порожньою множиною засновків. Об'єкт вважається породженим на певному кроці, якщо його отримують за допомогою певного породжувального правила із об'єктів, породжених на попередніх кроках.

Якщо на першому кроці процесу породження брати початкові об'єкти із певної множини A, то дістанемо числення з входом. Таке числення $\mathfrak{I}$ перетворює множину A у множину об'єктів B, породжених із об'єктів множини A за допомогою числення $\mathfrak{I}$.

На відміну від наказових правил алгоритму, які *однозначно* визначають перехід від одних об'єктів до інших, правила числення *дають змогу* переходити від одних об'єктів до інших.

Зв'язок понять алгоритму та числення полягає в такому.

1. Поняття числення можна звести до поняття алгоритму в сенсі зведення розгалуженого процесу породження до послідовного процесу переліку так, щоб алгоритм, який задає цей перелік, відтворив усі породжені численням об'єкти і тільки їх. Такий алгоритм послідовно додає до множини вже породжених об'єктів (спочатку порожньої) скінченну множину нових об'єктів, отриманих однократним застосуванням ПП до вже породжених об'єктів.

2. Поняття алгоритму можна звести до поняття числення в тому розумінні, що алгоритмічний процес можна розглядати як процес породження. Справді, кожний алгоритм можна трактувати як числення з входом, яке має такі породжувальні правила, що виконання кожного із них відповідає виконанню одного кроку алгоритму. Усі такі породжувальні правила мають один засновок, до кожного об'єкта застосовне не більш як одне породжувальне правило.

Поштовхом до виникнення теорії алгоритмів як окремого розділу математики стала невдача в знаходженні алгоритмів розв'язання низки масових проблем. Найвідомішими з них були проблема істинності для арифметичних формул, проблема істинності для формул числення предикатів 1-го порядку та 10-та проблема Гільберта про розв'язність діофантових рівнянь. Усе це підштовхнуло до думки про те, що деякі масові проблеми мають настільки загальний характер, що алгоритми

їх розв'язання взагалі не існують. Але для того, щоб доведести неіснування алгоритму, треба мати його точне математичне визначення. Тому після сформування поняття алгоритму як нової та окремої сутності першочерговою стала проблема знаходження адекватних формальних моделей алгоритму.

Варто зазначити, що відкриття поняття алгоритму як самостійного, окремого поняття не можна змішувати з відкриттям конкретних формальних моделей алгоритму чи алгоритмічно обчислюваної функції. Такі моделі запропоновані саме для адекватного формального уточнення інтуїтивного поняття алгоритму, яке є для них первісним.

Пошук формальних уточнень поняття алгоритму проводився в таких напрямах.

1. Опис точного математичного поняття алгоритмічної машини та обчислюваності на ній. Першою формальною моделлю алгоритмічної машини була машина Тьюрінга, яка моделює елементарні дії при реалізації алгоритму людиною (А. Тьюрінг, Е. Пост, 1936). Зараз відомо багато різновидів машин Тьюрінга, як детермінованих, так і недетермінованих. Із пізніших формальних моделей алгоритмів відзначимо також нормальні алгоритми (А. Марков, 1952) і регістрові машини (Д. Шепердсон, Г. Стерджіс, 1963). До регістрових машин можна віднести машини з натуральнозначними регістрами, що розглядаються в даному посібнику.

2. Опис певних класів функцій, для яких існує алгоритм знаходження функції за значеннями її аргументів. При такому підході уточнюється не первісне поняття алгоритму, а похідне поняття алгоритмічно обчислюваної функції. Спочатку такі описи були запропоновані для функцій, заданих на множині натуральних чисел, пізніше — для функцій, заданих на множинах інших об'єктів.

Першими формальними моделями алгоритмічно обчислюваних функцій були λ-означувані функції (А. Чорч, 1932) і загальнорекурсивні функції (К. Гьодель, 1934). Зазначені моделі визначались як функції, графіки яких породжуються відповідно численням λ-конверсій і численням Ербрана-Гьоделя. У 1936 р. С. Кліні поширив поняття загальнорекурсивної функції на випадок часткових функцій, увівши поняття частково рекурсивної функції. Він також описав клас частково рекурсивних функцій у чисто функціональних термінах. У 1943 р. Е. Пост запропонував модель обчислюваних функцій на основі введеного ним числення спеціального вигляду (канонічних систем).

Синонімом поняття алгоритму є поняття *програми.* В основі уточнення поняття програмування як процесу конструювання програм лежить запропоноване В. Редьком поняття програмної логіки (алгебри). Функції, задані за допомогою програмної логіки, називають *програмно заданими*, або *програмованими*.

Поняття програмованої функції природно вважати уточненням поняття відносно обчислюваної функції.

Поняття числення можна уточнити за допомогою поняття формальної системи. До формальних систем, крім класичних аксіоматичних теорій, належать канонічні та нормальні системи Поста, формальні граматики (Н. Хомський, 1952) тощо. Зауважимо, що, враховуючи зв'язок алгоритмів і числень, формальні моделі алгоритмів і алгоритмічно обчислюваних функцій можуть бути задані у вигляді формальних систем.

У 1936 р. А. Чорч і С. Кліні довели, що класи загальнорекурсивних і λ-означуваних функцій збігаються. На основі цього факту та аналізу ідей, які привели до зазначених понять, А. Чорч висунув відому *тезу* про збіг класу АОФ з класом загальнорекурсивних функцій. С. Кліні узагальнив цю тезу для часткових функцій. Доведений А. Тьюрінгом 1937 р. збіг класів частково рекурсивних функцій і функцій, обчислюваних на машинах Тьюрінга, став ще одним підтвердженням тези Чорча. Пізніше такі збіги були встановлені для всіх відомих формальних моделей АОФ. Тому маємо всі підстави вважати, що кожна із названих вище формальних моделей адекватно уточнює інтуїтивне поняття АОФ.

Сфера використання теорії алгоритмів дуже широка. Перші та найчисельніші застосування теорія алгоритмів має в математичній логіці, адже вона виникла саме як розділ математичної логіки. Виникнення швидкодіючих обчислювальних машин і бурхливий розвиток інформаційних технологій відкрили нові сфери застосування теорії алгоритмів. Поняття алгоритму є концептуальною основою процесів обробки інформації, теорія алгоритмів є теоретичним фундаментом програмування та інформатики. Методи й поняття теорії алгоритмів успішно використовуються в багатьох розділах науки, зокрема в основах математики, теорії інформації, теорії керування, обчислювальній математиці, конструктивному аналізі, теорії ймовірності, економіці, лінгвістиці.

1. ОСНОВНІ ПОНЯТТЯ ТА ВИЗНАЧЕННЯ

Вважаємо відомими базові математичні поняття, такі як поняття множини, відношення, відображення, функції, декартового добутку.

Уведемо потрібні для подальшого викладу поняття та позначення.

Множини натуральних, цілих, раціональних і дійсних чисел позначатимемо відповідно N, Z, Q і R.

Множину всіх підмножин довільної множини A позначатимемо 2^A.

Словом в алфавіті T називатимемо довільну скінченну послідовність символів із T.

Довжина слова — це кількість його символів.

Довжину слова α позначатимемо $|\alpha|$.

Слово, яке не містить жодного символу, називатимемо *порожнім*. Таке слово позначатимемо ε.

Для порожнього слова його довжина $|\varepsilon| = 0$.

Якщо a — символ, то позначаємо $a^0 = \varepsilon$, $a^{n+1} = a \cdot a^n$ для $n \geq 0$.

Множину всіх слів у алфавіті T позначатимемо T^*.

Мовою (вербальною множиною) в алфавіті T називатимемо довільну підмножину $L \subseteq T^*$.

X-Y-алгоритм *вербальний,* якщо X та Y — вербальні множини.

Функція (відображення) f: $X \to Y$ *вербальна,* якщо X і Y — вербальні множини.

Нехай A та R — довільні множини, f — довільна однозначна часткова функція вигляду f: $A \to R$.

Той факт, що значення $f(a)$ визначене, позначатимемо $f(a)\downarrow$.

Якщо $f(a)$ невизначене, то записуємо $f(a)\uparrow$.

Той факт, що значення $f(a)$ визначене та дорівнює b, позначатимемо $f(a)\downarrow = b$, або $f(a)\downarrow b$.

Для довільних часткових функцій f і g уведемо позначення $f(a) \cong g(b)$, якщо з умови $f(a)\downarrow$ та $g(b)\downarrow$ випливає $f(a) = g(b)$.

Функція f: $A \to R$ *тотальна,* або *всюди визначена,* якщо $f(a)\downarrow$ для всіх $a \in A$.

З кожною f: $A \to R$ пов'яжемо множини D_f і E_f, де $D_f \subseteq A$ і $E_f \subseteq R$.

Вважаємо, що для кожного $a \in D_f$ виконується умова $f(a)\downarrow$, для кожного $a \in A \setminus D_f$ — умова $f(a)\uparrow$, а умова $f(a)\downarrow$ рівносильна $f(a) \in E_f$.

Множини D_f та E_f називатимемо відповідно *множиною визначеності* та *множиною значень* (*результатів*) функції f.

Графіком функції f є множина $\Gamma_f = \{(a, f(a)) \mid a \in D_f\}$.

Функція вигляду $A^n \to A$ називається *n-арною функцією на A*.

Функція f: $N \to N$ *строго монотонна*, якщо для всіх $x, y \in D_f$ з умови $x < y$ випливає $f(x) < f(y)$.

Нехай $\boldsymbol{F}$ — множина функцій на деякій множині D. Довільну функцію вигляду $\boldsymbol{F}^n \to \boldsymbol{F}$ називатимемо *n-арною композицією*, або *n-арною операцією*.

Нехай $\boldsymbol{U}$ — деяка множина, яку трактуємо як універсум.

Характеристичною функцією довільної множини $A \subseteq \boldsymbol{U}$ називатимемо функцію χ_A: $\boldsymbol{U} \to \{0,1\}$ таку:

$$\chi_A(a) = \begin{cases} 1, & \text{якщо } a \in A, \\ 0, & \text{якщо } a \notin A. \end{cases}$$

Частковою характеристичною функцією довільної множини $A \subseteq \boldsymbol{U}$ називатимемо функцію $\chi_A^{\text{ч}}$: $\boldsymbol{U} \to \{1\}$ таку:

$$\chi_A^{\text{ч}}(a) = \begin{cases} 1, & \text{якщо } a \in A, \\ \text{невизначене}, & \text{якщо } a \notin A. \end{cases}$$

Записи вигляду χ_A і $\chi_A^{\text{ч}}$ завжди позначатимуть характеристичну та часткову характеристичну функцію множини А.

Під *формальною системою* (ФС) розуміють трійку (L, A, P).

Тут L — мова формальної системи, A — множина аксіом, P — множина правил виведення.

Мова задається алфавітом і правилами побудови слів мови, які звичайно називаються формулами.

Кожна аксіома є формулою.

Правила виведення ФС діють на множині формул.

Формула, отримувана з аксіом за допомогою правил виведення, називається *теоремою*.

Правила виведення ФС записуємо у вигляді $P_1, P_2, ..., P_n \vdash P$, де $P_1, P_2, ..., P_n$ — засновки, P — висновок.

Під *предикатом* на множині A розумітимемо довільне часткове відображення $P: A \to Bool$ множини A у множину $Bool = \{T, F\}$, елементи T і F якої інтерпретуються як "істина" і "фальш".

Предикат вигляду $A^n \to Bool$ називатимемо *n-арним предикатом на A*.

Для довільних предикатів P і Q уведемо такі позначення.

$P(a) \Rightarrow Q(b)$, якщо з $P(a)\downarrow$ і $Q(b)\downarrow$ та $P(a) = T$ випливає $Q(b) = T$;

$P \Rightarrow Q$, якщо $P(d) \Rightarrow Q(d)$ для довільних $d \in A$;

$P \cong Q$, якщо $P(a) \cong Q(a)$ для довільних $d \in A$.

Предикат $P: A \to Bool$ *істинний,* якщо для довільних $a \in A$ із умови $P(a)\downarrow$ випливає $P(a) = T$.

Характеристичною функцією довільного тотального предиката P на множині A називають функцію $\chi_P: A \to \{0,1\}$ таку:

$$\chi_P(a) = \begin{cases} 1, & \text{якщо } P(a) = T, \\ 0, & \text{якщо } P(a) = F. \end{cases}$$

Частковою характеристичною функцією довільного тотального предикату P на множині A називають функцію $\chi^{\text{ч}}_P: A \to \{1\}$ таку:

$$\chi^{\text{ч}}_P(a) = \begin{cases} 1, \text{ якщо } P(a) = T, \\ \text{невизначене, якщо } P(a) = F. \end{cases}$$

Областю істинності довільного тотального предикату P на множині A називають множину $\boldsymbol{I_P} = \{d \in A \mid P(d) = T\}$.

Записами вигляду χ_P, $\chi^{\text{ч}}_P$, $\boldsymbol{I_P}$ завжди позначатимемо характеристичну функцію, часткову характеристичну функцію, область істинності предиката P.

Засобами утворення складніших предикатів із простіших є пропозиційні композиції (логічні зв'язки) і композиції (операції) квантифікації $\exists \boldsymbol{x}$ та $\forall \boldsymbol{x}$.

Основними логічними зв'язками є заперечення $\neg$, диз'юнкція $\vee$, кон'юнкція **&**, імплікація $\to$, еквіваленція $\leftrightarrow$, роздільна диз'юнкція $\oplus$.

Визначення основних логічних зв'язків та операцій квантифікації $\exists \boldsymbol{x}$, $\forall \boldsymbol{x}$ наведено в [10; 11; 14; 25].

2. ФОРМАЛЬНІ МОДЕЛІ АЛГОРИТМІВ ТА АЛГОРИТМІЧНО ОБЧИСЛЮВАНИХ ФУНКЦІЙ

У цьому розділі розглянемо найпоширеніші формальні моделі алгоритмічних машин та алгоритмічно обчислюваних функцій.

2.1. Машини з натуральнозначними регістрами

Машина з натуральнозначними регістрами (МНР) є ідеалізованою моделлю комп'ютера.

МНР містить нескінченну кількість регістрів, вмістом яких є натуральні числа.

Регістри нумеруємо (іменуємо) натуральними числами, починаючи з 0, позначаючи їх R_0, R_1, ..., R_n, ...

Вміст регістру R_n позначимо $'R_n$.

Послідовність ($'R_0$, $'R_1$, ..., $'R_n$, ...) вмістів регістрів МНР назвемо *конфігурацією* МНР.

МНР може змінити вміст регістрів згідно з виконуваною нею командою. Скінченний список команд утворює *програму МНР*, або *МНР-програму*.

Команди програми послідовно нумеруємо (іменуємо) натуральними числами, починаючи з 1. Номер команди в програмі називаємо також *адресою* команди.

МНР-програму з командами I_1, I_2, ..., I_k позначатимемо $I_1I_2...I_k$.

Довжину (кількість команд) МНР-програми P позначатимемо $|P|$.

Команди МНР бувають чотирьох типів.

Тип 1. Обнулення n-го регістру $Z(n)$: $'R_n := 0$.

Тип 2. Збільшення вмісту n-го регістру на 1 $S(n)$: $'R_n := 'R_n + 1$.

Тип 3. Копіювання вмісту регістру $T(m, n)$: $'R_n := 'R_m$
(при цьому $'R_m$ не змінюється).

Команди типів 1–3 називають арифметичними.

Після виконання арифметичної команди МНР має виконувати наступну за списком команду програми.

Тип 4. Умовний перехід $J(m, n, q)$: якщо $'R_n = 'R_m$, то перейти до виконання q-ї команди, інакше виконувати наступну за списком команду програми.

Число q у команді $J(m, n, q)$ назвемо *адресою переходу*.

Виконання однієї команди МНР назвемо *кроком* МНР.

Зрозуміло, що саме МНР-програми є формальними моделями алгоритмів, а поняття МНР використовується для опису функціонування МНР-програм.

Виконання програми МНР починає, перебуваючи в деякій початковій конфігурації, з виконання першої за списком команди. Наступна для виконання команда програми визначається так, як описано вище. Виконання програми завершується (програма зупиняється), якщо наступна для виконання команда відсутня (тобто номер наступної команди перевищує номер останньої команди програми).

Конфігурація МНР у момент завершення виконання програми називається фінальною, вона визначає результат роботи МНР-програми над даною початковою конфігурацією.

Якщо МНР-програма P при роботі над початковою конфігурацією $(a_0, a_1, ...)$ ніколи не зупиняється, це позначаємо $P(a_0, a_1, \ldots)\uparrow$; якщо ж вона коли-небудь зупиниться, це позначимо $P(a_0, a_1, \ldots)\downarrow$.

Якщо МНР-програма P при роботі над початковою конфігурацією $(a_0, a_1, ...)$ зупиняється із фінальною конфігурацією $(b_0, b_1, ...)$, це позначаємо $P(a_0, a_1, ...)\downarrow(b_0, b_1, ...)$.

Кожна МНР-програма визначає відображення вигляду $N^N \to N^N$, де N^N — множина всіх нескінченних послідовностей натуральних чисел.

Зрозуміло, що таке відображення однозначне.

МНР-програми як моделі алгоритмів є фінітними об'єктами. Кожна МНР-програма у процесі виконання використовує лише скінченну множину регістрів, усі вони явно зазначені в МНР-програмі. Тому при розгляді відображень, які задаються МНР-програмами, природно обмежитися скінченними послідовностями натуральних чисел.

Отже, надалі розглядаємо тільки скінченні конфігурації.

Конфігурацію вигляду $(a_0, a_1, ..., a_n, 0, 0, ...)$, в якій $'R_m = 0$ для всіх $m > n$, назвемо *скінченною*.

Таку конфігурацію позначаємо у вигляді $(a_0, a_1, ..., a_n)$.

Якщо МНР-програма P починає роботу над скінченною початковою конфігурацією, то в процесі виконання P МНР перебуватиме тільки в скінченних конфігураціях.

МНР-програми P і Q *еквівалентні*, якщо вони визначають однакові відображення послідовностей натуральних чисел. Це означає, що при роботі над однаковими початковими конфігураціями вони або обидві зупиняються з однаковими фінальними конфігураціями, або обидві не зупиняються.

МНР-програму P назвемо *стандартною*, якщо в P для кожної команди вигляду $J(m, n, q)$ виконується умова $q \le |P| + 1$.

Конкатенацією стандартних МНР-програм $P = I_1 I_2 ... I_k$ та $Q = I_1 I_2 ... I_m$ назвемо стандартну МНР-програму $I_1 ... I_k I_{k+1} ... I_{k+m}$, де команди $I_{k+1}, ..., I_{k+m}$ — це по суті команди програми Q, в яких кожна команда вигляду $J(m, n, q)$ замінена командою $J(m, n, q + k)$.

МНР-програма P *обчислює* часткову n-арну функцію $f: N^n \to N$, якщо

$$f(a_1, a_2, ..., a_n) = b \Leftrightarrow P(a_1, a_2, ..., a_n)\downarrow(b, ...).$$

Замість $P(a_1, a_2, ...)\downarrow(b, ...)$ надалі писатимемо $P(a_1, a_2, ...)\downarrow b$.

Обчислюваність функції МНР-програмою означає, що значення аргументів функції послідовно розміщуються в початкових регістрах, починаючи з R_0, значення функції знімається з регістру R_0.

Неважко переконатись, що наведене визначення обчислюваності функції $f: N^n \to N$ МНР-програмою P еквівалентне такому:

при умові $(a_1, a_2, ..., a_n) \in D_f$ та $f(a_1, a_2, ..., a_n) = b$ маємо $P(a_1, a_2, ..., a_n)\downarrow b$;

при умові $(a_1, a_2, ..., a_n) \notin D_f$ маємо $P(a_1, a_2, ..., a_n)\uparrow$.

Функцію $f: N^n \to N$ називають *МНР-обчислюваною,* якщо існує МНР-програма, яка обчислює цю функцію.

Кожна МНР-програма обчислює безліч функцій натуральних аргументів і значень, але, зафіксовуючи наперед арність функцій (тобто кількість компонент початкових конфігурацій), отримуємо, що кожна МНР-програма обчислює єдину функцію заданої арності.

Розглянемо приклади МНР-програм.

Приклад 2.1.1. МНР-програма для функції $x + y$:

1) $J(1,2,5)$;
2) $S(0)$;
3) $S(2)$;
4) $J(0,0,1)$.

Приклад 2.1.2. МНР-програма для функції $x - y$:

1) $J(0,1,5)$;
2) $S(1)$;
3) $S(2)$;
4) $J(0,0,1)$;
5) $T(2,0)$.

Приклад 2.1.3. МНР-програма для всюди невизначеної функції $f_{\varnothing}$:

1) $J(0,0,1)$.

Приклад 2.1.4. МНР-програма для функції $[x/2]$:

1) $J(0,2,7)$;
2) $S(2)$;
3) $J(0,2,7)$;
4) $S(2)$;
5) $S(1)$;
6) $J(0,0,1)$;
7) $T(1,0)$.

Приклад 2.1.5. МНР-програма для функції

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

1) $J(0,1,7)$;
2) $J(0,2,6)$;

3) $S(1)$;

4) $S(2)$;

5) $J(0,0,1)$;

6) $Z(2)$;

7) $T(2,0)$.

Приклад 2.1.6. МНР-програма для функції $f(x, y) = x \cdot y$:

1) $J(3,1,9)$;

2) $J(0,2,6)$;

3) $S(2)$;

4) $S(4)$;

5) $J(0,0,2)$;

6) $Z(2)$;

7) $S(3)$;

8) $J(0,0,1)$;

9) $T(4,0)$.

Нехай P — стандартна МНР-програма для n-арної функції f.

Найбільший номер регістру, задіяного при обчисленні функції f, позначимо $\rho(P)$. При цьому для n-арної функції завжди вважаємо $\rho(P) \geq n - 1$.

Через $P[k_1, k_2, ..., k_n \to R]$ позначимо МНР-програму, яка обчислює ту саму функцію f, що й МНР-програма P, але якщо вважати, що початкові значення аргументів занесено в регістри $k_1, ..., k_n$, a значення функції знімається з регістру R. При цьому для обчислення f задіяні щонайбільше $\rho(P) + 1$ регістрів — з 0-го по $\rho(P)$-й включно.

МНР-програма $P[k_1, k_2, ..., k_n \to R]$ для $k_1 < k_2 < ... < k_n$ має такий вигляд (тут P' відрізняється від P тільки зміщенням адрес команд і адрес переходу на $\rho(P)$):

$T(k_i, i - 1)$, $1 < i \leq n$;

$Z(k)$, $n \leq k \leq \rho(P)$;

P';

$T(0, R)$.

2.2. Машини Тьюрінга

Англійський математик А. М. Тьюрінг в 1936 р. запропонував клас абстрактних обчислювальних машин, які задають алгоритми, що можуть бути виконані людиною з олівцем і достатнім запасом паперу.

Відомо дуже багато різних варіантів машин Тьюрінга та їх узагальнень (багатострічкові машини, машини з еластичною стрічкою тощо).

Розглянемо таке визначення машини Тьюрінга.

Під *машиною Тьюрінга* (МТ) розуміємо впорядковану п'ятірку (Q, T, δ, q_0, F), де:

- Q — скінченна множина внутрішніх станів;
- T — скінченний алфавіт символів стрічки, причому T містить спеціальний символ порожньої клітки λ;
- δ: $Q \times T \to Q \times T \times \{R, L, \varepsilon\}$ — функція переходів;
- $q_0 \in Q$ — початковий стан;
- $F \subseteq Q$ — множина фінальних станів.

Функцію переходів звичайно задають скінченною множиною команд одного з трьох видів:

$qa \to pbR$,

$qa \to pbL$,

$qa \to pb$,

де $p, q \in Q$, $a, b \in T$, $\to \notin Q \cup T$.

При цьому, як правило, не для всіх пар $(q, a) \in Q \times T$ існує команда з лівою частиною qa. Це означає, що функція δ не є тотальною. Але зручніше вважати δ тотальною, тому для всіх пар $(q, a) \notin D_\delta$ неявно, не додаючи явно відповідні команди вигляду $qa \to qa$ до множини команд, уводимо довизначення $\delta(q, a) = (q, a, \varepsilon)$.

Неформально МТ складається зі скінченної пам'яті, розділеної на клітки нескінченної з обох боків стрічки та головки зчитування-записування.

У кожній клітці стрічки міститься єдиний символ із T, причому в кожен момент стрічка містить скінченну кількість символів, відмінних від символу λ. Головка зчитування-записування в кожен момент оглядає єдину клітку стрічки.

Якщо МТ перебуває у стані q і головка зчитує символ a, то при виконанні команди $qa \to pbR$ (команди $qa \to pbL$, команди $qa \to pb$) МТ переходить у стан p, замість символу a записує на стрічці символ b

і зміщує головку на одну клітку направо (відповідно на одну клітку наліво, залишає головку на місці).

Конфігурація, або *повний стан МТ,* — це слово вигляду xqy, де $x, y \in T^*$, $q \in Q$.

Неформально це означає, що на стрічці записано слово xy, тобто зліва і справа від xy можуть стояти лише символи λ, МТ знаходиться в стані q, головка читає перший символ підслова y.

Конфігурацію вигляду q_0x, де перший та останній символи слова x відмінні від λ, називають *початковою*.

Конфігурацію вигляду xqy, де $q \in F$, називають *фінальною*.

Після переходу до фінального стану, а отже, до фінальної конфігурації, МТ зупиняється.

Нехай МТ перебуває в конфігурації $xcqay$, де $x, y \in T^*$, $a, c \in T$, $q \in Q$.

Після виконання команди $qa \to pbR$ МТ перейде до конфігурації $xcbpy$. Після виконання команди $qa \to pbL$ МТ перейде до конфігурації $xpcby$, після виконання команди $qa \to pb$ МТ перейде до конфігурації $xcpby$.

Кожна МТ задає деяке вербальне відображення (відображення множини слів у множину слів) $T^* \to T^*$ у такий спосіб.

Машина Тьюрінга M переводить слово $u \in T^*$ у слово $v \in T^*$, якщо вона з початкової конфігурації q_0u переходить до фінальної конфігурації xqy, де $q \in F$, $xy = \alpha v \beta$, $\alpha, \beta \in \{\lambda\}^*$. При цьому перший та останній символи слова v відмінні від λ, або $v = \varepsilon$.

Те, що МТ M переводить слово u у слово v, записуємо так: $v = M(u)$.

Якщо МТ M, починаючи роботу з початкової конфігурації q_0u, ніколи не зупиниться, кажуть, що M *зациклюється* при роботі над словом u. Тоді $M(u)$ не визначене.

МТ M_1 і M_2 *еквівалентні*, якщо вони задають те саме вербальне відображення.

Машина Тьюрінга називається *детермінованою*, якщо функція δ однозначна, інакше МТ називається *недетермінованою*.

Не обмежуючи загальності, можна вважати, що F складається з єдиного фінального стану q^*.

Справді, нехай $M = (Q, T, \delta, q_0, F)$. Візьмемо $q^* \notin Q$.

Тоді МТ $M' = (Q \cup \{q^*\}, T, \delta', q_0, \{q^*\})$, де $\delta' = \delta \cup \{qa \to q^*a \mid q \in F, a \in T\}$, еквівалентна початковій машині M.

Надалі розглядатимемо тільки детерміновані МТ з єдиним фінальним станом і позначатимемо їх у вигляді (Q, T, δ, q_0, q^*).

Конкретні МТ задаємо, вказуючи множину команд.

При цьому початковий стан позначаємо q_0, фінальний стан — q^*.

Машина Тьюрінга *M обчислює* часткову функцію f: $N^n \to N$, якщо вона кожне слово вигляду $|^{x_1}\#|^{x_2}\#\ldots\#|^{x_k}$ переводить у слово $|^{f(x_1,\ldots,x_k)}$ у випадку $(x_1, \ldots, x_k) \in D_f$, і $M(|^{x_1}\#|^{x_2}\#\ldots\#|^{x_k})$ невизначене при $(x_1, \ldots, x_k) \notin D_f$.

Функція називається *обчислюваною за Тьюрінгом*, або *МТ-обчислюваною*, якщо існує МТ, яка її обчислює.

Зауважимо, що кожна МТ обчислює безліч функцій натуральних аргументів і значень, але зафіксовуючи наперед арність функцій, дістаємо, що кожна МТ обчислює єдину функцію заданої арності.

Розглянемо приклади МТ.

Приклад 2.2.1. МТ, яка обчислює функцію $x + y$:

$q_0| \to q_1 \lambda R$;
$q_1| \to q_1 |R$;
$q_1\# \to q^*|$;
$q_0\# \to q^*\lambda$.

Приклад 2.2.2. МТ, яка обчислює функцію $x - y$:

$q_0| \to q_1 \lambda R$;
$q_1| \to q_1 |R$;
$q_1\# \to q_1 \# R$;
$q_1\lambda \to q_2 \lambda L$;
$q_2| \to q_3 \lambda L$;
$q_3| \to q_3 |L$;
$q_3\# \to q_3 \# L$;
$q_3\lambda \to q_0 \lambda R$;
$q_2\# \to q^*|$;
$q_0\# \to q_4 \lambda R$;
$q_4\lambda \to q^*\lambda$.

Приклад 2.2.3. МТ, яка обчислює функцію $f(x) = sg(x)$:

$q_0\lambda \to q^*\lambda$;
$q_0| \to q_1|R$;
$q_1| \to q_1\lambda R$;
$q_1\lambda \to q^*\lambda$.

Приклад 2.3.4. МТ, яка обчислює предикат "x парне":

$q_0| \to q_1\lambda R$;
$q_1| \to q_0\lambda R$;
$q_0\lambda \to q^*|$;
$q_1\lambda \to q^*\lambda$.

Приклад 2.2.5. МТ, яка обчислює функцію $f(x, y) = x \cdot y$:

$q_0\# \to q_1\lambda R$;
$q_1| \to q_1\lambda R$;
$q_1\lambda \to q^*\lambda$;
$q_1a \to q_1|R$;
$q_0| \to q_2\lambda R$;
$q_2| \to q_2|R$;
$q_2\# \to q_3\#R$;
$q_3| \to q_4\lambda R$;
$q_4| \to q_4|R$;
$q_4a \to q_4\,aR$;
$q_4\lambda \to q_5\,aL$;
$q_5| \to q_5|L$;
$q_5a \to q_5\,aL$;
$q_5\lambda \to q_3|R$;
$q_3a \to q_6\,aL$;
$q_6| \to q_6|L$;
$q_6\# \to q_6\#L$;
$q_6\lambda \to q_0\lambda R$;

$q_3\lambda \to q_7\lambda L$;
$q_7\# \to q_7\lambda L$;
$q_7| \to q_7\lambda L$;
$q_7\lambda \to q^*\lambda$.

Приклад 2.2.6. МТ, яка кожне слово $x \in T^*$ переводить у слово $x\#x$ (тут $\# \notin T$).

$q_0\, t \to q_0\, tR$ для всіх $t \in T$;
$q_0\lambda \to q_1\#L$;
$q_1\, t \to q_1\, tL$ для всіх $t \in T$;
$q_1\lambda \to q_2\lambda R$;
$q_2\, t \to q_t\, \lambda R$ для всіх $t \in T$;
$q_t\, p \to q_t\, pR$ для всіх $t \in T$, $p \in T \cup \{\#\}$;
$q_t\, \lambda \to q'_t\, tL$ для всіх $t \in T$;
$q'_t\, p \to q'_t\, pL$ для всіх $t \in T$, $p \in T \cup \{\#\}$;
$q'_t\, \lambda \to q_2 tR$ для всіх $t \in T$;
$q_2\# \to q^*\#$.

2.3. Нормальні алгоритми Маркова

Під *нормальним алгоритмом* (НА) *в алфавіті* T розуміють впорядковану послідовність продукцій (правил) вигляду $\alpha \to \beta$ або $\alpha \to \cdot\beta$, де $\alpha, \beta \in T^*$ та $\cdot \notin T$, $\to \notin T$.

Продукції вигляду $\alpha \to \cdot\beta$ називають *фінальними*.

Кожен НА в алфавіті T задає деяке вербальне відображення $T^* \to T^*$.

Слово, яке є результатом обробки слова x нормальним алгоритмом $\aleph$, позначимо $\aleph(x)$.

Обробка слова x нормальним алгоритмом $\aleph$ здійснюється поетапно таким чином.

Покладемо $x_0 = x$ і скажемо, що x_0 отримане із x після 0 етапів.

Нехай слово x_n отримане із слова x після n етапів. Тоді $(n + 1)$-й етап виконується так.

Шукаємо першу за порядком продукцію $\alpha \to \beta$ або $\alpha \to \cdot\beta$ таку, що α — підслово x_n. Застосуємо цю продукцію до x_n, тобто замінимо в x_n найлівіше входження α на β. Отримане слово позначимо x_{n+1}.

Якщо застосована на $(n+1)$-му етапі продукція нефінальна, тобто $\alpha\to\beta$, переходимо до $(n+2)$-го етапу.

Якщо ця продукція фінальна, тобто $\alpha\to\cdot\beta$, то після її застосування $\aleph$ зупиняється і $\aleph(x) = x_{n+1}$.

Якщо ж на $(n+1)$-му етапі жодна продукція $\aleph$ не застосовна до x_{n+1}, тобто в $\aleph$ немає продукції, ліва частина якої — підслово слова x_{n+1}, то $\aleph$ зупиняється і $\aleph(x) = x_n$.

Якщо в процесі обробки слова x НА $\aleph$ не зупиняється на жодному етапі, то вважаємо, що $\aleph(x)$ невизначене.

Нормальний алгоритм називають *нормальним алгоритмом над алфавітом T*, якщо він є нормальним алгоритмом у деякому розширенні $T'\supseteq T$.

Нормальний алгоритм над T задає певне відображення $T^*\to T^*$, використовуючи в процесі обробки слів допоміжні символи поза алфавітом T.

Зупинка НА $\aleph$ над T при роботі над словом $x\in T^*$ *результативна,* коли вона відбулась на слові $y\in T^*$, інакше вважаємо, що результат роботи $\aleph$ над x невизначений.

Нормальні алгоритми $\aleph$ і $\Re$ *еквівалентні відносно алфавіту T*, якщо для всіх $x\in T^*$ $\aleph(x)$ і $\Re(x)$ одночасно визначені або невизначені, причому в разі визначеності $\aleph(x) = \Re(x)$.

Відомо [10], що для кожного НА над алфавітом T існує еквівалентний йому відносно T НА в алфавіті $T\cup\{s\}$ з єдиним допоміжним символом $s\notin T$.

Відомо також [10], що вербальне відображення, яке кожне слово $x\in T^*$ переводить у слово xx, не може бути заданим жодним НА в алфавіті T.

Нормальний алгоритм $\aleph$ *обчислює* часткову функцію f: $N^n\to N$, якщо він кожне слово вигляду $|^{x_1}\#|^{x_2}\#\ldots\#|^{x_k}$ переводить у слово $|^{f(x_1,\ldots,x_k)}$ у випадку $(x_1, \ldots, x_k)\in D_f$, і $\aleph(|^{x_1}\#|^{x_2}\#\ldots\#|^{x_k})$, невизначене у випадку $(x_1, \ldots, x_k)\notin D_f$.

Функція є *обчислюваною за Марковим*, або *НА-обчислюваною*, якщо існує НА, який її обчислює.

Кожний НА обчислює безліч функцій натуральних аргументів та значень, але зафіксовуючи наперед арність функцій, дістаємо, що кожний НА обчислює єдину функцію заданої арності.

Розглянемо приклади НА.

Приклад 2.3.1. НА для функції $f(x, y) = x + y$:

$\#\to\varepsilon$.

Приклад 2.3.2. НА для функції $x - y$:

$|\#|\to\#$;
$\#|\to\#|$;
$\#\to\varepsilon$.

Приклад 2.3.3. НА для функції $x/2$:

$\#||\to|\#$;
$\#|\to\#|$;
$\#\to\cdot\varepsilon$;
$\varepsilon\to\#$.

Приклад 2.3.4. НА для функції $x\cdot y$:

$\#\to**$;
$|*\to*a$;
$*|\to b*$;
$*\to\varepsilon$;
$ab\to ba|$;
$|b\to b|$;
$a\to\varepsilon$;
$b\to\varepsilon$.

Приклад 2.3.5. НА для функції 2^x:

$*|\to|**$;
$|*\to*$;
$\#*\to|\#$;
$\#\to\cdot\varepsilon$;
$*\to\#*$;
$\varepsilon\to*$.

Приклад 2.3.6. НА, який кожне слово $x\in T^*$ переводить в його дзеркальне відображення — слово $x^R\in T^*$ (тут $\#\notin T$):

$\#ab\to b\#a$ для всіх $a, b\in T$;
$\#\#a\#\to a\#\#$ для всіх $a\in T$;
$\#\#\to\cdot\varepsilon$;
$\varepsilon\to\#$.

Приклад 2.3. 7. НА, який кожне $x \in T^*$ переводить у слово xx (тут $\# \notin T$):

$\#\#a \to a\#a\#\#$ для всіх $a \in T$;
$\#ab \to b\#a$ для всіх $a, b \in T$;
$\#a \to a$ для всіх $a \in T$;
$\#\# \to \cdot\varepsilon$;
$\varepsilon \to \#\#$.

2.4. Системи Поста. Комбінаторні системи

Канонічною системою Поста над алфавітом T називають формальну систему (T^*, A, P), в якої множина аксіом A є скінченною підмножиною множини T^*, а множина правил виведення P складається зі слів вигляду $\alpha_0 S_1 \alpha_1 ... \alpha_{m-1} S_m \alpha_m \to \beta_0 S_{j_1} \beta_1 ... \beta_{n-1} S_{j_n} \beta_n$.

Тут $\to \notin T$, усі α_k та β_i — фіксовані слова із T^*, усі символи $S_k \notin T$, причому всі $j_i \in \{1, ..., m\}$.

Символи S_k позначають довільні слова із T^*.

Системи Поста звичайно позначають у вигляді $\boldsymbol{P} = (T, A, P)$.

Множина правил P визначає на словах із T^* відношення безпосереднього виведення $\Rightarrow_P$ у такий спосіб: $\sigma \Rightarrow_P \tau$, якщо існує правило $\alpha_0 S_1 \alpha_1 ... \alpha_{m-1} S_m \alpha_m \to \beta_0 S_{j_1} \beta_1 ... \beta_{n-1} S_{j_n} \beta_n$ таке: для деяких $u_1, ..., u_m \in T^*$ $\sigma = \alpha_0 u_1 \alpha_1 ... \alpha_{m-1} u_m \alpha_m$ та $\tau = \beta_0 u_{j_1} \beta_1 ... \beta_{n-1} u_{j_n} \beta_n$.

Рефлексивно-транзитивне замикання відношення $\Rightarrow_P$ позначатимемо $\Rightarrow^*_P$. Таким чином, $\sigma \Rightarrow^*_P \tau$ означає, що слово τ отримане із слова σ за допомогою скінченної кількості застосувань правил із P.

Слово τ *породжується* системою Поста $\boldsymbol{P}$, якщо $\alpha \Rightarrow^*_P \tau$ для деякої $\alpha \in A$.

Це записуємо ${}_P|\!- \tau$

Назвемо таке слово τ *теоремою* системи Поста $\boldsymbol{P}$.

Множину $Th(\boldsymbol{P}) = \{\tau \in T^* |\, {}_P|\!- \tau\}$ назвемо *множиною теорем* системи Поста $\boldsymbol{P}$.

Зрозуміло, що для завдання системи Поста достатньо вказати множину правил та множину аксіом. У випадку необхідності вказуємо і алфавіт T.

Приклад 2.4.1. Система Поста із $A = \{a, b, \varepsilon\}$ та $P = \{S \to aSa, S \to bSb\}$ породжує всі слова-паліндроми в алфавіті $\{a, b\}$, тобто слова, які читаються однаково зліва направо і справа наліво.

Множина $X \in T^*$ *породжувана за Постом*, якщо існують алфавіт $T' \supseteq T$ та система Поста $\boldsymbol{P} = (T', A, P)$ такі, що $Th(\boldsymbol{P}) \cap (T^*) = X$.

Формальні системи вигляду (T^*, A, P), в яких множина аксіом A є скінченною підмножиною множини T^*, а правила виведення мають вигляд $\alpha \to \beta$, називають комбінаторними. В цьому плані системи Поста є комбінаторними системами досить загального вигляду.

Правило вигляду $gS \to Sh$ називається *правилом в нормальній формі*.

Система Поста, всі правила якої — в нормальній формі, називається *нормальною системою Поста*.

Відомо, що кожна породжувана за Постом множина може бути породжена нормальною системою Поста.

Система Туе — це система Поста, усі правила якої мають вигляд $S_1 \alpha S_2 \to S_1 \beta S_2$, причому для кожного правила $S_1 \alpha S_2 \to S_1 \beta S_2$ існує симетричне до нього правило $S_1 \beta S_2 \to S_1 S \alpha_2$.

Якщо симетричні правила системи Туе об'єднати в одне, отримаємо комбінаторні системи, які називають екваційними численнями. Для екваційних числень правила записують у вигляді $\alpha \equiv \beta$.

Окремим випадком систем Поста можна вважати породжувальні, або формальні, граматики [18, 20, 21].

При визначенні формальних граматик явно вказують основний (термінальний) алфавіт T, допоміжний (нетермінальний) алфавіт T_N, єдину аксіому $s \in T_N$ і правила виведення, які записують у вигляді $\alpha \to \beta$, де $\alpha, \beta \in (T \cup T_N)^*$.

Формальні граматики записують у вигляді $G = (T, T_N, s, P)$.

Отже, формальна граматика є системою Поста в алфавіті $T \cup T_N$, усі правила якої мають вигляд $S_1 \alpha S_2 \to S_1 \beta S_2$, а множина аксіом складається з єдиного слова s.

Мова $L(G)$, породжувана формальною граматикою G, визначається як множина усіх теорем такої системи Поста, які є словами термінального алфавіту.

Наведемо класифікацію формальних граматик за Хомським.

Формальна граматика, в якій немає обмежень на правила виведення, називається граматикою типу 0.

Формальна граматика, в якій для правил виведення виконується умова $|\alpha| \le |\beta|$, називається нескоротною, або граматикою типу 1.

Правила виведення для граматик типу 1 можна звести до вигляду $\varphi A \psi \to \varphi \gamma \psi$, де $A \in T_N$, $\gamma \not\equiv \varepsilon$. Такі граматики називають також контекст-

ними, контексто-чутливими, граматиками безпосередньо складових (БС-граматиками).

Формальна граматика, в якій правила виведення мають вигляд $A \to \gamma$, де $A \in T_N$, називається граматикою типу 2. Такі граматики називають безконтекстними (БК-граматиками), а також контексто-вільними (КВ-граматиками).

Формальна граматика, в якій правила виведення мають вигляд $A \to xB$ або $A \to x$, де $A \in T_N$ та $x \in T$, називається праволінійною.

Формальна граматика, в якій правила виведення мають вигляд $A \to Bx$ або $A \to x$, де $A \in T_N$ та $x \in T$, називається ліволінійною.

Праволінійні та ліволінійні граматики утворюють клас граматик типу 3. Такі граматики називають також регулярними, або скінченно-автоматними.

Наведеним класам граматик відповідають певні класи алгоритмічних машин (автоматів), які задають такі самі класи мов.

Граматикам типу 0 відповідають недетерміновані МТ.

Граматикам типу 1 відповідають лінійно-обмежені автомати — недетерміновані МТ, які не збільшують довжину робочої зони стрічки.

Граматикам типу 2 відповідають автомати з магазинною пам'яттю, або магазинні автомати.

Граматикам типу 3 відповідають скінченні автомати.

Детальніше про це можна прочитати в [18, 20, 21].

Перейдемо тепер до поняття обчислюваності функцій за Постом.

Обчислюваність функції за Постом означає породжуваність за Постом її графіка.

Функція $f\colon N^n \to N$ *обчислювана за Постом,* якщо множина $\{|^{x_1} \#|^{x_2} \# \ldots \# |^{x_k} \# |^{f(x_1,\ldots,x_k)} \mid (x_1, \ldots, x_k) \in D_f\}$ породжувана за Постом.

Наведемо приклади функцій, обчислюваних за Постом.

Приклад 2.4.2. Система Поста для функції $x + y$:

$A = \{\#\#\}$;
$P = \{X\#Y\#R \to X|\#Y\#R|,$
$X\#Y\#R \to X\#Y|\#R|\ \}$.

Приклад 2.4.3. Система Поста для функції $f(x, y) = x - y$:

$A = \{\#\#\}$;

$P = \{X\#Y\#R \to X|\#Y\#R|,$
$X\#Y\#R| \to X\#Y|\#R\ \}.$

Приклад 2.4.4. Система Поста для функції $f(x, y) = x \cdot y$:

$A = \{\#\#\};$
$P = \{X\#Y\#R \to X|\#Y\#RY,$
$X\#Y\#R \to X\#Y|\#RX\ \}.$

Приклад 2.4.5. Система Поста для функції x^2:

$A = \{\#\};$
$P = \{X\#R \to X|\#RXX|\ \}.$

Приклад 2.4.6. Система Поста для функції $f(x) = x^2 - 2x$:

$A = \{\#, ||\#\};$
$P = \{X|\#R \to X||\#RXX|\ \}.$
У цьому прикладі треба врахувати, що $f(1)\uparrow$.

Приклад 2.4.7. Система Поста для функції $f(x) = x^3$:

$A = \{**\};$
$P = \{X*Q*R \to X|*QXX|*RQQQXXX|,$
$X*Q*R \to X\#R\ \}.$

Для обчислення значення $f(x + 1) = (x + 1)^3 = f(x) + 3x^2 + 3x + 1$ потрібне x^2, тому теоремами мусять також бути коди трійок (x, x^2, x^3). Але теоремами в алфавіті $\{|, \#\}$ можуть бути тільки коди елементів графіка $f(x)$, тому для кодування трійок (x, x^2, x^3) використано $*$.

Приклад 2.4.8. Система Поста для функції $f(x) = x!$:

$A = \{\#|\};$
$P = \{X\#F \to X|*F***,$
$S*F*A*B*M \to S*F*A|*B*MB,$
$S*F*A*B*M \to S*F*A*B|*MA,$
$S*F*S*F*M \to S\#M\ \}.$

До теорем належать коди п'ятірок ($x + 1$, $x!$, a, b, ab), для їх кодування використано $*$. Із кодів п'ятірок ($x + 1$, $x!$, $x + 1$, $x!$, $(x + 1)\cdot x!$) можна вивести закодовані в алфавіті $\{|, \#\}$ коди пар $(x + 1, (x + 1)!)$.

Приклад 2.4.9. Система Поста для функції $f(x) = [\sqrt{x}]$:

$$A = \{**\};$$

$$P = \{X**\to X|**,$$

$$QS|*Q*R\to QS|*QRR|*R|,$$

$$X*X*R\to X\#R,$$

$$X*XS|*R|\to X\#R\ \}.$$

До теорем належать коди трійок (x, r^2, r), для їх кодування використано $*$. Із кодів трійок (x, x, r) і $(x, (r + 1)^2, r + 1)$ при умові $r^2 \le x < (r + 1)^2$ можна вивести закодовані в алфавіті $\{|, \#\}$ коди пар (x, r).

2.5. Примітивно рекурсивні, частково рекурсивні та рекурсивні функції

Розглянемо спосіб завдання обчислюваних функцій, який грунтується на породженні таких функцій за допомогою певних обчислюваних операцій (композицій) із певних базових функцій.

Основними обчислюваними операціями для n-арних функцій, заданих на множині N, є такі операції:

- *операція суперпозиції* $\boldsymbol{S}^{n+1}$,
- *операція примітивної рекурсії* $\boldsymbol{R}$,
- *операція мінімізації* $\boldsymbol{M}$.

Операція суперпозиції $\boldsymbol{S}^{n+1}$ n-арній функції $g(x_1, ..., x_n)$ і n функціям $g_1(x_1, ..., x_m)$, ..., $g_n(x_1, ..., x_m)$ однієї і тієї ж арності ставить у відповідність функцію

$$f(x_1, ..., x_m) = g(g_1(x_1, ..., x_m), ..., g_n(x_1, ..., x_m)).$$

Таку функцію позначають $\boldsymbol{S}^{n+1}(g, g_1, ..., g_n)$, її арність збігається з арністю функцій $g_1, ..., g_n$.

Нехай $\mathbf{F}$ — клас усіх k-арних функцій на N.

Нехай $\mathbf{F}^n$ — клас таких функцій фіксованої арності n, тобто функцій із N^m в N.

Операцію $\boldsymbol{S}^{n+1}$ можна розглядати як тотальну функцію із $\mathbf{F}^n \times (\mathbf{F}^m)^n$ у $\mathbf{F}^m$, або як часткову $(n+1)$-арну функцію на $\mathbf{F}$.

Твердження 2.5.1. *Якщо функції $g, g_1, ..., g_n$ тотальні та алгоритмічно обчислювані, то функція $\boldsymbol{S}^{n+1}(g, g_1, ..., g_n)$ також тотальна та алгоритмічно обчислювана.*

Операція примітивної рекурсії $\boldsymbol{R}$ n-арній функції g і $(n+2)$-арній функції h ставить у відповідність $(n+1)$-арну функцію f, яку позначають $\boldsymbol{R}(g, h)$, що задається рекурсивним визначенням

$$f(x_1, ..., x_n, 0) = g(x_1, ..., x_n);$$

$$f(x_1, ..., x_n, y+1) = h(x_1, ..., x_n, y, f(x_1, ..., x_n, y)).$$

Це означає, що для всіх $a_1, ..., a_n, b$ значення $f(a_1, ..., a_n, b)$ обчислюється так:

$$f(a_1, ..., a_n, 0) = g(a_1, ..., a_n);$$

$$f(a_1, ..., a_n, 1) = h(a_1, ..., a_n, 0, f(a_1, ..., a_n, 0));$$

... ...

$$f(a_1, ..., a_n, b) = h(a_1, ..., a_n, b-1, f(a_1, ..., a_n, b-1)).$$

Наведені співвідношення показують, що функція f однозначно визначається за функціями g і h.

Якщо для деяких $a_1, ..., a_n, b$ значення $f(a_1, ..., a_n, b)$ невизначене, то значення $f(a_1, ..., a_n, t)$ невизначені для всіх $t \geq b$.

При $n = 0$ за визначенням вважаємо, що функція g — це 1-арна константа.

Із наведених співвідношень випливає твердження.

Твердження 2.5.2. *Якщо функції g і h тотальні та алгоритмічно обчислювані, то функція $\boldsymbol{R}(g, h)$ тотальна та алгоритмічно обчислювана.*

Операцію примітивної рекурсії $\boldsymbol{R}$ можна розглядати як тотальну функцію із $\mathbf{F}^n \times \mathbf{F}^{m+2}$ у $\mathbf{F}^{n+1}$ (при $n = 0$ — із $\mathbf{F}^1 \times \mathbf{F}^2$ у $\mathbf{F}^1$) або як часткову бінарну функцію на $\mathbf{F}$.

Операція мінімізації $\boldsymbol{M}$ $(n+1)$-арній функції g ставить у відповідність n-арну функцію f, яку позначають $\boldsymbol{M}(g)$, що задається співвідношенням

$f(x_1, ..., x_n) = \mu_y(g(x_1, ..., x_n, y) = 0)$.

Це означає, що для всіх значень $x_1, ..., x_n$ значення функції $f(x_1, ..., x_n)$ обчислюється так.

Послідовно обчислюємо значення $g(x_1, ..., x_n, y)$ для $y = 0, 1, 2, ...$.

Перше таке значення y, для якого $g(x_1, ..., x_n, y) = 0$, буде шуканим значенням $f(x_1, ..., x_n)$. При цьому для всіх $t < y$ значення $g(x_1, ..., x_n, y)$ мусять бути визначені та не дорівнювати нулю.

Операцію $\boldsymbol{M}$ можна розглядати як тотальну функцію із $\mathbf{F}^{n+1}$ у $\mathbf{F}^n$ (при $n = 0$ — як тотальну функцію із $\mathbf{F}^1$ в $\mathbf{F}^1$) або як часткову 1-арну функцію на $\mathbf{F}$.

Із визначення зрозуміло, що процес знаходження значення $\mu_y(g(x_1, ..., x_n, y) = 0)$ ніколи не закінчиться в таких випадках:

- значення $g(x_1, ..., x_n, 0)$ невизначене;
- для всіх значень y значення $g(x_1, ..., x_n, y)$ визначене та не дорівнює нулю;
- для всіх $t < y$ значення $g(x_1, ..., x_n, t)$ визначене та $\neq 0$, а значення $g(x_1, ..., x_n, y)$ невизначене.

Зауваження. Для довільного значення x існує єдине значення $y = x + 1$ таке, що $y - (x + 1) = 0$. Але функція $\mu_y(y - (x + 1) = 0)$ всюди невизначена, бо вже $0 - (x + 1)$ завжди невизначене. Отже, не завжди найменше значення y таке, що $g(x_1, ..., x_n, y) = 0$, збігається із $\mu_y(g(x_1, ..., x_n, y) = 0)$, яке може бути невизначеним, бо у випадку операції мінімізації таке перше значення y, для якого $g(x_1, ..., x_n, y) = 0$, знаходять за допомогою чітко описаного і не залежного від g алгоритму.

З алгоритмічності процесу одержання такого першого y маємо

Твердження 2.5.3. *Якщо функція g алгоритмічно обчислювана, то функція* $\boldsymbol{M}(g)$ *також алгоритмічно обчислювана*.

Функція g може бути тотальною, а функція $f = \boldsymbol{M}(g)$ — навіть всюди невизначеною.

Наприклад, такою є функція $f(x) = \mu_y(x + y + 1 = 0)$.

Базовими обчислюваними n-арними функціями називаються найпростіші функції $\mathbf{o}(x) = 0$, $\boldsymbol{s}(x) = x + 1$ та функції-селектори $\boldsymbol{I}_m^n(x_1, ..., x_n) = x_m$, де $n \geq m \geq 1$.

Зрозуміло, що базові обчислювані n-арні функції тотальні та алгоритмічно обчислювані.

Функцію, яку можна отримати із базових функцій за допомогою скінченної кількості застосувань операцій суперпозиції та примітивної рекурсії, називають *примітивно рекурсивною функцією* (ПРФ).

Функцію, яку можна отримати із базових функцій за допомогою скінченної кількості застосувань операцій суперпозиції, примітивної рекурсії та мінімізації, називають *частково рекурсивною* функцією (ЧРФ).

Тотальну ЧРФ називають *рекурсивною* функцією (РФ).

Із тверджень 2.5.1–2.5.3 випливає:

1) кожна ПРФ — тотальна алгоритмічно обчислювана функція;

2) кожна ЧРФ — алгоритмічно обчислювана функція;

3) кожна РФ — тотальна алгоритмічно обчислювана функція.

Із визначень ПРФ, ЧРФ і РФ маємо такі співвідношення між зазначеними класами функцій:

$$\text{ПРФ} \subseteq \text{РФ} \subseteq \text{ЧРФ}.$$

Алгебра $(\mathfrak{R}; \boldsymbol{R}, \boldsymbol{M}, \boldsymbol{S}^2, \boldsymbol{S}^3, ...)$, носієм $\mathfrak{R}$ якої є клас усіх ЧРФ, а операціями — операції $\boldsymbol{R}$, $\boldsymbol{M}$ та $\boldsymbol{S}^{n+1}$, де $n \geq 1$, називається *алгеброю ЧРФ*, або *алгеброю Чорча*.

Алгебра $(\mathfrak{R}_\mathbf{p}; \boldsymbol{R}, \boldsymbol{S}^2, \boldsymbol{S}^3, ...)$, носієм $\mathfrak{R}_\mathbf{p}$ якої є клас усіх ПРФ, а операціями — операції $\boldsymbol{R}$ і $\boldsymbol{S}^{n+1}$, де $n \geq 1$, називається *алгеброю ПРФ*.

Уведемо поняття *операторного терма* (ОТ) *алгебри ЧРФ* та *операторного терма алгебри ПРФ*.

Алфавіт мови алгебри ЧРФ складатиметься із символів базових функцій o, s та I_m^n, де $n \geq m \geq 1$, символів операцій R, M і S^{n+1}, де $n \geq 1$, а також допоміжних символів "(", ")" і ",".

Дамо індуктивне визначення ОТ алгебри ЧРФ.

1) кожен символ базової функції є ОТ;

такі ОТ називають *атомарними*;

2) якщо $t_0, t_1, ..., t_n$ — ОТ, то $S^{n+1}(t_0, t_1, ..., t_n)$ — ОТ;

3) якщо t_1 і t_2 — ОТ, то $R(t_1, t_2)$ — ОТ;

4) якщо t — ОТ, то $M(t)$ — ОТ.

Аналогічно можна дати індуктивне визначення операторного терма алгебри ПРФ (залишаються пункти 1)–3)).

Інтерпретуючи символи природним чином, маємо, що кожна ЧРФ є значенням деякого ОТ алгебри ЧРФ.

Проте не кожен ОТ алгебри ЧРФ має певне значення.

Наприклад, операторні терми $R(\text{o}, I_2^4)$ і $S^3(I_1^2, I_2^3, I_2^2)$ не мають значення.

Якщо функція f є значенням ОТ τ, то кажуть, що τ — операторний терм функції f, або що f задана операторним термом τ.

Зауважимо, що задання ЧРФ операторними термами не є однозначним.

Наприклад, операторні терми o, $S^2(s)$, $S^2(\text{o}, \text{o})$ і $S^3(\text{o}, S^2(\text{o}, s))$ задають ту саму функцію $\mathbf{o}(x)$.

Розглянемо приклади ПРФ, ЧРФ і РФ.

Приклад 2.5.1. Функції-константи — ПРФ:

n-арна нуль-функція $\mathbf{o}^n(x_1, ..., x_n) = 0$ задається ОТ $S^2(\text{o}, I_1^n)$;

n-арна константа $\mathbf{k}^n(x_1, ..., x_n) = k$ задається ОТ $S^2(s, S^2(s, ..., S^2(\text{o}, I_1^n)...)$.

Приклад 2.5.2. Функція $f(x_1, x_2) = x_1 + x_2$ — ПРФ.

Справді, маємо

$$f(x_1, 0) = x_1 = \boldsymbol{I}_1^1(x_1);$$

$$f(x_1, x_2 + 1) = x_1 + (x_2 + 1) = (x_1 + x_2) + 1 = \mathbf{s}(x_1 + x_2) = \boldsymbol{s}(f(x_1, x_2)).$$

Отже, функція $f(x_1, x_2) = x_1 + x_2$ виникає примітивною рекурсією із функцій $g(x_1) = \boldsymbol{I}_1^1(x_1)$ і $h(x_1, x_2, x_3) = x_3 + 1 = \boldsymbol{s}(x_3) = \boldsymbol{S}^2(\boldsymbol{s}, \boldsymbol{I}_3^3)(x_1, x_2, x_3)$.

Операторний терм функції $x_1 + x_2$ має вигляд $R(I_1^1, S^2(s, I_3^3))$.

Приклад 2.5.3. Функція $f(x_1, x_2) = x_1 \cdot x_2$ — ПРФ.

Справді, маємо

$$f(x_1, 0) = 0 = \mathbf{o}(x_1);$$

$$f(x_1, x_2 + 1) = x_1 \cdot (x_2 + 1) = x_1 \cdot x_2 + x_1 = f(x_1, x_2) + x_1.$$

Отже, функція $f(x_1, x_2) = x_1 \cdot x_2$ виникає примітивною рекурсією із функцій $g(x_1) = \mathbf{o}(x_1)$ і $h(x_1, x_2, x_3) = x_3 + x_1$. Згідно з прикладом 2 функція h є ПРФ, тому f — ПРФ.

ОТ функції $x_1 \cdot x_2$ має вигляд $R(\text{o}, S^3(\oplus, I_3^3, I_1^3))$, де $\oplus$ — операторний терм функції $x_1 + x_2$.

Приклад 2.5.4. Функція $sg(x_1) = \begin{cases} 0, & \text{якщо } x_1 = 0, \\ 1, & \text{якщо } x_1 \geq 1, \end{cases}$ — ПРФ.

Справді, маємо

$sg(0) = 0 = \mathbf{o}(x_1)$;

$sg(x_1 + 1) = 1$.

ОТ функції $sg(x_1)$ має вигляд $R(\text{o}, S^2(s, S^2(\text{o}, I_1^2)))$.

Приклад 2.5.5. Функція $nsg(x_1) = \begin{cases} 1, & \text{якщо } x_1 = 0, \\ 0, & \text{якщо } x_1 \geq 1, \end{cases}$ є ПРФ.

Справді, маємо

$nsg(0) = 1 = \boldsymbol{s}(\mathbf{o}(x_1))$;

$nsg(x_1 + 1) = 0$.

ОТ функції $nsg(x_1)$ має вигляд $R(S^2(s, \text{o}), S^2(\text{o}, I_1^2))$.

Приклад 2.5.6. Функція

$$f(x_1, x_2) = x_1 \dot{-} x_2 = \begin{cases} x_1 - x_2, & \text{якщо } x_1 \geq x_2, \\ 0, & \text{якщо } x_1 \leq x_2, \end{cases}$$

є ПРФ.

Спочатку покажемо, що функція $x_1 \dot{-} 1$ є ПРФ.

Справді, маємо

$0 \dot{-} 1 = 0 = \mathbf{o}(x_1)$;

$(x_1 + 1) \dot{-} 1 = x_1 = \boldsymbol{I}_1^2(x_1, x_2)$.

Операторний терм функції $x_1 \dot{-} 1$ має вигляд $R(\text{o}, I_1^2)$.

Тепер покажемо, що функція $f(x_1, x_2) = x_1 \dot{-} x_2$ є ПРФ.

Маємо

$f(x_1, 0) = x_1 \dot{-} 0 = \boldsymbol{I}_1^1(x_1)$;

$f(x_1, x_2 + 1) = x_1 \dot{-} (x_2 + 1) = (x_1 \dot{-} x_2) \dot{-} 1 = f(x_1, x_2) \dot{-} 1$.

Отже, функція $f(x_1, x_2) = x_1 \dot{-} x_2$ виникає примітивною рекурсією із функцій $g(x_1) = \boldsymbol{I}_1^1(x_1)$ і $h(x_1, x_2, x_3) = x_3 \dot{-} 1$.

Звідси ОТ функції $x_1 \dot{-} x_2$ має вигляд $R(I_1^1, S^2(R(\text{o}, I_1^2), I_3^3))$.

Приклад 2.5.7. Функція $f(x_1, x_2) = |x_1 - x_2| = (x_1 \dot{-} x_2) + (x_2 \dot{-} x_1)$ є ПРФ.

Приклад 2.5.8. Функція $f(x_1, x_2) = x_1 - x_2$ є ЧРФ.
Справді, $x_1 - x_2 = \mu_{x_3}(x_1 = x_2 + x_3) = \mu_{x_3}(|x_1 - (x_2 + x_3)| = 0)$.

Приклад 2.5.9. Функція $f(x_1, x_2) = [x_1/x_2]$ — ЧРФ.
Справді, $[x_1/x_2] = \mu_{x_3}(x_2 \cdot (x_3 + 1) > x_1) = \mu_{x_3}(nsg(x_2 \cdot (x_3 + 1) \dot{-} x_1) = 0)$.
Зауважимо, що функція $[x_1/x_2]$ невизначена при $x_2 = 0$, тому вона не РФ і не ПРФ.

Приклад 2.5.10. Функція $f(x_1) = [\sqrt{x_1}\,]$ — РФ.
Маємо

$$[\sqrt{x_1}\,] = \mu_{x_2}((x_2 + 1)\cdot(x_2 + 1) > x_1) = \mu_{x_2}(nsg((x_2 + 1)\cdot(x_2 + 1) \dot{-} x_1) = 0).$$

Зрозуміло, що функція $[\sqrt{x_1}\,]$ тотальна.

Приклад 2.5.11. Всюди невизначена функція $f_\varnothing$ є ЧРФ.
Справді, $f_\varnothing(x_1) = \mu_{x_1}(x_1 + 1 = 0)$, тому $f_\varnothing$ є значенням ОТ **M(*s*).**

Розглянемо деякі властивості ПРФ і РФ.
Для спрощення звичайно позначаємо x_{n+1} і x_{n+2} як y і z.

При $z < y$ домовимося вважати $\sum_{k=y}^{z} a_k = 0$.

Теорема 2.5.1. *Нехай* $(n + 1)$*-арна функція* g *є ПРФ. Тоді* $(n + 1)$*-арна функція* f*, задана умовою*

$$f(x_1, ..., x_n, y) = \sum_{k=0}^{y} g(x_1, ..., x_n, k), \text{ — } \textit{також ПРФ.}$$

Маємо

$$f(x_1, ..., x_n, 0) = g(x_1, ..., x_n, 0);$$
$$f(x_1, ..., x_n, t + 1) = f(x_1, ..., x_n, t) + g(x_1, ..., x_n, t + 1).$$

Отже, функція f виникає примітивною рекурсією із функцій $g(x_1, ..., x_n, 0)$ і $h(x_1, ..., x_n, y, z) = z + g(x_1, ..., x_n, y + 1)$ ■

Теорема 2.5.2. *Нехай* $(n + 1)$*-арна функція* g *є ПРФ. Тоді* $(n + 2)$*-арна функція* f*, задана умовою* $f(x_1, ..., x_n, y, z) = \sum_{k=y}^{z} g(x_1, ..., x_n, k)$, — *також ПРФ.*

Справді, маємо

$$f(x_1, ..., x_n, y, z) = \sum_{k=0}^{z} g(x_1,...,x_n, k) \dot{-} \sum_{k=0}^{y} g(x_1,...,x_n,k) + nsg(y \dot{-} z) \cdot g(x_1, ..., x_n, y)$$ ■

Теорема 2.5.3. *Нехай* $(n + 1)$*-арна функція* g *і* n*-арні функції* p *та* q *— ПРФ. Тоді* n*-арна функція* h*, задана умовою*

$$h(x_1, ..., x_n) = \sum_{k=p(x_1,...,x_n)}^{q(x_1,...,x_n)} g(x_1,...,x_n,k)$$, *— також ПРФ.*

Справді, $h(x_1, ..., x_n) = f(x_1, ..., x_n, p(x_1, ..., x_n), q(x_1, ..., x_n))$, де f — функція із теореми 2.5.2 ■

Зауваження 1. Теореми 2.5.1–2.5.3 називають теоремами про підсумовування. Замінивши в цих теоремах символ суми Σ символом добутку П, одержимо теореми про мультиплікацію.

Приклад 2.5.12. Довизначимо функцію $f(x_1, x_2) = [x_1/x_2]$ так: $[x_1/0] = x_1$. Тоді функція $[x_1/x_2]$ є ПРФ.

Справді, значення $[a/b]$ дорівнює кількості нулів у послідовності $1 \cdot b \dot{-} a$, $2 \cdot b \dot{-} a$, ..., $a \cdot b \dot{-} a$. Тому $[x_1/x_2] = \sum_{k=1}^{x_1} nsg(k \cdot x_2 \dot{-} x_1)$, звідки $[x_1/x_2]$ є ПРФ за теоремою 2.5.3.

Приклад 2.5.13. Функція $mod(x_1, x_2)$ — остача від ділення x_1 на x_2, — є ЧРФ.

Справді, $mod(x_1, x_2) = x_1 \dot{-} (x_2 \cdot [x_1/x_2])$.

Беручи тут довизначену $[x_1/x_2]$, дістаємо довизначену функцію $mod(x_1, x_2)$: $mod(x_1, 0) = 0$. Така довизначена функція $mod(x_1, x_2)$ є ПРФ.

Теорема 2.5.4 (про кускове завдання). *Нехай* n*-арні функції* $f_1, ..., f_k, f_{k+1}$ та $g_1, ..., g_k$ *— ПРФ, причому для жодних значень* $x_1, ..., x_n$ *жодні з двох функцій* $g_1, ..., g_k$ *одночасно не набувають значення* 0. *Тоді* n*-арна функція* f*, задана умовою*

$$f(x_1, ..., x_n) = \begin{cases} f_1(x_1,...,x_n), & \text{якщо } g_1(x_1,...,x_n) = 0, \\ \quad ... \quad ... \quad ... \quad ... \quad ... \quad ... \quad ... \\ f_k(x_1,...,x_n), & \text{якщо } g_k(x_1,...,x_n) = 0, \\ f_{k+1}(x_1,...,x_n) & \text{в інших випадках,} \end{cases}$$

є ПРФ.

Справді, маємо $f(x_1, ..., x_n) =$

$$= \sum_{m=1}^{k} f_m(x_1,...,x_n) \cdot nsg(g_m(x_1, ..., x_n)) + f_{k+1}(x_1, ..., x_n) \cdot sg(\prod_{m=1}^{k} g_m(x_1,...,x_n)) \blacksquare$$

Кажуть, що $(n + 1)$-арну функцію f отримують із $(n + 1)$-арної функції g за допомогою операції *обмеженої мінімізації,* якщо f задається такою умовою:

$$f(x_1, ..., x_n, y) = \begin{cases} \text{перше, починаючи з 0, значення таке, що } z \le y; \\ \text{та } g(x_1,...,x_n) = 0, \text{ якщо таке значення } z \text{ існує,} \\ \text{значення у, якщо такого значення } z \text{ не існує.} \end{cases}$$

Цей факт позначаємо так: $f(x_1, ..., x_n, y) = \mu_{z \le y}((g(x_1, ..., x_n, z) = 0)$.

Теорема 2.5.5 (про обмежену мінімізацію). *Нехай* $(n + 1)$*-арна функція g є ПРФ. Тоді* $(n + 1)$*-арна функція f, задана співвідношенням*

$$f(x_1, ..., x_n, y) = \mu_{z \le y}((g(x_1, ..., x_n, z) = 0),$$

також є ПРФ.

Функцію $sg(\prod_{k=1}^{z} g(x_1,...,x_n,k))$ позначимо $q(x_1, ..., x_n, z)$. Зафіксуємо значення $x_1, ..., x_n$ і y. Нехай $\mu_{z \le y}((g(x_1, ..., x_n, z) = 0) = b$.

Тоді $q(x_1, ..., x_n, z) = \begin{cases} 1, \text{ якщо } z < b, \\ 0, \text{ якщо } z \ge b. \end{cases}$

Звідси $\mu_{z \le y}((g(x_1, ..., x_n, z) = 0) = b = \sum_{z=0}^{y} q(x_1,...,x_n,z)$.

Отже, $f(x_1, ..., x_n, y) = \sum_{z=0}^{y} q(x_1,...,x_n,z)$.

За теоремами про підсумовування та про мультиплікацію f є ПРФ ■

Наслідок. *Нехай функції* $g(x_1, ..., x_n, y)$ *і* $h(x_1, ..., x_n)$ є ПРФ. *Тоді функція* $f(x_1, ..., x_n) = \mu_{y \le h(E_1, ..., x_n)}((g(x_1, ..., x_n, y) = 0)$ *також є ПРФ.*

Зауваження 2. Твердження теорем про підсумовування, мультиплікацію, кускове завдання та обмежену мімімізацію залишаються вірними, якщо слово "ПРФ" замінити на "РФ".

Приклад 2.5.14. Функція $f(x_1) = [\sqrt{x_1}\,]$ є ПРФ.

Справді, маємо $[\sqrt{x_1}\,] = \mu_{x_2 \le x_1}\,(nsg((x_2 + 1)\cdot(x_2 + 1) \dot{-} x_1) = 0)$, тому за теоремою 2.5.5 $[\sqrt{x_1}\,]$ є ПРФ.

2.6. Програмовані функції. Примітивні програмні алгебри

Поняття програми можна вважати синонімом поняття алгоритму. За дещо вужчого трактування під програмою розуміють алгоритм, призначений для виконання на комп'ютері. В основі уточнення поняття програмування як процесу конструювання програм лежить уведене В. Редьком [35] поняття програмної логіки (алгебри).

Програмна алгебра (**P**, **C**) задається парою (**B**, **C**), де множина функцій **P** — основа алгебри, **C** — множина композицій (операцій) над функціями із **P**, **B** $\subseteq$ **P** — множина базових функцій.

Функції, отримані з базових функцій програмної алгебри за допомогою її композицій, називаються *програмно заданими*, або *програмованими.*

На нижньому рівні ієрархії програмних алгебр розміщені *примітивні програмні алгебри* (ППА) — програмні алгебри функцій з простими типами даних. До таких функцій належать, зокрема, функції, задані на неструктурованих множинах (наприклад, на N, на R).

Композиції (операції) ППА адекватно уточнюють засоби конструювання програм як функцій з простими типами даних.

Композиціями ППА є операції *суперпозиції, циклу* та *розгалуження.*

Для ППА n-арних функцій операції суперпозиції, циклу та розгалуження уточнимо у такий спосіб.

Операції суперпозиції — це описані вище операції суперпозиції n-арних функцій $\boldsymbol{S}^{n+1}$, де $n \geq 1$.

Зрозуміло, що операції $\boldsymbol{S}^{n+1}$ однаково визначаються для довільних класів n-арних функцій, заданих на неструктурованих множинах (наприклад, n-арних функцій на N і n-арних функцій на R).

Операція *розгалуження* Δ n-арним функціям g, h і n-арному предикату p ставить у відповідність n-арну функцію f, яку позначають $\Delta(p, g, h)$.

Для кожного набору значень $x_1, ..., x_n$ значення $f(x_1, ..., x_n)$ визначається так:

$$f(x_1, ..., x_n) = \begin{cases} g(x_1,...,x_n), & \text{якщо } p(x_1,...,x_n) = T, \\ h(x_1,...,x_n), & \text{якщо } p(x_1,...,x_n) = F, \\ & \text{не визначене інакше.} \end{cases}$$

Операція *циклу* ☼ n-арній функції g і n-арному предикату p ставить у відповідність n-арну функцію f, яку позначимо ☼(p, g).

Для кожного набору значень $x_1, ..., x_n$ значення $f(x_1, ..., x_n)$ визначається як перший елемент a_m послідовності $a_0 = x_1$, $a_1 = g(a_0, x_2, ..., x_n)$, $a_2 = g(a_1, x_2, ..., x_n)$, ..., $a_k = g(a_{k-1}, x_2, ..., x_n)$, ... такий, що $p(a_m, x_2, ..., x_n) = F$ та для всіх $k < m$ $p(a_k, x_2, ..., x_n) = T$, якщо такий елемент a_m існує.

Якщо такий елемент a_m не існує, то $f(x_1, ..., x_n)\uparrow$.

Функцію ☼(p, g) позначають також *while* p *do* g.

Із визначення операції циклу випливає твердження.

Твердження 2.6.1. Функція ☼(p, g) алгоритмічно обчислювана відносно функцій p і g.

Базовими програмованими функціями для n-арних функцій, заданих на множині R, вважатимемо такі функції:

- функції-селектори $\boldsymbol{I}_m^n(x_1, ..., x_n) = x_m$, де $n \geq m \geq 1$;
- натуральнозначні константи $\boldsymbol{k}^n(x_1, ..., x_n) = k$, де $k \in N$;
- стандартні функції додавання + , віднімання – , множення × і ділення / на множині R;
- обмежені на множині N функції $[x_1/x_2]$ і $mod(x_1, x_2)$;
- бінарні предикати порівняння $<$, $\leq$, $>$, $\geq$, $=$ і $\neq$.

Функція *програмована на R*, якщо її можна отримати із зазначених базових функцій за допомогою скінченної кількості застосувань операцій суперпозиції $\boldsymbol{S}^{n+1}$, розгалуження Δ, циклу ☼ і логічних операцій ¬, **&**, ∨, →.

Твердження 2.6.2. Кожна програмована на *R* *n*-арна функція алгоритмічно обчислювана відносно базових функцій.

Зауважимо, що наведена множина базових функцій та операцій не мінімальна. Наприклад, із логічних операцій достатньо залишити ¬ і **&**, із базових предикатів — предикати < і = .

Алгебра ($\wp_{\boldsymbol{R}}$; ¬, **&**, ∨, →, Δ, ☼, $\boldsymbol{S}^2$, $\boldsymbol{S}^3$, ...), носієм $\wp_{\boldsymbol{R}}$ якої є клас всіх програмованих на *R* *n*-арних функцій, а операціями — операції Δ, ☼ і $\boldsymbol{S}^{n+1}$, де $n \geq 1$, і логічні операції ¬, **&**, ∨, →, називається *примітивною програмною алгеброю програмованих на R n-арних функцій* (скорочено ППА-*AR*-*R*).

Розглянемо програмовані *n*-арні функції на *N*.

Для *n*-арних функцій на *N* можна не залучати до розгляду предикати. Кожну таку функцію можна трактувати як предикат, інтерпретуючи значення функції 0 як істиннісне значення "*F*", а довільне значення функції $a \neq 0$ — як істиннісне значення "*T*".

Тому для *n*-арних функцій на *N* операції $^{\mathbf{N}}\Delta$ і $^{\mathbf{N}}$☼ задаємо так.

Операція $^{\mathbf{N}}\Delta$ *n*-арним функціям *g*, *h* і *p* ставить у відповідність *n*-арну функцію *f*, значення $f(x_1, ..., x_n)$ якої для кожного набору значень $x_1, ..., x_n$ визначається так:

$$f(x_1, ..., x_n) = \begin{cases} g(x_1, ..., x_n), & \text{якщо } p(x_1, ..., x_n) \text{ визначене та } \neq 0, \\ h(x_1, ..., x_n), & \text{якщо } p(x_1, ..., x_n) = 0, \\ \text{не визначене інакше.} \end{cases}$$

Операція $^{\mathbf{N}}$☼ *n*-арним функціям *g* і *p* ставить у відповідність *n*-арну функцію *f*, значення $f(x_1, ..., x_n)$ якої для кожного набору значень $x_1, ..., x_n$ визначається як перший елемент a_m послідовності $a_0 = x_1$, $a_1 = g(a_0, x_2, ..., x_n)$, $a_2 = g(a_1, x_2, ..., x_n)$, ..., $a_k = g(a_{k-1}, x_2, ..., x_n)$, ... такий, що $p(a_m, x_2, ..., x_n) = 0$ та для всіх $k < m$ $p(a_k, x_2, ..., x_n)\downarrow \neq 0$, якщо такий елемент a_m існує.

Якщо такий елемент a_m не існує, то $f(x_1, ..., x_n)\uparrow$.

Із визначення операцій $^{\mathbf{N}}\Delta$ та $^{\mathbf{N}}$☼ випливають твердження.

Твердження 2.6.3. *Якщо функції p, g, h алгоритмічно обчислювані, то функція* $^{\mathbf{N}}\Delta(p, g, h)$ *також алгоритмічно обчислювана.*

Твердження 2.6.4. *Якщо функції p і g алгоритмічно обчислювані, то функція* $^{\mathbf{N}}\☼(p, g)$ *також алгоритмічно обчислювана.*

Базовими програмованими функціями для *n*-арних функцій на *N* вважатимемо функції **o**, ***s*** і $\boldsymbol{I}_m^n$, де $n \geq m \geq 1$, а також $+$, $\cdot$, $\div$.

Функцію називаємо *програмованою на N*, якщо її можна отримати із зазначених вище базових функцій за допомогою скінченної кількості застосувань операцій $\boldsymbol{S}^{n+1}$, $^{\mathbf{N}}\Delta$ та $^{\mathbf{N}}\☼$.

Ураховуючи твердження 2.5.3, 2.6.3, 2.6.4 та алгоритмічну обчислюваність базових програмованих на *N n*-арних функцій, маємо

Твердження 2.6.5. *Кожна програмована на N n-арна функція алгоритмічно обчислювана.*

Алгебра $(\wp_{\mathbf{N}}; {}^{\mathbf{N}}\Delta, {}^{\mathbf{N}}\☼, \mathbf{S}^2, \mathbf{S}^3, ...)$, носієм $\wp_{\mathbf{N}}$ якої є клас усіх програмованих на *N n*-арних функцій, а операціями — операції $^{\mathbf{N}}\Delta$, $^{\mathbf{N}}\☼$ і $\boldsymbol{S}^{n+1}$, де $n \geq 1$, називається *примітивною програмною алгеброю програмованих на N n-арних функцій* (скорочено ППА-*AR*-*N*).

Дамо визначення операторного терма ППА-*AR*-*N*.

Алфавіт мови ППА-*AR*-*N* складається із символів базових функцій o, *s*, $+$, $\times$, $\div$ та I_m^n, де $n \geq m \geq 1$, символів операцій $^{N}\Delta$, $^{N}\☼$ та S^{n+1}, де $n \geq 1$, а також допоміжних символів "(", ")", ",".

Індуктивне визначення ОТ ППА-*AR*-*N* таке:

1) кожен символ базової функції є ОТ; такі ОТ називають атомарними;

2) якщо $t_0, t_1, ..., t_n$ — ОТ, то $S^{n+1}(t_0, t_1, ..., t_n)$ є ОТ;

3) якщо t_0, t_1 та t_2 — ОТ, то $^{N}\Delta(t_0, t_1, t_2)$ є ОТ;

4) якщо t_0 та t_1 — ОТ, то $^{N}\☼(t_0, t_1)$ є ОТ.

Інтерпретуючи символи у природний спосіб, маємо, що кожна програмована на *N n*-арна функція є значенням деякого ОТ ППА-*AR*-*N*.

Проте, як і у випадку ОТ алгебри *n*-арних ЧРФ, через порушення умов арності не кожен ОТ ППА-*AR*-*N* має певне значення.

Як і у випадку ОТ алгебри *n*-арних ЧРФ, подання програмованих на *N n*-арних функцій ОТ ППА-*AR*-*N* неоднозначне.

Логічні операції над предикатами можна промоделювати.

Нехай функції p і q трактуються як предикати. Тоді функції $nsg(p)$, $p \cdot q$ і $p + q$ можна трактувати як предикати $\neg p$, $p\&q$ і $p \vee q$.

Позаяк $p \to q$ і $p \leftrightarrow q$ можна подати як $\neg p \vee q$ та $(p \to q)\&(q \to p)$, легко отримати функції, які моделюють зазначені предикати.

Приклад 2.6.1. Функції-константи програмовані.

Їх отримують із базових функцій **o**, $\boldsymbol{s}$ та $\boldsymbol{I}_m^n$ за допомогою операцій $\boldsymbol{S}^{n+1}$.

Приклад 2.6.2. Функції $nsg(x_1)$ і $sg(x_1)$ програмовані.

Справді, $nsg(x_1) = 1 \dot{-} x_1$, $sg(x_1) = 1 \dot{-} (1 \dot{-} x_1)$.

Приклад 2.6.3. Функція $|x_1 - x_2|$ програмована.

Справді, $|x_1 - x_2| = (x_1 \dot{-} x_2) + (x_2 \dot{-} x_1)$.

Приклад 2.6.4. Бінарні предикати $x_1 > x_2$, $x_1 \geq x_2$, $x_1 = x_2$, $x_1 \neq x_2$ програмовані.

Предикат $x_1 > x_2$ моделюється функцією $x_1 \dot{-} x_2$;

предикат $x_1 \geq x_2$ моделюється функцією $(x_1 + 1) \dot{-} x_2$;

предикат $x_1 = x_2$ можна подати у вигляді $(x_1 \geq x_2)\&(x_2 \geq x_1)$;

предикат $x_1 \neq x_2$ можна подати у вигляді $\neg(x_1 = x_2)$.

Приклад 2.6.5. Функція $mod(x_1, x_2)$ програмована.

ОТ для функції $mod(x_1, x_2)$: ${}^{N}\!\!\Rightarrow(S^3(\dot{-}, S^2(s, I_1^2), I_2^2), \dot{-})$.

Приклад 2.6.6. Функцію $x_1 + x_2$ можна не включати [40] до базових програмованих на N n-арних функцій.

Справді, $x_1 + x_2$ можна подати у вигляді $\boldsymbol{S}^4({}^{\mathbf{N}}\!\!\Rightarrow(p, g), \mathbf{o}^2, \boldsymbol{I}_2^2, \boldsymbol{I}_1^2)$, де p — предикат $x_1 < x_2 + x_3$, g — 3-арна функція $x_1 + 1$. Але $x_1 < x_2 + x_3 \Leftrightarrow nsg(((x_1 + 1) \dot{-} x_2) \dot{-} x_3) = 1$, звідси функція $x_1 + x_2$ отримується із базових функцій **o**, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $\times$, $\dot{-}$ за допомогою операцій ${}^{\mathbf{N}}\!\!\Rightarrow$ та $\boldsymbol{S}^{n+1}$.

Ураховуючи приклад 2.6.6, можна обмежитися базовими функціями **o**, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $\times$, $\dot{-}$.

Приклад 2.6.7. Операцію ${}^{\mathbf{N}}\Delta$ можна промоделювати, використовуючи базові функції ППА-*AR*-*N* і операції суперпозиції та циклу.

Справді, функцію $f = {}^{N}\Delta(p, g, h)$ можна подати у вигляді

$$f = g \cdot sg(\alpha) + h \cdot nsg(\alpha).$$

Отже, $f = {}^{N}\Delta(p, g, h)$ можна отримати із базових програмованих на N n-арних функцій і функцій p, g і h за допомогою операцій ${}^{\mathrm{N}}$☼ та $\boldsymbol{S}^{n+1}$.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Що таке МНР?
2. Що таке конфігурація МНР?
3. Що таке програма МНР?
4. Опишіть команди МНР.
5. Як виконується програма МНР?
6. Дайте визначення еквівалентних МНР-програм.
7. Дайте визначення стандартної МНР-програми.
8. Як визначається конкатенація стандартних МНР-програм?
9. Як визначається обчислюваність функції f: $N^n \to N$ за допомогою МНР-програми P?
10. Що таке МНР-обчислювана функція?
11. Дайте визначення машини Тьюрінга.
12. Які ви знаєте варіанти МТ?
13. Опишіть команди МТ.
14. Що таке конфігурація МТ?
15. Що таке початкова конфігурація МТ? Фінальна конфігурація МТ?
16. Як змінюється конфігурація МТ при виконанні команди МТ відповідного типу?
17. Як МТ задає вербальне відображення?
18. Дайте визначення еквівалентних МТ.
19. Дайте визначення детермінованих і недетермінованих МТ.
20. Як визначається обчислюваність функції f: $N^n \to N$ за допомогою МТ?
21. Що таке МТ-обчислювана функція?
22. Дайте визначення нормального алгоритму Маркова.
23. Як визначається обробка слова нормальним алгоритмом?
24. Як НА задає вербальне відображення?

25. У чому полягає відмінність між НА в алфавіті T і НА над алфавітом T?
26. Як визначається обчислюваність функції f: $N^n \to N$ за допомогою НА?
27. Що таке НА-обчислювана функція?
28. Дайте визначення канонічної системи Поста.
29. Як визначається множина теорем системи Поста?
30. Дайте визначення множини, породжуваної за Постом.
31. Що таке нормальна система Поста?
32. Які ви знаєте комбінаторні системи?
33. Що таке система Туе?
34. Що таке формальна граматика?
35. Які ви знаєте типи формальних граматик?
36. Як визначається обчислюваність функції f: $N^n \to N$ за допомогою системи Поста?
37. Дайте визначення операцій суперпозиції $\boldsymbol{S}^{n+1}$, примітивної рекурсії $\boldsymbol{R}$, мінімізації $\boldsymbol{M}$.
38. Зазначте властивості операцій $\boldsymbol{S}^{n+1}$, $\boldsymbol{R}$ та $\boldsymbol{M}$?
39. Дайте визначення базових обчислюваних n-арних функцій.
40. Дайте визначення ПРФ, ЧРФ і РФ.
41. Зазначте властивості ПРФ, ЧРФ і РФ.
42. Дайте визначення алгебри ЧРФ та алгебри ПРФ.
43. Дайте визначення операторного терма алгебри ЧРФ.
44. Дайте визначення операторного терма алгебри ПРФ.
45. Сформулюйте теореми про підсумовування.
46. Сформулюйте теореми про мультиплікацію.
47. Сформулюйте теореми про кускове завдання.
48. Дайте визначення операції обмеженої мінімізації.
49. Сформулюйте теорему про обмежену мінімізацію.
50. Що таке програмна алгебра?
51. Що таке програмована функція?
52. Що таке примітивна програмна алгебра?
53. Дайте визначення операції циклу $^{\mathbf{N}}\!\!\odot$ для n-арних функцій на N.
54. Опишіть властивості операції $^{\mathbf{N}}\!\!\odot$.
55. Зазначте базові програмовані функції для n-арних функцій на N.
56. Дайте визначення програмованої на N n-арної функції.
57. Опишіть властивості програмованих на N функцій.
58. Дайте визначення ППА-AR-N та операторного терма ППА-AR-N.
59. Як базовими засобами ППА-AR-N промоделювати функцію + ?

ВПРАВИ

1. Наведіть МНР-програми для таких функцій:
   1) $f(x, y) = x - 2y$;
   2) $f(x, y, z) = (x - y) + z$;
   3 $f(x, y) = nsg(x + y)$;
   4) $f(x, y) = sg(x \cdot y)$;
   5) $f(x, y, z) = max(x, y) + z$;
   6) $f(x, y, z) = x - min(y, z)$;
   7) $f(x, y) = max(x, 2y)$;
   8) $f(x) = (x + 1)/3$;
   9) $f(x) = x!$;
   10) $f(x, y) = x^y$.

2. Доведіть, що для кожної МНР-програми можна збудувати еквівалентну їй МНР-програму без команд $T(m, n)$ (елімінація команд $T(m, n)$).

3. Дайте визначення машини з цілочисельними регістрами (МЦР) і МЦР-програм.

4. За допомогою належного кодування розширте поняття МНР-обчислюваності на функції, задані на множині Z.

Визначте МЦР-обчислюваність так, щоб вона була еквівалентна МНР-обчислюваності на Z.

5. Наведіть МТ для таких функцій:
   1)$f(x, y)) = x \dot{-} y$;
   2) $f(x) = sg(x/2)$;
   3) $f(x, y) = sg(x + y)$;
   4) $f(x, y) = nsg(x \cdot y)$;
   5) $f(x, y) = max(x, y)$;
   6) $f(x) = x/2$;
   7) $f(x, y) = x + 2y$;
   8) $f(x) = 2^x$;
   9) $f(x, y) = x^y$;
   10) $f(x) = x!$.

6. Наведіть НА, який:
   1) дописує фіксоване слово на початок вхідного слова;
   2) дописує фіксоване слово в кінець вхідного слова;
   3) стирає перший символ вхідного слова;
   4) стирає останній символ вхідного слова;
   5) кожне слово $x \in T^*$ переводить у слово xx^R.

7. Наведіть НА для таких функцій:
   1) $f(x, y) = min(x, y)$;
   2) $f(x, y) = 3x - 2y$;
   3) $f(x, y) = 2x + 3y$;
   4) $f(x, y, z) = (x + y) - z$;
   5) $f(x, y) = 3^x - 2y$;
   6) $f(x, y) = 2^x + 3^y$;
   7) $f(x, y) = x^y$;
   8) $f(x) = x!$.
8. Наведіть системи Поста для таких функцій:
   1) $f(x, y) = max(x, y)$;
   2) $f(x, y) = x \dot{-} y$;
   3) $f(x, y) = x - y^2$;
   4) $f(x, y) = x^2 \cdot (y + 1)$;
   5) $f(x) = x^3 - 3x$;
   6) $f(x) = x^4 + 2x$;
   7) $f(x, y) = x + 2^y$;
   8) $f(x, y) = 2^x - 3^y$;
   9) $f(x, y) = (x + 1) \cdot 2^y$;
   10) $f(x, y, z) = x \cdot y \cdot z$;
   11) $f(x) = [log_2 x]$;
   12) $f(x, y) = x^y$.
9. Встановіть:
- чи може бути тотальною функція $\boldsymbol{S}^{n+1}(g, g_1, ..., g_n)$, якщо g нетотальна?
- чи може бути тотальною функція $\boldsymbol{S}^{n+1}(g, g_1, ..., g_n)$, якщо одна з функцій $g_1, ..., g_n$ нетотальна?
- чи може бути тотальною функція $\boldsymbol{R}(g, h)$, якщо g нетотальна?
- чи може бути тотальною функція $\boldsymbol{R}(g, h)$, якщо h нетотальна?
- чи може бути тотальною функція $\boldsymbol{M}(g)$, якщо g нетотальна?

10. Вкажіть ОТ алгебри n-арних ЧРФ для функцій:
   1) $f(x_1) = (x_1)!$;
   2) $f(x_1, x_2) = x_1^{x_2}$;
   3) $f(x_1, x_2, x_3) = (x_2 + x_3)!$;
   4) $f(x_1, x_2, x_3) = x_2^{x_1 + x_3}$;
   5) $f(x_1) = [log_3(x_1)]$;

6) $f(x_1) = [\sqrt[3]{x_1}]$;
9) $f(x_1) = (2x_1 + 1)!!$;
10) $f(x_1, x_2) = (2x_2)!!$.

11. Доведіть, що наступні функції є ПРФ:
1) $nd(x)$ — кількість дільників числа x (беремо $nd(0) = 1$);
2) $\sigma(x)$ — сума дільників числа x (беремо $\sigma(0) = 0$);
3) $p(x)$ — x-ве просте число (беремо $p(0) = 2$, $p(1) = 3$ і т. д.);
4) $spd(x)$ — сума простих дільників числа x (беремо $spd(0) = 0$);
5) $kpd(x)$ — кількість простих дільників числа x (тут $kpd(0) = 0$);
6) $ex(x, y)$ — степінь числа $p(x)$ у розкладі числа y на множники, які є степенями простих чисел;
7) $HCD(x_1, x_2)$ та $HCK(x_1, x_2)$.

12. Встановіть:
- чи може бути тотальною функція ${}^{N}\!\!\Box(p, g)$, якщо p нетотальна?
- чи може бути тотальною функція ${}^{N}\!\!\Box(p, g)$, якщо g нетотальна?

13. Вкажіть ОТ ППА-*Ar*-*N* для функцій:
1) $f(x_1, x_2) = max(x_1, x_2)$;
2) $f(x_1, x_2) = min(x_1, x_2)$;
3) $f(x_1, x_2) = HCD(x_1, x_2)$;
4) $f(x_1, x_2) = HCK(x_1, x_2)$;
5) $f(x_1, x_2) = [x_1/x_2]$;
6) $f(x_1) = [\sqrt{x_1}]$;
7) $f(x_1, x_2, x_3) = mod(x_1 + x_2, x_3)$;
8) $f(x_1, x_2, x_3) = max(mod(x_1, x_2 + 1), x_3)$.

3. КОДУВАННЯ ТА НУМЕРАЦІЇ. КАНТОРОВІ НУМЕРАЦІЇ. ТЕЗА ЧОРЧА

3.1. Кодування та нумерації. Універсальні класи алгоритмів

Різні формальні моделі алгоритмів можуть діяти на різних множинах об'єктів. Наприклад, МТ і нормальні алгоритми є вербальними алгоритмами, МНР-програми визначають функції натуральних аргументів і значень.

Для порівняння різних формальних моделей та для переходу від одної моделі до іншої треба *кодувати* елементи одної множини елементами іншої множини.

Під *кодуванням множини A у множині B* розумітимемо ін'єктивне відображення $\varphi: A \to B$ таке, що існують алгоритми $\aleph$ і $\Re$:

- для кожного $a \in A$ $\aleph(a) \in \varphi(a)$;
- алгоритм $\Re$ для кожного $b \in B$ визначає, чи $b \in \varphi(A)$, і якщо так, то знаходить $\varphi^{-1}(b)$.

Під *кодуванням множини A на множині B* розумітимемо ін'єктивне та сюр'єктивне відображення $\varphi: A \to B$ таке, що існують алгоритми $\aleph$ і $\Re$:

- для кожного $a \in A$ $\aleph(a) \in \varphi(a)$;
- для кожного $b \in B$ $\Re(b) = \varphi^{-1}(b)$.

Під *однозначним кодуванням множини A у множині B* розумітимемо ін'єктивне функціональне відображення $\varphi: A \to B$ таке, що існують алгоритми $\aleph$ і $\Re$:

- для кожного $a \in A$ $\aleph(a) = \varphi(a)$;
- алгоритм $\Re$ для кожного $b \in B$ визначає, чи $b \in \varphi(A)$, і якщо так, то знаходить $\varphi^{-1}(b)$.

Під *однозначним кодуванням множини A на множині B* розумітимемо бієктивне відображення відображення $\varphi: A \to B$ таке, що існують алгоритми $\aleph$ та $\Re$:

- для кожного $a \in A$ $\aleph(a) = \varphi(a)$;
- для кожного $b \in B$ $\Re(b) = \varphi^{-1}(b)$.

Іншими словами, однозначне кодування A на B — це бієктивне відображення $\varphi: A \to B$ таке, що φ і φ^{-1} алгоритмічні.

Нехай $\aleph$ — X-A-алгоритм, $\Re$ — Y-B-алгоритм, φ — однозначне кодування X на Y, ψ — однозначне кодування A на B.

Алгоритми $\aleph$ і $\Re$ називають *еквівалентними з точністю до кодувань* φ *і* ψ, якщо:

- для кожного $x \in X$ $\aleph(x) = \psi^{-1}(\Re(\varphi(x)))$;
- для кожного $y \in Y$ $\Re(y) = \psi(\aleph(\varphi^{-1}(y)))$.

Алгоритми $\aleph$ і $\Re$ називають *еквівалентними*, якщо $\aleph$ і $\Re$ еквівалентні з точністю до деяких кодувань φ і ψ.

Клас алгоритмів $\wp$ називають *універсальним*, якщо кожний алгоритм еквівалентний деякому алгоритму із $\wp$.

Аналогічно можна ввести поняття універсального класу чисел.

Розглянуті формальні моделі алгоритмів — МНР-програми, машини Тьюрінга, нормальні алгоритми Маркова — визначають універсальні класи алгоритмів.

З поняттям кодування тісно пов'язане поняття ефективної нумерації.

Розглянемо спочатку поняття нумерації в загальному вигляді.

Нумерацією множини A називають сюр'єктивне функціональне відображення $\xi: N \to A$.

Це означає, що кожний елемент $b \in A$ має номер із N (можливо, не єдиний) і кожне $n \in N$ є номером єдиного елемента $\xi(n) \in A$.

Однозначною нумерацією множини A називають бієктивне відображення $\xi: N \to A$.

Для теорії алгоритмів особливо важливі ефективні нумерації, які дають змогу за кожним $b \in A$ ефективно (тобто за допомогою певного алгоритму) визначити його номер і за кожним $n \in N$ ефективно визначити той елемент $\xi(n) = b \in A$, який нумерується цим n.

Нумерація $\xi: N \to A$ *ефективна*, якщо існують алгоритми $\aleph$ і $\Re$:

- для кожного $a \in A$ $\aleph(a) \in \xi^{-1}(a)$;
- для кожного $n \in N$ $\Re(n) = \xi(n)$.

Отже, $\xi: N \to A$ — ефективна нумерація $\Leftrightarrow \xi^{-1}: A \to N$ — кодування A на N.

3.2. Канторові нумерації

Розглянемо нумерації пар та n-ок натуральних чисел. Такі нумерації називають *канторовими*.

Канторові нумерації є однозначними ефективними нумераціями.

Усі пари натуральних чисел розташуємо у послідовність так:

пара (x, y) передує парі $(u, v) \Leftrightarrow x + y < u + v$ або $(x + y = u + v$ і $x < u)$.

Отже, маємо таку послідовність: (0,0); (0,1); (1,0); (0,2); (1,1); … .

Номер пари (x, y) у такій послідовності позначають $C(x, y)$ та називають *канторовим номером* пари (x, y).

Ліву та праву компоненти пари з номером n позначають відповідно $\boldsymbol{l}(n)$ і $\boldsymbol{r}(n)$.

Функції $\boldsymbol{l}(n)$ та $\boldsymbol{r}(n)$ називають лівою та правою координатними функціями.

Наприклад, $C(0,0) = 0$, $C(0,2) = 3$, $\boldsymbol{l}(4) = 1$, $\boldsymbol{r}(3) = 2$.

Теорема 3.2.1. *Функції* $C(x, y)$, $\boldsymbol{l}(n)$ і $\boldsymbol{r}(n)$ є *ПРФ*.

Пара (x, y) розміщена на x-му місці після пари $(0, x + y)$. Перед парою $(0, x + y)$ знаходиться $x + y$ груп пар з однаковою сумою компонент, причому в групі з сумою компонент m міститься $m + 1$ пара. Тому перед $(0, x + y)$ знаходиться $n = 1 + 2 + ... + (x + y) = (x + y + 1)\cdot(x + y)/2$ пар.

Звідси $C(x, y) = n + x$, тобто

$C(x, y) = [(x + y + 1)\cdot(x + y)/2] + x = [((x + y)^2 + 3\cdot x + y)/2]$.

Отже, $C(x, y)$ є ПРФ.

Нехай $x = \boldsymbol{l}(n)$, $y = \boldsymbol{r}(n)$. Тоді $2n = 2\cdot C(x, y) = ((x + y)^2 + 3\cdot x + y)/2$, звідки $8n + 1 = (2x + 2y + 1)^2 + 8x = (2x + 2y + 3)^2 - 8y - 8$. Звідси маємо $2x + 2y + 1 \le [\sqrt{8n+1}\,] < 2x + 2y + 3$, тому $x + y + 1 \le [([\sqrt{8n+1}\,] + 1)/2] < x + y + 2$. Отже, $x + y + 1 = [([\sqrt{8n+1}\,] + 1)/2]$. Але $n = C(x, y) = [(x + y + 1)\cdot(x + y)/2] + x$, звідки маємо $\boldsymbol{l}(n) = x = n \dot{-} [(x + y + 1)\cdot(x + y)/2] = n \dot{-} [[([\sqrt{8n+1}\,] + 1)/2]\cdot[([\sqrt{8n+1}\,] \dot{-} 1)/2]/2]$.

Таким чином, функція $\boldsymbol{l}(n)$ є ПРФ.

Маємо $\boldsymbol{r}(n) = y = (x + y + 1) \dot{-} (x + 1) = [([\sqrt{8n+1}\,] + 1)/2] \dot{-} (\boldsymbol{l}(n) + 1)$.

Тому $\boldsymbol{r}(n)$ є ПРФ ■

Функція $C(x, y)$ задає бієкцію $N\times N \to N$, пара функцій $(\boldsymbol{l}(n), r(n))$ — бієкцію $N \to N\times N$.

Функції C, $\boldsymbol{l}$ і $\boldsymbol{r}$ пов'язані такими тотожностями:

$$C(\boldsymbol{l}(n), \boldsymbol{r}(n)) = n;$$

$$\begin{aligned} &\boldsymbol{l}(C(x, y)) = x; \\ &\boldsymbol{r}(C(x, y)) = y \end{aligned} \qquad (1)$$

Зауважимо, що якщо функції $\boldsymbol{C}$, $\boldsymbol{L}$, $\boldsymbol{R}$ пов'язані тотожностями

$$\begin{aligned} &\boldsymbol{C}(\boldsymbol{L}(n), \boldsymbol{R}(n)) = n; \\ &\boldsymbol{L}(\boldsymbol{C}(x, y)) = x; \\ &\boldsymbol{R}(\boldsymbol{C}(x, y)) = y, \end{aligned} \qquad (2)$$

то, називаючи $\boldsymbol{C}(x, y)$ номером пари (x, y), дістаємо однозначну нумерацію $N \times N$.

З іншого боку, нехай задано однозначну нумерацію $N \times N$, тобто всі пари натуральних чисел без повторів розташовані у послідовність

$$(x_0, y_0), (x_1, y_2), ..., (x_n, y_n), ...$$

Поклавши $\boldsymbol{L}(n) = x_n$, $\boldsymbol{R}(n) = y_n$ та $\boldsymbol{C}(x_n, y_n) = n$, дістанемо функції, пов'язані тотожностями (2).

Маючи нумерацію пар натуральних чисел, можна ввести нумерацію n-ок натуральних чисел для довільного n:

$$C^3(x_1, x_2, x_3) = C(C(x_1, x_2), x_3);$$
$$C^{k+1}(x_1, x_2, ..., x_{k+1}) = C^k(C(x_1, x_2), x_3, ..., x_{k+1}) =$$
$$= C(... C(C(x_1, x_2), x_3), ...), x_{k+1}), k > 2.$$

Координатні функції $C_{n1}, ..., C_{nn}$ уводимо так:

Нехай $C^n(x_1, x_2, ..., x_n) = m$. Тоді

$$C_{n1}(m) = x_1; C_{n2}(m) = x_2; ..., C_{nn}(m) = x_n.$$

Для функцій C^n, $C_{n1, ..., } C_{nn}$ маємо такі тотожності:

$$C^n(C_{n1}(x), ..., C_{nn}(x)) = x;$$
$$C_{nk}(C^n(x_1, x_2, ..., x_n)) = x_k, 1 \le k \le n.$$

Теорема 3.2.2. *Функції C^n, C_{n1}, ..., C_{nn} є ПРФ.*

Справді,

$$C_{nn}(m) = x_n = \boldsymbol{r}(m); C_{nn-1}(m) = x_{n-1} = \boldsymbol{r}(\boldsymbol{l}(m)); ...$$
$$C_{n2}(m) = x_2 = \boldsymbol{r}(\boldsymbol{l}(... \boldsymbol{l}(m))..); C_{n1}(m) = x_1 = \boldsymbol{l}(\boldsymbol{l}(... \boldsymbol{l}(m))..) \blacksquare$$

Приклад 3.2.1. Знайдемо $\boldsymbol{l}(100)$ і $\boldsymbol{r}(100)$.

Для $x = \boldsymbol{l}(100)$ і $y = \boldsymbol{r}(100)$ маємо рівняння $C(x, y) = 100$. Нехай $x + y = p$. Тоді p є найбільшим натуральним числом, для

якого $p\cdot(p+1) \le 2\cdot 100$. Звідси $p = 13$. Тому маємо $x + y = 13$, $x = 100 - [(13\cdot 14)/2] = 9$, $y = 13 - 9 = 4$.

Отже, $\boldsymbol{l}(100) = 9$ і $\boldsymbol{r}(100) = 4$.

Приклад 3.2.2. Розв'яжемо рівняння $C^4(x, y, z, v) = 207$.

За визначенням $C(C(C(x, y), z), v) = 207$. Нехай $C(C(x, y), z) = a$, маємо $C(a, v) = 207$. Нехай $a + v = p$. Тоді p є найбільшим натуральним числом, для якого $p\cdot(p+1) \le 2\cdot 207$. Звідси $p = 19$, отже, $a = 17$ і $v = 2$. Тепер маємо $C(C(x, y), z) = 17$. Нехай $C(x, y) = b$, тоді $C(b, z) = 17$. Нехай $b + z = q$. Тоді q є найбільшим натуральним числом, для якого $q\cdot(q+1) \le 2\cdot 17$. Звідси $q = 5$, отоже, $b = 2$ і $z = 3$. Маємо $C(x, y) = 2$, звідки $x = 1$ і $y = 0$.

Приклад 3.2.3. На основі канторових нумераційних функцій задамо однозначну ефективну нумерацію всіх скінченних послідовностей натуральних чисел.

Спочатку задамо кодування κ таких послідовностей:

$$\kappa(\varnothing) = 0;$$

$$\kappa(a_1, ..., a_n) = C(C^n(a_1, ..., a_n), n - 1) + 1.$$

Зрозуміло, що таке відображення $\kappa: \bigcup_{k\ge 0} N^k \to N$ бієктивне. Тепер обернене відображення $\eta = \kappa^{-1}$ — шукана однозначна нумерація.

Приклад 3.2.4. Задамо тепер наступне кодування σ скінченних послідовностей:

$$\sigma(\varnothing) = 0;$$

$$\sigma(a_1, ..., a_n) = 2^{a_1} + 2^{a_1+a_2+1} + ... + 2^{a_1+a_2+...+a_n+n-1}$$

Бієктивність відображення $\sigma: \bigcup_{k\ge 0} N^k \to N$ випливає з однозначності подання кожного натурального числа у двійковій системі числення. Тоді обернене відображення $\lambda = \sigma^{-1}$ — однозначна ефективна нумерація всіх скінченних послідовностей натуральних чисел.

Приклад 3.2.5. Модифікуючи кодування σ, дістаємо однозначне кодування всіх непорожніх скінченних послідовностей натуральних чисел ν: $\bigcup_{k\geq 1} N^k \to N$:

$$\nu(a_1, ..., a_n) = 2^{a_1} + 2^{a_1+a_2+1} + ... + 2^{a_1+a_2+...+a_n+n-1} - 1.$$

Обернене відображення $\zeta = \nu^{-1}$ — однозначна ефективна нумерація всіх непорожніх скінченних послідовностей натуральних чисел.

Приклад 3.2.6. Ефективну нумерацію ϕ множини формул довільної мови 1-го порядку зі зліченним алфавітом уведемо наступним чином.

Занумеруємо множини предметних імен x_0, x_1, ..., константних символів c_0, c_1, ..., функціональних символів f_0, f_1, ... відповідної арності та предикатних символів p_0, p_1, ... відповідної арності.

Знаючи k — номер функціонального чи предикатного символа, можна знайти його арність n.

Кодування термів τ і формул κ задамо так:

$$\tau(x_k) = 3\cdot k;\ \kappa;$$
$$\tau(c_k) = 3\cdot k + 1;$$
$$\tau(f_k t_1 ... t_n) = 3\cdot C(C^n(\tau(t_1), ..., \tau(t_n)), k) + 2;$$
$$\kappa(p_k t_1 ... t_n) = 4\cdot C(C^n(\tau(t_1), ..., \tau(t_n)), k);$$
$$\kappa(\neg\Phi) = 4\cdot C(k, \kappa(\Phi)) + 1;$$
$$\kappa(\Phi\vee\Psi) = 4\cdot C(\kappa(\Phi), \kappa(\Psi)) + 2;$$
$$\kappa(\exists x_k\Phi) = 4\cdot C(k, \kappa(\Phi)) + 3.$$

Обернене відображення $\phi = \kappa^{-1}$ — однозначна ефективна нумерація всіх формул мови 1-го порядку.

3.3. Функція Гьоделя. Елімінація примітивної рекурсії

Функція Гьоделя Г дає змогу кодувати одним натуральним числом довільну скінченну послідовність натуральних чисел.

Функція Гьоделя визначається так:

$$\Gamma(x, y) = mod(\boldsymbol{l}(x), 1 + (y + 1)\cdot\boldsymbol{r}(x)).$$

Отже, функція Г є ПРФ.

Теорема 3.3.1 (про основну властивість функції Гьоделя). *Для довільної скінченної послідовності натуральних чисел* $b_0, b_1, ..., b_n$ існує натуральне число t таке, що $\Gamma(t, i) = b_i$ для всіх $i \in \{0, ..., n\}$.

Доведення можна прочитати в [7, 8, 9].

Кодування за допомогою функції Гьоделя $\Gamma(x, y)$ довільних скінченних послідовностей натуральних чисел одним натуральним числом дозволяє при розширенні множини базових функцій промоделювати операцію примітивної рекурсії.

Теорема 3.3.2 (про елімінацію операції примітивної рекурсії). *Функція* $f = \boldsymbol{R}(g, h)$ *може бути отримана із функцій* g, h, *базових функцій і функцій* $+, \times$ *та* $\div$ *за допомогою скінченної кількості застосувань операцій суперпозиції* $\boldsymbol{S}^{n+1}$ *і мінімізації* $\boldsymbol{M}$.

Доводимо для випадку, коли $f = \boldsymbol{R}(g, h)$ — бінарна функція.

У загальному випадку доведення аналогічне.

Зафіксуємо x і y.

Згідно зі схемою примітивної рекурсії для функції $f(x, y)$ маємо:

$$\begin{aligned} &f(x, 0) = g(x);\\ &f(x, 1) = h(x, 0, f(x, 0));\\ &\ldots\ \ldots\ \ldots\ \ldots\ \ldots\\ &f(x, y) = h(x, y - 1, f(x, y - 1)). \end{aligned} \tag{1}$$

За основною властивістю функції Гьоделя існує таке t, що

$$\begin{aligned} &\Gamma(t, 0) = f(x, 0);\\ &\ldots\ \ldots\ \ldots\ \ldots\\ &\Gamma(t, y) = f(x, y). \end{aligned} \tag{2}$$

Урахувавши (1), перепишемо (2) у вигляді

$$\begin{aligned} &\Gamma(t, 0) = g(x);\\ &\Gamma(t, 1) = h(x, 0, \Gamma(t, 0));\\ &\ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\\ &\Gamma(t, y) = h(x, y - 1, \Gamma(t, y - 1)). \end{aligned} \tag{3}$$

Можна розглядати (3) як систему рівнянь відносно t.

Згідно з (3) для всіх $u \in \{0, ..., y - 1\}$ маємо $\Gamma(t, u + 1) = h(x, u, \Gamma(t, u))$, звідки $y = \mu_u(y = u$ або $\Gamma(t, u + 1) \neq h(x, u, \Gamma(t, u)))$.

Тому (3) рівносильне системі

$$\Gamma(t, 0) = g(x);$$

$$y = \mu_u(|y - u| \cdot nsg(|\Gamma(t, u + 1) - h(x, u, \Gamma(t, u))|) = 0). \quad (4)$$

Позначимо $\mu_u(|y - u| \cdot nsg(|\Gamma(t, u + 1) - h(x, u, \Gamma(t, u))|) = 0)$ як $F(x, y, t)$.

Тоді система (4) рівносильна рівнянню

$$|\Gamma(t, 0) - g(x)| + |y - F(x, y, t)| = 0. \quad (5)$$

За основною властивістю функції Гьоделя система (3), а отже і рівносильне їй рівняння (5), має розв'язок відносно t для всіх значень x і y, тому існує мінімальний розв'язок

$$t = \mu_z(|\Gamma(z, 0) - g(x)| + |y - F(x, y, z)| = 0).$$

Позначивши таке t як $\varphi(x, y)$, матимемо $f(x, y) = \Gamma(t, y) = \Gamma(\varphi(x, y), y)$.

Функція φ отримана із функцій g, h і функцій Γ, $+$, $\times$, $\div$, nsg і базових функцій $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$ за допомогою операцій $\boldsymbol{S}^{n+1}$ та $\boldsymbol{M}$.

Функцію Γ отримують із функцій $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$ і функцій $+$, $\times$, $\div$, mod, $\boldsymbol{l}$, $\boldsymbol{r}$ за допомогою операцій $\boldsymbol{S}^{n+1}$ і $\boldsymbol{M}$.

У свою чергу, функції $\boldsymbol{l}$ та $\boldsymbol{r}$ отримані із функцій $[\sqrt{x}]$, $[x/2]$, $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $+$, $\div$, $\times$ за допомогою операцій $\boldsymbol{S}^{n+1}$.

Функції $[\sqrt{x}]$, $[x/2]$ і mod отримують із функцій $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $+$, $\div$, $\times$ за допомогою операцій $\boldsymbol{S}^{n+1}$ та $\boldsymbol{M}$.

Ураховуючи, що $nsg(x) = 1 \div x$, функцію Γ можна отримати із функцій $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $+$, $\div$, $\times$ за допомогою операцій $\boldsymbol{S}^{n+1}$ і $\boldsymbol{M}$.

Отже, функцію φ, а отже і функцію f, отримують із функцій g, h, $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $+$, $\div$, $\times$ за допомогою операцій суперпозиції $\boldsymbol{S}^{n+1}$ і мінімізації $\boldsymbol{M}$. ■

Наслідок. *Клас ЧРФ збігається з класом функцій, отриманих із функцій* $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $+$, $\div$, $\times$ *за допомогою скінченної кількості застосувань операцій* $\boldsymbol{S}^{n+1}$ *і* $\boldsymbol{M}$.

3.4. Теза Чорча

Розглянемо співвідношення між різними формальними уточненнями поняття алгоритмічно обчислюваної функції.

Теорема 3.4.1. *Клас ЧРФ збігається з класом програмованих на N n-арних функцій.*

Доведемо теорему.

1. ЧРФ $\subseteq$ програмовані на N.

Базові функції **o**, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$ програмовані на N. Покажемо, що функції, отримані із програмованих за допомогою операцій $\boldsymbol{R}$ і $\boldsymbol{M}$, також програмовані.

Нескладно переконатись, що функції $+$, $\div$, $\times$ програмовані. За теоремою 3.3.2 про елімінацію операції $\boldsymbol{R}$ досить показати, що якщо g програмована, то й $f = \boldsymbol{M}(g)$ програмована.

Але операція $\boldsymbol{M}$ — по суті окремий випадок операції циклу.

У випадку $(n+1)$-арної функції g ОТ ППА-AR-N для функції $f = \boldsymbol{M}(g)$ має вигляд

$$S^{n+2}({}^{N}\!\!\Cup(S^{n+2}(g, I_2^{n+1}, ..., I_{n+1}^{n+1}, I_1^{n+1}), S^2(s, I_1^{n+1})), \text{o}^n, I_1^n, ..., I_n^n).$$

2. Програмовані на $N \subseteq$ ЧРФ.

Базові програмовані на N n-арні функції **o**, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $+$, $\div$, $\times$ є ЧРФ. Покажемо, що якщо функції p і f є ЧРФ, то й функція ${}^{\mathbf{N}}\!\!\Cup(p, f)$ — ЧРФ.

Нехай маємо ЧРФ $p(x_1, ..., x_n)$ і $f(x_1, ..., x_n)$.

Розглянемо тепер функцію $q(x_1, ..., x_n, y)$ — y-кратну ітерацію функції f за x_1. Її задають такою схемою примітивної рекурсії:

$$q(x_1, ..., x_n, y) = x_1;$$

$$q(x_1, ..., x_n, k+1) = f(q(x_1, ..., x_n, k), x_2, ..., x_n).$$

Отже, q можна подати у вигляді $\boldsymbol{R}(g, h)$, де функції g і h — суть $g(x_1, ..., x_n) = x_1$ та $h(x_1, ..., x_n, y, z) = f(z, x_2, ..., x_n)$. ОТ алгебри ЧРФ для функції q має вигляд $R(I_1^n, S^{n+1}(f, I_{n+2}^{n+2}, I_2^{n+2}, ..., I_n^{n+2}))$. Звідси маємо ${}^{\mathbf{N}}\!\!\Cup(p, f)(x_1, ..., x_n) = q(x_1, ..., x_n, \mu_y(p(q(x_1, ..., x_n, y), x_2, ..., x_n) = 0))$.

Отже, ${}^{\mathbf{N}}\!\!\Cup(p, f)$ — ЧРФ ■

Теорема 3.4.2. *Кожна ЧРФ — МНР-обчислювана функція.*

Базові функції **o**, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$ обчислювані такими МНР-програмами:

- $Z(0)$ — для функції **o**;
- $S(0)$ — для функції $\boldsymbol{s}$;
- $T(m-1, 0)$ — для функції $\boldsymbol{I}_m^n$.

Нехай k-арна функція g та n-арні функції $g_1, ..., g_k$ обчислювані відповідно МНР-програмами $G, G_1, ..., G_k$.

Тоді $\boldsymbol{S}^{n+1}(g, g_1, ..., g_k)$ обчислюється МНР-програмою такого вигляду:

$T(j, m + j + 1)$ для всіх $j \in \{0, ..., n - 1\}$;

$G_j[m + 1, ..., m + n \to m + n + j]$ для всіх $j \in \{0, ..., n - 1\}$;

$G[m + n + 1, ..., m + n + k \to 0]$.

Тут $m = max(\rho(G), \rho(G_1), ..., \rho(G_k)))$.

Нехай n-арна функція g та $(n + 2)$-арна функція h обчислювані відповідно МНР-програмами G та H. Тоді функція $\boldsymbol{R}(g, h)$ обчислюється МНР-програмою такого вигляду (тут $m = max(\rho(G), \rho(H))$:

$T(j, m + j + 1)$ для всіх $j \in \{0, ..., n\}$;

$p)$ $G[m + 1, ..., m + n \to m + n + 3]$;

$J(m + n + 1, m + n + 2, q)$;

$H[m + 1, ..., m + n + 2, m + n + 3 \to m + n + 3]$;

$S(m + n + 2)$;

$J(0, 0, p)$;

$q)$ $T(m + n + 3, 0)$.

Нехай $(n + 1)$-арна функція g обчислювана МНР-програмою G. Тоді функція $\boldsymbol{M}(g)$ обчислюється МНР-програмою такого вигляду (тут $m = \rho(G)$):

$T(j, m + j + 1)$ для всіх $j \in \{0, ..., n - 1\}$;

$p)$ $G[m + 1, ..., m + n + 1 \to 0]$;

$J(0, m + n + 2, q)$;

$S(m + n + 1)$;

$J(0, 0, p)$;

$q)$ $T(m + n + 1, 0)$.

Отже, кожна ЧРФ — МНР-обчислювана функція ■

Вірне і зворотне твердження.

Теорема 3.4.3. *Кожна МНР-обчислювана функція є ЧРФ.*

Дамо ідею доведення.

Нехай $f(x_1, ..., x_n)$ обчислюється МНР-програмою P.

Позначимо $\eta(x_1, ..., x_n, t)$ функцію, значенням якої є вміст 0-го регістра після t кроків роботи P над $x_1, ..., x_n$, або після $q \le t$ кроків, якщо на q-му кроці програма P при роботі над $x_1, ..., x_n$ зупинилась. Позначимо через $\omega(x_1, ..., x_n, t)$ функцію, значенням якої є номер команди після t кроків роботи програми P над $x_1, ..., x_n$, або 0, якщо P при роботі над $x_1, ..., x_n$ зупинилась на кроці $q \le t$.

Моделюючи роботу P над $x_1, ..., x_n$ за t кроків, можна довести, що η і ω є ПРФ. Але $f(x_1, ..., x_n) = \eta(x_1, ..., x_n, \mu_t(\omega(x_1, ..., x_n, t) = 0))$, тому f є ЧРФ ■

Наслідок 1. *Клас ЧРФ збігається з класом МНР-обчислюваних функцій.*

Ураховуючи результат теореми 3.4.1, маємо:

Наслідок 2. *Класи ЧРФ, МНР-обчислюваних функцій та програмованих на N функцій збігаються.*

За кожною МТ можна збудувати еквівалентний їй НА, який задає те саме вербальне відображення (нескладна вправа). З іншого боку, за кожним НА можна збудувати еквівалентну йому МТ. Отже:

Твердження 3.4.1. *Клас МТ-обчислюваних функцій збігається з класом НА-обчислюваних функцій.*

Ще 1937 р. А. Тьюрінг встановив (доведення див. [9]):

Твердження 3.4.2. *Клас МТ-обчислюваних функцій збігається з класом ЧРФ.*

Маємо також наступне твердження (див., наприклад, [5]):

Твердження 3.4.3. *Клас функцій, обчислюваних за Постом, збігається з класом НА-обчислюваних функцій.*

Підсумовуючи названі результати, дістаємо наступну теорему.

Теорема 3.4.4. *Наступні класи функцій збігаються*:
1) *клас ЧРФ*;
2) *клас програмованих на N n-арних функцій*;
3) *клас МНР-обчислюваних функцій*;
4) *клас функцій, обчислюваних за Тьюрінгом*;
5) *клас функцій, обчислюваних за Марковим*;
6) *клас функцій, обчислюваних за Постом.*

Розглянуті вище формалізми задають той самий клас n-арних функцій на N. При цьому визначення формалізмів забезпечують ефективну обчислюваність описуваних ними функцій.

Тому є всі підстави вважати, що такі формалізми є різними математичними уточненнями інтуїтивного поняття алгоритмічно обчислюваної функції (АОФ).

Уперше таке твердження стосовно рекурсивних функцій висунув у 1936 р. А. Чорч, тому воно дістало назву "*теза Чорча*". У цьому самому році С. Кліні узагальнив тезу Чорча для часткових функцій.

У такому розширеному вигляді тезу Чорча (скорочено ТЧ) можна сформулювати так:

Теза Чорча. **Клас ЧРФ збігається з класом *n*-арних АОФ, заданих на множині натуральних чисел.**

Поняття АОФ не є строго визначеним математичним поняттям, тому теза Чорча математичному доведенню не підлягає.

Теза Чорча є *природно-науковим фактом*, підтвердженим такими результатами:

1) істотно різні формальні уточнення поняття АОФ, запропоновані різними авторами в різний час, виявилися еквівалентними в сенсі завдання того самого класу функцій;

2) перехід від завдання функції в одному формалізмі до завдання цієї самої функції в іншому формалізмі конструктивний, тобто він здійснюється певним алгоритмом (наприклад, за операторним термом алгебри ЧРФ ефективно будується МНР-програма, що задає ту саму функцію);

3) для кожного з таких формалізмів усі задані в ньому функції алгоритмічно обчислювані в інтуїтивному смислі;

4) усі відомі досі алгоритмічно обчислювані *n*-арні функції на N виявилися частково рекурсивними. Нікому ще не вдалося навести приклад функції, яку можна було б вважати алгоритмічно обчислюваною в інтуїтивному смислі, але яка не є ЧРФ.

Із тези Чорча як наслідок випливає: *клас РФ збігається з класом тотальних АОФ, заданих на множині натуральних чисел.*

Значення тези Чорча полягає в наступному.

1. Прийняття ТЧ перетворює інтуїтивні поняття алгоритму, обчислюваності, розв'язності в об'єкти математичного вивчення. Це дає змогу ставити і розв'язувати питання про алгоритмічну обчислюваність функцій, алгоритмічну розв'язність чи нерозв'язність масових проблем.

2. Використання ТЧ як своєрідної аксіоми дає змогу в багатьох випадках замінити формальні завдання алгоритмів неформальними їх описами. Це істотно спрощує доведення, дозволяє виокремити основну ідею доведення чи побудови, звільняючи його від зайвих деталей.

Проте доведення на основі ТЧ завжди має бути ретельно аргументованим! При виникненні сумнівів треба бути здатним провести чисто формальне доведення.

Отже, у нас є всі підстави прийняти тезу Чорча.

Активно використовуватимемо тезу Чорча в подальшому викладі.

Розглянемо приклад використання тези Чорча.

Приклад 3.4.1. Нехай функція f є ЧРФ.

Доведемо, що тоді функція

$$h(x) = \begin{cases} 1, \text{ якщо } x \in E_f, \\ \text{не визначене в інших випадках.} \end{cases}$$

також є ЧРФ.

Розглянемо процес глобального обчислення всіх значень функції f. Такий процес розіб'ємо на етапи. На кожному етапі починаємо обчислення для наступного значення аргументу.

На етапі 0 робимо 1-й крок обчислення $f(0)$.

На етапі 1 робимо 1-й крок обчислення $f(1)$ та 2-й крок обчислення $f(0)$ і т. д.

На етапі n робимо 1-й крок обчислення $f(n)$, 2-й крок обчислення $f(n-1)$, …, $(n+1)$-й крок обчислення $f(0)$.

Якщо на якомусь етапі обчислення певного $f(m)$ завершується, порівнюємо $f(m)$ і x. При $f(m) = x$ процес глобальних обчислень завершується, адже тоді $x \in E_f$, тому результатом нашої роботи буде число 1.

При $f(m) \neq x$ продовжуємо процес глобальних обчислень.

Ми описали алгоритм для обчислення функції $h(x)$, звідки за тезою Чорча функція $h(x)$ є ЧРФ.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Що таке кодування множини A у множині B? Кодування множини A на множині B?
2. Що таке однозначне кодування множини A у множині B? Однозначне кодування множини A на множині B?
3. Дайте визначення еквівалентних з точністю до кодувань алгоритмів, еквівалентних алгоритмів.

4. Що таке універсальний клас алгоритмів?
5. Що таке нумерація?
6. Що таке однозначна нумерація?
7. Які нумерації називають ефективними?
8. Який зв'язок ефективних нумерацій та кодувань?
9. Як визначаються канторові нумерації пар і n-ок натуральних чисел?
10. Наведіть тотожності для нумераційних функцій C, $\boldsymbol{l}$ і $\boldsymbol{r}$.
11. Наведіть тотожності для нумераційних функцій C^n, C_{n1}, ..., C_{nn}.
12. Задайте однозначні ефективні нумерації всіх скінченних послідовностей та всіх скінченних непорожніх послідовностей натуральних чисел на основі канторових нумераційних функцій.
13. Задайте однозначні ефективні нумерації всіх скінченних послідовностей та всіх скінченних непорожніх послідовностей натуральних чисел на основі подання натуральних чисел у двійковій системі числення.
14. Задайте ефективну нумерацію множини формул мови 1-го порядку із зліченним алфавітом.
15. Дайте визначення функції Гьоделя.
16. Сформулюйте основну властивість функції Гьоделя.
17. Сформулюйте теорему про елімінацію операції примітивної рекурсії.
18. Наведіть співвідношення між різними формальними уточненнями поняття алгоритмічно обчислюваної функції.
19. Сформулюйте тезу Чорча.
20. Як ви можете аргументувати вірність тези Чорча?
21. У чому полягає значення тези Чорча?

ВПРАВИ

1. Знайдіть $\boldsymbol{l}(150)$, $\boldsymbol{r}(150)$ і $\boldsymbol{l}(200)$, $\boldsymbol{r}(200)$.
2. Розв'яжіть такі рівняння:
   1) $C^3(x, y, z) = 131$;
   2) $C^3(x, y, z) = 226$;
   3) $C^4(x, y, z, v) = 123$;
   4) $C^4(x, y, z, v) = 282$.

3. Задайте ефективну нумерацію множини формул мови арифметики.

4. Задайте ефективну нумерацію множини формул мови теорії множин.

5. Задайте ефективну нумерацію множини формул мови частково впорядкованих множин.

6. Доведіть, що класи МТ-обчислюваних і НА-обчислюваних функцій збігаються.

7. Доведіть, що кожна n-арна ЧРФ обчислювана за Постом.

8. Доведіть рекурсивність функції f, заданої такою умовою: $f(n)$ є $(n+1)$-ю цифрою після коми в десятковому розкладі числа e.

9. Доведіть рекурсивність функції f, заданої такою умовою: $f(n)$ є $(n+1)$-ю цифрою після коми в десятковому розкладі числа π.

10. Нехай функції f та g є ЧРФ. Доведіть:

1) функція $h(x) = \begin{cases} 1, \text{ якщо } x \in D_f \cup E_g, \\ \text{не визначене в інших випадках,} \end{cases}$

є ЧРФ;

2) функція $h(x) = \begin{cases} 1, \text{ якщо } x \in E_f \cap D_g, \\ \text{не визначене в інших випадках,} \end{cases}$

є ЧРФ.

11. Нехай функції f і g є ЧРФ. Чи вірно, що функція $h(x)$, задана умовою

$$h(x) = \begin{cases} 1, \text{ якщо } x \in D_f \setminus D_g, \\ \text{не визначене в інших випадках,} \end{cases}$$

завжди є ЧРФ?

4. НУМЕРАЦІЇ ЧРФ. УНІВЕРСАЛЬНІ ФУНКЦІЇ. ТЕОРЕМИ КЛІНІ ПРО НЕРУХОМУ ТОЧКУ

У цьому розділі розглянемо приклади ефективних нумерацій формальних моделей алгоритмів та АОФ. Задамо стандартні нумерації ЧРФ. Далі введемо фундаментальне поняття універсальної функції та відповідні поняття універсальної алгоритмічної машини та універсальної програми.

У розділі також розглянуто теореми Кліні про нерухому точку. Твердження про існування нерухомих точок мають у математиці універсальний характер. Зокрема, у теорії алгоритмів вони набувають вигляду теорем про нерухому точку для рекурсивних функцій, які будуть розглянуті далі, і теорем про нерухому точку для ефективних операцій на функціях і множинах, про що можна прочитати в [5; 8; 12].

4.1. Ефективні нумерації формальних моделей алгоритмів і АОФ

Розглянемо приклади ефективних нумерацій формальних моделей алгоритмів та АОФ.

Приклад 4.1.1. Однозначну ефективну нумерацію всіх МНР-програм задамо на основі кодування МНР-програм як скінченних послідовностей команд МНР.

Кодування команд θ задамо так:

$$\theta(Z(n)) = 4\cdot n;$$
$$\theta(S(n)) = 4\cdot n + 1;$$
$$\theta(T(m, n)) = 4\cdot C(m, n) + 2;$$
$$\theta(J(m, n, q + 1)) = 4\cdot C(C(m, n), q) + 3.$$

Зрозуміло, що таке θ — бієктивне відображення множини всіх команд МНР на N.

Використовуючи розглянуту вище бієкцію $\nu: \bigcup_{k \geq 1} N^k \to N$, задамо кодування τ усіх МНР-програм так.

Якщо P — МНР-програма $I_1 I_2 \ldots I_k$, то $\tau(P) = \nu(\theta(I_1), \ldots, \theta(I_k))$, але ν і θ бієктивні, тому τ також бієкція. Тоді $\varphi = \tau^{-1}$ — шукана однозначна нумерація.

Нумерація φ ефективна. Справді, за кожною МНР-програмою P ефективно обчислюється її номер $\tau(P)$.

З іншого боку, за кожним $n \in N$ ефективно визначаємо МНР-програму $P = \varphi(n)$. Справді, спочатку подамо число $n + 1$ як суму зростаючих степенів числа 2: $n + 1 = 2^{b_1} + 2^{b_2} + \ldots + 2^{b_k}$, де $0 \leq b_1 < \ldots < b_k$. Далі визначимо послідовність чисел $a_1, \ldots, a_k$: $a_1 = b_1$, $a_{i+1} = b_{i+1} - b_i - 1$ для $1 < i < k$.

За числами $a_1, \ldots, a_k$ як за кодами команд МНР відновимо відповідні команди. Послідовність цих команд є шуканою МНР-програмою P.

МНР-програму з кодом n позначатимемо P_n.

Знайдемо, наприклад, $P = \varphi(5119)$. Маємо $5119 + 1 = 2^{10} + 2^{12}$, звідки $a_1 = 10$, $a_2 = 12 - 10 - 1 = 1$. Але $10 = 2 + 4 \cdot C(1,0)$, тому P має вигляд

1) $T(1, 0)$;
2) $S(0)$.

Приклад 4.1.2. Ефективну нумерацію всіх МТ задамо на основі кодування МТ.

Кожну МТ можна задати послідовністю її команд такою, що перша команда містить у лівій частині символ q_0, а остання — у правій частині символ q^*. Таке завдання неоднозначне, бо множину команд МТ можна впорядкувати у вигляді послідовності із зазначеною властивістю багатьма способами. Тому наша нумерація МТ неоднозначна.

Занумеруємо внутрішні стани МТ і символи стрічки.

Нехай $Q = \{q_1, \ldots, q_f\}$, $T = \{a_1, \ldots, a_m\}$.

Кодування μ команд МТ задамо так:

$$\mu(q_i a_j \to q_k a_l) = 3 \cdot C^4(i, j, k, l);$$
$$\mu(q_i a_j \to q_k a_l L) = 3 \cdot C^4(i, j, k, l) + 1;$$
$$\mu(q_i a_j \to q_k a_l R) = 3 \cdot C^4(i, j, k, l) + 2.$$

Таке μ є бієктивним відображенням множини всіх можливих команд машин Тьюрінга на N.

Використовуючи розглянуту вище бієкцію $\nu: \bigcup_{k\geq 1} N^k \to N$, визначаємо код МТ M, заданої послідовністю команд $I_1 I_2 ... I_k$: $\rho(M) = \nu(\mu(I_1), ..., \mu(I_k))$. Але ν та μ бієктивні, тому ρ також бієкція.

Тоді $\varphi = \rho^{-1}$ — шукана однозначна нумерація послідовностей команд МТ, але неоднозначна нумерація всіх МТ.

Неважко переконатись, що така нумерація ефективна.

Для прикладу знайдемо код МТ M, яка обчислює функцію $x + y$.

Нехай $a_0 = \lambda$, $a_1 = |$, $a_2 = \#$, $q^* = q_2$.

$q_0| \to q_0|R$; $\mu(q_0| \to q_0|R) = 3 \cdot C^4(0,1,0,1) + 2 = 3 \cdot C(2,1) + 2 = 3 \cdot 8 + 2 = 26$;

$q_0\# \to q_0|R$; $\mu(q_0\# \to q_0|R) = 3 \cdot C^4(0,2,0,1) + 2 = 3 \cdot C(9,1) + 2 = 3 \cdot 64 + 2 = 194$;

$q_0\lambda \to q_1\lambda L$; $\mu(q_0\lambda \to q_1\lambda L) = 3 \cdot C^4(0,0,1,0) + 1 = 3 \cdot C(1,0) + 1 = 3 \cdot 2 + 1 = 7$;

$q_1| \to q^*\lambda$; $\mu(q_1| \to q^*\lambda) = 3 \cdot C^4(1,1,2,0) = 3 \cdot C(25,0) = 3 \cdot 350 = 1050$.

Тепер $\tau(M) = \nu(26, 194, 7, 1050) = 2^{26} + 2^{221} + 2^{229} + 2^{1280} - 1$.

Приклад 4.1.3. Ефективну нумерацію ϕ усіх ЧРФ задамо на основі кодування γ операторних термів алгебри ЧРФ.

Завдання ЧРФ операторними термами неоднозначне, бо кожну ЧРФ можна описати нескінченною кількістю різних ОТ, тому нумерація ϕ неоднозначна.

Кодування γ операторних термів задамо так:

$\gamma(o) = 0$;

$\gamma(s) = 4$;

$\gamma(I^n_m) = 4 \cdot (C(n, m) - 2)$;

$\gamma(S^{n+1}(t_0, t_1, ..., t_n)) = 4 \cdot C(C^{n+1}(\gamma(t_0), \gamma(t_1), ..., \gamma(t_n)), n - 1) + 1$;

$\gamma(R(t_0, t_1)) = 4 \cdot C(\gamma(t_0), \gamma(t_1)) + 2$;

$\gamma(M(t)) = 4 \cdot \gamma(t) + 3$.

Число $n \in N$ вважаємо номером ЧРФ f, якщо n є кодом якогось ОТ, і значенням цього ОТ є функція f.

Числа, які не є кодами ОТ, і коди тих ОТ, які не задають ЧРФ, вважаємо номерами всюди невизначеної функції $f_\varnothing$.

Зрозуміло, що за кожною ЧРФ можна ефективно знайти її номер.

З іншого боку, за кожним $n \in N$ ефективно визначається ЧРФ f така, що $\phi(n) = f$. Справді, за n як за кодом будуємо ОТ; якщо ОТ з таким кодом існує та задає ЧРФ, то $\phi(n)$ — саме така ЧРФ f; якщо ОТ з таким кодом існує, але не задає ЧРФ, то $\phi(n) = f_{\varnothing}$; якщо ОТ з таким кодом не існує, то $\phi(n) = f_{\varnothing}$.

Отже, ϕ є ефективною нумерацією всіх ЧРФ.

Для прикладу знайдемо код ОТ алгебри n-арних ЧРФ для всюди невизначеної функції $f_{\varnothing}$.

Для $f_{\varnothing}$ маємо ОТ $M(s)$, звідки $\gamma(M(s)) = 4 \cdot \gamma(s) + 3 = 4 \cdot 4 + 3 = 19$.

Приклад 4.1.4. Ефективну нумерацію всіх ПРФ задамо на основі кодування операторних термів алгебри ПРФ. Така нумерація ПРФ неоднозначна.

Кодування π ОТ алгебри ПРФ задамо так:

$$\pi(o) = 0;$$
$$\pi(s) = 3;$$
$$\pi(I_m^n) = 3 \cdot (C(n, m) - 2);$$
$$\pi(S^{n+1}(t_0, t_1, ..., t_n)) = 3 \cdot C(C^{n+1}(\pi(t_0), \pi(t_1), ..., \pi(t_n)), n - 1) + 1;$$
$$\pi(R(t_0, t_1)) = 3 \cdot C(\pi(t_0), \pi(t_1)) + 2.$$

Число $n \in N$ вважаємо номером ПРФ f, якщо n є кодом якогось ОТ і значенням цього ОТ є функція f.

Числа, які не є кодами ОТ, і коди тих ОТ, які не задають ПРФ, вважаємо номерами функції **o**.

Приклад 4.1.5. Ефективну нумерацію всіх програмованих на N n-арних функцій задамо на основі кодування операторних термів ППА-*Ar*-*N*. Зрозуміло, що така нумерація неоднозначна.

Єдина істотна відмінність від нумерації прикладу 4.1.3 — інше кодування γ операторних термів ППА-*Ar*-*N*:

$$\gamma(o) = 0;$$
$$\gamma(s) = 4;$$
$$\gamma(+) = 8;$$
$$\gamma(\cdot) = 12;$$
$$\gamma(\div) = 16:$$
$$\gamma(I_m^n) = 4 \cdot (C(n, m) + 1);$$

$$\gamma(S^{n+1}(t_0, t_1, ..., t_n)) = 4 \cdot C(C^{n+1}(\gamma(t_0), \gamma(t_1), ..., \gamma(t_n)), n-1) + 1;$$
$$\gamma({}^{N}\Delta(t_0, t_1, t_2)) = 4 \cdot C^3(\gamma(t_0), \gamma(t_1), \gamma(t_2)) + 2;$$
$$\gamma({}^{N}\!\!\Box(t_0, t_1)) = 4 \cdot C(\gamma(t_0), \gamma(t_1)) + 3.$$

Приклад 4.1.6. Ефективну нумерацію φ^n усіх n-арних МНР-обчислюваних функцій задамо на основі кодування МНР-програм (див. приклад 4.1.1).

Номером функції f буде код МНР-програми, яка обчислює f.

Кожна МНР-програма визначає єдину функцію фіксованої арності, тому така нумерація буде нумерацією функцій фіксованої арності n. Але кожна n-арна МНР-обчислювана функція може бути задана нескінченною кількістю різних МНР-програм, тому така нумерація неоднозначна.

Приклад 4.1.7. Ефективну нумерацію всіх МНР-обчислюваних функцій можна ввести на основі кодування МНР-програм.

Номером n-арної функції f буде число $C(k, n)$, де k — код МНР-програми для f.

Приклад 4.1.8. Ефективну нумерацію всіх n-арних обчислюваних за Тьюрінгом функцій задамо на основі кодування МТ (див. приклад 4.1.2).

Номером функції f буде код МТ, яка обчислює f.

Кожна МТ визначає єдину функцію, якщо зазначено її арність, тому це нумерація функцій фіксованої арності. Кожна n-арна обчислювана за Тьюрінгом функція може бути задана нескінченною кількістю різних МТ, тому така нумерація неоднозначна.

Приклад 4.1.9. Ефективну нумерацію всіх обчислюваних за Тьюрінгом функцій можна ввести на основі кодування МТ.

Номером n-арної функції f буде число $C(k, n)$, де k — код МТ для f.

Через збіг класів ЧРФ, програмованих на N функцій, МНР-обчислюваних функцій і МТ-обчислюваних функцій, нумерації прикладів 4.1.5, 4.1.6, 4.1.8 можна розглядати як ефективні нумерації всіх n-арних ЧРФ для фіксованого n, а нумерації прикладів 4.1.3, 4.1.7, 4.1.9 — як ефективні нумерації всіх ЧРФ.

Зафіксуємо для кожного $n \geq 1$ деяку ефективну нумерацію n-арних ЧРФ. Найчастіше такою нумерацією буде нумерація на основі кодування МНР-програм (приклад 4.1.6). Інколи використовуватимемо нумерацію на основі кодування МТ (приклад 4.1.8).

Такі нумерації називають *стандартними нумераціями n-арних ЧРФ.*

Для стандартних нумерацій введемо такі позначення:

n-арну ЧРФ з номером m позначатимемо φ^n_m.

У випадку $n = 1$ вживатимемо також спрощене позначення φ_m.

Область визначення та область значень функції φ^n_m позначаємо відповідно D^n_m та E^n_m.

У випадку $n = 1$ вживатимемо також позначення D_m і E_m.

Номер функції називають також *індексом* функції.

Номер функції у стандартній нумерації називають *стандартним індексом* функції.

Маючи деякий індекс ЧРФ f, можна ефективно знайти як завгодно великий індекс тієї самої функції f:

Твердження 4.1.1. *Для кожної ЧРФ* φ^n_m *і для кожного* $j \in N$ *ефективно знаходиться* $k \in N$ *таке, що* $k > j$ *та* $\varphi^n_k = \varphi^n_m$.

Обмежимось розглядом випадку нумерації n-арних ЧРФ кодами МНР-програм.

У кінець заданої МНР-програми P_m допишемо j команд вигляду $T(0, 0)$. Нехай k — код отриманої у такий спосіб МНР-програми. Тоді $k > j$ і $\varphi^n_k = \varphi^n_m$ ■

Перехід від однієї з описаних вище нумерацій ЧРФ до іншої такої нумерації ефективний, тобто існує алгоритм, який за номером функції f в одній нумерації φ дає змогу знайти номер f в іншій нумерації ψ.

Наприклад, алгоритм переходу від нумерації прикладу 4.1.6 до нумерації прикладу 4.1.8 такий: за номером k як за кодом МНР-програми будуємо програму P_k; за P_k будуємо МТ, що задає ту ж функцію; код такої МТ і є шуканим номером у нумерації прикладу 4.1.8.

Нумерації з описаною вище властивістю називають гьоделевими.

Уточнимо поняття гьоделевої нумерації для n-арних ЧРФ.

Нумерація ξ *гьоделева*, якщо існують рекурсивні функції f і g такі:

- для кожного $m \in N$ $\varphi^n_m = \xi_{f(m)}$;
- для кожного $k \in N$ $\xi_k = \varphi^n_{g(k)}$.

Це означає, що існує пара алгоритмів, перший з яких за стандартним індексом функції знаходить її ξ-індекс, а другий за ξ-індексом функції знаходить її стандартний індекс.

Твердження 4.1.2. *Кожна гьоделева нумерація ефективна.*

Нехай нумерація ξ гьоделева. За ЧРФ, заданою стандартним індексом (кодом МНР-програми) m як φ^n_m, ефективно (використовуючи РФ *f)* знаходимо її ξ-індекс $f(m)$; за ξ-індексом k ефективно (використовуючи РФ *g)* знаходимо ЧРФ, задану кодом МНР-програми (стандартним індексом) $g(m)$ ■

Уведемо тепер важливе поняття обчислюваної нумерації.

Нехай **F** — деякий клас функцій вигляду $X \to Y$, для якого задана нумерація ξ: $N \to \mathbf{F}$.

З кожною такою нумерацією ξ пов'язана функція u: $N \times X \to Y$, що визначається умовою $u(n, x) = \xi_n(x)$.

Таку функцію u називають *спряженою* з нумерацією ξ.

Нумерація *обчислювана*, якщо спряжена з нею функція є ЧРФ.

Твердження 4.1.3. *Кожна гьоделева нумерація обчислювана.*

Справді, нехай нумерація ξ гьоделева, нехай g — РФ із визначення гьоделевої нумерації. Тоді спряжена до нумерації ξ функція $u(n, x) = \xi_n(x) = \varphi^n_{g(k)}$ є ЧРФ за тезою Чорча ■

Обернене твердження, узагалі, невірне [13]. Кожна обчислювана нумерація в певному сенсі зводиться до гьоделевої, водночас існують приклади негьоделевих обчислюваних нумерацій. Проте такі приклади досить неприродні, вони використовують дуже штучні конструкції.

4.2. Універсальні функції. Універсальна ЧРФ

Для довільного класу k-арних функцій **F** клас усіх функцій із **F** фіксованої арності n позначатимемо $\mathbf{F}^n$.

Функція $u(y, x_1, ..., x_n)$ *універсальна* для класу $\mathbf{F}^n$, якщо:

- для кожного значення m функція $u(m, x_1, ..., x_n) \in \mathbf{F}^n$;
- для кожної $f \in \mathbf{F}^n$ існує таке m, що $f(x_1, ..., x_n) = u(m, x_1, ..., x_n)$ для всіх значень $x_1, ..., x_n$.

Теорема 4.2.1. *Нехай* $\mathbf{T}$ *— деякий клас тотальних n-арних функцій на N, який містить функції* $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$ *і замкнений відносно суперпозиції. Нехай функція u універсальна для* $\mathbf{T}^n$. *Тоді* $u \notin \mathbf{T}^{n+1}$.

Припустимо супротивне: така $u \in \mathbf{T}$, тобто $u \in \mathbf{T}^{n+1}$. Визначимо функцію g таким чином: $g(x_1, ..., x_n) = u(x_1, x_1, ..., x_n) + 1$. Тоді $g \in \mathbf{T}^n$. Через універсальність функції u існує таке m, що для всіх значень $x_1, ..., x_n$ маємо $g(x_1, ..., x_n) = u(m, x_1, ..., x_n)$. Звідси $g(m, x_2, ..., x_n) = u(m, m, x_2, ..., x_n)$. Але $g(m, x_2, ..., x_n) = u(m, m, x_2, ..., x_n) + 1$ за визначенням функції g. Дістали суперечність, бо g і u тотальні ■

Наслідок 1. *Функція, універсальна для класу n-арних РФ, не є ЧРФ.*

Справді, така функція тотальна і не є РФ.

Наслідок 2. *Функція, універсальна для класу n-арних ПРФ, не є ПРФ.*

Теорема 4.2.2. *Існує РФ, універсальна для класу n-арних ПРФ.*

На основі розглянутої вище ефективної нумерації ПРФ задамо алгоритм для обчислення функції $u(y, x_1, ..., x_n)$, універсальної для класу n-арних ПРФ.

За n як за кодом ОТ алгебри ПРФ побудуємо відповідний ОТ і перевіримо, чи задає він n-арну ПРФ. Якщо ні, то видаємо значення 0 (тобто $u(y, x_1, ..., x_n)$ суть функція $\mathbf{o}^{n)}$. Якщо так, то обчислимо значення заданої цим термом функції над $x_1, ..., x_n$.

Отже, u — тотальна АОФ, за тезою Чорча вона є РФ ■

Наслідок 1. *Існує РФ, яка не є ПРФ.*

Наслідок 2. *Для відповідних класів функцій маємо строгі включення*

ПРФ $\subset$ РФ $\subset$ ЧРФ.

Теорема 4.2.3. *Існує ЧРФ, універсальна для класу n-арних ЧРФ.*

Розглянемо функцію $u(y, x_1, ..., x_n) = \varphi_y^n(x_1, ..., x_n)$. Вона є універсальною для класу n-арних ЧРФ. Справді,

- для кожного m функція $u(m, x_1, ..., x_n) = \varphi_m^n(x_1, ..., x_n)$ є ЧРФ;
- кожна n-арна ЧРФ f — суть функція φ_m^n для деякого m, тобто

$$f(x_1, ..., x_n) = \varphi_m^n(x_1, ..., x_n) = u(m, x_1, ..., x_n) \text{ для всіх } x_1, ..., x_n.$$

Задамо алгоритм для обчислення функції u.

За y як за кодом МНР-програми відновимо програму P_y для функції φ_C^n. Потім запустимо P_y над значеннями $x_1, ..., x_n$.

Отримане значення — це значення $\varphi_y^n(x_1, ..., x_n)$.

За тезою Чорча u — n-арна ЧРФ ■

МНР-програма, яка обчислює універсальну ЧРФ, називається універсальною МНР-програмою.

Універсальність зовсім не означає, що така програма містить усі програми для обчислення n-арних ЧРФ!

Універсальна програма вміє декодувати довільне число y у програму P_y, а далі вона моделює роботу P_y. Тому така універсальна програма може бути задана в явному вигляді.

Отже, можна ефективно знайти індекс k універсальної функції u у стандартній нумерації $(n + 1)$-арних ЧРФ, тобто u суть функція φ_k^{n+1}.

Машина Тьюрінга, яка обчислює універсальну ЧРФ, називається універсальною МТ.

Таку МТ, здатну моделювати роботу довільної МТ за її кодом, також можна задати в явному вигляді.

Універсальна МНР-програма та універсальна МТ є абстрактними моделями сучасних комп'ютерів. Вони реалізують в абстрактному вигляді *принцип програмного керування:* виконання заданої програми над заданими даними.

4.3. *s-m-n*-теорема

Нехай маємо $(m + n)$-арну ЧРФ $\varphi_z^{m+n}(x_1, ..., x_m, y_1, ..., y_n)$. Тоді для кожного фіксованого значення $a_1, ..., a_m$ аргументів $x_1, ..., x_m$ функція φ_z^{m+n} стає n-арною ЧРФ $\varphi_k^n(y_1, ..., y_n)$. Покажемо, що її індекс k може бути ефективно знайдений за z і $a_1, ..., a_m$. Це означає, що існує $(m + 1)$-арна РФ, яка обчислює цей індекс. Таку функцію традиційно позначають s_n^m, тому відповідне твердження називають s-m-n-теоремою.

Теорема 4.3.1 (s-m-n-теорема). *Для довільних* $m, n \geq 1$ *існує* $(m + 1)$*-арна РФ* $s_n^m(z, x_1, ..., x_m)$ *така, що для всіх* $z, x_1, ..., x_m, y_1, ..., y_n$ *маємо* $\varphi_z^{m+n}(x_1, ..., x_m, y_1, ..., y_n) = \varphi_{s_n^m(z, x_1, ..., x_m)}(y_1, ..., y_n)$.

Зафіксуємо значення z, $x_1, ..., x_m$. За значенням z визначимо МНР-програму P_z для функції φ_z^{m+n}.

Задамо МНР-програму Q, яка працює так. Перепишемо вміст регістрів з 0-го по $(n-1)$-й у регістри із m-го по $(m+n-1)$-й відповідно, а в регістри із 0-го по $(m-1)$-й запишемо відповідно $x_1, ..., x_m$. Далі виконуємо програму P_z. МНР-програма Q обчислює n-арну функцію f таку, що $f(y_1, ..., y_n) = \varphi_z^{m+n}(x_1, ..., x_m, y_1, ..., y_n)$ для всіх $y_1, ..., y_n$.

Код k програми Q ефективно обчислюється за $z, x_1, ..., x_m$, тому за тезою Чорча функція, яка обчислює цей код, рекурсивна. Позначимо таку функцію s_n^m. Але функція f суть φ_k^n, тому $k = s_n^m(z, x_1, ..., x_m)$ ■

Зауваження 1. Із доведення *s-m-n*-теореми випливає, що для всіх $x_1, ..., x_m \in N$ маємо $s_n^m(z, x_1, ..., x_m) > x_1 + ... + x_m$.

Зауваження 2. Твердження теореми 4.3.1 можна посилити, знявши залежність функції s_n^m від n. Це стає зрозумілим, коли для завдання ЧРФ використовувати МТ.

Справді, за z визначаємо МТ з кодом z для функції φ_z^{m+n}. Задамо нову МТ M, яка зліва від початкового вмісту стрічки дописує слово $|^{x_1} \# |^{x_2} \# ... \# |^{x_m}$, а далі моделює роботу МТ з кодом z. Така МТ M при вході $|^{y_1} \# |^{y_2} \# ... \# |^{y_n}$ обчислює n-арну функцію φ_k^n, причому k — код МТ M, — не залежить від n ■

s-m-n-Теорему часто формулюють у спрощеній формі.

Теорема 4.3.2. *Для кожної ЧРФ* $f(x_1, ..., x_m, y_1, ..., y_n)$ *існує РФ* $s(x_1, ..., x_m)$ *така*: $f(x_1, ..., x_m, y_1, ..., y_n) = \varphi_{s(x_1, ..., x_m)}^n(y_1, ..., y_n)$ *для всіх* $x_1, ..., x_m, y_1, ..., y_n$.

При $m = n = 1$ спрощена *s-m-n*-теорема формулюється інакше:

Теорема 4.3.3. *Для кожної ЧРФ* $f(x, y)$ *існує РФ* $s(x)$ *така, що для всіх значень* x, y *маємо* $f(x, y) = \varphi_{s(x)}(y)$.

Справді, f суть φ_b^2 для деякого b. Тому за теоремою 4.3.1 існує РФ s_1^1 така, що $\varphi_b^2(x, y) = \varphi_{s_1^1(b,x)}(y)$ для всіх значень x, y.

Звідси функція $s(x)$ суть функція $s_1^1(b, x)$ ■

Розглянемо приклади застосування *s-m-n*-теореми.

Приклад 4.3.1. Існує РФ $s(x, y)$ така: $\varphi_{s(x, y)}(z) = \varphi_x(z) + \varphi_y(z)$ для всіх $x, y, z \in N$.

Функція $f(x, y, z) = \varphi_x(z) + \varphi_y(z)$ є ЧРФ, тому за *s-m-n*-теоремою існує РФ $s(x, y)$: $f(x, y, z) = \varphi_{s(x, y)}(z) = \varphi_x(z) + \varphi_y(z)$ для всіх $x, y, z \in N$.

Приклад 4.3.2. Існує РФ $s(x)$ така, що для всіх $x \in N$ маємо $E_{s(x)} = D_x$.

Функція

$$f(x, y) = \begin{cases} y, \text{ якщо } y \in D_x, \\ \text{не визначене інакше,} \end{cases}$$

є ЧРФ за ТЧ; за *s-m-n*-теоремою існує РФ $s(x)$ така, що $f(x, y) = \varphi_{s(x)}(y)$ для всіх $x, y \in N$.

Зафіксуємо x. За побудовою функції f маємо $D_{s(x)} = E_{s(x)}$. Тепер отримаємо $y \in E_{s(x)} \Leftrightarrow y \in D_{s(x)} \Leftrightarrow \varphi_{s(x)}(y)\downarrow \Leftrightarrow f(x, y)\downarrow \Leftrightarrow y \in D_x$. Звідси $E_{s(x)} = D_x$.

Приклад 4.3.3. Існує РФ $t(x)$ така, що для всіх $x \in N$ маємо $D_{t(x)} = E_x$.

Функція

$$f(x, y) = \begin{cases} y, \text{ якщо } y \in E_x, \\ \text{не визначене інакше,} \end{cases}$$

є ЧРФ за ТЧ; за *s-m-n*-теоремою існує РФ $t(x)$ така: $f(x, y) = \varphi_{t(x)}(y)$ для всіх $x, y \in N$.

Зафіксуємо x. Маємо $y \in D_{t(x)} \Leftrightarrow \varphi_{t(x)}(y)\downarrow \Leftrightarrow f(x, y)\downarrow \Leftrightarrow y \in E_x$. Тому $D_{t(x)} = E_x$.

Приклад 4.3.4. Існує РФ $s(x, y)$ така: $D_{s(x, y)} = D_x \cup D_y$ для всіх $x, y \in N$.

Функція

$$f(x, y, z) = \begin{cases} z, \text{ якщо } z \in D_x \text{ або } z \in D_y, \\ \text{не визначене інакше,} \end{cases}$$

є ЧРФ за ТЧ. За *s-m-n*-теоремою існує РФ $s(x, y)$ така: $f(x, y, z) = \varphi_{s(x, y)}(z)$ для всіх $x, y, z \in N$.

Зафіксуємо x і y. Маємо $z \in D_{s(x,y)} \Leftrightarrow \varphi_{s(x,y)}(z)\downarrow \Leftrightarrow f(x, y, z)\downarrow \Leftrightarrow z \in D_x \cup D_y$. Звідси випливає $D_{s(x, y)} = D_x \cup D_y$.

Приклад 4.3.5. Існує РФ $s(x, y)$ така: $D_{s(x, y)} = E_{s(x, y)} = D_x \cap D_y$ для всіх $x, y \in N$.

Функція

$$f(x, y, z) = \begin{cases} z, \text{ якщо } z \in D_x \text{ та } z \in D_y \\ \text{не визначене інакше,} \end{cases}$$

є ЧРФ за ТЧ. За s-m-n-теоремою існує РФ $s(x, y)$ така, що $f(x, y, z) = \varphi_{s(x, y)}(z)$ для всіх $x, y, z \in N$.

Зафіксуємо x і y. За побудовою функції f маємо $D_{s(x, y)} = E_{s(x, y)}$. Тепер $z \in E_{s(x, y)} \Leftrightarrow z \in D_{s(x, y)} \Leftrightarrow \varphi_{s(x, y)}(z)\downarrow \Leftrightarrow f(x, y, z)\downarrow \Leftrightarrow z \in D_x \cap D_y$. Звідси $D_{s(x, y)} = E_{s(x, y)} = D_x \cap D_y$.

Приклад 4.3.6. Для кожної 1-арної ЧРФ f існує РФ $s(x)$ така, що для всіх $x \in N$ маємо $D_{s(x)} = f^{-1}(D_x)$.

Функція $g(x, y) = \varphi_x(f(y))$ є ЧРФ за ТЧ, тому за s-m-n-теоремою існує РФ $s(x)$ така: $g(x, y) = \varphi_{s(x)}(y)$ для всіх $x, y \in N$. Зафіксуємо x. Маємо $y \in D_{s(x)} \Leftrightarrow \varphi_{s(x)}(y)\downarrow \Leftrightarrow g(x, y)\downarrow \Leftrightarrow \varphi_x(f(y))\downarrow \Leftrightarrow f(y) \in D_x \Leftrightarrow y \in f^{-1}(D_x)$. Звідси $D_{s(x)} = f^{-1}(D_x)$.

4.4. Теореми Кліні про нерухому точку

Розглянемо теореми про нерухому точку для рекурсивних функцій. Такі теореми є твердженнями про індекси обчислюваних функцій, їх доведення може розглядатися як розвиток діагонального методу, воно фактично використовує тільки s-m-n-теорему та обчислюваність універсальної ЧРФ.

Теорема 4.4.1. *Нехай f — $(n + 1)$-арна РФ. Тоді існує n-арна РФ g така, що $\varphi_{g(x_1,\ldots,x_n)} = \varphi_{f(g(x_1,\ldots,x_n),x_1,\ldots,x_n)}$ для всіх значень $x_1, \ldots, x_n$.*

За s-m-n-теоремою існує РФ $s(u, x_1, \ldots, x_n)$ така, що для всіх $u, x_1, \ldots, x_n, y$ маємо

$$\varphi_{\varphi_u^{n+1}(u, x_1, \ldots, x_n)}(y) = \varphi_{s(u, x_1, \ldots, x_n)}(y) \qquad (1)$$

Нехай функція $f(s(u, x_1, \ldots, x_n), x_1, \ldots, x_n)$ має індекс k у нумерації $(n + 1)$-арних ЧРФ, тобто це функція $\varphi_k^{n+1}(u, x_1, \ldots, x_n)$. За тотальністю

функцій f і s функція φ_k^{n+1} тотальна. Отже, при $u = k$ для всіх $x_1, ..., x_n$ маємо

$$f(s(u, x_1, ..., x_n), x_1, ..., x_n) = \varphi_k^{n+1}(k, x_1, ..., x_n). \quad (2)$$

Із (1) при $u = k$ та із (2) для всіх $x_1, ..., x_n$ маємо

$$\varphi_{f(s(x_1,...,x_n),\, x_1,...,x_n)} = \varphi_{\varphi_k^{n+1}(k,\, x_1,\, ...,\, x_n)} = \varphi_{s(k,\, x_1,\, ...,\, x_n)}.$$

Тому $g(x_1, ..., x_n) = s(k, x_1, ..., x_n)$ — це шукана РФ ■

Для випадку $n = 0$ теорема 4.4.1 переформульовується так:

Теорема 4.4.2. *Нехай $f(x)$ — РФ. Тоді існує $n \in N$ таке, що* $\varphi_n = \varphi_{f(n)}$.

Із теореми 4.4.2 маємо кілька цікавих наслідків.

Наслідок 1. *Нехай $f(x)$ — РФ. Тоді існує $n \in N$ таке*: $D_n = D_{f(n)}$ *і* $E_n = E_{f(n)}$.

Справді, згідно з теоремою 4.4.2 візьмемо $n \in N$ таке, що $\varphi_n = \varphi_{f(n)}$ ■

Наслідок 2. *Нехай $h(z, x)$ — ЧРФ. Тоді існує $n \in N$ таке, що для всіх x маємо $h(n, x) = \varphi_n(x)$.*

За *s-m-n*-теоремою існує РФ $s(z)$ така, що $h(z, x) = \varphi_{s(z)}(x)$ для всіх z, x. За теоремою 4.4.2 існує таке n, що $\varphi_n = \varphi_{s(n)}$, тобто для всіх x маємо $h(n, x) = \varphi_{s(n)}(x) = \varphi_n(x)$ ■

Формулювання наслідку 2 — це первісне формулювання самого С. Кліні теореми про нерухому точку.

Покажемо, що наслідок 2 і теорема 4.4.2 еквівалентні. Для цього виведемо теорему 4.4.2 із наслідку 2.

Нехай $f(x)$ — РФ. За тезою Чорча функція $h(z, x) = \varphi_{g(z)}(x)$ є ЧРФ. За наслідком 2 існує $n \in N$ таке, що $h(n, x) = \varphi_n(x)$ для всіх x, тобто для всіх x $h(n, x) = \varphi_{g(n)}(x) = \varphi_n(x)$ ■

Теорему Кліні про нерухому точку краще називати теоремою про псевдонерухому точку.

Справді, умова $\varphi_n = \varphi_{f(n)}$ зовсім не означає, що $n = f(n)$, а свідчить тільки про те, що n і $f(n)$ — індекси тієї самої ЧРФ.

Теорему про нерухому точку називають також теоремою про рекурсію, бо вона виражає рекурсивне визначення найзагальнішого вигляду.

Наприклад, визначимо функцію φ_n через задану РФ f так: $\varphi_n = \varphi_{f(n)}$. Тоді φ_n ефективно визначена через n — код МНР-програми для її обчислення, бо таке n існує згідно з теоремою 4.4.2.

МНР-програму P називають *самотворною*, якщо для довільного $x \in N$ маємо $P(x)\downarrow\tau(P)$, де $\tau(P)$ — код програми P.

На перший погляд, таких програм бути не може, бо для побудови P треба знати $\tau(P)$, тобто саму програму P.

Проте самотворні програми існують!

Теорема 4.4.3. *Існує МНР-програма P така, що $P(x)\downarrow\tau(P)$ для всіх $x \in N$.*

Візьмемо функцію $h(z, x) = z$. За наслідком 2 існує таке n, що для всіх x $h(n, x) = \varphi_n(x)$. Отже, $\varphi_n(x) = h(n, x) = n$ для всіх x. Тому програма P з кодом n шукана ■

Покажемо тепер, що нерухома точка кожної РФ φ_n ефективно залежить від її індексу n. Це посилює твердження теореми 4.4.2.

Теорема 4.4.4. *Існує РФ $\alpha(x)$ така: для кожного $n \in N$ якщо $\varphi_n \in$ РФ, то $\varphi_{\alpha(n)} = \varphi_{\varphi_n(\alpha(n))}$.*

За s-m-n-теоремою існує РФ $s(x)$ така:

$$\varphi_{\varphi_x(x)}(y) = \varphi_{s(y)}(y) \text{ для всіх } x, y \in N. \tag{1}$$

Тоді за s-m-n-теоремою існує РФ $v(x)$ така:

$$\varphi_n(s(x)) = \varphi_{v(n)}(x) \text{ для всіх } n, x \in N \tag{2}$$

Якщо φ_n є РФ, то кожне значення $\varphi_n(x)$ визначене.

Узявши $x = v(n)$, із (1) маємо

$$\varphi_{\varphi_{v(n)}(v(n))} = \varphi_{s(v(n))} \tag{3}$$

Із (2) маємо $\varphi_n(s(v(n))) = \varphi_{v(n)}(v(n))$.

Звідси та із (3) дістаємо $\varphi_{s(v(n))} = \varphi_{\varphi_{v(n)}(v(n))}$.

Поклавши $\alpha(x) = s(v(x))$, маємо $\varphi_{\alpha(n)} = \varphi_{\varphi_n(\alpha(n))}$ ■

Покажемо, що для кожної РФ можна ефективно знайти монотонно зростаючу послідовність її нерухомих точок.

Звідси, зокрема, випливає нескінченність множини нерухомих точок кожної РФ.

Теорема 4.4.5. *Для кожної РФ $f(x)$ існує строго монотонна РФ $\alpha(x)$ така, що для кожного $n \in N$ маємо* $\varphi_{\alpha(n)} = \varphi_{f(\alpha(n))}$.

За *s-m-n*-теоремою існує РФ $s(x)$ така: $\varphi_{f(\varphi_x(x))}(y) = \varphi_{s(x)}(y)$ для всіх $x, y \in N$. Нехай m — деякий індекс функції $s(x)$, тобто $s(x)$ суть $\varphi_m(x)$.

Звідси $\varphi_{f(\varphi_m(m))}(y) = \varphi_{s(m)}(y) = \varphi_{\varphi_m(m)}(y)$ для всіх $y \in N$, тобто $\varphi_m(m)$ — нерухома точка функції f.

Згідно із зауваженням 1 до *s-m-n*-теореми $s(x) \geq x$ для всіх $x \in N$, тому $\varphi_m(m) \geq m$.

Функцію $\alpha(x)$ задамо так:

- $\alpha(0) = \varphi_{m_0}(m_0)$, де m_0 — довільний індекс функції s;
- $\alpha(k+1) = \varphi_{m_{k+1}}(m_{k+1})$, де m_{k+1} — такий індекс функції s, що $m_{k+1} > \alpha(k)$ (за твердженням 4.1.1, m_{k+1} можна знайти ефективно).

Кожне значення функції α є нерухомою точкою функції f.

Функція α алгоритмічно обчислювана, вона тотальна через тотальність функції s. Згідно з ТЧ функція α є РФ, причому для всіх $k \in N$ маємо $\alpha(k+1) = \varphi_{m_{k+1}}(m_{k+1}) \geq m_{k+1} > \alpha(k)$. Тому α — строго монотонна РФ ■

Наслідок. *Для кожної РФ $f(x)$ і для кожного $k \in N$ існує $n \in N$ таке, що $n > k$ і $\varphi_n = \varphi_{f(n)}$.*

Розглянуті ефективні нумерації ЧРФ неоднозначні. Однозначні ефективні нумерації ЧРФ існують [9], але немає в певному сенсі "природних" однозначних ефективних нумерацій ЧРФ.

Теорема 4.4.6. *Нехай $f(x)$ — строго монотонна тотальна функція така*:

- *якщо $m \neq n$, то $\varphi_{f(m)} \neq \varphi_{f(n)}$;*
- *$f(n)$ — найменший індекс функції $\varphi_{f(n)}$.*

Тоді функція f не є ЧРФ.

Функція f не може бути тотожною, бо тоді із умови $m \neq n$ випливає $\varphi_m \neq \varphi_n$. Тому існує таке $k \in N$, що $f(n) > n$ при $n \geq k$. Але $f(n)$ — найменший індекс функції $\varphi_{f(n)}$, тому $\varphi_{f(m)} \neq \varphi_{f(n)}$ для всіх $n \geq k$. Якщо f рекурсивна, то за наслідком теореми 4.4.5 існує $n \in N$ таке, що $n > k$ і $\varphi_{f(n)} = \varphi_n$. Дістали суперечність, тому функція f не є РФ та не є ЧРФ ■

Розглянемо приклади застосування теореми Кліні про нерухому точку.

Приклад 4.4.1. Існує $n \in N$ таке, що для всіх x маємо $\varphi_n(x) = 2n + x^{3n}$.

Візьмемо функцію $h(z, x) = 2z + x^{3z}$. За теоремою 4.4.3 існує таке n, що для всіх x маємо $h(n, x) = \varphi_n(x)$. Отже, $\varphi_n(x) = h(n, x) = 2n + x^{3n}$ для всіх x.

Приклад 4.4.2. Існує $n \in N$ таке, що $D_n = E_n = \{n\}$.

Візьмемо функцію

$$h(z, x) = \begin{cases} x, \text{ якщо } x = z, \\ \text{не визначене інакше.} \end{cases}$$

Така h є ЧРФ. За теоремою 4.4.3 існує таке n, що для всіх x маємо $h(n, x) = \varphi_n(x)$. Тоді

$$\varphi_n(x) = \begin{cases} x, \text{ якщо } x = n, \\ \text{не визначене інакше.} \end{cases}$$

Звідси $D_n = E_n = \{n\}$.

Приклад 4.4.3. Існує РФ $g(x)$ така: $D_{g(x)} = E_{g(x)} = \{3g(x) + 2^x\}$ для всіх $x \in N$.

Функція

$$h(t, x, y) = \begin{cases} y, \text{ якщо } y = 3t + 2^x, \\ \text{не визначене інакше,} \end{cases}$$

є ЧРФ за ТЧ. За s-m-n-теоремою існує РФ $s(t, x)$ така: $h(t, x, y) = \varphi_{s(t,x)}(y)$ для всіх $t, x, y \in N$. За теоремою 4.4.1 існує РФ g така, що $\varphi_{g(x)} = \varphi_{s(g(x), x)}$ для всіх $x \in N$. Тоді

$$\varphi_{g(x)}(y) = \varphi_{s(g(x), x)}(y) = h(g(x), x, y) = \begin{cases} y, \text{ якщо } y = 3g(x) + 2^x, \\ \text{не визначене інакше.} \end{cases}$$

Тому для кожного $x \in N$ маємо $D_{g(x)} = E_{g(x)} = \{3g(x) + 2^x\}$.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Як задають кодування та нумерацію МНР програм?
2. Як задають кодування та нумерацію МТ?
3. Опишіть кодування ОТ алгебри n-арних ЧРФ.
4. Опишіть кодування ОТ алгебри n-арних ПРФ.
5. Опишіть кодування ОТ ППА-Ar-N.
6. Як задати ефективну нумерацію всіх МНР-обчислюваних функцій фіксованої арності n?
7. Як задати ефективну нумерацію всіх МТ-обчислюваних функцій фіксованої арності n?
8. Як задати ефективну нумерацію всіх n-арних ЧРФ?
9. Як задати ефективну нумерацію всіх n-арних ПРФ?
10. Які нумерації n-арних ЧРФ вважають стандартними?
11. Що таке стандартний індекс ЧРФ?
12. Дайте визначення гьоделевої нумерації n-арних ЧРФ.
13. Дайте визначення спряженої з нумерацією функції.
14. Дайте визначення обчислюваної нумерації.
15. Дайте визначення універсальної функції.
16. Сформулюйте теореми про універсальні функції.
17. Що таке універсальна ЧРФ?
18. Що таке універсальна МНР-програма?
19. Що таке універсальна МТ?
20. Опишіть принцип роботи універсальної МНР-програми.
21. Як пов'язані універсальні алгоритмічні машини з програмуванням?
22. Сформулюйте s-m-n-теорему в загальному вигляді.
23. Сформулюйте s-m-n-теорему у спрощеній формі.
24. Сформулюйте теорему Кліні про нерухому точку для РФ.
25. Наведіть первісне формулювання С. Кліні теореми про нерухому точку.
26. Що таке самотворна МНР-програма?
27. Чи існують РФ із скінченними множинами нерухомих точок?
28. Чи існують "природні" однозначні ефективні нумерації ЧРФ? Як уточнити "природність"?

ВПРАВИ

1. Наведіть МНР-програму та її код для функцій:
   1) $f(x) = 2x$;

2) $f(x, y) = x - y$;
3) $f(x) = x/2$;
4) $f(x) = nsg(x)$;
5) $f(x, y) = sg(x + y)$;
6) $f(x, y) = max(x, y)$.

2. Наведіть всі МНР-програми з кодами від 0 до 100 включно.

3. Наведіть МТ та її код для функцій і предикатів:
1) $f(x) = sg(x)$;
2) $f(x, y) = nsg(x + y)$;
3) $f(x) = sg(x/2)$;
4) "x — парне число".

4. Наведіть приклад однозначної нумерації n-арних ЧРФ.

5. Доведіть існування таких РФ s:
1) $D^3_{s(x)} = \{(u, v, w) \mid x = u^2 + v^2 + w^2\}$ для всіх $x \in N$;
2) $E_{s(x, y)} = D_x \cup E_y$ для всіх $x, y \in N$;
3) $D_{s(x, y)} = E_{2x} \cap D_{3y}$ для всіх $x, y \in N$;
4) $E_{s(x, y)} = (D_{3x} \cap E_{2y}) \cup \{x, y\}$ для всіх $x, y \in N$;
5) $D_{s(x, y, z)} = (D_x \cup E_y) \cap D_z$ для всіх $x, y, z \in N$;
6) $E_{s(x, y)} = f^{-1}(E_x \cap D_y)$ для всіх $x, y \in N$ (тут f — фіксована ЧРФ);
7) $D_{s(x, y)} = f(E_y \cup D_x)$ для всіх $x, y \in N$ (тут f — фіксована ЧРФ).

6. Спробуйте побудувати самотворну МНР-програму.

7. Доведіть, що існує $n \in N$ таке:
1) $D_n = \{x \mid \varphi_n(x)\downarrow\} \cap \{x \mid x$ є простим числом$\}$;
2) $Dn = E_n = N \setminus \{1, 2, 3, ..., n\}$;
3) $D_n = E_n = \{n, 2n, 3n, ..., n^2\}$;
4) $D_n = E_n = \{x \mid x$ є парним числом$\} \setminus \{2n, 4n, 6n\}$;
5) $E_n = \{x \mid x$ є простим числом$\} \cup \{2n, 3n, 5n\}$.

8. Доведіть, що існує РФ $g(x)$ така, що для всіх $x \in N$ маємо:
1) $D_{g(x)} = E_{g(x)} = \{2g(x) + x^3\}$;
2) $E_{g(x)} = \{5x + (2 + x)g(x)\}$;
3) $D_{g(x)} = E_{g(x)} = \{x \cdot g(x) + 3^x + 2\}$.

9. Доведіть, що для кожних РФ $g(x)$ і $m \in N$ існує $n \in N$ таке, що справджується $D_n = D_m \cup \{g(n)\}$.

10. Чи існують $m, n \in N$ такі, що $m \neq n$, $D_m = \{n\}$ і $D_n = \{m\}$?

11. Збудуйте нескінченну послідовність попарно різних натуральних чисел $n_0, n_1, \ldots$ таку, що для всіх $i \in N$ маємо $D_{n_{i+1}} = \{n_i\}$.

12. Збудуйте нескінченну послідовність попарно різних натуральних чисел $n_0, n_1, \ldots$ таку, що для всіх $i \in N$ маємо $D_{n_i} = \{n_{i+1}\}$.

13. Чи завжди існує спільна нерухома точка для двох довільних РФ?

14. Доведіть твердження (теорема про парну рекурсію): для кожної пари РФ $f(x)$ і $g(x)$ існують $m, n \in N$ такі, що $\varphi_m = \varphi_{f(C(m, n))}$ і $\varphi_n = \varphi_{g(C(m, n))}$.

5. РОЗВ'ЯЗНІСТЬ, ЧАСТКОВА РОЗВ'ЯЗНІСТЬ, НЕРОЗВ'ЯЗНІСТЬ

Перенесемо введені раніше для функцій поняття примітивної рекурсивності, рекурсивності та часткової рекурсивності на класи множин і предикатів.

5.1. Примітивно рекурсивні, рекурсивні, рекурсивно перелічні множини

Множину $M \subseteq N^n$ називають *рекурсивною* (РМ), якщо її характеристична функція χ_M рекурсивна.

Множину $M \subseteq N^n$ називають *примітивно рекурсивною* (ПРМ), якщо її характеристична функція χ_M примітивно рекурсивна.

Зрозуміло, що кожна ПРМ є рекурсивною множиною.

Множину $M \subseteq N$ називають *рекурсивно перелічною* (РПМ), якщо $M = \varnothing$ або $M = E_f$ для деякої рекурсивної функції f.

Множину $M \subseteq N^n$ називають РПМ, якщо $M = \varnothing$ або існують 1-арні РФ $g_1, ..., g_n$ такі, що $M = \{(g_1(x), ..., g_n(x)) \mid x \in N\}$.

Як наслідки тези Чорча дістаємо такі твердження:

- клас РМ збігається з класом алгоритмічно розв'язних множин натуральних чисел;
- клас РПМ збігається з класом алгоритмічно перелічних множин натуральних чисел.

Для кожної $L \subseteq N^n$ визначимо множину-згортку

$$C^n(L) = \{C^n(x_1, ..., x_n) \mid (x_1, ..., x_n) \in L\}.$$

Нехай $L \subseteq N^n$ та $M \subseteq N^n$. Безпосередньо із визначень випливає:

$$C^n(L \cup M) = C^n(L) \cup C^n(M),$$
$$C^n(L \cap M) = C^n(L) \cap C^n(M),$$
$$C^n(N^n \setminus L) = N \setminus C^n(L).$$

Використовуючи множини-згортки, дамо еквівалентні визначення РМ, ПРМ і РПМ, заданих на N^n. Це дозволяє обмежитися розглядом рекурсивних, примітивно рекурсивних і рекурсивно перелічних множин на N.

Теорема 5.1.1. *Множина $L \subseteq N^n$ є РМ (ПРМ, РПМ) $\Leftrightarrow$ множина $C^n(L)$ є РМ (ПРМ, РПМ).*

Доводимо для випадку РМ (для ПРМ доведення аналогічне).

Нехай $L \subseteq N$ є РМ, тобто функція $\chi_L(x_1, ..., x_n)$ є РФ. Тоді маємо $\chi_{C^n(L)}(x) = \chi_L(C_{n1}(x), ..., C_{nn}(x))$. Справді, $x \in C^n(L) \Leftrightarrow x = C^n(x_1, ..., x_n)$ для деякої n-ки $(x_1, ..., x_n) \in L \Leftrightarrow (C_{n1}(x), ..., C_{nn}(x)) \in L$.

Отже, $C^n(L)$ є РМ.

Нехай тепер $C^n(L)$ є РМ, тобто функція $\chi_{C^n(L)}$ — РФ. Тоді $\chi_L(x_1, ..., x_n) = \chi_{C^n(L)}(C^n(x_1, ..., x_n))$, звідки L є РМ.

Доводимо для РПМ.

Нехай $L \subseteq N^n$ є РПМ, тоді $L = \{(g_1(x), ..., g_n(x)) \mid x \in N\}$ для деяких РФ $g_1, ..., g_n$. Тоді $C^n(L) = \{C^n(g_1(x), ..., g_n(x)) \mid x \in N\}$ є РПМ, бо $C^n(g_1(x), ..., g_n(x))$ є РФ.

Нехай тепер $C^n(L)$ є РПМ, тобто $C^n(L) = \{g(x) \mid x \in N\}$ для деякої РФ g. Тоді $L = \{(C_{n1}(g(x)), ..., C_{nn}(g(x))) \mid x \in N\}$ є РПМ, бо $C_{n1}(g(x)), ..., C_{nn}(g(x))$ — РФ ■

Співвідношення між класами ПРМ, РМ і РПМ встановлює теорема:

Теорема 5.1.2. 1) *кожна скінченна множина є ПРМ*;

2) *кожна рекурсивна множина є РПМ*;

3) *клас ПРМ строго включається в клас РМ*.

1. Нехай $L = \{a_1, ..., a_n\}$. Тоді $\chi_L(x) = nsg((\prod_{i=0}^{n} | x - a_i |))$ — ПРФ.

2. Нехай $L \subseteq N$ є РМ. Якщо $L = \varnothing$, то L, за визначенням, є РПМ. Якщо $L \neq \varnothing$, то зафіксуємо якийсь елемент $b \in L$. Функція χ_L рекурсивна, тому $f(x) = x \cdot \chi_L(x) + b \cdot nsg(\chi_L(x))$ також рекурсивна, причому $L = E_f$. Отже, L є РПМ.

3. Кожна ПРФ рекурсивна, тому кожна ПРМ є РМ. Нехай $u(t, x)$ — рекурсивна універсальна функція для ПРФ[1]. Тоді $f(x) = nsg(u(x, x))$ — характеристична функція деякої РМ L. Якщо $f(x)$ є ПРФ, то за універсальністю $u(t, x)$ існує $k \in N$ таке: $f(x) = u(k, x)$ для всіх x. Тоді $f(k) = u(k, k) = nsg(u(k, k))$. Маємо суперечність, тому $f(x)$ не є ПРФ, звідки L не є ПРМ ■

Теорема 5.1.3. *Класи ПРМ і РМ замкнені відносно операцій $\cup$, $\cap$ та доповнення.*

Нехай χ_A і χ_B є РФ (ПРФ).

Маємо $\chi_{A\cup B}(x) = sg(\chi_A(x) + \chi_B(x))$, $\chi_{A\cap B}(x) = \chi_A(x)\cdot\chi_B(x)$, $\chi_{\bar{A}}(x) = nsg(\chi_A(x))$. Тому $\chi_{A\cup B}$, $\chi_{A\cap B}$, $\chi_{\bar{A}}$ є РФ (ПРФ) ■

Теорема 5.1.4. *Множина L є нескінченною РМ $\Leftrightarrow$ $L = E_f$ для деякої строго монотонної РФ f.*

Для строго монотонних функцій для всіх $x \in D_f$ виконується умова $f(x) \geq x$. Тому $\chi_{E_f}(x) = nsg(\prod_{k=0}^{x} |x - f(k)|)$. Якщо f рекурсивна, то χ_{E_f} також рекурсивна, звідки E_f є РМ.

Нехай L є нескінченною РМ. Задамо функцію f такою схемою примітивної рекурсії:

$f(0) = \mu_k(\chi_L(k) = 1)$;

$f(x + 1) = \mu_k(\chi_L(k) = 1$ і $k > f(x))$.

За побудовою $L = E_f$ і f строго монотонна. За нескінченністю множини L функція f тотальна. Отже, f — строго монотонна РФ ■

Теорема 5.1.5. *Нехай L — нескінченна РПМ. Тоді існує нескінченна рекурсивна множина M така, що $M \subseteq L$.*

Нехай L — нескінченна РПМ. Тоді $L = E_g$ для деякої РФ g.

Розглянемо функцію f, задану такою схемою примітивної рекурсії:

$f(0) = g(0)$;

$f(x + 1) = g(\mu_k(g(k) > f(x))$.

За побудовою функція f строго монотонна, причому f тотальна через нескінченність E_g. Отже, f — строго монотонна рекурсивна функція, тому E_f — некінченна РМ. Зрозуміло, що $E_f \subseteq E_g = L$, тому E_f — шукана множина M ■

Теорема 5.1.6. *Нехай L — нескінченна РПМ. Тоді існує ін'єктивна РФ f така, що $L = E_f$.*

Маємо $L = E_g$ для деякої РФ g.

Розглянемо функцію f, задану такою схемою примітивної рекурсії:

$f(0) = g(0)$;

$$f(x+1) = g(\mu_k(g(k)\neq f(0), \ldots, g(k)\neq f(x))) =$$
$$= g(\sum_{i=0}^{x} \mu_k(nsg(|g(k) - f(i)| = 0))).$$

Через нескінченність E_g функція f ін'єктивна й тотальна, причому $E_f = E_g = L$. Отже, f — шукана ін'єктивна РФ ■

Теорема 5.1.7. *Існує РФ* α *така, що для кожного* $x \in N$ $E_{\alpha(x)} = D_x$, *причому* $\varphi_{\alpha(x)}$ *є РФ при* $D_x \neq \varnothing$.

Зафіксуємо довільне $x \in N$.

Задамо ефективний процес поетапного породження множини D_x, формуючи список елементів D_x з повторами.

Виконання однієї команди МНР-програми (команди МТ) при обчисленні певної ЧРФ називають кроком обчислення.

Етап 1. Робимо 1-й крок обчислення $\varphi_x(0)$; якщо при цьому $\varphi_x(0)$ обчислено, то заносимо 0 до списку.

Етап $n+1$. Робимо по $n+1$ кроків обчислення для $\varphi_x(0)$, $\varphi_x(1)$, ..., $\varphi_x(n)$; усі такі $k \le n$, для яких $\varphi_x(k)$ обчислено, заносимо до списку.

Задамо функцію $f(x, y)$ так.

Для кожного фіксованого $x \in N$ покладемо:

$f(x, 0)$ є першим елементом списку;

$$f(x, y+1) =$$
$$\begin{cases} \mu_z(z \text{ занесено до списку на етапі } y+1 \text{ і } \neq f(x,0), \ldots, \neq f(x,y)), \\ \text{якщо таке } z \text{ існує;} \\ f(x,\ 0)\ , \text{ таке } z \text{ не існує.} \end{cases}$$

За тезою Чорча так задана $f(x, y)$ є ЧРФ. Тому за s-m-n-теоремою існує така РФ $\alpha(x)$, що $f(x, y) = \varphi_{\alpha(x)}(y)$ для всіх значень x, y.

За побудовою $E_{\alpha(x)} = D_x$.

Якщо $D_x = \varnothing$, то $\varphi_{\alpha(x)} = f_\varnothing$.

Якщо $D_x \neq \varnothing$, то при такому фіксованому x функція $f(x, y)$ визначена для всіх $y \in N$, тому $\varphi_{\alpha(x)}$ тотальна, отже, функція $\varphi_{\alpha(x)}$ є РФ ■

Теорема 5.1.8. *Існують РФ* s *і* t *такі*: *для кожного* $x \in N$ $E_{s(x)} = D_x$ *та* $D_{t(x)} = E_x$.

Задамо функцію

$$f(x, y) = \mu_z(P_x(z)\downarrow y) = \begin{cases} \text{визначене, якщо } y \in E_x, \\ \text{не визначене, якщо } y \notin E_x. \end{cases}$$

За ТЧ $f(x, y)$ є ЧРФ. Тому за *s-m-n*-теоремою існує така РФ $t(x)$, що $f(x, y) = \varphi_{t(x)}(y)$ для всіх значень x, y. Тоді $y \in E_x \Leftrightarrow f(x, y)$ визначене $\Leftrightarrow$ $\varphi_{t(x)}(y)$ визначене $\Leftrightarrow y \in D_{t(x)}$. Звідси $D_{t(x)} = E_x$.

Задамо функцію $g(x, y) = \begin{cases} y, & \text{якщо } y \in D_x, \\ \text{не визначене}, & \text{якщо } y \notin D_x. \end{cases}$

За ТЧ $g(x, y)$ є ЧРФ. Тому за *s-m-n*-теоремою існує така РФ $s(x)$, що $g(x, y) = \varphi_{s(x)}(y)$ для всіх значень x, y. За побудовою $E_{s(x)} = D_{s(x)}$.

Маємо $y \in D_x \Leftrightarrow f(x, y)$ визначене $\Leftrightarrow \varphi_{s(x)}(y)$ визначене $\Leftrightarrow y \in D_{s(x)} \Leftrightarrow y \in E_{s(x)}$. Звідси $D_x = E_{s(x)}$ ■

Теорема 5.1.9. *Наступні визначення РПМ еквівалентні*:

*df*1) $L = \varnothing$ *або* L *є областю значень деякої РФ*;

*df*2) L *є областю значень деякої ЧРФ*;

*df*3) L *є областю визначення деякої ЧРФ*;

*df*4) *часткова характеристична функція множини* L *є ЧРФ*.

Імплікації $df1 \Rightarrow df2$ та $df4 \Rightarrow df3$ є очевидними.

Покажемо $df3 \Rightarrow df4$. Нехай $L = D_f$ для деякої ЧРФ f. Тоді маємо $\chi^{\text{ч}}_L(x) = \boldsymbol{s}(\mathbf{o}(f(x)))$.

Тепер покажемо $df3 \Rightarrow df1$. Нехай множина L є областю визначення деякої ЧРФ, нехай x — індекс такої ЧРФ, тобто $L = D_x$.

Візьмемо РФ $\alpha(x)$ із теореми 5.1.7, тоді $E_{\alpha(x)} = D_x$, причому $\varphi_{\alpha(x)}$ є РФ при $D_x \neq \varnothing$. Отже, або $D_x = \varnothing$, або $D_x = E_{\alpha(x)}$ і $\varphi_{\alpha(x)}$ є РФ.

Твердження $df2 \Leftrightarrow df3$ випливає із теореми 5.1.8.

Звідси та із вже доведеного $df3 \Rightarrow df1$ отримуємо $df2 \Rightarrow df1$ ■

Зауважимо, що *df*3 та *df*4 можна без зміни використовувати для РПМ, заданих на N^n.

Теорема 5.1.10. *Клас РПМ замкнений відносно операцій* $\cup$ *та* $\cap$.

Нехай A і B — РПМ, а f і g — РФ такі, що $A = E_f$ та $B = E_g$.

Задамо функцію h так: $h(2 \cdot x) = f(x)$, $h(2 \cdot x + 1) = g(x)$. Тоді h є РФ, причому $E_h = E_f \cup E_g = A \cup B$. Тому $A \cup B$ — РПМ.

Маємо $\chi^{\text{ч}}_{A \cap B}(x) = \chi^{\text{ч}}_A(x) \cdot \chi^{\text{ч}}_B(x)$, тому $\chi^{\text{ч}}_{A \cap B}$ — ЧРФ. Згідно з *df*4 $A \cap B$ є РПМ ■

Теорема 5.1.11 (теорема Поста). *Якщо множини* L *і* $\overline{L}$ *рекурсивно перелічні, то множини* L *і* $\overline{L}$ *рекурсивні.*

Вважаємо, що $L \neq \varnothing$ і $\overline{L} \neq \varnothing$, інакше твердження теореми тривіальне.

Нехай f і g — такі РФ, що $L = E_f$ і $\overline{L} = E_g$. Вкажемо алгоритм $\aleph$, який за довільним $b \in N$ визначає, $b \in L$ чи $b \notin L$.

Функцію $h(x)$ задамо так: $h(2 \cdot x) = f(x)$, $h(2 \cdot x + 1) = g(x)$. Тоді h є РФ, причому $N = E_h = E_f \cup E_g = L \cup \overline{L}$.

Поступово обчислюємо значення $h(0)$, $h(1)$, ... Позаяк $N = E_h = L \cup \overline{L}$, то $b = h(n)$ для деякого n. Якщо n парне, то $b = h(n) = f(n/2)$, звідки $b \in L$. Якщо n непарне, то $b = h(n) = g((n-1)/2)$, звідки $b \in \overline{L}$, тому $b \notin L$. Отже, множина L алгоритмічно розв'язна. За тезою Чорча L рекурсивна.

Аналогічно доводимо рекурсивність множини $\overline{L}$ ■

Для довільної множини $L \subseteq N$ уведемо позначення

$$L_{2x} = \{2x \mid x \in L\} \text{ і } L_{2x+1} = \{2x+1 \mid x \in L\}.$$

Для множин на N визначимо операції сполучення $\oplus$ та добутку $\otimes$:

$$A \oplus B = \{2x \mid x \in A\} \cup \{2x+1 \mid x \in B\} = A_x \cup B_{2x+1};$$

$$A \otimes B = \{C(x, y) \mid x \in A, y \in B\}.$$

Теорема 5.1.12. 1. *Множини A і B є РМ/РПМ $\Leftrightarrow$ $A \oplus B$ є РМ/РПМ.*

2. *Якщо $A \neq \varnothing$ і $B \neq \varnothing$, то A* і *B є РМ/РПМ $\Leftrightarrow$ $A \otimes B$ є РМ/РПМ.*

Доведемо для випадку РПМ. Для випадку РМ доведення аналогічне, тільки замість часткових характеристичних функцій беремо характеристичні функції відповідних множин.

Нехай A і B є РПМ. Маємо $x \in A \oplus B \Leftrightarrow$ (x парне і $x/2 \in A$) або (x непарне і $(x-1)/2 \in B$). За ТЧ $\chi^{\text{ч}}_{A \otimes B}$ є ЧРФ, тому $A \oplus B$ є РПМ.

Маємо $x \in A \Leftrightarrow 2x \in A \oplus B$ і $x \in B \Leftrightarrow 2x+1 \in A \oplus B$. Звідси

$$\chi^{\text{ч}}_A(x) = \chi^{\text{ч}}_{A \oplus B}(2x), \ \chi^{\text{ч}}_B(x) = \chi^{\text{ч}}_{A \oplus B}(2x+1).$$

Тому якщо $A \oplus B$ є РПМ, то A і B також є РПМ.

Маємо $x \in A \otimes B \Leftrightarrow \boldsymbol{l}(x) \in A$ і $\boldsymbol{r}(x) \in B$, звідки $\chi^{\text{ч}}_{A \otimes B}(x) = \chi^{\text{ч}}_A(l(x)) \cdot \chi^{\text{ч}}_B(r(x))$. Тому якщо A і B є РПМ, то $A \otimes B$ є РПМ.

Зафіксуємо довільні $a \in A$, $b \in B$. Маємо $x \in A \Leftrightarrow C(x, b) \in A \otimes B$ і $x \in B \Leftrightarrow C(a, x) \in A \otimes B$. Тому $\chi^{\text{ч}}_A(x) = \chi^{\text{ч}}_{A \otimes B}(C(x,b))$ і $\chi^{\text{ч}}_B(x) = \chi^{\text{ч}}_{A \otimes B}(C(a,x))$.

Отже, якщо $A \oplus B$ є РПМ, то A і B також є РПМ ■

Ефективну нумерацію РПМ вводимо на основі нумерацій n-арних ЧРФ згідно з *df*3.

Таку нумерацію називають *стандартною нумерацією РПМ*.

Номером РПМ $L \subseteq N^n$ є номер n-арної ЧРФ f такої, що $L = D_f$.

РПМ $L \subseteq N^n$ з номером (індексом) m позначаємо D_m^n, або D_m для випадку $n = 1$.

Множину $\{L \subseteq N^n \mid L \in$ РПМ$\}$ позначаємо РПМ^n.

Аналогічно вводимо позначення РМ^n і ПРМ^n.

5.2. Примітивно рекурсивні, рекурсивні, частково рекурсивні предикати

n-арний предикат на N називають *рекурсивним* (РП), якщо його характеристична функція рекурсивна.

n-арний предикат на N називають *примітивно рекурсивним* (ПРП), якщо його характеристична функція є ПРФ.

n-арний предикат на N називають *частково рекурсивним* (ЧРП), якщо його часткова характеристична функція є ЧРФ.

Безпосередньо із визначень випливає, що кожний ПРП є РП.

Замість "$P(x_1, ..., x_n) = T$" записуватимемо "$P(x_1, ..., x_n)$".

Теорема 5.2.1. 1) *$P \in$ ЧРП (РП, ПРП) $\Leftrightarrow$ $I_P \in$ РПМ (РМ, ПРМ);*

2) *класи ПРП і РП замкнені відносно логічних операцій* $\vee$, & *і* $\neg$;

3) *клас ЧРП замкнений відносно операцій* $\vee$ *і* &;

4) *клас ПРП строго включається до класу РП*;

5) *кожний рекурсивний предикат є ЧРП*;

6) *якщо P і $\neg P$* — ЧРП, *то P і $\neg P$* — РП.

Твердження 1) безпосередньо випливає із визначень.

Нехай I_P та I_Q — області істинності предикатів P і Q.

Тоді $I_P \cup I_Q$ — область істинності предиката $P \vee Q$, $I_P \cap I_Q$ — область істинності предиката $P\&Q$, $\overline{I}_P$ — область істинності предиката $\neg P$. Ураховуючи твердження 1), твердження 2) випливає з теореми 5.1.3, твердження 3) — з теореми 5.1.10, твердження 4) і 5) випливають із теореми 5.1.2, твердження 6) випливає з теореми Поста ■

Теорема 5.2.2. *Предикат $Q(x_1, ..., x_n)$ частково рекурсивний тоді і тільки тоді, коли існує рекурсивний предикат $R(x_1, ..., x_n, y)$ такий*: $Q(x_1, ..., x_n) \Leftrightarrow \exists y R(x_1, ..., x_n, y)$.

Нехай Q — ЧРП, нехай $\chi_Q^{\text{ч}}$ обчислюється МНР-програмою P.

Уведемо предикат $R(x_1, ..., x_n, y)$, що означає: $P(x_1, ..., x_n)\downarrow$ за k кроків.

Тоді $Q(x_1, ..., x_n) \Leftrightarrow \exists y R(x_1, ..., x_n, y)$. Але $\chi^{ч}_Q$ алгоритмічно обчислювана, тому $\chi^{ч}_Q$ є РФ за ТЧ. Звідси R є РП.

Нехай $R(x_1, ..., x_n, y)$ — РП, нехай $Q(x_1, ..., x_n) \Leftrightarrow \exists y R(x_1, ..., x_n, y)$.

Тоді $Q(x_1, ..., x_n) \Leftrightarrow \chi_R(x_1, ..., x_n, y) = 1$.

Тому $f(x_1, ..., x_n) = \boldsymbol{s}(\mathbf{o}(\mu_y(nsg(\chi_R(x_1, ..., x_n, y)) = 0))$ — часткова характеристична функція предиката $\exists y R(x_1, ..., x_n, y)$.

Функція χ_R є РФ, тому f є ЧРФ. Але $f = \chi^{ч}_Q$, тому Q є ЧРП ■

Теорема 5.2.3. *Нехай $Q(x_1, ..., x_n, y)$ є ЧРП. Тоді $\exists y Q(x_1, ..., x_n, y)$ також ЧРП.*

Нехай $Q(x_1, ..., x_n, y)$ є ЧРП.

За теоремою 5.2.2 існує РП $R(x_1, ..., x_n, y, z)$ такий: $Q(x_1, ..., x_n, y) \Leftrightarrow$ $\Leftrightarrow \exists z R(x_1, ..., x_n, y, z)$. Поклавши $u = C(y, z)$, отримаємо $\exists y Q(x_1, ..., x_n, y) \Leftrightarrow$ $\Leftrightarrow \exists y \exists z R(x_1, ..., x_n, y, z) \Leftrightarrow \exists u R(x_1, ..., x_n, \boldsymbol{l}(u), \boldsymbol{r}(u))$. Але $\boldsymbol{l}$ і $\boldsymbol{r}$ є ПРФ, $R(x_1, ..., x_n, y, z)$ є РП, тому $R(x_1, ..., x_n, \boldsymbol{l}(u), \boldsymbol{r}(u))$ також РП. Тоді $\exists u R(x_1, ..., x_n, \boldsymbol{l}(u), \boldsymbol{r}(u))$, а отже і $\exists y Q(x_1, ..., x_n, y)$, є ЧРП за теоремою 5.2.2 ■

Наслідок. *Якщо $Q(x_1, ..., x_n, y)$ є ЧРП, то $\exists y_1 ... \exists y_k Q(x_1, ..., x_n, y_1, ..., y_k)$ також ЧРП.*

Приклад 5.2.1. Предикат "x є числом Ферма" є ЧРП.

"x є числом Ферма" $\Leftrightarrow \exists u \exists v \exists w (u > 0 \,\&\, v > 0 \,\&\, w > 0 \,\&\, u^x + v^x = w^x)$. Але предикат у дужках є РП, тому за теоремою 5.2.2 наш предикат є ЧРП.

Приклад 5.2.2. Предикат "$y \in E_x$" є ЧРП.

$y \in E_x \Leftrightarrow \exists z \exists k (P_x(z) \downarrow y$ за k кроків). Предикат у дужках є РП, тому за теоремою 5.2.2 наш предикат є ЧРП.

Приклад 5.2.3. Предикат "$D_x \neq \varnothing$" є ЧРП.

$D_x \neq \varnothing \Leftrightarrow \exists z \exists k (P_x(z) \downarrow$ за k кроків). Предикат у дужках є РП, тому за теоремою 5.2.2 наш предикат є ЧРП.

Приклад 5.2.4. Предикат "$\{x, y\} \subseteq D_z$" є ЧРП.

Маємо $\{x, y\} \subseteq D_z \Leftrightarrow x \in D_z \,\&\, y \in D_z \Leftrightarrow \exists k (P_z(x) \downarrow y$ за k кроків) $\&$ $\exists k (P_z(y) \downarrow y$ за k кроків). У дужках РП, тому наш предикат ЧРП.

Приклад 5.2.5. Предикат "φ_x неін'єктивна" є ЧРП.

Маємо φ_x неін'єктивна $\Leftrightarrow \exists a \exists b \exists c\ (a \neq b\ \&\ \varphi_x(a) = c\ \&\ \varphi_x(b) = c) \Leftrightarrow$ $\exists a \exists b \exists c \exists k \exists l\ (a \neq b\ \&(P_x(a)\downarrow c$ за k кроків) & $(P_x(b)\downarrow c$ за l кроків)).

Теорема 5.2.4. *Функція* $f(x_1, ..., x_n)$ *є ЧРФ* $\Leftrightarrow$ *предикат* "$y = f(x_1, ..., x_n)$" *є ЧРП.*

Нехай $f(x_1, ..., x_n)$ є ЧРФ, P — МНР-програма для f.

Тоді $y = f(x_1, ..., x_n) \Leftrightarrow \exists k(P(x_1, ..., x_n)\downarrow y$ за k кроків).

Але "$P(x_1, ..., x_n)\downarrow y$ за k кроків" — РП. Тому предикат "$y = f(x_1, ..., x_n)$", тобто $\exists k(P(x_1, ..., x_n)\downarrow y$ за k кроків), є ЧРП.

Нехай "$y = f(x_1, ..., x_n)$" — ЧРП. За теоремою 5.2.2. існує РП R: $y = f(x_1, ..., x_n) \Leftrightarrow \exists z R(x_1, ..., x_n, y, z)$. Покладемо $t = C(y, z)$, тоді $y = f(x_1, ..., x_n) \Leftrightarrow \exists t R(x_1, ..., x_n, \boldsymbol{l}(t), \boldsymbol{r}(t))$. Покладемо $g(x_1, ..., x_n, t) =$ $= nsg(\chi_R(x_1, ..., x_n, \boldsymbol{l}(t), \boldsymbol{r}(t)))$. Тоді $\exists t R(x_1, ..., x_n, \boldsymbol{l}(t), \boldsymbol{r}(t)) \Leftrightarrow \exists t(y = \boldsymbol{l}(t)$ і $g(x_1, ..., x_n, t) = 0)$. Узявши перше таке t, матимемо $\exists t(y = \boldsymbol{l}(t)$ і $g(x_1, ..., x_n, t) = 0) \Leftrightarrow y = \boldsymbol{l}(\mu_t(g(x_1, ..., x_n, t) = 0))$.

Отримаємо $y = f(x_1, ..., x_n) \Leftrightarrow y = \boldsymbol{l}(\mu_t(g(x_1, ..., x_n, t) = 0))$, тому f є ЧРФ ■

Теорема 5.2.5 (теорема Кліні про нормальну форму). *Для кожної n-арної ЧРФ f існує* $(n + 1)$*-арна РФ g така*: *для всіх* $x_1, ..., x_n \in N$ *маємо* $f(x_1, ..., x_n) = \boldsymbol{l}(\mu_t(g(x_1, ..., x_n, t) = 0))$.

У цьому випадку зображення $\boldsymbol{l}(\mu_t(g(x_1, ..., x_n, t) = 0))$ називають *нормальною формою* функції $f(x_1, ..., x_n)$.

За теоремою 5.2.4 предикат "$y = f(x_1, ..., x_n)$" є ЧРП, тому для деякого РП R маємо $y = f(x_1, ..., x_n) \Leftrightarrow \exists z R(x_1, ..., x_n, y, z)$.

Покладемо $g(x_1, ..., x_n, t) = nsg(\chi_R(x_1, ..., x_n, \boldsymbol{l}(t), \boldsymbol{r}(t)))$. Повторюючи доведення теореми 5.2.4, маємо $y = f(x_1, ..., x_n) \Leftrightarrow$ $\Leftrightarrow y = \boldsymbol{l}(\mu_t(g(x_1, ..., x_n, t) = 0))$.

Отже, $f(x_1, ..., x_n) = \boldsymbol{l}(\mu_t(g(x_1, ..., x_n, t) = 0))$ ■

Функцію g в формулюванні теореми Кліні про нормальну форму насправді можна брати ПРФ. Доведення такого посиленого варіанта теореми 5.2.5 див. у [5, 9]. У цьому разі теорема Кліні про нормальну форму свідчить про те, що кожну ЧРФ можна отримати з деякої ПРФ не більше ніж одним застосуванням операції мінімізації, причому у певний стандартний спосіб.

Теорема 5.2.6 (про графік). *Функція* $f(x_1, ..., x_n)$ *є ЧРФ* $\Leftrightarrow$ Γ_f *є РПМ.*

$\Gamma_f = \{(x_1, ..., x_n, y) \mid (x_1, ..., x_n) \in D_f$ і $y = f(x_1, ..., x_n)\}$ є областю істинності предиката "$y = f(x_1, ..., x_n)$". Але предикат є ЧРП $\Leftrightarrow$ його область істинності є РПМ. За теоремою 5.2.4 маємо потрібний результат ■

5.3. Алгоритмічна нерозв'язність проблем зупинки та самозастосовності. Наслідки

Масову проблему називають *алгоритмічно розв'язною*, або *розв'язною*, якщо відповідний предикат рекурсивний.

Масова проблема *алгоритмічно нерозв'язна*, якщо відповідний предикат не рекурсивний.

Масову проблему називають *частково алгоритмічно розв'язною*, або *частково розв'язною*, або *напівrозв'язною*, якщо відповідний предикат частково рекурсивний.

Наприклад, проблеми "x є квадратом натурального числа" і "$P(x_1, ..., x_n)\downarrow$ за k кроків" алгоритмічно розв'язні.

Прикладами алгоритмічно нерозв'язних проблем є проблема зупинки та проблема самозастосовності.

Проблема зупинки формулюється так: за x і y встановити, чи є визначеним значення $\varphi_x(y)$.

Проблема самозастосовності формулюється так: за x встановити, чи є визначеним значення $\varphi_x(x)$.

Неформально проблема зупинки означає: встановити за x і y, чи зупиниться МНР-програма з кодом x при роботі над y.

Проблема самозастосовності неформально означає: встановити за x, чи зупиниться МНР-програма з кодом x при роботі над власним кодом.

Предикат "$\varphi_x(y)$ визначене" позначимо $Q(x, y)$.

Предикат "$\varphi_x(x)$ визначене" позначимо $S(x)$.

Зрозуміло, що $S(x) \Leftrightarrow Q(x, x)$.

Теорема 5.3.1. *Проблема самозастосовності алгоритмічно нерозв'язна.*

Покажемо, що предикат $S(x)$ нерекурсивний.

Припустимо супротивне, тоді

$$\chi_S(x) = \begin{cases} 1, & \text{якщо } \varphi_x(x) \text{ визначене,} \\ 0, & \text{якщо } \varphi_x(x) \text{ не визначене,} \end{cases}$$

є РФ.

Задамо функцію

$$f(x) = 0 \text{ — } \chi_S(x) = \begin{cases} \text{не визначене, якщо } \varphi_x(x) \text{ визначене,} \\ 0, \text{ якщо } \varphi_x(x) \text{ не визначене.} \end{cases}$$

Функція f є ЧРФ, нехай n — її індекс у нумерації ЧРФ[1], тобто функція $f(x)$ суть $\varphi_n(x)$. Тоді маємо

$$\varphi_n(n) = \begin{cases} \text{не визначене, якщо } \varphi_n(n) \text{ визначене,} \\ 0, \text{ якщо } \varphi_n(n) \text{ не визначене,} \end{cases}$$

суперечність! ■

Наслідок. *Проблема зупинки алгоритмічно нерозв'язна.*

Справді, алгоритмічна розв'язність проблеми зупинки означає, що предикат $Q(x, y)$ є рекурсивним, звідки предикат $S(x)$ рекурсивний. Це суперечить теоремі 5.3.1 ■

Теорема 5.3.2. *Проблеми зупинки та самозастосовності частково розв'язні.*

Для доведення часткової рекурсивності предиката $Q(x, y)$ задамо алгоритм обчислення $\chi^{\text{ч}}_Q$: за x як за кодом МНР-програми відновимо P_x і почнемо обчислення $P_x(y)$. Якщо $P_x(y)\downarrow$, то алгоритм видає 1 як результат; якщо $P_x(y)\uparrow$, то алгоритм ніколи не видасть результату, тобто $\chi^{\text{ч}}_Q(x, y)$ тоді невизначене. За ТЧ $\chi^{\text{ч}}_Q$ є ЧРФ, тому Q є ЧРП. Але $\chi^{\text{ч}}_S(x) = \chi^{\text{ч}}_Q(x, x)$, тому S також ЧРП ■

Теорема 5.3.3. *Множина $D = \{x \mid \varphi_x(x)$ визначене$\}$ — нерекурсивна РПМ.*

Характеристична функція множини D є характеристичною функцією предиката S, яка нерекурсивна згідно з теоремою 5.3.1. Множина D є областю визначення ЧРФ $u(x) = \varphi_x(x)$, тому множина D є РПМ ■

Наслідок 1. *Множина $\overline{D} = \{x \mid \varphi_x(x)$ невизначене$\}$ не є РПМ.*

Припустимо супротивне: $\overline{D}$ є РПМ. Тоді за теоремою Поста множини $\overline{D}$ і D рекурсивні, що суперечить теоремі 5.3.3 ■

Наслідок 2. *Предикат* $\neg S(x)$ *не є ЧРП.*

Справді, областю істинності предиката $\neg S$ є множина $\bar{D}$ ■

Наслідок 3. *Клас РПМ незамкнений відносно операції доповнення.*

Наслідок 4. *Клас ЧРП незамкнений відносно логічної операції* $\neg$.

На основі отриманих результатів дістаємо такі співвідношення для відповідних класів функцій, множин і предикатів (тут СМ позначає клас скінченних множин).

Теорема 5.3.4. *Справджуються такі строгі включення*:

ПРФ $\subset$ РФ $\subset$ ЧРФ;

СМ $\subset$ ПРМ $\subset$ РМ $\subset$ РПМ;

ПРП $\subset$ РП $\subset$ ЧРП.

Функція g називається *розширенням* функції f, якщо $D_f \subseteq D_g$ і для всіх $x \in D_f$ маємо $f(x) = g(x)$.

Функцію f тоді називають звуженням функції g.

Цей факт позначатимемо $f \subseteq g$.

Тотальне розширення функції називається *довизначенням* цієї функції.

Рекурсивні функції є тотальними ЧРФ, тому виникає питання, чи можна так довизначити кожну ЧРФ, щоб вона стала рекурсивною.

Теорема 5.3.5. *Функція* $\varphi_x(x)$ *не має рекурсивних довизначень.*

Припустимо супротивне: $\varphi_x(x)$ має рекурсивне довизначення $f(x)$. Тоді функція $nsg(\varphi_x(x))$ має рекурсивне довизначення $g(x)$. Нехай k — індекс функції g у нумерації 1-арних ЧРФ, тобто g суть функція φ_k. Тоді значення $\varphi_k(k) = g(k)$ визначене, бо g є РФ. Тому $nsg(\varphi_k(k))$ визначене.

Маємо $nsg(\varphi_k(k)) = g(k) = \varphi_k(k)$ — суперечність ■

Теорема 5.3.6. *Існують ЧРФ з нерекурсивним графіком.*

Такими є, зокрема, функції $\chi_L^{\text{ч}}$ для нерекурсивних РПМ L.

Справді, якщо $\chi_L^{\text{ч}} = \{(x, 1) \mid x \in L\}$ рекурсивна, то за ТЧ множина $L = pr_1(\Gamma_{\chi_L^{\text{ч}}})$ рекурсивна, що суперечить нерекурсивності L ■

Покажемо, що операція мінімізації μ_y істотно відрізняється від неконструктивної, узагалі кажучи, операції min_y для знаходження найменшого значення y, яке задовольняє певну умову.

Функція $f(x_1, ..., x_n)$ виникає з функції $g(x_1, ..., x_n, y)$ за допомогою операції min_y, якщо для всіх значень $x_1, ..., x_n$ маємо

$$f(x_1, ..., x_n) =$$

$$\begin{cases} \text{найменше } y \text{ таке, що } g(x_1,...,x_n,y) = 0, & \text{якщо таке } y \text{ існує,} \\ \text{не визначене,} & \text{якщо таке } y \text{ не існує.} \end{cases}$$

Теорема 5.3.7. *Існує ЧРФ h така, що* $f(x) = min_y(h(x, y) = 0)$ *не є ЧРФ.*

Функція $h(x, y) = \begin{cases} 0, & \text{якщо } y = 1 \text{ або } (y = 0 \text{ та } x \in D), \\ & \text{не визначене інакше.} \end{cases}$

є ЧРФ за ТЧ.

Функція $f(x) = min_y(h(x, y) = 0)$ тотальна, бо для всіх $x \in N$ маємо $f(x) = 1$ або $f(x) = 0$.

Якщо $x \in D$, то $h(x, 0) = 0$, тому $f(x) = 0$.

Якщо $x \notin D$, то $h(x, 0)$ невизначене, але $h(x,1) = 1$, тому $f(x) = 1$.

Таким чином, $f(x) = nsg(\chi_D(x))$, але тоді $f(x)$ не є РФ і не є ЧРФ ■

5.4. Індексні множини. Теореми Райса та Райса-Шапіро

Після введення ефективних нумерацій ЧРФ природно виникають питання, які властивості ЧРФ можна розпізнати або частково розпізнати за номерами функцій, тобто чи будуть множини номерів відповідних класів ЧРФ рекурсивними або рекурсивно перелічними.

Нехай φ: $N \to \Im$ — ефективна нумерація множини об'єктів $\Im$.

Для довільної $\Re \subseteq \Im$ визначимо множину номерів усіх об'єктів із $\Re$:

$$N(\Re) = \varphi^{-1}(\Re).$$

Множини вигляду $N(\Re)$, де $\Re \subseteq ЧРФ^n$ (зокрема, $\Re \subseteq ЧРФ^1$), називають індексними множинами.

Теорема 5.4.1 (теорема Райса). *Нехай* $\Re \subset ЧРФ^n$ *і* $\Re \neq \varnothing$. *Тоді множина* $N(\Re)$ *нерекурсивна.*

Вважаємо, що всюди невизначена функція $f_\varnothing \notin \mathfrak{R}$ (інакше замість $\mathfrak{R}$ візьмемо $\mathfrak{R}' = \text{ЧРФ}^n \setminus \mathfrak{R}$, тоді $N(\mathfrak{R})$ рекурсивна $\Leftrightarrow$ $N(\mathfrak{R}')$ рекурсивна, причому $\mathfrak{R}' \subset \text{ЧРФ}^n$ та $\mathfrak{R}' \neq \varnothing$).

Зафіксуємо довільну функцію $g \in \mathfrak{R}$. Задамо функцію f так:

$$f(z, x_1, ..., x_n) = \begin{cases} g(x_1,...,x_n), \text{ якщо } z \in D, \\ \text{не визначене, якщо } z \notin D. \end{cases}$$

Предикат "$z \in D$" є ЧРП, тому за ТЧ функція f є ЧРФ.

За s-m-n-теоремою існує РФ s така:

$f(z, x_1, ..., x_n) = \varphi^n_{s(z)}(x_1, ..., x_n)$ для всіх $z, x_1, ..., x_n$.

При $z \in D$ маємо $\varphi^n_{s(z)}(x_1, ..., x_n) = f(z, x_1, ..., x_n) = g(x_1, ..., x_n)$, тобто $\varphi^n_{s(z)}$ — це функція g. Отже, $\varphi^n_{s(z)} \in \mathfrak{R}$, звідки $s(z) \in N(\mathfrak{R})$.

При $z \notin D$ значення $\varphi^n_{s(z)}(x_1, ..., x_n) = f(z, x_1, ..., x_n)$ невизначене, тобто $\varphi^n_{s(z)}$ — це функція $f_\varnothing$. Отже, $\varphi^n_{s(z)} \notin \mathfrak{R}$, звідки $s(z) \notin N(\mathfrak{R})$.

Отже, $z \in D \Leftrightarrow s(z) \in N(\mathfrak{R})$.

Припустимо, що $N(\mathfrak{R})$ рекурсивна. Тоді предикат "$z \in N(\mathfrak{R})$" рекурсивний, звідки предикат "$s(z) \in N(\mathfrak{R})$" також рекурсивний через рекурсивність функції s. Звідси предикат "$z \in D$" рекурсивний, що суперечить нерекурсивності D. Отже, $N(\mathfrak{R})$ нерекурсивна ■

Наслідок. *Нехай* $\mathfrak{R} \subset \text{РПМ}$ *і* $\mathfrak{R} \neq \varnothing$. *Тоді* $N(\mathfrak{R})$ *не є РМ.*

Отже, теорема Райса стверджує: жодна нетривіальна властивість у класах усіх n-арних ЧРФ і всіх РПМ не може бути ефективно розпізнана!

Приклад 5.4.1. Множина $\{x \mid D_x \neq \varnothing\}$ — нерекурсивна РПМ.

Предикат "$D_x \neq \varnothing$" є ЧРП, бо $D_x \neq \varnothing \Leftrightarrow \exists y \exists k (P_x(y)\downarrow$ за k кроків), а предикат "$P_x(y)\downarrow$ за k кроків" є РП. Тому $\{x \mid D_x \neq \varnothing\}$ є РПМ. Але за теоремою Райса $\{x \mid D_x \neq \varnothing\}$ не є РМ.

У доведенні теореми Райса ми вважали, що $f_\varnothing \notin \mathfrak{R}$. Припустивши $f_\varnothing \in \mathfrak{R}$, дістанемо твердження, певною мірою дуальне до теореми Райса.

Теорема 5.4.2. *Нехай* $\mathfrak{R} \subset \text{ЧРФ}^n$ *і* $f_\varnothing \in \mathfrak{R}$. *Тоді* $N(\mathfrak{R})$ *не є РПМ.*

Зафіксуємо довільну функцію $g \notin \mathfrak{R}$. Задамо функцію f так:

$$f(z, x_1, ..., x_n) = \begin{cases} g(x_1,...,x_n), \text{ якщо } z \in D, \\ \text{не визначене, якщо } z \notin D. \end{cases}$$

За ТЧ f є ЧРФ. За s-m-n-теоремою існує РФ s така:

$f(z, x_1, ..., x_n) = \varphi^n_{s(z)}(x_1, ..., x_n)$ для всіх $z, x_1, ..., x_n$.

При $z \in D$ $\varphi^n_{s(z)}(x_1, ..., x_n) = f(z, x_1, ..., x_n) = g(x_1, ..., x_n)$, тобто $\varphi^n_{s(z)}$ — це функція g. Отже, $\varphi^n_{s(z)} \notin \Re$, звідки $s(z) \notin N(\Re)$.

При $z \notin D$ значення $\varphi^n_{s(z)}(x_1, ..., x_n) = f(z, x_1, ..., x_n)$ невизначене, тобто $\varphi^n_{s(z)}$ — це функція $f_\varnothing$. Отже, $\varphi^n_{s(z)} \in \Re$, звідки $s(z) \in N(\Re)$.

Отже, $z \notin D \Leftrightarrow s(z) \in N(\Re)$.

Якщо $N(\Re)$ є РПМ, то предикат "$z \in N(\Re)$" є ЧРП, звідки предикат "$s(z) \in N(\Re)$" також ЧРП через рекурсивність функції s. Звідси предикат "$z \notin D$" є ЧРП, а це суперечить тому факту, що $\bar{D}$ не є РПМ.

Отже, $N(\Re)$ не є РПМ ■

Наслідок. *Множини* $\{x \mid D_x$ є РМ$\}$ *та* $\{x \mid D_x$ *скінченна*$\}$ *не є РПМ.*

Множина $\varnothing$ скінченна і рекурсивна. Тому $f_\varnothing \in \{\varphi_x \mid D_x$ скінченна$\}$ та $f_\varnothing \in \{\varphi_x \mid D_x$ є РМ$\}$, звідки за теоремою 5.4.2 множини $\{x \mid D_x$ є РМ$\}$ і $\{x \mid D_x$ скінченна$\}$ не є РПМ ■

Скориставшись наслідком теореми 5.4.2, покажемо, що в загальному випадку за індексом x рекурсивної множини D_x неможливо ефективно знайти індекс РМ $\bar{D}_x$.

Теорема 5.4.3. *Не існує ЧРФ* f *такої, що для всіх* $x \in N$

$$f(x) = \begin{cases} \text{індекс множини } \bar{D}_x, \text{ якщо } D_x \quad PM, \\ \text{не визначене, якщо } D_x \text{ не } \quad PM. \end{cases}$$

Якщо така ЧРФ f існує, то D_x є РМ $\Leftrightarrow f(x)$ визначене.

Тоді $D_f = \{x \mid D_x$ є РМ$\}$, звідки за наслідком теореми 5.4.2 D_f не є РПМ. Дістали суперечність із припущенням, що f є ЧРФ ■

Сформулюємо критерій належності функції до конструктивної індексної множини, тобто до множини ЧРФ із рекурсивно перелічною множиною індексів. Виявляється, що для цього достатньо скінченної інформації про функцію.

Теорема 5.4.4 (теорема Райса-Шапіро). *Нехай* $\mathfrak{R} \subseteq$ ЧРФ^n *така, що* $N(\mathfrak{R})$ *є РПМ. Тоді для довільної функції* $f \in$ ЧРФ^n *маємо*: $f \in \mathfrak{R} \Leftrightarrow$ *існує скінченна функція* θ *така, що* $\theta \subseteq f$ *та* $\theta \in \mathfrak{R}$.

Випадок $f = f_{\varnothing}$ тривіальний, тому розглядаємо випадок $f \neq f_{\varnothing}$. Зауважимо, що при $f_{\varnothing} \in \mathfrak{R}$ множина $N(\mathfrak{R})$ є РПМ лише у випадку $\mathfrak{R} =$ ЧРФ^n.

Доводимо $\Rightarrow$.

Припустимо супротивне: $f \in \mathfrak{R}$, але не існує скінченної функції θ такої, що $\theta \subseteq f$ та $\theta \in \mathfrak{R}$. Нехай P — МНР-програма така, що $P(z)\downarrow \Leftrightarrow z \in D$.

Задамо функцію $g(z, x_1, ..., x_n) =$

$$= \begin{cases} f(x_1,...,x_n), & \text{якщо } P(z) \text{ не зупиниться за } \le C^n(x_1, ..., x_n) \text{ кроків,} \\ \text{не визначене, якщо } P(z)\downarrow \text{ за } \le C^n(x_1, ..., x_n) \text{ кроків.} \end{cases}$$

За ТЧ така g є ЧРФ. За s-m-n-теоремою існує РФ s така:
$g(z, x_1, ..., x_n) = \varphi^n_{s(z)}(x_1, ..., x_n$ для всіх $z, x_1, ..., x_n$.
Тоді $\varphi^n_{s(z)} \subseteq f$ для всіх z.

Нехай $z \in D$. Тоді $P(z)\downarrow$, звідки існує t таке, що $P(z)\downarrow$ за t кроків. Але тоді для кожного $k \ge t$ $P(z)\downarrow$ за $\le k$ кроків, тому для всіх $x_1, ..., x_n$ таких, що $C^n(x_1, ..., x_n) \ge t$, маємо $\varphi^n_{s(z)}(x_1, ..., x_n)\uparrow$. Тому $\varphi^n_{s(z)}$ скінченна. Але $\varphi^n_{s(z)} \subseteq f$, тому за припущенням $\varphi^n_{s(z)} \notin \mathfrak{R}$, звідки $s(z) \notin N(\mathfrak{R})$.

Нехай $z \notin D$. Тоді $P(z)\uparrow$, звідки $P(z)$ не зупиниться за $\le C^n(x_1, ..., x_n)$ кроків для кожних $x_1, ..., x_n$. Отже, $\varphi^n_{s(z)}$ — це функція f, за припущенням $\varphi^n_{s(z)} \in \mathfrak{R}$, звідки $s(z) \in N(\mathfrak{R})$.

Маємо $z \notin D \Leftrightarrow s(z) \in N(\mathfrak{R})$. Якщо $N(\mathfrak{R})$ є РПМ, то предикат "$z \in N(\mathfrak{R})$" є ЧРП, звідки "$s(z) \in N(\mathfrak{R})$" також ЧРП через рекурсивність функції s. Звідси предикат "$z \notin D$" є ЧРП, а це суперечить тому факту, що $\bar{D}$ не є РПМ.

Отримали суперечність, бо $N(\mathfrak{R})$ є РПМ і $\bar{D}$ не є РПМ.

Доводимо $\Leftarrow$.

Припустимо супротивне: маємо функцію f таку, що $f \notin \mathfrak{R}$, але існує скінченна функція θ така: $\theta \subseteq f$ і $\theta \in \mathfrak{R}$.

Задамо функцію

$$h(z, x_1, ..., x_n) = \begin{cases} f(x_1, ..., x_n), \text{ якщо } (x_1, ..., x_n) \in D_\theta \text{ або } z \in D, \\ \text{не визначене інакше.} \end{cases}$$

За ТЧ h є ЧРФ. За s-m-n-теоремою існує РФ s така:

$h(z, x_1, ..., x_n) = \varphi^n_{s(z)}(x_1, ..., x_n)$ для всіх значень $z, x_1, ..., x_n$.

Зрозуміло, що $\varphi^n_{s(z)} \subseteq f$ для всіх z.

Нехай $z \in D$. Тоді $\varphi^n_{s(z)}$ — це функція f, звідси за припущенням маємо $\varphi^n_{s(z)} \notin \Re$, тому $s(z) \notin N(\Re)$.

Нехай $z \notin D$. При $(x_1, ..., x_n) \in D_\theta$ маємо $\varphi^n_{s(z)}(x_1, ..., x_n) = f(x_1, ..., x_n)$, при $(x_1, ..., x_n) \notin D_\theta$ маємо $\varphi^n_{s(z)}(x_1, ..., x_n)\uparrow$. Отже, $\varphi^n_{s(z)}$ — це функція θ, тому за припущенням $\varphi^n_{s(z)} \in \Re$, звідки $s(z) \in N(\Re)$.

Маємо $z \notin D \Leftrightarrow s(z) \in N(\Re)$. Знову отримуємо суперечність ■

Приклад 5.4.2. Множина $\{x \mid \varphi_x \in \text{РФ}\}$ не є РПМ.

Припустимо, що $\{x \mid \varphi_x \in \text{РФ}\}$ є РПМ. Тоді за теоремою Райса-Шапіро для кожної РФ g існує скінченна функція θ така, що $\theta \subseteq f$ і θ — 1-арна РФ. Але скінченні функції не можуть бути рекурсивними. Маємо суперечність.

Приклад 5.4.3. Множина $\{x \mid D_x \text{ нескінченна}\}$ не є РПМ.

Припустимо, що $\{x \mid D_x \text{ нескінченна}\}$ є РПМ. За теоремою Райса-Шапіро для кожної нескінченної φ_x існує скінченна функція θ така, що $\theta \in \{\varphi_x \mid D_x \text{ нескінченна}\}$ та $\theta \subseteq \varphi_x$. Але тоді θ нескінченна. Знову отримали суперечність.

5.5. Складність обчислень

При виконанні реальних обчислень на перший план виходять питання, пов'язані з практичною реалізовністю таких обчислень на тій чи іншій моделі алгоритму. Основне: чи вистачить для цього ресурсів, передусім часу та місця?

Розглянемо дуже просту на перший погляд функцію, задану такою схемою примітивної рекурсії:

$$f(0) = 1;$$
$$f(x + 1) = 2^{f(x)}.$$

Таку фунцію називають надсхідчастою. Вона надзвичайно швидко зростає. Маємо $f(1) = 2, f(2) = 4, f(3) = 16, f(4) = 65536$. Спробуйте тепер хоча б оцінити величину значення $f(10)$.

Ми бачимо, що реальна обчислюваність може дуже відрізнятися від абстрактної, яка передбачає тільки принципову можливість виконати обчислення, не беручи до уваги необхідні для цього ресурси.

Питання практичної обчислюваності та оцінювання складності обчислень вивчаються в розділі теорії алгоритмів, який називають прикладною теорією алгоритмів, або теорією складності обчислень.

Кожне конкретне обчислення здійснюється певним пристроєм у певному місці фізичного простору, займаючи відповідний об'єм і тривaючи певний час. Тому для формалізації інтуїтивної уяви про об'єм та час обчислень природно зафіксувати певну алгоритмічну модель א, а також зазначити спосіб вимірювання часу, витраченого на ці обчислення, і спосіб вимірювання необхідного для цього об'єму простору (пам'яті).

Отже, отримуємо спеціальні функції від вхідних даних, які природно назвати часом і пам'яттю обчислень. Такі функції називають функціями складності обчислень, або мірами обчислювальної складності, або мірами складності обчислень. Досить поширеним є також термін "сигналізуюча функція" (часу чи пам'яті).

Найпопулярнішою моделлю для дослідження складності обчислень є багатострічкові машини Тьюрінга [2, 37]. Такі машини відрізняються від МТ, розглянутих у розділі 2, наявністю окремої вхідної стрічки і, як правило, вихідної, а також робочих стрічок. Існує багато різновидів таких машин: наприклад, робочі стрічки можуть бути необмежені в обидва боки або лише в один бік, вхідна стрічка може читатися тільки в один бік або в обидва боки і т. п. Зауважимо, що для оцінювання складності обчислень такі відмінності МТ не дуже істотні.

Час роботи МТ визначається як кількість кроків, що виконуються МТ для отримання результату. Крок — це виконання однієї команди МТ. Об'єм пам'яті звичайно визначають як максимум довжин використаних ділянок робочих стрічок.

Відмінність між МТ із однією робочою стрічкою та МТ із n робочими стрічками, де $n \geq 2$, не дуже істотна. Кожну функцію, обчислювану на n-стрічковій МТ із необхідною пам'яттю $S(x)$, можна обчислити на 1-стрічковій МТ із пам'яттю $c \cdot S(x)$, де c — константа. Цей факт видається інтуїтивно зрозумілим. Проте із часом обчислень ситуація дещо відмінна: кожну функцію, обчислювану на n-стрічковій

МТ за час $T(x)$, можна обчислити на 1-стрічковій МТ за час $c \cdot T^2(x)$, де c — константа.

Між об'ємом необхідної пам'яті та часом обчислення існує безпосередній зв'язок, що також зрозуміло. Справді, жодне обчислення з невеликим вхідним даним і невеликою пам'яттю не може бути надто тривалим без повторів, а жодне нетривале за часом обчислення не може потребувати великої памяті. Точніше кажучи, для кожної МТ існує константа k така, що $S(x) \le k \cdot T(x)$ і $T(x) \le k^{S(x)+x}$.

Аналогічні функції, що є мірами обчислювальної складності, вводяться також для інших моделей алгоритмів, зокрема для МНР-програм [5].

Найкращими з практичного погляду є функції, час обчислення яких лінійно залежить від розміру вхідного даного — функції, обчислювані за лінійний час. На жаль, переважна більшість практично цікавих функцій не можуть бути обчислені за лінійний час. Проте досить широкі класи таких функцій допускають обчислення за час, обмежений поліномами. Неважко переконатись, що час обчислення суперпозиції таких функцій також обмежений поліномом. Це дає нам поняття одного з найважливіших класів алгоритмічно обчислюваних функцій — класу ***P*** поліноміально обчислюваних функцій [2; 13].

До класу ***P*** належать усі функції, які можна обчислити на n-стрічковій МТ за поліноміально обмежений (від довжини вхідного даного) час.

Ураховуючи, що при переході від n-стрічкових до 1-стрічкових МТ часова міра складності зростає від $T(x)$ до $c \cdot T^2(x)$, а це не виводить за межі поліноміальної обчислюваності, при визначенні класу ***P*** можна обмежитись 1-стрічковими МТ.

Аналогічно можна ввести поняття класу ***NP***, який складається з функцій, обчислюваних за поліноміальний час на недетермінованих МТ.

Можна також задати клас вербальних множин **NP** як клас усіх множин (мов), що породжуються формальними граматиками типу 0 за поліноміальний час. Це означає, що кожне слово такої множини можна вивести за кількість кроків, що поліноміально залежить від його довжини.

Можна дати еквівалентне визначення **NP** як класу всіх вербальних множин, які розпізнаються недетермінованими МТ за поліноміальний час.

Для вербальних множин клас **P** визначається як клас усіх множин, які розпізнаються детермінованими МТ за поліноміальний час.

Прикладами множин класу **P** є, зокрема, довільна КС-мова, а також множина пар ізоморфних графів зі степенями вершин, що не перевищують фіксованого значення d [13].

Задачі, пов'язані з функціями (множинами, предикатами) класу $\boldsymbol{P}$, можна трактувати як практично розв'язні. Водночас набагато більше задач, що реально виникають у різних областях математики та інформатики, належать до класу $\boldsymbol{NP}$. Тому надзвичайний інтерес викликає питання збіжності класів $\boldsymbol{P}$ і $\boldsymbol{NP}$. Зрозуміло, що $\boldsymbol{P} \subseteq \boldsymbol{NP}$, але відповідь на питання, чи вірно $\boldsymbol{P} = \boldsymbol{NP}$, невідома досі, це одна з найважливіших відкритих проблем прикладної теорії алгоритмів.

Множину (предикат) S із класу $\boldsymbol{NP}$ називають NP-повною, якщо до неї зводиться кожна множина (предикат) із $\boldsymbol{NP}$.

Прикладом NP-повного предиката є проблема виконуваності пропозиційних формул: за даною ПФ встановити, чи буде вона виконуваною (тобто не буде суперечністю) [2].

Інші приклади NP-повних предикатів можна знайти в [2].

Аналогічно класам $\boldsymbol{P}$ і $\boldsymbol{NP}$ можна ввести класи $\boldsymbol{P\text{-}Sp}$ і $\boldsymbol{NP\text{-}Sp}$, які складаються з функцій, обчислюваних із поліноміальною пам'яттю відповідно на детермінованих МТ і недетермінованих МТ.

Поліноміальна оцінка складності за пам'яттю веде, загалом, до експоненціальної оцінки за часом, тому з практичного погляду обчислення з поліноміальною пам'яттю не завжди можуть бути реально здійсненими, адже для практичної реалізації потрібний поліноміальний час.

Відомо [2], що $\boldsymbol{P\text{-}Sp} = \boldsymbol{NP\text{-}Sp}$ і $\boldsymbol{NP} \subseteq \boldsymbol{P\text{-}Sp}$.

Складність обчислення конкретної функції f можна оцінити зверху та знизу. Для знаходження верхньої оцінки звичайно задають алгоритм обчислення такої функції на відповідній формальній моделі (МНР-програма, n-стрічкова МТ) і доводять потім, що оцінювана міра складності (за часом чи за пам'яттю) не перевищує значень деякої функції μ для всіх значень аргументу. Таку функцію μ називають верхньою оцінкою складності обчислення функції f.

Знаходження нижньої оцінки (на даній алгоритмічній моделі) складності обчислення функції f означає, що жоден алгоритм обчислення функції f на такій моделі не може мати міру складності (за ча-

сом чи за пам'яттю), меншу за значення певної функції φ. Як правило, для знаходження таких оцінок використовується діагональний метод. При цьому виявляється, що для переважної більшості логічних теорій нижня оцінка складності розв'язності формули (як функція її довжини) є експонентою.

Покращання верхніх оцінок складності не завжди веде до практичного покращання, адже оцінки складності дають, як правило, з точністю до деякого константного множника, а він може зростати дуже відчутно, незважаючи на покращання самої оцінки. Наприклад, множення матриць класичним алгоритмом має верхню оцінку порядку n^3, точніше, $k_1 \cdot n^3$. Відомий алгоритм Штрассена [2] дає оцінку порядку $n^{2.81}$, точніше, $k_2 \cdot n^{2.81}$. Останнім часом розроблено алгоритм [13] із ще кращою оцінкою порядку $n^{2.5}$, точніше, $k_3 \cdot n^{2.5}$. Проте алгоритм Штрассена стає кращим за класичний для матриць порядку не менше 14, а зазначений алгоритм з оцінкою $k_3 \cdot n^{2.5}$ стає кращим для матриць, порядок яких набагато перевищує 10^{10}, так що його практичне використання видається сумнівним.

Різні алгоритмічні моделі можуть давати, загалом, різні оцінки складності обчислення конкретної функції. Тому виникає проблема дослідження таких властивостей оцінок складності, які не залежать від конкретних алгоритмічних моделей. Цими питаннями займається інваріантна, або машинно-незалежна, теорія складності обчислень [5, 13]. Основним напрямом розвитку такої теорії є аксіоматичний. Він базується на формулюванні аксіом, яким задовольняє довільна розумна оцінка складності обчислень. При цьому поняття складності обчислення формалізується у вигляді міри обчислювальної складності.

У загальному випадку поняття міри обчислювальної складності можна ввести таким чином.

Мірою обчислювальної складності називають довільний клас функцій $\{\Phi_m^n\}$ із такими властивостями:

1) $D_{\Phi_m^n}$ для кожного m;

2) предикат "$y = \Phi_m^n$" є алгоритмічно розв'язним.

Прикладами мір обчислювальної складності для n-стрічкових МТ є часова міра $T(x)$ і пам'ять $S(x)$.

Аналогічні міри складності можна ввести для МНР-програм.

Часова міра

$$T_m^n(\bar{x}) = \mu_t(P_m(\bar{x}) \text{ за } t \text{ кроків}).$$

Міру складності $S_m^n(\bar{x})$, пов'язану з пам'яттю, визначають як максимальне значення вмісту регістрів МНР за весь час роботи МНР-програми P_m над вхідним даним $\bar{x}$, якщо $P_m(\bar{x})\downarrow$, інакше значення $S_m^n(\bar{x})$ невизначене.

Для 1-арних функцій звичайно вживають позначення T_m та S_m.

Неважко переконатись, що існують які завгодно складні обчислювані функції.

Теорема 5.5.1. *Для кожної РФ* β *існує РФ* $f = \varphi_m$ *така, що* $T_m(x) > \beta(x)$ *для всіх* x*, окрім, можливо, їх скінченної кількості.*

Зауважимо, що твердження теореми не можна посилити до такого: для кожної РФ β існує РФ $f = \varphi_m$ така, що $T_m(x) > \beta(x)$ для всіх x.

Справді, для кожної РФ f можна написати МНР-програму, яка дуже швидко обчислює $f(a)$ для довільного наперед вибраного конкретного значення a, просто вносячи $f(a)$ у програму.

Точніше кажучи, така програма може обчислити $f(a)$ за $a + f(a) + 2$ кроки.

Можна вказати МНР-програму, яка дуже швидко обчислює значення функції $f(x)$ для довільної наперед вибраної скінченної множини значень x.

Дуже важливим результатом інваріантної теорії обчислюваності є теорема Блюма про прискорення. Для 1-арних функцій теорема про прискорення формулюється так:

Теорема 5.5.2 (про прискорення). *Для кожної РФ* β *існує РФ* f *така, що для кожної МНР-програми* P_k *для функції* f *існує МНР-програма* P_m *для цієї самої функції* f *така*: $\beta(T_m(x)) < T_k(x)$ *для всіх* x*, окрім, можливо, їх скінченної кількості.*

Отже, для довільної МНР-програми P_k функції f можна знайти МНР-програму P_m для тієї самої функції, яка працює більше ніж у β разів краще за програму P_k майже для всіх вхідних значень! Іншими словами, найкращої МНР-програми для обчислення функції f не існує. Це робить вельми проблемним визначення складності РФ f, а не складності конкретного алгоритму для обчислення f, адже найкращої, найшвидшої МНР-програми для обчислення f може просто не існувати.

З іншого боку, практичне обчислення значень кожної РФ $f(x)$ звичайно виконується для скінченної множини відносно невеликих зна-

чень аргументу. Якщо нам відомий розподіл частот для різних значень x, то існує, причому може бути ефективно знайдена, найшвидша програма для обчислення $f(x)$.

Важливим прикладом обчислюваних функцій, які можуть бути охарактеризовані в термінах складності обчислень, є елементарні за Кальмаром функції [5, 8].

Функція елементарна, якщо вона може бути отримана із базових функцій $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $+$, $\dot{-}$, $\times$ за допомогою операцій суперпозиції, підсумовування та мультиплікації.

Елементарні функції утворюють власний підклас ПРФ.

Прикладом неелементарної ПРФ є розглянута вище надсхідчаста функція.

Відомо, що кожна функція, обчислювана за елементарний час, є елементарною. З іншого боку, кожна елементарна функція може бути обчислена за елементарний час.

Для подальшого ознайомлення з теорією складності обчислень можна, зокрема, порекомендувати [2; 5; 8; 13; 37].

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Дайте визначення РМ, ПРМ, РПМ.
2. Сформулюйте наслідки тези Чорча для РМ і РПМ.
3. Що таке множина-згортка?
4. Наведіть співвідношення між класами ПРМ, РМ і РПМ.
5. Відносно яких теоретико-множинних операцій замкнені класи ПРМ і РМ?
6. Відносно яких теоретико-множинних операцій замкнений клас ПРМ?
7. Наведіть властивості РМ і РПМ
8. Сформулюйте еквівалентні визначення РПМ.
9. Сформулюйте теорему Поста для множин.
10. Як формулюється принцип редукції?
11. Як задається стандартна нумерація РПМ?
12. Дайте визначення операцій $\oplus$ і $\otimes$.
13. Чи замкнений щодо операцій $\oplus$ і $\otimes$ клас ПРМ? клас РМ? клас РПМ?
14. Дайте визначення РП, ПРП і ЧРП.
15. Відносно яких логічних зв'язок замкнені класи ПРП і РП?

16. Відносно яких логічних зв'язок замкнений клас ЧРП?
17. Зазначте співвідношення між класами ПРП, РП і ЧРП.
18. Сформулюйте теорему Поста для предикатів.
19. Який зв'язок між РП і ЧРП?
20. Який зв'язок між ЧРП і ЧРФ?
21. Сформулюйте теорему Кліні про нормальну форму.
22. Що таке алгоритмічно розв'язна масова проблема? частково алгоритмічно розв'язна масова проблема?
23. Як формулюється проблема зупинки?
24. Як формулюється проблема самозастосовності?
25. Сформулюйте наслідки алгоритмічної нерозв'язності проблеми самозастосовності.
26. Наведіть приклади нерекурсивних РПМ і множин, які не є РПМ.
27. Наведіть приклади нерекурсивних ЧРП і предикатів, які не є ЧРП.
28. Наведіть приклади ЧРФ, які не мають рекурсивних довизначень.
29. Поясніть принципову різницю між операціями мінімізації μ_y і неконструктивної операції min_y.
30. Сформулюйте теорему про графік.
31. Які множини називають індексними?
32. Чи є індексною діагональна множина D?
33. Сформулюйте теорему Райса.
34. У чому полягає значення теореми Райса?
35. Сформулюйте теорему, дуальну до теореми Райса.
36. Сформулюйте теорему Райса-Шапіро.
37. Наведіть приклади використання теорем про індексні множини.
38. Що таке функція складності обчислень?
39. Який зв'язок між об'ємом необхідної пам'яті та часом обчислення?
40. Що таке функція, обчислювана за лінійний час?
41. Дайте визначення класів ***P*** і ***NP***.
42. Яке співвідношення між класами ***P*** та ***NP***?
43. Що таке *NP*-повна множина (предикат)?
44. Дайте визначення класів ***P-Sp*** і ***NP-Sp***.
45. Дайте визначення міри обчислювальної складності.
46. Сформулюйте теорему про прискорення.
47. Дайте визначення класу елементарних функцій.
48. Який зв'язок між елементарними функціями та функціями, обчислюваними за елементарний час?

ВПРАВИ

1. Нехай A та B — РПМ, C — РМ, причому $A\cap B = \varnothing$, $A \subseteq C \subseteq A\cup B$. Доведіть, що тоді A є РМ.

2. Нехай f — РФ, g — ін'єктивна РФ така, що E_g — РМ, причому $f(x) \geq g(x)$ для всіх x. Доведіть, що тоді E_f є РМ.

3. Нехай A — РМ, f — сюр'єктивна РФ така, що $f(A)\cap f(N\setminus A) = \varnothing$. Доведіть, що множина $f(A)$ рекурсивна.

4. Доведіть, що множина $L \neq \varnothing$ рекурсивна $\Leftrightarrow$ існує нестрого монотонно зростаюча РФ g така, що $L = E_g$.

5. Доведіть узагальнену теорему Поста: нехай множини A та B — РПМ, причому $A\cap B = \varnothing$ та множина $A\cup B$ рекурсивна. Тоді A та B — рекурсивні множини.

6. Доведіть твердження (принцип редукції): для довільних РПМ A та B існують РПМ L і M такі: $L\subseteq A$, $M\subseteq B$, $L\cap M = \varnothing$, $L\cup M = A\cup B$.

7. Нехай f є ЧРФ, A є РПМ.

1) Доведіть, що тоді множини $f^{-1}(A)$ та $f(A)$ є РПМ.

2) Чи залишиться зазначене вище твердження вірним, якщо слова "ЧРФ" і "РПМ" відповідно замінити "РФ" і "РМ"?

8. Доведіть: якщо A є РПМ, то $\bigcup\limits_{x\in A} D_x$ та $\bigcup\limits_{x\in A} E_x$ також РПМ.

9. Доведіть, що для кожного $k\in N$ множина ${}^kD_n = \{x \mid \varphi_n(x) = k\}$ є РПМ. Доведіть, що при фіксованому k послідовність ${}^kD_0, {}^kD_1, \ldots, {}^kD_n, \ldots$ є переліком усіх РПМ.

10. Чи існує РФ $f(x, y)$ така: якщо $P_x(y)\downarrow$, то це за $\leq f(x, y)$ кроків?

11. Чи має рекурсивні довизначення функція $\varphi_x(y) + \varphi_y(x)$?

12. Чи існує РФ $s(x)$ така, що для всіх $x, y\in N$ маємо

$$\varphi_{s(x)}(y) = \begin{cases} 1, & \text{якщо } \varphi_x(y) = 1, \\ 0 & \text{в усіх інших випадках} \end{cases} ?$$

13. Нехай предикат $R(x, y)$ є РП.

1) Чи завжди $\forall x R(x, y)$ є РП?

2) Чи завжди $\forall x R(x, y)$ є ЧРП?

14. Нехай всі множини R_m, де $m\in N$, рекурсивні.

1) Чи завжди множина $\bigcap\limits_{m\in N} R_m$ є РМ?

2) Чи завжди множина $\bigcap_{m \in N} R_m$ є РПМ?

15. Доведіть:

1) Якщо функція f є ПРФ/РФ, то Γ_f є ПРМ/РМ.

2) Існують нерекурсивні ЧРФ із скінченним графіком.

3) Існує РФ f така, що Γ_f не є ПРМ.

16. Доведіть теорему Сколема: Γ_f є РМ/ПРМ $\Leftrightarrow$ існує РФ/ПРФ g така, що для всіх $x_1, ..., x_n \in N$ маємо $f(x_1, ..., x_n) = \mu_t(g(x_1, ..., x_n, t) = 0)$.

17. Доведіть:

1) Існують РФ f, які не можна подати у вигляді $f(x_1, ..., x_n) = \mu_t(g(x_1, ..., x_n, t) = 0)$ для деякої ПРФ g.

2) Існують РФ h такі, які не є ПРФ, але Γ_h є ПРМ.

18. Доведіть, що існують ЧРФ f такі, які не можна подати у вигляді $f(x_1, ..., x_n) = \mu_t(g(x_1, ..., x_n, t) = 0)$ для деякої РФ g.

19. Чи існує РФ s така:

1) для всіх $x, y \in N$ $D_{s(x,y)} = (E_{2x} \cup D_{x+2y}) \cap D_{3y}$?

2) для всіх $x, y \in N$ $E_{s(x,y)} = (D_{3x} \cap E_{2y}) \cup \{3y, x+y\}$?

3) для всіх $x, y \in N$ $D_{s(x,y)} = (E_{4x} \cup D_{3y+x}) \setminus \{x, y\}$?

4) для всіх $x, y, z \in N$ $E_{s(x,y,z)} = (D_{5x} \cap E_{x+y}) \cup E_{y+3z}$?

5) для всіх $x \in N$ $D_{s(x)} = C(D_x^2)$?

6) для всіх $x \in N$ $D_{s(x)}^2 = C^{-1}(D_x)$?

7) для всіх $x, y \in N$ $E_{s(x,y)} = f^{-1}(D_{2x} \cup E_y)$? (тут f — фіксована ЧРФ)

8) для всіх $x, y \in N$ $D_{s(x,y)} = f(D_x \cap E_{3y})$? (тут f — фіксована ЧРФ)

9) для всіх $x, y, z \in N$ $E_{s(x,y,z)} = E_x \setminus (D_y \cup E_z)$?

10) для всіх $x, y, z \in N$ $D_{s(x,y,z)} = (E_x \cap \bar{D}_y) \setminus D_z$?

11) для всіх $x, y, z \in N$ $E_{s(x,y,z)} = D_x \cup E_y \setminus E_z$?

20. Доведіть теорему Райса на основі теореми Кліні про нерухому точку.

21. Чи будуть РПМ наступні множини:

1) $\{x \mid 4 \in D_x\}$?

2) $\{x \mid x \in E_x\}$?

3) $\{x \mid 1 \in E_x^2\}$?

4) $\{x \mid \{1,2\} \subseteq E_x\}$?

5) $\{3x \mid x \in D_x\}$?

6) $\{C(x, y) \mid x \in D_y\}$?
7) $\{C(x, y) \mid x \in E_y\}$?
9) $\{x \mid E_x$ скінченна$\}$?
10) $\{x \mid \varphi_x$ є ПРФ$\}$?
11) $\{x \mid \varphi_x$ не є ПРФ$\}$?
12) $\{x \mid \varphi_x$ ін'єктивна$\}$?
13) $\{x \mid \varphi_x$ сюр'єктивна$\}$?
14) $\{x \mid \varphi_x$ не є сюр'єктивна$\}$?
15) $\{x \mid E_x$ є ПРМ$\}$?
16) $\{x \mid \varphi_x$ не є поліном$\}$?
17) $\{x \mid \varphi_x$ є поліном$\}$?
18) $\{x \mid D_x$ скінченна та $\neq\varnothing\}$?
19) $\{x \mid \varphi_x = \mathbf{o}\}$?
20) $\{x \mid E_x = \{1,2\}\}$?
21) $\{x \mid x$ непарне$\} \oplus \{x \mid D_x \neq\varnothing\}$?
22) $D \otimes \{x \mid 2x \in E_x\}$?

22. Чи будуть ЧРП такі предикати:

1) "$\varphi_x(y + 2)$ просте"?
2) "$\varphi_{2x}(3y)$ парне"?
3) "$(0,1) \in D_x^2$"?
4) "$\{2, y + 1\} \subseteq D_{x + 5y}$"?
5) "$\{x, y\} \subseteq E_{3x + 2y}$"?
6) "$\{0,1\} \subseteq D_x$"?
7) "$\{0,1\} \neq D_x$"?
8) "$\{0,1\} = D_x$"?
9) "$D_x = D_y$"?
10) "$D_x \neq D_y$"?
11) "$D_x = N$"?
12) "$E_x \neq N$"?
13) "$E_x = E_y$"?
14) "$E_x \neq E_y$"?
15) "$E_x \neq D_y$"?
16) "$E_x = D_y$"?

23. Доведіть теорему 5.5.1.

24. Вкажіть МНР-програму, яка для заданої РФ f обчислює $f(a)$ за $a + f(a) + 2$ кроки.

6. ЗВІДНОСТІ. ВІДНОСНА ОБЧИСЛЮВАНІСТЬ

Для доведення розв'язності чи нерозв'язності масових проблем часто використовують метод звідності одних проблем до інших.

Кажуть, що проблема α *зводиться* до проблеми β, якщо із розв'язності β випливає розв'язність α.

Отже, якщо нерозв'язна проблема α зводиться до проблеми β, то β також нерозв'язна.

Метод нумерацій дає змогу масові проблеми подавати за допомогою певних числових множин, тому далі розглядатимемо саме звідність множин.

Існують різні уточнення поняття звідності множини A до множини B. Ці уточнення відрізняються за способом використання та обсягом інформації про множину B, яку можна використати для вирішення питання про множину A.

6.1. *m*-звідність і 1-звідність

Спочатку розглянемо сильні звідності: m-звідність та її окремий випадок — 1-звідність.

Неформально m-звідність множини A до множини B означає, що для вирішення питання "$x \in A$" треба задати єдине питання до множини B, причому заздалегідь визначеним ефективним способом, який можна уточнити як певну рекурсивну функцію g, тобто питання "$g(x) \in B$".

Дамо точне визначення m-звідності.

Множина A *m-зводиться* до множини B, якщо існує РФ g така, що для всіх $x \in N$ маємо $x \in A \Leftrightarrow g(x) \in B$.

Цей факт записуватимемо у вигляді $A \leq_m B$, або g: $A \leq_m B$, якщо треба вказати, що саме РФ g m-зводить A до B.

Окремим випадком m-звідності є 1-звідність.

Множина A 1-*зводиться* до множини B, якщо існує ін'єктивна РФ g така, що для всіх $x \in N$ маємо $x \in A \Leftrightarrow g(x) \in B$.

Цей факт записуватимемо у вигляді $A \le_1 B$.

Розглянемо елементарні властивості m-звідності та 1-звідності.

$r1$) Якщо $A \le_1 B$, то $A \le_m B$.

$r2$) Відношення $\le_1$ і $\le_m$ рефлексивні й транзитивні.

$r3$) $A \le_m B \Leftrightarrow \overline{A} \le_m \overline{B}$; те саме вірне для $\le_1$.

Справді, якщо g: $A \le_m B$, то $x \in A \Leftrightarrow g(x) \in B$, тому $x \in \overline{A} \Leftrightarrow g(x) \in \overline{B}$.

$r4$) Якщо $A \le_m B$ і B є РМ, то A є РМ; те саме для $\le_1$.

Нехай g: $A \le_m B$; тоді $\chi_A(x) = \chi_B(g(x))$ — РФ, бо χ_B та g є РФ.

$r5$) Якщо $A \le_m B$ і B є РПМ, то A є РПМ; те саме для $\le_1$.

Нехай РФ g: $A \le_m B$; тоді $\chi_A^{ч}(x) = \chi_B^{ч}(g(x))$ — ЧРФ, бо $\chi_B^{ч}$ є ЧРФ.

$r6$) якщо A є нерекурсивна РПМ, то невірно $A \le_m \overline{A}$ і невірно $\overline{A} \le_m A$; те саме для $\le_1$.

На основі теореми Поста $\overline{A}$ не є РПМ. За $r5$) якщо $\overline{A} \le_m A$, то $\overline{A}$ є РПМ, тому невірно $\overline{A} \le_m A$. Звідси за $r3$) невірно $A \le_m \overline{A}$.

$r7$) $A \le_m N \Leftrightarrow A = N$; те саме для $\le_1$.

Нехай g: $A \le_m N$, тоді $x \in A \Leftrightarrow g(x) \in N$. Але $g(x) \in N$ вірне завжди;

$r8$) $A \le_m \varnothing \Leftrightarrow A = \varnothing$; те саме для $\le_1$.

За $r3$) $A \le_m \varnothing \Leftrightarrow \overline{A} \le_m N$; за $r7$) $\overline{A} \le_m N \Leftrightarrow \overline{A} = N$, звідки $A = \varnothing$.

$r9$) $N \le_m A \Leftrightarrow A \ne \varnothing$.

Якщо РФ g: $N \le_m A$, то $A \supseteq E_g \ne \varnothing$. Якщо $A \ne \varnothing$, то зафіксуємо $a \in A$ і покладемо $g(x) = a$ для всіх $x \in N$; тоді g: $N \le_m A$.

$r10$) $\varnothing \le_m A \Leftrightarrow A \ne N$.

За $r3$) $\varnothing \le_m A \Leftrightarrow N \le_m \overline{A}$; за $r9$) $N \le_m \overline{A} \Leftrightarrow \overline{A} \ne \varnothing$, звідки $A \ne N$.

$r11$) $N \le_1 A \Leftrightarrow A$ містить нескінченну РПМ.

Нехай g: $N \le_1 A$. Тоді $x \in N \Leftrightarrow g(x) \in A$, звідки $E_g \subseteq A$. Але E_g є нескінченною РПМ як область значень ін'єктивної РФ g. Якщо L — нескінченна РПМ і $L \subseteq A$, то $L = E_g$ для деякої ін'єктивної РФ g. Тоді $g(x) \in A$ для всіх $x \in N$, звідки g: $N \le_1 A$.

$r12$) Якщо A рекурсивна і $B \ne \varnothing$ та $B \ne N$, то $A \le_m B$.

Виберемо $b \in B$ і $a \notin B$. Тоді РФ $g(x) = b \cdot \chi_A(x) + a \cdot nsg(\chi_A(x))$ m-зводить A до B.

$r13$) Для довільної множини B маємо $A \le_m A \oplus B$ і $A \le_m B \oplus A$.

Справді, $x \in A \Leftrightarrow 2x \in A \oplus B$ і $x \in A \Leftrightarrow 2x + 1 \in B \oplus A$.

Отже, для $f(x) = 2x$ і $g(x) = 2x + 1$ маємо f: $A \le_m A \oplus B$ та g: $A \le_m B \oplus A$.

$r14$) Для довільної множини $B \ne \varnothing$ маємо $A \le_m A \otimes B$ і $A \le_m B \otimes A$.

Візьмемо довільний $b \in B$. Нехай $f(x) = C(x, b)$, $g(x) = C(b, x)$. Тоді $x \in A \Leftrightarrow f(x) \in A \otimes B$ і $x \in A \Leftrightarrow g(x) \in B \otimes A$. Отже, f: $A \le_m A \otimes B$ і g: $A \le_m B \otimes A$;

$r15$) Якщо A є РПМ, то $A \le_m D$.

Пропонується довести самостійно як вправу.

Розглянемо кілька прикладів.

Приклад 6.1.1. Покажемо, що $\{x \mid \varphi_x = \mathbf{o}\} \equiv_m \{x \mid \varphi_x \text{ є РФ}\}$

Позначимо $A = \{x \mid \varphi_x = \mathbf{o}\}$ і $B = \{x \mid \varphi_x \text{ є РФ}\}$.

Розглянемо ЧРФ $f(x, y) = 0 - \varphi_x(y)$. За s-m-n-теоремою існує РФ $s(x)$ така, що $f(x, y) = \varphi_{s(x)}(y)$ для всіх x, y. Зафіксуємо x. Тоді для всіх y маємо: $\varphi_x(y) = 0 \Leftrightarrow \varphi_{s(x)}(y)\downarrow$. Звідси $\varphi_x = \mathbf{o} \Leftrightarrow \varphi_{s(x)}$ є РФ. Отже, $x \in A \Leftrightarrow s(x) \in B$. Тому РФ $s(x)$: $A \le_m B$.

Розглянемо ЧРФ $g(x, y) = \mathbf{o}(\varphi_x(y))$. За s-m-n-теоремою існує РФ $t(x)$ така, що $g(x, y) = \varphi_{t(x)}(y)$ для всіх x, y. Зафіксуємо x. Тоді для всіх y маємо: $\varphi_x(y)\downarrow \Leftrightarrow \varphi_{t(x)}(y) = 0$. Звідси φ_x є РФ $\Leftrightarrow \varphi_{t(x)} = \mathbf{o}$. Отже, $x \in B \Leftrightarrow t(x) \in A$. Тому РФ $t(x)$: $B \le_m A$.

Маємо $A \le_m B$ і $B \le_m A$. Звідси $A \equiv_m B$.

На множині всіх підмножин множини N уведемо відношення *m-еквівалентності*:

$$A \equiv_m B \Leftrightarrow A \le_m B \text{ та } B \le_m A.$$

Згідно з $r2$) відношення $\equiv_m$ є насправді відношенням еквівалентності. Тому введемо класи еквівалентності відносно $\equiv_m$:

$$d_m(A) = \{B \mid A \equiv_m B\}.$$

Такі класи еквівалентності називатимемо *m-степенями*.

Записуватимемо $A <_m B$, якщо $A \le_m B$ і невірно $B \le_m A$.

Писатимемо $A|_m B$, якщо невірно $A \le_m B$ і невірно $B \le_m A$.

Відношення $\le_m$ індукує на множині m-степенів відношення $\le_{\boldsymbol{m}}$:

$\boldsymbol{a} \le_{\boldsymbol{m}} \boldsymbol{b}$, якщо $A \le_m B$ для деяких $A \in \boldsymbol{a}$, $B \in \boldsymbol{b}$.

Легко бачити, що $\boldsymbol{a} \le_{\boldsymbol{m}} \boldsymbol{b} \Leftrightarrow A \le_m B$ для всіх $A \in \boldsymbol{a}$, $B \in \boldsymbol{b}$.

Відношення $\le_{\boldsymbol{m}}$ на множині m-степенів є відношенням часткового порядку.

Справді, рефлексивність і транзитивність маємо за $r1$), тому залишилось показати антисиметричність.

Маємо $\boldsymbol{a} \le_{\boldsymbol{m}} \boldsymbol{b}$ і $\boldsymbol{b} \le_{\boldsymbol{m}} \boldsymbol{a} \Leftrightarrow A \le_m B$ і $B \le_m A$ для деяких $A \in \boldsymbol{a}$ і $B \in \boldsymbol{b} \Leftrightarrow A \equiv_m B$ для деяких $A \in \boldsymbol{a}$ та $B \in \boldsymbol{b} \Leftrightarrow \boldsymbol{a} = \boldsymbol{b}$.

Записуватимемо $\boldsymbol{a} <_m \boldsymbol{b}$, якщо $\boldsymbol{a} \le_m \boldsymbol{b}$ і $\boldsymbol{a} \ne \boldsymbol{b}$.

Записуватимемо $\boldsymbol{a}|_m \boldsymbol{b}$, якщо невірно $\boldsymbol{a} \le_m \boldsymbol{b}$ і невірно $\boldsymbol{b} \le_m \boldsymbol{a}$.

Аналогічно вводиться відношення 1-*еквівалентності* $\equiv_1$, визначаються 1-степені, вводиться відношення часткового порядку $\le_1$ на множині 1-степенів.

Зрозуміло, що кожний m-степінь складається із 1-степенів.

m-степінь *рекурсивний*, якщо він містить РМ.

m-степінь *рекурсивно перелічний* (РП), якщо він містить РПМ.

Аналогічно визначаємо рекурсивні та РП 1-степені.

Із $r4$) і $r5$) випливає, що кожний рекурсивно перелічний m-степінь складається тільки з РПМ, кожний рекурсивний m-степінь — тільки з РМ.

Те саме вірне для 1-степенів.

Згідно з $r7$) – $r10$) існують 2 специфічні рекурсивні m-степені, які складаються з єдиної множини: $\mathbf{0} = d_m(\varnothing) = \{\varnothing\}$ і $\boldsymbol{n} = d_m(N) = \{N\}$.

Згідно з $r4$) і $r12$) усі інші РМ утворюють рекурсивний m-степінь $\mathbf{0}_m$.

Визначимо також рекурсивно перелічний m-степінь $\mathbf{0}'_m = d_m(D)$.

Ураховуючи властивості $r4$), $r5$), $r7$), $r8$), $r12$) і $r15$), дістаємо такі елементарні властивості m-степенів:

$d1$) $\mathbf{0}_m \le_m \boldsymbol{a}$ для всіх m-степенів $\boldsymbol{a} \ne \mathbf{0}, \ne \boldsymbol{n}$;

$d2$) $\boldsymbol{n} \le_m \boldsymbol{a}$ для всіх m-степенів $\boldsymbol{a} \ne \mathbf{0}$;

$d3$) $\mathbf{0} \le_m \boldsymbol{a}$ для всіх m-степенів $\boldsymbol{a} \ne \boldsymbol{n}$;

$d4$) якщо $\boldsymbol{a} \le_m \boldsymbol{b}$ і m-степінь $\boldsymbol{b}$ рекурсивно перелічний, то $\boldsymbol{a}$ — рекурсивно перелічний m-степінь;

$d5$) існує найбільший рекурсивно перелічний m-степінь $\mathbf{0}'_m$ такий, що $\boldsymbol{b} \le \mathbf{0}'_m$ для кожного рекурсивно перелічного m-степеня $\boldsymbol{b}$.

Точною верхньою гранню, або *сюпремумом,* m-степенів $\boldsymbol{a}$ і $\boldsymbol{b}$ (позначаємо $\boldsymbol{a} \cup \boldsymbol{b}$) називають m-степінь $\boldsymbol{c}$ такий, що:

- $\boldsymbol{a} \le_m \boldsymbol{c}$ і $\boldsymbol{b} \le_m \boldsymbol{c}$;
- $\boldsymbol{c} \le_m \boldsymbol{d}$ для кожного m-степеня $\boldsymbol{d}$ такий, що $\boldsymbol{a} \le_m \boldsymbol{d}$ і $\boldsymbol{b} \le_m \boldsymbol{d}$.

Теорема 6.1.1 (про сюпремум). *Для кожної пари m-степенів $\boldsymbol{a}$ та $\boldsymbol{b}$ існує єдина точна верхня грань.*

Покладемо $\boldsymbol{c} = d_m(A \oplus B)$, де $A \in \boldsymbol{a}$, $B \in \boldsymbol{b}$. Тоді функція $f(x) = 2x$ m-зводить A до $A \oplus B$, функція $g(x) = 2x + 1$ m-зводить B до $A \oplus B$. Отже, $\boldsymbol{a} \le_m \boldsymbol{c}$, $\boldsymbol{b} \le_m \boldsymbol{c}$.

Зауважимо, що коли $\boldsymbol{a}$ та $\boldsymbol{b}$ — рекурсивно перелічні m-степені, то m-степінь $\boldsymbol{c}$ також рекурсивно перелічний.

Нехай $\boldsymbol{d}$ — довільний m-степінь такий, що $\boldsymbol{a} \le_m \boldsymbol{d}$ і $\boldsymbol{b} \le_m \boldsymbol{d}$. Нехай $M \in \boldsymbol{d}$, f та g — такі РФ, що f: $A \le_m M$ і g: $B \le_m M$.

Маємо $x \in A \oplus B \Leftrightarrow x$ парне та $x/2 \in A$ або x непарне та $(x+1)/2 \in B \Leftrightarrow x$ парне та $f(x/2) \in M$ або x непарне та $g((x-1)/2) \in M$.

$$\text{Отже, РФ } h(x) = \begin{cases} f(x/2), & \text{якщо } x \text{ парне,} \\ g((x-1)/2), & \text{якщо } x \text{ непарне,} \end{cases}$$

m-зводить $A \oplus B$ до M. Тому $\boldsymbol{c} = d_m(A \oplus B) \le_m \boldsymbol{d}$. Звідси $\boldsymbol{c}$ — точна верхня грань m-степенів $\boldsymbol{a}$ та $\boldsymbol{b}$ ■

Приклад 6.1.2. Покажемо: якщо $\boldsymbol{a} \le_m \boldsymbol{b}$, то $\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{b}$.

Нехай $\boldsymbol{a} \le_m \boldsymbol{b}$. Тоді існує РФ f: $A \le_m B$ для деяких $A \in \boldsymbol{a}$ та $B \in \boldsymbol{b}$. Задамо

$$\text{РФ } g(x) = \begin{cases} f(x/2), & \text{якщо } x \text{ парне,} \\ (x-1)/2, & \text{якщо } x \text{ непарне.} \end{cases}$$

Тоді g: $A \oplus B \le_m B$.

Але $B \le_m A \oplus B$, звідки $A \oplus B \equiv_m B$. Отже, $\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{b}$.

6.2. Формалізація відносної обчислюваності. Релятивізація теорем

У цьому розділі розглянемо важливе поняття відносної обчислюваності та тісно пов'язане з ним поняття тьюрінгової звідності, або T-звідності.

Обмежимося розглядом відносної обчислюваності n-арних функцій на N, причому обчислюваності відносно тотальних функцій.

Неформально кажучи, функція f обчислювана відносно тотальної функції α, яку називають оракулом, якщо існує алгоритм для обчислення f, який може при необхідності брати потрібні значення функції α.

Формально поняття відносної обчислюваності уточнимо через поняття *МНР з оракулом* (МНРО).

Порівняно із МНР, МНРО використовують новий тип команд $O(n)$ — звернення до оракула. Для виконання таких команд МНРО мусить з'єднатися з певним оракулом α.

Виконання команди $O(n)$ означає, що $'R_n := \alpha('R_n)$, тобто вміст n-го регістра засилається в оракул α, який повертає в n-й регістр значення функції α від цього вмісту. Після виконання команди $O(n)$ далі виконується чергова за списком команда програми МНРО.

Програмою МНРО називають скінченну послідовність команд МНРО.

Зрозуміло, що смисл МНРО-програми залежить від конкретного оракула. Тому МНРО-програму P, яка виконується МНРО з оракулом α, позначатимемо P^α.

МНРО-програма P *обчислює* функцію f: $N^n \to N$ *відносно оракула* α, або α-*обчислює* функцію f, якщо

$$f(a_1, a_2, ..., a_n) = b \Leftrightarrow P^\alpha(a_1, a_2, ..., a_n)\downarrow b.$$

Функція f *МНРО-обчислювана відносно* α, або α-*обчислювана*, якщо існує МНРО-програма P, яка обчислює f відносно α.

Дещо інший підхід до відносної обчислюваності веде до поняття α-ЧРФ.

Функцію називають *частково рекурсивною відносно* α, або α-*ЧРФ*, якщо її отримують із функцій $\mathbf{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$ і α за допомогою скінченної кількості застосувань операцій $\boldsymbol{S}^{n+1}$, $\boldsymbol{R}$ і $\boldsymbol{M}$.

Тотальну α-ЧРФ називають α-РФ.

Аналогічно нерелятивному випадку, можна довести еквівалентність формальних понять α-обчислюваності і часткової рекурсивності відносно α:

Теорема 6.2.1. *Функція $f \in \alpha$-ЧРФ $\Leftrightarrow$ функція f МНРО-обчислювана відносно* α.

Клас усіх α-ЧРФ позначимо ЧРФ^α.

Наведемо деякі елементарні властивості α-ЧРФ.

о1) $\alpha \in \text{ЧРФ}^\alpha$;

о2) Для довільного оракула α маємо $\text{ЧРФ} \subseteq \text{ЧРФ}^\alpha$.

о3) Якщо тотальна функція φ є α-ЧРФ, то $\text{ЧРФ}^\varphi \subseteq \text{ЧРФ}^\alpha$.

Справді, при обчисленні $f \in \text{ЧРФ}^\varphi$ виконання кожного звернення до оракула φ зводиться до виконання скінченної кількості команд для

обчислення потрібного значення φ, серед яких можуть бути команди звернення до α. Тому f — α-обчислювана і є α-ЧРФ.

о4) якщо α рекурсивна, то $\text{ЧРФ}^{\alpha} = \text{ЧРФ}$.

Безпосередньо випливає із о3).

Про інші уточнення поняття відносної обчислюваності можна прочитати в [12], а також у [9, 13, 39].

Для відносно обчислюваних функцій можна сформулювати релятивний аналог тези Чорча, який називають тезою Тьюрінга.

Теза Тьюрінга. **Клас α-ЧРФ збігається з класом *n*-арних функцій на *N*, алгоритмічно обчислюваних відносно α.**

Тезу Чорча можна розглядати як окремий випадок тези Тьюрінга.

Ефективну нумерацію *n*-арних α-ЧРФ можна ввести на основі кодування МНРО-програм аналогічно відповідній нумерації *n*-арних ЧРФ.

Кодування команд МНРО можна задати так:

$$\theta(Z(n)) = 5\cdot n;$$

$$\theta(S(n)) = 5\cdot n + 1;$$

$$\theta(T(m, n)) = 5\cdot C(m, n) + 2;$$

$$\theta(J(m, n, q + 1)) = 5\cdot C(C(m, n), q) + 3;$$

$$\theta(O(n)) = 5\cdot n + 4.$$

Вживатимемо позначення $\varphi_m^{\alpha,n}$ для *n*-арної α-ЧРФ з індексом m, $D_m^{\alpha,n}$ — для області визначення $\varphi_m^{\alpha,n}$, $E_m^{\alpha,n}$ для області значень $\varphi_m^{\alpha,n}$.

Якщо $n = 1$, то відповідно застосовуватимемо позначення φ_m^{α}, D_m^{α}, E_m^{α}.

Множину L називають α-*РМ*, якщо χ_L є α-РФ.

Множину L називають α-*РПМ*, якщо $L = \varnothing$ або $L = E_f$ для деякої α-рекурсивної функції f.

Предикат P називають α-*РП*, якщо χ_P є α-РФ.

Предикат P називають α-*ЧРП*, якщо $\chi_P^{\text{ч}}$ є α-ЧРФ.

Майже дослівним повторенням доведень відповідних теорем розділів 4 і 5 можна довести їх релятивні варіанти.

Наведемо релятивні варіанти основних теорем.

*R*1) Релятивна *s-m-n*-теорема. Для довільних $m, n > 1$ існує $(m + 1)$-арна РФ $s_n^m(z, x_1, ..., x_m)$ така, що для всіх $z, x_1, ..., x_m, y_1, ..., y_n$ маємо $\varphi_z^{\alpha,m+n}(x_1, ..., x_m, y_1, ..., y_n) = \varphi_{s_n^m(z,x_1,...,x_m)}^{\alpha,n}(y_1, ..., y_n)$.

*R*2) Релятивна *s-m-n*-теорема (спрощена форма). Для кожної α-ЧРФ $f(x, y)$ існує РФ $s(x)$ така, що для всіх x, y маємо $f(x, y) = \varphi_{s(x)}^{\alpha}(y)$.

*R*3) Функція, універсальна для класу *n*-арних α-РФ, не є α-ЧРФ.

*R*4) Існує α-ЧРФ, універсальна для класу *n*-арних α-ЧРФ.

*R*5) Релятивна теорема Кліні про нерухому точку.

Нехай f — $(n + 1)$-арна РФ. Тоді існує *n*-арна РФ g така: для всіх $x_1, ..., x_n$ маємо $\varphi_{g(x_1, ..., x_n)}^{\alpha} = \varphi_{f(g(x_1, ..., x_n),\ x_1, ...x_n)}^{\alpha}$.

*R*6) Релятивна теорема Поста.

Якщо множини L і $\overline{L}$ є α-РПМ, то L і $\overline{L}$ є α-РМ.

*R*7) Наступні визначення α-РПМ еквівалентні:

*df*1) $L = \varnothing$ або L є областю значень деякої α-РФ;

*df*2) L є областю значень деякої α-ЧРФ;

*df*3) L є областю визначення деякої α-ЧРФ;

*df*4) часткова характеристична функція множини L є α-ЧРФ.

*R*8) Предикат $Q(x_1, ..., x_n)$ є α-ЧРП тоді і тільки тоді, коли існує α-РП $R(x_1, ..., x_n, y)$ такий, що $Q(x_1, ..., x_n) \Leftrightarrow \exists y R(x_1, ..., x_n, y)$.

*R*9) Якщо $Q(x_1, ..., x_n, y)$ є α-ЧРП, то предикат $\exists y_1 ... \exists y_k Q(x_1, ..., x_n, y_1, ..., y_k)$ теж є α-ЧРП.

*R*10) Множина $D^{\alpha} = \{x \mid \varphi_x^{\alpha}(x)$ визначене$\}$ є α-РПМ і не є α-РМ.

*R*11) Множина $\overline{D^{\alpha}} = \{x \mid \varphi_x^{\alpha}(x)$ невизначене$\}$ не є α-РПМ.

Обчислюваність відносно довільної множини B визначають як обчислюваність відносно її характеристичної функції χ_B.

Функцію називають *B-рекурсивною*, якщо вона χ_B-рекурсивна.

Функцію називають *B-ЧРФ*, якщо вона χ_B-ЧРФ.

Множину A називають *B-рекурсивною*, якщо $\chi_A \in \chi_B$-РФ.

Множину A називають *B-РПМ*, якщо $\chi_A^{\text{ч}} \in \chi_B$-ЧРФ.

Предикат P називають *B-рекурсивним*, якщо $\chi_P \in \chi_B$-РФ.

Предикат P називають *B-ЧРП*, якщо $\chi_P^{\text{ч}} \in \chi_B$-ЧРФ.

Функцію $\varphi_m^{\chi_B, n}$ і множину $D_m^{\chi_B, n}$ позначаємо $\varphi_m^{B,n}$ та $D_m^{B,n}$.

Якщо $n = 1$, то відповідно застосовуватимемо позначення φ_m^B та D_m^B.

Класи функцій ЧРФ$^{\chi_B}$ і РФ$^{\chi_B}$ позначатимемо ЧРФ^B і РФ^B.

Теорема 6.2.2. 1. *Множина $A \in \overline{A}$-РМ.*

2. *Якщо $A \in B$-РМ і $B \in C$-РМ, то $A \in C$-РМ.*

3. *Якщо $A \in B$-РПМ і $B \in C$-РМ, то $A \in C$-РПМ.*

4. *Якщо $A \in B$-РМ і $B \in C$-РПМ, то не завжди $A \in C$-РПМ.*

Доводимо 1. Маємо $\chi_{\overline{A}}(x) = nsg(\chi_A(x))$, тому $\overline{A} \in A$-РМ.

Доводимо 2. Якщо $B \in C$-РМ, то $\chi_B \in \chi_C$-РФ, звідки ЧРФ$^B \subseteq$ЧРФ^C. Але $A \in B$-РМ, тобто $\chi_A \in$ ЧРФ^B, звідки $\chi_A \in$ ЧРФ^C.

Доводимо 3. Якщо $B \in C$-РМ, то ЧРФ$^B \subseteq$ЧРФ^C. Але $A \in B$-РПМ, тому маємо $\chi_A^{\text{ч}} \in$ ЧРФ$^B \subseteq$ЧРФ^C.

Доводимо 4. Візьмемо $A = \overline{D^C}$ і $B = D^C$. Тоді $\overline{D^C} \in D^C$-РМ згідно 1 та $D^C \in C$-РПМ за $R10$, але за $R11$ $\overline{D^C}$ не є C-РПМ ■

6.3. *T*-звідність

Поняття m-звідності має кілька патологічних властивостей: специфічна поведінка множин $\varnothing$ і N, не завжди $A \equiv_m \overline{A}$. Така неприємна ситуація виникає внаслідок обмеженості природи m-звідності: g: $A \leq_m B$, якщо для вирішення питання "$x \in A$" треба задати єдине питання до B, причому заздалегідь зазначеним способом "$g(x) \in B$".

Найадекватніше інтуїтивне поняття звідності відбиває поняття тьюрінгової звідності, або T-звідності. Поняття T-звідності тісно пов'язане із поняттям відносної обчислюваності.

Неформально кажучи, множина A T-зводиться до множини B, що позначаємо $A \leq_T B$, якщо для вирішення питання "$x \in A$" необхідно відповісти на скінченну кількість питань про B, але їх кількість і природа заздалегідь невідомі.

Отже, отримуємо таке визначення.

Множина A *T-зводиться* до множини B, якщо множина $A \in B$-рекурсивною.

Цей факт позначатимемо $A \leq_T B$.

Уведемо відношення *T-еквівалентності* $\equiv_T$:

$$A \equiv_T B, \text{ якщо } A \leq_T B \text{ і } B \leq_T A.$$

Писатимемо $A <_T B$, якщо $A \le_T B$ та невірно, що $B \le_T A$.

Писатимемо $A|_T B$, якщо невірно $A \le_T B$ та невірно $B \le_T A$.

Визначення T-звідності цілком узгоджується з інтуїтивним поняттям звідності.

Справді, нехай маємо МНРО-програму P^B з оракулом χ_B для обчислення χ_A. За тотальністю χ_A при обчисленні кожного значення $\chi_A(x)$ виконується скінченна кількість команд P^B, деякі з них можуть бути командами звернення до оракула, тобто питаннями вигляду "$z \in B$". Звідси маємо:

$$x \in A \Leftrightarrow \chi_A(x) = 1 \Leftrightarrow P^B(x)\downarrow 1;$$
$$x \notin A \Leftrightarrow \chi_A(x) = 0 \Leftrightarrow P^B(x)\downarrow 0.$$

Отже, для вирішення питання "$x \in A$" використовується скінченна кількість питань про B.

Наведемо елементарні властивості T-звідності.

t1) $A \le_T A$.

t2) Якщо $A \le_T B$ і $B \le_T C$, то $A \le_T C$.

Випливає із п. 2 теореми 6.2.2.

t3) Для кожної множини A маємо $A \le_T \overline{A}$ і $\overline{A} \le_T A$.

Випливає із п. 1 теореми 6.2.2.

t4) $A \equiv_T A$ для кожної множини A.

Випливає із t3).

t5) якщо $A \le_m B$, то $A \le_T B$.

Нехай РФ g: $A \le_m B$. Тоді $\chi_A(x) = \chi_B(g(x))$, звідки $\chi_A \in \chi_B$-РФ.

t6) Якщо $B \in$ РМ і $A \le_T B$, то $A \in$ РМ.

$\chi_B \in$ РФ, тому ЧРФ^B = ЧРФ і РФ^B = РФ.

Якщо $A \le_T B$, то $\chi_A \in$ РФ^B = РФ.

t7) Якщо $A \in$ РМ, то $A \le_T B$ для кожної множини B.

За умовою $\chi_A \in$ РФ. Ураховуючи о2), для довільної B маємо $\chi_A \in$ РФ $\subseteq$ ЧРФ $\subseteq$ ЧРФ^B, звідки $A \in B$-рекурсивною.

t8) Якщо $A \in$ РПМ, то $A \le_T D$.

За r15) $A \le_m D$ для кожної РПМ A, звідки за t5) маємо $A \le_T D$.

Теорема 6.3.1. *Множина $B \in A$-РПМ* $\Leftrightarrow B \le_m D^A$.

Нехай $B \in A$-РПМ. Функція $f(x, y) = \chi_B^{\text{ч}}(x) + \mathbf{o}(y) \in A$-ЧРФ, бо $\chi_B^{\text{ч}} \in A$-ЧРФ. За релятивною s-m-n-теоремою існує РФ s така:

$f(x, y) = \varphi^{A}_{s(x)}(y)$ для всіх x, y. При $x \in B$ маємо $\varphi^{A}_{s(x)}(y) = 1$ для всіх y, звідки $\varphi^{A}_{s(x)}(s(x))\downarrow$, тому $s(x) \in D^A$. При $x \notin B$ $\varphi^{A}_{s(x)}(y)\uparrow$ для всіх y, тому $\varphi^{A}_{s(x)}(s(x))\uparrow$, звідки $s(x) \notin D^A$.

Отже, $x \in B \Leftrightarrow s(x) \in D^A$, тому $B \leq_m D^A$.

Нехай РФ f: $B \leq_m D^A$. Тоді $x \in B \Leftrightarrow s(x) \in D^A$.

Але D^A є A-РПМ, f є РФ, звідки предикат "$x \in B$" є A-ЧРП.

Отже, B є A-РПМ ■

Наслідок 1. *Якщо B є A-РПМ, то $B \leq_T D^A$.*

Наслідок 2. *$A <_T D^A$ для кожної множини A.*

Маємо $A \leq_T D^A$, бо A є A-РПМ. Згідно з R10) D^A не є A-РМ, тому невірно $D^A \leq_T A$ ■

Наведемо кілька прикладів.

Приклад 6.3.1. Існують множини A та B: $A <_T A \otimes B$ і $B \equiv_m A \otimes B$.

Наприклад, візьмемо $A = N$ і $B = D$.

Приклад 6.3.2. Покажемо, що $A \otimes B \leq_T A \oplus B$.

Маємо $x \in A \otimes B \Leftrightarrow \boldsymbol{l}(x) \in A$ & $\boldsymbol{r}(x) \in B \Leftrightarrow 2\boldsymbol{l}(x) \in A \oplus B$ & $2\boldsymbol{r}(x) + 1 \in A \oplus B$.

Отже, $\chi_{A\otimes B}(x) = \chi_{A\oplus B}(2\boldsymbol{l}(x)) \cdot \chi_{A\oplus B}(2\boldsymbol{r}(x) + 1)$. Тому $\chi_{A\otimes B}$ є $\chi_{A\oplus B}$-РФ.

Приклад 6.3.3. Покажемо, що $D \equiv_T \bar{D} \equiv_T D \oplus \bar{D} \equiv_T D \otimes \bar{D}$.

За t4) $D \equiv_T \bar{D}$. За r13) $D \leq_m D \oplus \bar{D}$, за r14) $D \leq_m D \otimes \bar{D}$, тому за t5) $D \leq_T D \oplus \bar{D}$ і $D \leq_T D \otimes \bar{D}$. З огляду t2) досить $D \oplus \bar{D} \leq_T D$ і $D \otimes \bar{D} \leq_T D$.

Маємо $x \in D \oplus \bar{D} \Leftrightarrow (x$ парне та $x/2 \in D) \vee (x$ непарне та $(\tilde{x}1)/2 \notin D)$. Тому "$x \in D \oplus \bar{D}$" є D-РП, звідки $\chi_{D\oplus\bar{D}}$ є χ_D-РФ. Але "$x \in D \otimes \bar{D}$" також D-РП, бо $x \in D \otimes \bar{D} \Leftrightarrow \boldsymbol{l}(x) \in D$ & $\boldsymbol{r}(x) \notin D$. Отже, $\chi_{D\otimes\bar{D}}$ є χ_D-РФ.

Відношення $\equiv_T$ є відношенням еквівалентності, тому вводимо класи еквівалентності $d_T(A) = \{B \mid A \equiv_T B\}$ відносно $\equiv_T$.

Такі класи називають *T-степенями*, або *степенями нерозв'язності*.

T-степінь *рекурсивний*, якщо він містить РМ.

T-степінь *рекурсивно перелічний*, або *РП-T-степінь*, якщо він містить РПМ.

На множині T-степенів уведемо відношення часткового порядку, яке також позначатимемо $\leq$:

$\boldsymbol{a} \leq \boldsymbol{b}$, якщо $A \leq_T B$ для деяких $A \in \boldsymbol{a}$, $B \in \boldsymbol{b}$.

Зрозуміло, що $\boldsymbol{a} \leq \boldsymbol{b} \Leftrightarrow A \leq_T B$ для всіх $A \in \boldsymbol{a}$, $B \in \boldsymbol{b}$.

Записуватимемо $\boldsymbol{a} < \boldsymbol{b}$, якщо $\boldsymbol{a} \leq \boldsymbol{b}$ та $\boldsymbol{a} \neq \boldsymbol{b}$.

Записуватимемо $\boldsymbol{a} \mid \boldsymbol{b}$, якщо невірно $\boldsymbol{a} < \boldsymbol{b}$ і невірно $\boldsymbol{b} \leq \boldsymbol{a}$.

Вкажемо деякі властивості T-степенів:

s1) Існує єдиний рекурсивний T-степінь $\mathbf{0}$, який складається з усіх РМ. Він є найменшим T-степенем: $\mathbf{0} < \boldsymbol{b}$ для кожного T-степеня $\boldsymbol{b} \neq \mathbf{0}$.

s2) Існує найбільший рекурсивно перелічний T-степінь $\mathbf{0}' = d_T(D)$ така, що $\boldsymbol{b} \leq \mathbf{0}'$ для кожного рекурсивно перелічного T-степеня $\boldsymbol{b}$.

s3) Кожний нерекурсивний РП-степінь містить множини, які не є РПМ.

s4) Якщо $d_m(A) \leq_m d_m(B)$, то $d_T(A) \leq_T d_T(B)$.

s5) $d_m(A) \subseteq d_T(A)$ для довільної множини A.

Твердження **s1)** випливає із t6) і t7); твердження **s2)** — із t8); твердження **s3)** — із t4; твердження **s4**) і **s5)** випливають із t5).

У 1944 р. Е. Пост поставив проблему: чи існує рекурсивно перелічна T-степінь $\boldsymbol{b}$ така, що $\mathbf{0} < \boldsymbol{b} < \mathbf{0}'$? Позитивна відповідь була отримана тільки в 1956 р. (про це можна прочитати в [7, 9, 12]).

Теорема 6.3.2 (Мучник, Фрідберг). *Існують РПМ A та B*: $A \mid_T B$.

Визначивши T-степені $\boldsymbol{a} = d_T(A)$ і $\boldsymbol{b} = d_T(B)$ для РПМ A і B із теореми, отримуємо наслідок.

Наслідок. *Існують РП-T-степені $\boldsymbol{a}$ та $\boldsymbol{b}$ такі*: $\mathbf{0} < \boldsymbol{a} < \mathbf{0}'$, $\mathbf{0} < \boldsymbol{b} < \mathbf{0}'$ *і* $\boldsymbol{a} \mid \boldsymbol{b}$.

Для T-степенів справджується теорема про сюпремум.

Теорема 6.3.3. *Для кожної пари T-степенів $\boldsymbol{a}$ та $\boldsymbol{b}$ існує єдина точна верхня грань $\boldsymbol{a} \cup \boldsymbol{b} = d_T(A \oplus B)$, де $A \in \boldsymbol{a}$, $B \in \boldsymbol{b}$.*

Маємо $A \leq_m A \oplus B$ і $B \leq_m A \oplus B$, тому за t5) $A \leq_T A \oplus B$ та $B \leq_T A \oplus B$. Звідси $\boldsymbol{a} \leq \boldsymbol{a} \cup \boldsymbol{b}$ і $\boldsymbol{b} \leq \boldsymbol{a} \cup \boldsymbol{b}$.

Нехай $\boldsymbol{d}$ — довільний T-степінь такий, що $\boldsymbol{a} \leq \boldsymbol{d}$ та $\boldsymbol{b} \leq \boldsymbol{d}$.

Тоді для довільних $A \in \boldsymbol{a}$, $B \in \boldsymbol{b}$ і $L \in \boldsymbol{d}$ маємо, що A і B є L-РМ. Але $x \in A \oplus B \Leftrightarrow x$ парне та $x/2 \in A$ або x непарне та $(x-1)/2 \in B$, тому функція $\chi_{A \oplus B}$ є L-РФ. Звідси $\boldsymbol{a} \cup \boldsymbol{b} \leq \boldsymbol{d}$ ■

6.4. Операція стрибка

Згідно з наслідком 2 теореми 6.3.1 маємо: $A <_T D^A$ для кожної множини A.

Неформально $A <_T D^A$ означає, що при переході від A до D^A складність множини стрибкоподібно зростає, тому D^A називають *стрибком* множини A.

Операцію, яка з кожною множиною $A \subseteq N$ зіставляє множину D^A, називають *операцією стрибка*.

Теорема 6.4.1. $A \le_T B \Leftrightarrow D^A \le_m D^B$.

Нехай $A \le_T B$. Тоді A є B-РМ. Але D^A є A-РПМ, тому за теоремою 6.2.3 D^A є B-РПМ. За теоремою 6.3.1 $D^A \le_m D^B$.

Нехай $D^A \le_m D^B$. Позаяк A та $\overline{A}$ є A-РПМ, за теоремою 6.3.1 маємо $A \le_m D^A$ та $\overline{A} \le_m D^A$. Урахувавши $D^A \le_m D^B$, маємо $A \le_m D^B$ та $\overline{A} \le_m D^B$.

Згідно з теоремою 6.3.1 множини A та $\overline{A}$ є B-РПМ, тому за релятивною теоремою Поста A є B-РМ. Отже, $A \le_T B$ ■

Наслідок 1. $A \equiv_T B \Leftrightarrow D^A \equiv_m D^B$.

Наслідок 2. *Якщо* $A \equiv_T B$, *то* $D^A \equiv_T D^B$.

Обернене до наслідку 2 твердження невірне (див. [12]). Можливі випадки $A <_T B$ та $A|_T B$, для яких також виконується $D^A \equiv_T D^B$.

Операцію стрибка поширимо на множину T-степенів.

Стрибком T-степені $\boldsymbol{b}$ називають степінь $\boldsymbol{b}' = d_T(D^B)$, де $B \in \boldsymbol{b}$.

Таке визначення коректне, бо за наслідком 2 теореми 6.4.1 $\boldsymbol{b}'$ не залежить від вибору конкретного представника $B \in \boldsymbol{b}$.

Наведемо деякі властивості операції стрибка.

*jm*1) $\boldsymbol{b} < \boldsymbol{b}'$ для довільного T-степеня $\boldsymbol{b}$.

Випливає із $B <_T D^B$ (наслідок 2 теореми 6.3.1).

*jm*2) Якщо $\boldsymbol{a} \le \boldsymbol{b}$, то $\boldsymbol{a}' \le \boldsymbol{b}'$.

Випливає із теореми 6.4.1.

*jm*3) $\boldsymbol{0} < \boldsymbol{b}'$ для довільного T-степеня $\boldsymbol{b}$.

*jm*4) Якщо $\boldsymbol{a} = \boldsymbol{b}$, то $\boldsymbol{a}' = \boldsymbol{b}'$.

Випливає із наслідка 2 теореми 6.4.1.

*jm*5) Якщо $A \in \boldsymbol{a}$, $B \in \boldsymbol{b}$ та B є A-РПМ, то $\boldsymbol{b} \le \boldsymbol{a}'$.

Випливає з теореми 6.3.1.

Для множин і степенів уведемо операцію *n-кратного стрибка*.

Для довільної $A \subseteq N$ покладемо

$$A^{(0)} = A,$$
$$A^{(k+1)} = D^{A^{(k)}}.$$

Для довільного T-степеня $\boldsymbol{a}$ покладемо

$$\boldsymbol{a}^{(0)} = \boldsymbol{a},$$
$$\boldsymbol{a}^{(k+1)} = (\boldsymbol{a}^{(k)})'.$$

Наведемо деякі властивості операції n-кратного стрибка.

Ураховуючи $A <_T D^A$ і $\boldsymbol{a} < \boldsymbol{a}'$, дістаємо:

*jn*1) $A^{(0)} <_T A^{(1)} <_T \ldots <_T A^{(k)} <_T A^{(k+1)} <_T \ldots$ для довільної $A \subseteq N$;

*jn*2) $\boldsymbol{a}^{(0)} < \boldsymbol{a}^{(1)} < \ldots < \boldsymbol{a}^{(k)} < \boldsymbol{a}^{(k+1)} < \ldots$ для довільного T-степеня $\boldsymbol{a}$;

*jn*3) Якщо $A \leq_T B$, то $A^{(n)} \leq_m B^{(n)}$ для всіх $n \geq 1$.

Властивість *jn*3 отримаємо, використовуючи n раз теорему 6.4.1.

Введемо операцію ω-стрибка.

ω-*стрибком* множини $A \subseteq N$ називають множину

$$A^{(\omega)} = \{C(x, y) \mid x \in A^{(y)}\}$$

ω-*стрибком* T-степені $\boldsymbol{a}$ називають T-степінь $\boldsymbol{a}^{(\omega)} = d_T(A^{(\omega)})$, де $A \in \boldsymbol{a}$.

Теорема 6.4.2. *$A^{(n)} <_T A^{(\omega)}$ для всіх A, n.*

Маємо $C(x, y) \in A^{(\omega)} \Leftrightarrow x \in A^{(y)}$. Отже, для кожного $n \in N$ маємо $C(x, n)$: $A^{(n)} \leq_1 A^{(\omega)}$, тому для всіх n $A^{(n)} \leq_T A^{(\omega)}$. Але неможливо $A^{(\omega)} \equiv_T A^{(k)}$ для деякого k, бо тоді $A^{(\omega)} \equiv_T A^{(k)} <_T A^{(k+1)} \leq_T A^{(\omega)}$ — суперечність. Отже, $A^{(n)} \leq_T A^{(\omega)}$ для всіх A, n ■

Теорема 6.4.3. *Якщо $A \leq_T B$, то $A^{(\omega)} \leq_m B^{(\omega)}$.*

Доведення можна прочитати в [5, 8].

Теорема 6.4.4. *Існують множини A і B такі: $A^{(\omega)} \leq_m B^{(\omega)}$ і $B <_T A$.*

Візьмемо довільні $B \subseteq N$ і $n > 0$. Покладемо $A = B^{(n)}$. Тоді $A^{(x)} = B^{(x+n)}$ для всіх x. Звідси $u \in A^{(\omega)} \Leftrightarrow \boldsymbol{l}(u) \in A^{(\boldsymbol{r}(u))} \Leftrightarrow \boldsymbol{l}(u) \in B^{(\boldsymbol{r}(u)+n)} \Leftrightarrow C(\boldsymbol{l}(u), \boldsymbol{r}(u) + n) \in B^{(\omega)}$. Отже, $A^{(\omega)} \leq_m B^{(\omega)}$. Але згідно з *jn*1 маємо $B <_T B^{(n)}$, тобто $B <_T A$ ■

Отже, обернене до теореми 6.4.5 твердження невірне.

Зауважимо, що для множин A та B із доведення теореми 6.4.6 маємо $A^{(\omega)} \equiv_m B^{(\omega)}$, адже $B^{(\omega)} \leq_m A^{(\omega)}$.

Структура T-степенів, зокрема РП-T-степенів, дуже складна. Наведемо для прикладу деякі результати (детальніше про це див. [12 і 39]).

Теорема 6.4.5. *Для кожного РП-Т-степеня* $\boldsymbol{a}$ *такого, що* $\mathbf{0} < \boldsymbol{a} < \mathbf{0}'$, *існує РП-Т-степінь* $\boldsymbol{b}$ *такий, що* $\boldsymbol{a} \mid \boldsymbol{b}$.

Теорема 6.4.6 (теорема щільності). *Для кожної пари РП-Т-степенів* $\boldsymbol{a}$ *та* $\boldsymbol{b}$ *існує РП-Т-степінь* $\boldsymbol{c}$ *такий, що* $\boldsymbol{a} < \boldsymbol{c} < \boldsymbol{b}$.

Т-степінь $\boldsymbol{m}$ *мінімальний,* якщо $\mathbf{0} < \boldsymbol{m}$ і не існує такого *Т*-степеня $\boldsymbol{a}$, що $\mathbf{0} < \boldsymbol{a} < \boldsymbol{m}$.

Теорема 6.4.7. *Існує мінімальний степінь* $\boldsymbol{m}$ *такий, що* $\boldsymbol{m} < \mathbf{0}'$.

Зауважимо, що за теоремою щільності така мінімальна *Т*-степінь ніяк не може бути рекурсивно перелічною!

Теорема 6.4.8 (теорема про розбиття). *Для кожної РП-Т-степені* $\boldsymbol{c}$ *існують РП-Т-степені* $\boldsymbol{a} < \boldsymbol{c}$ *і* $\boldsymbol{b} < \boldsymbol{c}$ *такі, що* $\boldsymbol{c} = \boldsymbol{a} \cup \boldsymbol{b}$.

Теорема 6.4.9 (С. Купер). *Для кожної Т-степені* $\boldsymbol{b} \geq \mathbf{0}'$ *існує мінімальний степінь* $\boldsymbol{m}$ *такий, що* $\boldsymbol{m}' = \boldsymbol{m} \cup \mathbf{0}' = \boldsymbol{b}$.

Упорядкування *Т*-степенів досить нетривіальне. Це ілюструють теореми С. Кліні та Е. Поста (доведення див. [12]).

Теорема 6.4.10. *Існує зліченна сукупність Т-степенів, розташованих між* $\mathbf{0}$ *і* $\mathbf{0}'$, *лінійно впорядкована за типом раціональних чисел.*

Теорема 6.4.11. *Існують Т-степені* $\boldsymbol{a}$ *та* $\boldsymbol{b}$ *такі*:
1) $\boldsymbol{a} < \mathbf{0}^{(\omega)}$, $\boldsymbol{b} < \mathbf{0}^{(\omega)}$ *та* $\boldsymbol{a} \mid \boldsymbol{b}$;
2) $\mathbf{0}^{(n)} < \boldsymbol{a}$ *та* $\mathbf{0}^{(n)} < \boldsymbol{b}$ *для кожного* $n \in N$;
3) *для кожного Т-степеня* $\boldsymbol{d}$ *такого, що* $\boldsymbol{d} \leq \boldsymbol{a}$ і $\boldsymbol{d} \leq \boldsymbol{b}$, *існує* $n \in N$ *таке*: $\boldsymbol{d} \leq \mathbf{0}^{(n)}$.

Наслідок. 1. *Т-степені* $\boldsymbol{a}$ *і* $\boldsymbol{b}$ *не мають найбільшої нижньої грані*.
2. $\mathbf{0}^{(\omega)}$ *не є найменшою верхньою гранню Т-степенів* $\mathbf{0}$, $\mathbf{0}'$, ..., $\mathbf{0}^{(n)}$, ...

Для подальшого ознайомлення з результатами теорії звідності та степенів можна рекомендувати [12; 13; 39].

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Як можна уточнити поняття звідності множини A до множини B?
2. Дайте визначення m-звідності.
3. Дайте визначення 1-звідності.
4. Наведіть елементарні властивості m-звідності.
5. Наведіть елементарні властивості 1-звідності.
6. Дайте визначення відношення m-еквівалентності.
7. Дайте визначення відношення 1-еквівалентності.
8. Що таке m-степінь? 1-степінь?
9. Опишіть елементарні властивості m-степенів.
10. Що таке рекурсивний m-степінь?
11. Що таке рекурсивно перелічний m-степінь?
12. Які ви знаєте рекурсивні m-степені?
13. Що таке сюпремум (точна верхня грань) m-степенів?
14. Сформулюйте теорему про сюпремум.
15. Як можна уточнити поняття відносної обчислюваності?
16. Дайте визначення МНР з оракулом.
17. Що таке програма МНРО?
18. Що таке α-обчислювана функція?
19. Що таке α-ЧРФ?
20. Наведіть елементарні властивості α-ЧРФ.
21. Сформулюйте тезу Тьюрінга.
22. Як задають кодування команд МНРО-програм?
23. Уведіть ефективну нумерацію n-арних α-ЧРФ.
24. Дайте визначення α-РФ, α-ЧРФ, α-РМ, α-РПМ, α-РП, α-ЧРП.
25. Сформулюйте релятивні варіанти теорем.
26. Як визначається обчислюваність відносно множини?
27. Дайте визначення A-РФ, A-ЧРФ, A-РМ, A-РПМ, A-РП, A-ЧРП.
28. Наведіть властивості обчислюваності відносно множини.
29. Дайте визначення T-звідності.
30. Чому поняття T-звідності найадекватніше відбиває інтуїтивне поняття звідності?
31. Наведіть елементарні властивості T-звідності.
32. Дайте визначення відношення T-еквівалентності.
33. Що таке T-степінь?
34. Охарактеризуйте елементарні властивості T-степенів.
35. Що таке рекурсивний T-степінь? Опишіть рекурсивні T-степені.

36. Що таке рекурсивно перелічний T-степінь?
37. Що таке сюпремум (точна верхня грань) T-степенів?
38. Сформулюйте теорему про сюпремум.
39. Як визначається операція стрибка на множинах?
40. Як визначається операція стрибка на степенях?
41. Наведіть властивості операції стрибка.
42. Як визначається операція n-кратного стрибка на множинах?
43. Як визначається операція n-кратного стрибка на степенях?
44. Наведіть властивості операції n-кратного стрибка.
45. Як визначається операція ω-стрибка на множинах?
46. Як визначається операція ω-стрибка на степенях?
47. Як співвідносяться операції n-кратного стрибка та ω-стрибка?
48. Наведіть властивості операції ω-стрибка.

ВПРАВИ

1. Доведіть:
   1) для кожної $B \subseteq N$ існує не більше ніж зліченна кількість множин $A \subseteq N$ таких, що $A \leq_m B$;
   2) що кожний m-степінь скінченний або зліченний;
   3) для кожного m-степеня $\boldsymbol{b}$ існує не більше ніж зліченна множина m-степенів $\boldsymbol{a}$ таких, що $\boldsymbol{a} \leq_{\boldsymbol{m}} \boldsymbol{b}$.

2. Оцініть потужності множин усіх m-степенів і всіх рекурсивно перелічних m-степенів.

3. Нехай $A = \{x \mid \varphi_x \text{ є поліном}\}$. Чи вірно $A \leq_m D$? $A \leq_m \overline{D}$?

4. Нехай $A = \{x \mid E_x \text{ є ПРМ}\}$. Чи вірно $A \leq_m D$? $A \leq_m \overline{D}$?

5. Доведіть $D \leq_m A$, якщо:
   1) $A = \{x \mid \varphi_x$ сюр'єктивна$\}$;
   2) $A = \{x \mid D_x \text{ є РМ}\}$;
   3) $A = \{x \mid \varphi_x \text{ є РФ}\}$;
   4) $A = \{x \mid D_x$ не є ПРМ$\}$;
   5) $A = \{x \mid \varphi_x$ не є ПРФ$\}$;
   6) $A = \{x \mid E_x$ скінченна$\}$.

6. Доведіть: якщо A та B — такі РПМ, що $A \cap B \neq \varnothing$ і $A \cup B = N$, то $A \leq_m A \cap B$ і $B \leq_m A \cap B$.

7. Доведіть, що $A \oplus A \equiv_1 \overline{A \oplus \overline{A}}$.

8. Доведіть, що $\boldsymbol{a} \cup \boldsymbol{b} = \boldsymbol{b} \cup \boldsymbol{a}$.

9. Встановіть, в якому відношенні щодо m-звідності перебувають множини D, $\overline{D}$, $D \oplus \overline{D}$ і $D \otimes \overline{D}$.

10. Доведіть релятивні варіанти теорем $R1 - R11$.

11. Чи існує РФ s така:

1) для всіх $x, y \in N$ $D^{\alpha}_{s(x,y)} = (D^{\alpha}_{x+y} \cap E^{\alpha}_{y}) \setminus \{3x, y+2\}$?

2) для всіх $x, y \in N$ $E^{\alpha}_{s(x,y)} = E^{\alpha}_{2x} \cup (D^{\alpha}_{y} \setminus E^{\alpha}_{2x})$?

3) для всіх $x, y, z \in N$ $D^{\alpha}_{s(x,y,z)} = E^{\alpha}_{y} \cup (E^{\alpha}_{z} \setminus D^{\alpha}_{x})$?

12. Доведіть:

1) для кожної $B \subseteq N$ існує зліченна кількість множин $A \subseteq N$ таких, що $A \leq_T B$;

2) кожний T-степінь — зліченна множина;

3) для кожного T-степеня $\boldsymbol{b}$ існує не більше ніж зліченна множина T-степенів $\boldsymbol{a}$ таких, що $\boldsymbol{a} \leq \boldsymbol{b}$.

13. Порівняйте потужності множини всіх T-степенів і множини всіх рекурсивно перелічних T-степенів.

14. Доведіть: якщо $A \neq \varnothing$ та $B \neq \varnothing$, то $A \otimes B \equiv_T A \oplus B$.

15. Чи існують множини A та B такі:

1) $A \otimes B <_T A$ і $A \otimes B \equiv_m A \oplus B$?

2) $A <_m A \oplus B$ і $A \equiv_T A \oplus B$?

3) $A <_T A \otimes B$ і $A \otimes B \equiv_m A \oplus B$?

4) $A <_m A \otimes B$ і $A \equiv_T A \oplus B$?

5) $A <_T A \oplus B$ і $A \equiv_m A \otimes B$?

6) $B <_T A \oplus B$ і $A \otimes B \equiv_m A \oplus B$?

7) $A \otimes B <_m A \oplus B$ і $A \equiv_T A \otimes B$?

8) $A \otimes B <_T A \oplus B$ і $A \equiv_m A \oplus B$?

9) $B <_T A \otimes B$ і $B \equiv_m A \oplus B$?

10) $A <_m A \otimes B$ і $A \otimes B \equiv_T A \oplus B$?

16. Чи вірні такі твердження:

1) якщо $A <_m B$, то $A <_T B$?

2) якщо $A \mid_m B$, то $A \mid_T B$?

3) якщо $A \mid_T B$, то $A \mid_m B$?

17. Доведіть:

1) $\overline{D} \leq_T \{x \mid D_x$ не є РМ$\}$;

2) $D \leq_T \{x \mid E_x$ є РМ$\}$.

18. Доведіть, що $A \leq_T B \Leftrightarrow A \leq_m D^B$ та $\overline{A} \leq_m D^B$.

19. Доведіть, що для РПМ A та B таких, що $A \cap B = \varnothing$, маємо:
   1) $A \oplus B \leq_T A \cup B$;
   2) $d_T(A \cup B) = \boldsymbol{a} \cup \boldsymbol{b}$, де $\boldsymbol{a} = d_T(A)$ та $\boldsymbol{b} = d_T(B)$.
20. Доведіть, що існують множини A і B такі: $A^{(\omega)} \leq_m B^{(\omega)}$ і $B <_T A$.
21. Доведіть:
   1) теорему 6.4.3;
   2) наслідок теореми 6.4.11.

7. АРИФМЕТИЧНІСТЬ. АРИФМЕТИЧНА ІЄРАРХІЯ

В цьому розділі вивчається найпервісніша математична структура — множина натуральних чисел. При цьому використовуються ідеї та методи як власне математичної логіки, так і теорії алгоритмів.

Нагадаємо (див., наприклад, [11]), що мова арифметики L_{ar} визначається сигнатурою $\sigma_{ar} = \{0, 1, +, \times, =\}$, де 0 і 1 – константні символи, + та $\times$ — бінарні функціональні символи, = — бінарний предикатний символ.

Формулу мови арифметики називають арифметичною формулою.

Стандартною інтерпретацією (стандартною моделлю) мови арифметики називають множину натуральних чисел N з виділеними константами 0 та 1, визначеними на N стандартними бінарними функціями додавання + і множення $\times$ та стандартним бінарним предикатом рівності. Іншими словами, стандартна інтерпретація L_{ar} – це алгебраїчна система (АС) $\boldsymbol{N} = (N, \sigma_{ar})$.

Арифметична формула, яка істинна на $\boldsymbol{N}$, називається істинною арифметичною формулою (ІАФ).

Предикати, множини та функції, виразні в $\boldsymbol{N} = (N, \sigma_{ar})$, називають арифметичними.

7.1. Арифметичність ЧРФ і РПМ. Теорема Тарського

При інтерпретації арифметичних формул на стандартній моделі $\boldsymbol{N} = (N, \sigma_{ar})$ іменами натуральних чисел можуть бути замкнені терми 0, 1, 1 + 1, ..., 1 + ... + 1, ...

Таке ім'я натурального числа n позначатимемо $\overline{n}$.

Зрозуміло, що ці імена можна визначити так: $\overline{0} = 0$, $\overline{1} = 1$, $\overline{n+1} = \overline{n} + 1$ для $n \geq 1$.

Використовуючи введені імена, можна визначити виразність на $\boldsymbol{N}$ предикатів, множин і функцій, використовуючи тільки замкнені арифметичні формули. Зокрема, для n-арних арифметичних функцій

і предикатів і для арифметичних множин вигляду $L \subseteq N^k$ можна дати традиційні [39] визначення. Такі визначення цілком узгоджуються з наведеними в [11] визначеннями виразних в АС предикатів, множин і функцій.

Предикат P: $N^k \to \{T, F\}$ арифметичний, якщо існує арифметична формула $\Phi(x_1, ..., x_k)$ з вільними іменами $x_1, ..., x_k$ така:

$$P(n_1, ..., n_k) = T \Leftrightarrow N \models \Phi_{x_1, ..., x_k}[\overline{n}_1, ..., \overline{n}_k].$$

Така формула Φ *виражає* предикат P.

Множина $L \subseteq N^k$ *арифметична* (скорочено АМ), якщо існує арифметична формула $\Phi(x_1, ..., x_k)$ з вільними іменами $x_1, ..., x_k$ така:

$$(\overline{n}_1, ..., \overline{n}_k) \in L \Leftrightarrow N \models \Phi_{x_1, ..., x_k}[\overline{n}_1, ..., \overline{n}_k].$$

Така формула Φ *виражає* множину L.

Класи арифметичних множин і предикатів позначаємо ***АМ*** і ***АП***.

Функція f: $N^k \to N$ арифметична, якщо її графік Γ_f — арифметична множина.

Арифметична формула Φ *виражає* функцію f, якщо формула Φ виражає Γ_f.

Теорема 7.1.1. *Кожна ЧРФ арифметична.*

Арифметичними є такі функції:

1) функція $x + y$ виражається арифметичною формулою $z = x + y$;

2) функція $x \times y$ виражається арифметичною формулою $z = x \times y$;

3) функція $\boldsymbol{o}(x) = 0$ виражається арифметичною формулою $z = 0 \,\&\, x = x$;

4) функція $\boldsymbol{s}(x) = x + 1$ виражається арифметичною формулою $z = x + 1$;

5) функція $\boldsymbol{I}^n_m(x_1, ..., x_n) = x_m$ виражається арифметичною формулою $(z = x_m) \& (x_1 = x_1) \& ... \& (x_n = x_n)$;

6) функція $x \dot{-} y$ виражається арифметичною формулою $(\exists v(x + v = y) \to z = 0) \,\&\, \exists v(y + v = x) \to y + z = x)$.

Із розділу 3 відомо, що кожну ЧРФ отримують із функцій $\boldsymbol{o}$, $\boldsymbol{s}$, $\boldsymbol{I}^n_m$, $+$, $\times$, $\dot{-}$ за допомогою операцій $\boldsymbol{S}^{n+1}$ і $\boldsymbol{M}$.

Функції $\boldsymbol{o}$, $\boldsymbol{s}$, $\boldsymbol{I}^n_m$, $+$, $\times$, $\dot{-}$ арифметичні, тому покажемо, що операції $\boldsymbol{S}^{n+1}$ і $\boldsymbol{M}$ зберігають арифметичність функцій.

Нехай $f = \boldsymbol{S}^{n+1}(g, g_1, ..., g_n)$ та функції $g(x_1, ..., x_n)$, $g_1(x_1, ..., x_m)$, ..., $g_n(x_1, ..., x_m)$ виражені формулами $G(x_1, ..., x_n, z)$, $G_1(x_1, ..., x_m, z)$, ..., $G_n(x_1, ..., x_m, z)$ відповідно. Тоді функцію $z = f(x_1, ..., x_m)$ виражає формула $\exists z_1 ... \exists z_n (G_{x_1,...,x_n}[z_1, ..., z_n] \& (G_1)_z[z_1] \& ... \& (G_n)_z[z_n])$.

Нехай функція $g(x_1, ..., x_n, y)$ виражена арифметичною формулою $G(x_1, ..., x_n, y, z)$. Але $z = \mu_y(g(x_1, ..., x_n, y) = 0) \Leftrightarrow (g(x_1, ..., x_n, z) = 0)$ & $(\forall u(u<z \rightarrow g(x_1, ..., x_n, u) \neq 0)$, тому функцію $z = \mu_y(g(x_1, ..., x_n, y) = 0)$ виражає формула $G_{y,z}[z,0] \& \forall u(u<z \rightarrow \exists t(G_{y,z}[u, t] \& (t \neq 0)))$ ■

Як наслідок звідси отримуємо теорему.

Теорема 7.1.2. *Кожна РПМ арифметична.*

Нехай $L \subseteq N^k$ є РПМ. Тоді $L = D_f$ для деякої ЧРФ f. Згідно з теоремою 7.1.1 функція f арифметична, нехай f виражається арифметичною формулою $\Phi(x_1, ..., x_n, z)$. Тоді D_f виражається арифметичною формулою $\exists z \Phi$ ■

Теорема 7.1.3. *Клас арифметичних множин замкнений відносно операцій* $\cup$, $\cap$ *і доповнення*.

Нехай множини A і B виражаються арифметичними формулами Φ і Ψ. Тоді $A \cup B$, $A \cap B$ і $\overline{A}$ виражаються відповідно арифметичними формулами $\Phi \vee \Psi$, $\Phi \& \Psi$ і $\neg \Phi$ ■

Наслідок. *Для класів РПМ і АМ маємо строге включення* $РПМ \subset АМ$.

Множина $D = \{x | \varphi_x(x)\downarrow\}$ є РПМ (див. підрозділ 5.3), тому за теоремою 7.1.2 D арифметична, звідки $\overline{D}$ арифметична, але ж $\overline{D}$ не є РПМ ■

Деякі властивості арифметичних формул є розв'язними, наприклад властивості замкненості, атомарності чи безкванторності формули. Але властивість бути ІАФ не є навіть частково розв'язною. Більше того, властивість бути ІАФ навіть не арифметична.

Нехай задана деяка ефективна нумерація множини арифметичних формул. Нехай **T** — множина номерів усіх ІАФ.

Теорема 7.1.4 (теорема Тарського). *Множина* **T** *неарифметична.*

Припустимо супротивне: множина **T** арифметична. Тоді **T** можна виразити деякою арифметичною формулою $U(x)$ з єдиним вільним іменем x: $n \in \mathbf{T} \Leftrightarrow \boldsymbol{N} \models U_x[\overline{n}]$.

Розглянемо функцію $f(m, n) =$

$$= \begin{cases} \kappa((\phi_m)_x[\bar{n}]), & \text{якщо } \phi_m - \text{арифметична формула з єдиним} \\ & \text{вільним іменем } x, \\ & \text{не визначене інакше.} \end{cases}$$

За тезою Чорча $f \in$ ЧРФ, тому f арифметична.

Нехай f виражена арифметичною формулою $B(x, y, z)$ з трьома вільними іменами x, y, z. Тоді $\boldsymbol{N} \models B_{x,y,z}[\bar{m}, \bar{n}, \bar{k}] \Leftrightarrow k = \kappa((\phi_m)_x[\bar{n}])$ і ϕ_m має єдине вільне ім'я x.

Арифметичну формулу $\exists z(B(x, y, z) \& U_x[z])$ позначимо $A(x, y)$. Тоді отримуємо: $\boldsymbol{N} \models A_{x,y}[\bar{m}, \bar{n}] \Leftrightarrow \phi_m$ має єдине вільне ім'я x та $\kappa((\phi_m)_x[\bar{n}]) \in \mathbf{T} \Leftrightarrow \phi_m$ має єдине вільне ім'я x і $\boldsymbol{N} \models (\phi_m)_x[\bar{n}]$.

Арифметичну формулу $\neg A_{x,y}[x, x]$ позначимо $\vartheta(x)$. Нехай k — її номер, тобто ϑ — це формула ϕ_k. Тоді $\boldsymbol{N} \models (\phi_k)_x[\bar{k}] \Leftrightarrow \boldsymbol{N} \models \vartheta_x[\bar{k}]$. Але $\boldsymbol{N} \models \vartheta_x[\bar{k}] \Leftrightarrow \boldsymbol{N} \models \neg A_{x,y}[\bar{k}, \bar{k}] \Leftrightarrow$ невірно, що ϕ_k має єдине вільне ім'я x і $\boldsymbol{N} \models (\phi_k)_x[\bar{k}] \Leftrightarrow$ невірно, що $\boldsymbol{N} \models (\phi_k)_x[\bar{k}]$ (бо ϕ_k — формула $\vartheta(x)$ з єдиним вільним іменем x) $\Leftrightarrow \boldsymbol{N} \models \neg(\phi_k)_x[\bar{k}]$.

Маємо $\boldsymbol{N} \models \vartheta_x[\bar{k}] \Leftrightarrow \boldsymbol{N} \models (\phi_k)_x[\bar{k}] \Leftrightarrow \boldsymbol{N} \models \neg(\phi_k)_x[\bar{k}]$. Отримали суперечність, тому множина $\mathbf{T}$ неарифметична.

Семантично формула $\vartheta_x[\bar{k}]$ стверджує: "мій номер $\notin \mathbf{T}$", тобто "я хибна". Отже, $\vartheta_x[\bar{k}]$ виражає відомий парадокс брехуна ■

Теорема Тарського засвідчує, що не існує "універсальної" ІАФ, яка дозволяла б отримувати довільну ІАФ за її номером.

Фундаментальне значення теореми Тарського полягає в тому, що вона доводить неможливість повної формалізації поняття істини в достатньо багатих мовах, які включають або можуть моделювати мову арифметики.

7.2. Арифметична ієрархія

Розглянемо класифікацію арифметичних множин і предикатів, яка пов'язує теорію рекурсивних функцій з математичною логікою.

Σ_n-*префіксом* називають послідовність кванторних префіксів із $n - 1$ зміною однотипних кванторів, який починається квантором $\exists$.

Π_n-*префіксом* називають послідовність кванторних префіксів із $n - 1$ зміною однотипних кванторів, який починається квантором $\forall$.

Наприклад, $\exists x\exists y\forall u\forall v\forall w\exists t\exists z$ — Σ_3-префікс; $\forall x\exists y\exists z$ — Π_2-префікс; $\exists x\exists y\forall u\forall v$ — Σ_2-префікс; $\exists x\exists y\exists u$ — Σ_1-префікс; $\forall u\forall z$ — Π_1-префікс.

Нехай $\mathfrak{R}$ — множина арифметичних формул, значеннями яких є рекурсивні предикати.

Для всіх $n \geq 0$ введемо класи предикатів Σ_n, Π_n і Δ_n.

Покладемо $\Sigma_0 = \Pi_0 = \Delta_0$ = множина всіх РП.

Для $n \geq 1$ визначаємо так:

- Σ_n складається з усіх предикатів, виражених формулами вигляду $\sigma\Phi$, де $\sigma \in \Sigma_n$ і $\Phi \in \mathfrak{R}$;
- Π_n складається з усіх предикатів, виражених формулами вигляду $\sigma\Phi$, де $\sigma \in \Pi_n$ і $\Phi \in \mathfrak{R}$;
- $\Delta_n = \Sigma_n \cap \Pi_n$.

Відомо [12, 28], що для $n \geq 1$ маємо:

- $P \in \Sigma_n \Leftrightarrow P = (\sigma\Psi)_N$ для деяких $\sigma \in \Sigma_n$ і атомарної формули Ψ;
- $P \in \Pi_n \Leftrightarrow P = (\sigma\Psi)_N$ для деяких $\sigma \in \Pi_n$ і атомарної формули Ψ.

Уведені класи предикатів Σ_n, Π_n і Δ_n індукують відповідні класи множин $\Sigma_n = \{I_P \mid P \in \Sigma_n\}$, $\Pi_n = \{I_P \mid P \in \Pi_n\}$, $\Delta_n = \Sigma_n \cap \Pi_n$.

Зрозуміло, що $\Sigma_0 = \Pi_0 = \Delta_0$ = множина всіх РМ, Σ_1 — це множина всіх РПМ, Π_1 — це множина всіх доповнень до РПМ. За теоремою Поста $\Delta_1 = \Sigma_1 \cap \Pi_1$ — це множина всіх РМ. Отже, $\Delta_1 = \Delta_0$.

Наведемо елементарні властивості введених класів предикатів.

Теорема 7.2.1. $P \in \Sigma_n \Leftrightarrow \neg P \in \Pi_n$.

Нехай $P = (\sigma\Phi)_N$, де $\sigma \in \Sigma_n$ і $\Phi \in \mathfrak{R}$. Використовуючи пренексні операції пронесення $\neg$ через квантори, дістаємо формулу $\sigma'\neg\Phi$, де $\sigma' \in \Pi_n$, таку: $\sigma'\neg\Phi \sim \neg\sigma\Phi$. Але $\neg\Phi \in \mathfrak{R}$, тому $\neg P = (\sigma'\neg\Phi)_N \in \Pi_n$.

Теорема 7.2.2. $\Sigma_n \cup \Pi_n \subseteq \Delta_{n+1}$.

Нехай $P \in \Sigma_n$. Тоді $P = (\sigma\Phi)_N$ для деяких $\sigma \in \Sigma_n$ і $\Phi \in \mathfrak{R}$. Візьмемо предметне ім'я $u \notin \sigma\Phi$. Ім'я u неістотне для $\sigma\Phi$, тому $\forall u\sigma\Phi \sim \sigma\Phi$. Звідси $P = (\forall u\sigma\Phi)_N$. Але $\forall u\sigma \in \Pi_{n+1}$, тому $P \in \Pi_{n+1}$.

Ім'я u неістотне для Φ, тому $\forall u\Phi \sim \Phi$ і $\exists u\Phi \sim \Phi$. Звідси $\sigma\forall u\Phi \sim \sigma\Phi$ і $\sigma\exists u\Phi \sim \sigma\Phi$. Отже, $P = (\sigma\forall u\Phi)_N = (\sigma\exists u\Phi)_N$. Але при n непарному $\sigma\forall u \in \Sigma_{n+1}$, при n парному $\sigma\exists u \in \Sigma_{n+1}$. Тому $P \in \Sigma_{n+1}$. Отже, $\Sigma_n \subseteq \Pi_{n+1}$ і $\Sigma_n \subseteq \Sigma_{n+1}$, звідки $\Sigma_n \subseteq \Sigma_{n+1} \cap \Pi_{n+1} = \Delta_{n+1}$.

Нехай $P \in \Pi_{\boldsymbol{n}}$. Тоді $P = (\sigma\Phi)_N$ для деяких $\sigma \in \Pi_n$ і $\Phi \in \Re$. Візьмемо предметне ім'я $u \notin \sigma\Phi$. Ім'я u неістотне для $\sigma\Phi$, тому $\exists u \sigma\Phi \sim \sigma\Phi$. Звідси $P = (\exists u \sigma\Phi)_N$. Але $\exists u \sigma \in \Sigma_{n+1}$, тому $P \in \Sigma_{\boldsymbol{n+1}}$.

Ім'я u неістотне для Φ, тому $\forall u\Phi \sim \Phi$ і $\exists u\Phi \sim \Phi$. Звідси $\sigma\forall u\Phi \sim \sigma\Phi$ і $\sigma\exists u\Phi \sim \sigma\Phi$. Отже, $P = (\sigma\forall u\Phi)_N = (\sigma\exists u\Phi)_N$. Але при n парному $\sigma\forall u \in \Pi_{n+1}$, при n непарному $\sigma\exists u \in \Pi_{n+1}$. Тому $P \in \Pi_{\boldsymbol{n+1}}$.

Отже, $\Pi_{\boldsymbol{n}} \subseteq \Sigma_{\boldsymbol{n+1}}$ і $\Pi_{\boldsymbol{n}} \subseteq \Pi_{\boldsymbol{n+1}}$, звідки $\Pi_{\boldsymbol{n}} \subseteq \Sigma_{\boldsymbol{n+1}} \cap \Pi_{\boldsymbol{n+1}} = \Delta_{\boldsymbol{n+1}}$.

Теорема 7.2.3. $\bigcup_{n\geq 0} \Sigma_n = \bigcup_{n\geq 0} \Pi_n = \boldsymbol{A\Pi}$.

Нехай $P \in \bigcup_{n\geq 0} \Sigma_n$. Тоді $P \in \Sigma_{\boldsymbol{n}}$ для деякого $n \geq 0$, звідки $P \in \boldsymbol{A\Pi}$.

За теоремою 7.2.2 $P \in \Pi_{\boldsymbol{n+1}}$, тому $P \in \bigcup_{n\geq 0} \Pi_n$.

Отже, $\bigcup_{n\geq 0} \Sigma_n \subseteq \boldsymbol{A\Pi}$ та $\bigcup_{n\geq 0} \Sigma_n \subseteq \bigcup_{n\geq 0} \Pi_n$.

Аналогічно $\bigcup_{n\geq 0} \Pi_n \subseteq \boldsymbol{A\Pi}$ і $\bigcup_{n\geq 0} \Pi_n \subseteq \bigcup_{n\geq 0} \Sigma_n$.

Нехай $P \in \boldsymbol{A\Pi}$. Тоді $P = (\mathrm{A})_N$ для деякої арифметичної формули A. Звівши A до пренексної форми, дістанемо пренексну формулу Φ таку, що A~Φ. Така Φ має вигляд $\sigma\Psi$ для деяких $\sigma \in \Sigma_n \cup \Pi_n$ та атомарної формули Ψ.

Якщо $\sigma \in \Sigma_n$, то $P \in \bigcup_{n\geq 0} \Sigma_n$. Якщо $\sigma \in \Pi_n$, то $P \in \bigcup_{n\geq 0} \Pi_n$.

Отже, $\boldsymbol{A\Pi} \subseteq \bigcup_{n\geq 0} \Sigma_n = \bigcup_{n\geq 0} \Pi_n$.

Теорема 7.2.4 (теорема Кліні про ієрархію). *Для кожного $n>0$ існує арифметичний предикат ϑ такий, що $\vartheta \in \Sigma_{\boldsymbol{n}} \backslash \Pi_{\boldsymbol{n}}$ та $\neg\vartheta \in \Pi_{\boldsymbol{n}} \backslash \Sigma_{\boldsymbol{n}}$.*

Твердження теорем 7.2.1 – 7.2.4 повністю переносяться на відповідні класи арифметичних множин.

Позначимо ${}^{\Sigma}\mathbf{T}_{\boldsymbol{n}}$ і ${}^{\Pi}\mathbf{T}_{\boldsymbol{n}}$ множини номерів тих ІАФ, що мають пренексну форму з Σ_n-префіксом та Π_n-префіксом відповідно.

Теорема 7.2.5. ${}^{\Sigma}\mathbf{T}_{\boldsymbol{n}} \equiv_1 \varnothing^{(n)}$.

Для T-степенів звідси дістаємо

Наслідок. $^{\Sigma}\mathbf{T}_n \in \mathbf{0}^{(n)}$ та $^{\Pi}\mathbf{T}_n \in \mathbf{0}^{(n)}$.

Для множини номерів усіх ІАФ маємо:

Теорема 7.2.6. $\mathbf{T} \in \mathbf{0}^{(\omega)}$.

Встановити належність множини до класів Σ_n чи Π_n, тобто визначити її місце в арифметичній ієрархії, можна за допомогою *алгоритму Тарського-Куратовського* [12].

Суть алгоритму: використовуючи пренексні операції, подаємо предикат "$x \in M$" у вигляді $(\sigma\Phi)_N$, після чого встановлюємо $\sigma \in \Sigma_n$ чи $\sigma \in \Pi_n$ для деякого $n>0$.

Приклад 7.2.1. $M = \{x \mid D_x$ нескінченна$\} \in \Pi_2$.

D_x нескінченна $\Leftrightarrow \forall z \exists y (y>z \ \& \ y \in D_x) \Leftrightarrow \forall z \exists y (y>z \ \& \ \exists k (P_x(y)\downarrow$ за k кроків$) \Leftrightarrow \forall z \exists y \exists k (y>z \ \& \ P_x(y)\downarrow$ за k кроків$)$.

Предикати $y > z$ і $(P_x(y)\downarrow$ за k кроків$)$ є РП.

Приклад 7.2.2. $M = \{x \mid \varphi_x$ не є РФ$\} \in \Sigma_2$.

$x \in M \Leftrightarrow \varphi_x$ не є РФ $\Leftrightarrow \exists y (\varphi_x(y)\uparrow) \Leftrightarrow \exists y \neg \exists k (P_x(y)\downarrow$ за k кроків$) \Leftrightarrow \exists y \forall k \neg (P_x(y)\downarrow$ за k кроків$)$. Предикат $\neg (P_x(y)\downarrow$ за k кроків$)$ є РП.

Приклад 7.2.3. $M = \{x \mid D_x$ є РМ$\} \in \Sigma_3$.

Предикат $(P_u(v)\downarrow$ за w кроків$)$ позначимо $P(u, v, w)$.

Використаємо співвідношення $A \leftrightarrow \neg B \sim (\neg A \vee \neg B) \& (A \vee B)$.

Тепер маємо: D_x є РМ $\Leftrightarrow \exists z (D_x = \overline{D_z}) \Leftrightarrow$

$\exists z \forall y (y \in D_x \leftrightarrow \neg (y \in D_z)) \Leftrightarrow \exists z \forall y (\exists k \, P(x, y, k) \leftrightarrow \neg \exists n P(z, y, n)) \Leftrightarrow$

$\exists z \forall y ((\neg \exists k \, P(x, y, k) \vee \neg \exists n P(z, y, n)) \ \& \ (\exists k \, P(x, y, k) \vee \exists n P(z, y, n)) \Leftrightarrow$

$\exists z \forall y (\forall k \forall n (\neg P(x, y, k) \vee \neg P(z, y, n)) \ \& \ \exists k \exists n (P(x, y, k) \vee P(z, y, n)) \Leftrightarrow$

$\exists z \forall y \forall k \forall n \exists l \exists m ((\neg P(x, y, k) \vee \neg P(z, y, n)) \ \& \ (P(x, y, l) \vee P(z, y, m)))$.

Предикат у дужках після кванторних префіксів є РП.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Дайте визначення арифметичного предиката, арифметичної множини, арифметичної функції.
2. Покажіть арифметичність функцій $\boldsymbol{o}(x)$, $\boldsymbol{s}(x)$, $\boldsymbol{I}_m^n(x_1, ..., x_n)$, $x + y$, $x \times y$, $x \div y$.

3. Покажіть, що операції $\boldsymbol{S}^{n+1}$ і $\boldsymbol{M}$ зберігають арифметичність функцій.
4. Відносно яких теоретико-множинних операцій замкнений клас арифметичних множин?
5. Яке співвідношення між класами РПМ і АМ?
6. Сформулюйте теорему Тарського. Що засвідчує теорема Тарського?
7. У чому полягає значення теореми Тарського?
8. Що таке Σ_n-префікс? Π_n-префікс?
9. Дайте визначення класів предикатів Σ_n, Π_n і Δ_n.
10. Дайте визначення класів множин Σ_n, Π_n і Δ_n.
11. Наведіть елементарні властивості класів Σ_n, Π_n і Δ_n.
12. Зобразіть арифметичну ієрархію класів арифметичних предикатів і арифметичних множин.
13. Сформулюйте теорему Кліні про ієрархію.
14. Який зв'язок існує між класами арифметичних множин і T-степенями?
15. Опишіть алгоритм Тарського-Куратовського.
16. Наведіть приклади використання алгоритма Тарського-Куратовського.

ВПРАВИ

1. Доведіть арифметичність таких множин і предикатів:
   1) $\{x \mid D_x$ скінченна$\}$;
   2) $\{x \mid E_x$ нескінченна$\}$;
   3) $\{x \mid E_x \in$ РМ$\}$;
   4) $\{x \mid \varphi_x \in$ РФ$\}$;
   5) "$D_x \neq N$";
   6) "$E_x = N$";
   7) "$D_x = D$";
   8) $\{x \mid \varphi_x$ неін'єктивна$\}$;
   9) $\{x \mid \varphi_x$ несюр'єктивна$\}$;
   10) $\{C(x, y) \mid x \in D_y\}$.

2. Визначте місце в арифметичній ієрархії таких множин і предикатів:
   1) $\{x \mid \varphi_x \in$ РФ$\}$;

2) “φ_x ін’єктивна”;
3) $\{x \mid E_x = \varnothing\}$;
4) $\{x \mid D_x \neq \varnothing\}$;
5) $\{x \mid D_x$ скінченна$\}$;
6) $\{x \mid E_x$ нескінченна$\}$;
7) $\{x \mid \varphi_x$ є ПРФ$\}$;
8) $\{x \mid E_x$ є ПРМ$\}$;
9) $\{x \mid \varphi_x$ сюр’єктивна$\}$;
10) $\{x \mid \varphi_x$ бієктивна$\}$;
11) $\{x \mid E_x = D\}$;
12) “$D_x \neq D_y$”;
13) “$E_x = D_y$”;
14) “$D_x \neq E_y$”;
15) “$E_x = E_y$”.

СПИСОК ЛІТЕРАТУРИ

Основна

1. *Андерсон Д. А.* Дискретная математика и комбинаторика. — М.: Вильямс, 2003. — 960 с.
2. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979. — 536 с.
3. *Булос Дж., Джеффри Р.* Вычислимость и логика. — М.: Мир, 1994. — 396 с.
4. *Капітонова Ю. В., Кривий С. Л., Летичевський О. А. та ін.* Основи дискретної математики. — К.: Наук. думка, 2002. — 579 с.
5. *Катленд Н.* Вычислимость. Введение в теорию рекурсивных функций. — М.: Мир, 1983. — 256 с.
6. *Лавров И. А., Максимова Л. Л.* Задачи по теории множеств, математической логике и теории алгоритмов. — М.: Наука, 1975. — 240 с.
7. *Лісовик Л. П., Редько В. Н.* Алгоритмы и формальные системы. — К.: КГУ, 1981. — 112 с.
8. *Лісовик Л. П., Шкільняк С. С.* Теорія алгоритмів. — К.: ВПЦ Київ. ун-т. — 163 с.
9. *Мальцев А. И.* Алгоритмы и рекурсивные функции. — М.: Наука, 1965. — 392 с.
10. *Мендельсон Э.* Введение в математическую логику. — М.: Наука, 1976. — 320 с.
11. *Нікітченко М. С., Шкільняк С. С.* Основи математичної логіки. — К.: ВПЦ Київ. ун-т. — 2006. — 246 с.
12. *Роджерс Х.* Теория рекурсивных функций и эффективная вычислимость. — М.: Мир, 1972. — 624 с.
13. *Успенский В. А., Семенов А. Л.* Теория алгоритмов: основные открытия и приложения. — М.: Наука, 1987. — 288 с.
14. *Шкільняк С. С.* Математична логіка. Приклади і задачі. — К.: ВПЦ Київ. ун-т. — 2007. — 144 с.
15. *Шкільняк С. С.* Теорія алгоритмів: приклади і задачі. — К.: ВПЦ Київ. ун-т. — 2003. — 93 с.

Додаткова

16. *Басараб И. А., Никитченко Н. С., Редько В. Н.* Композиционные базы данных. — К.: Либідь, 1992. — 192 с.
17. *Гильберт Д., Бернайс П.* Основания математики. Т. 1, Т. 2. — М.: Наука, 1982.
18. *Гладкий А. В.* Формальные грамматики и языки. — М.: Наука, 1973. — 368 с.
19. *Глушков В. М.* Введение в кибернетику. — К.: АН УССР, 1964. — 324 с.

20. *Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л.* Алгебра, языки, программирование. — К.: Наук. думка, 1974. — 328 с.
21. *Гросс М., Лантен А.* Теория формальних грамматик. — М.: Мир, 1971. — 294 с.
22. *Ершов Ю. Л.* Теория нумераций. — М., 1977.
23. *Кривий С. Л.* Курс дискретної математики: Навч. посібник. — К.: НАУ, 2007. — 432 с.
24. *Клини С.* Введение в метаматематику. — М.: Иностр. лит-ра, 1957. — 526 с.
25. *Клини С.* Математическая логика. — М.: Мир, 1973. — 480 с.
26. *Колмогоров А. Н., Драгалин А. Г.* Введение в математическую логику. — М.: МГУ, 1982. — 120 с.
27. *Колмогоров А. Н., Драгалин А. Г.* Математическая логика. Дополнительные главы. — М.: МГУ, 1984. — 120 с.
28. *Манин Ю. И.* Вычислимое и невычислимое. — М.: Сов. радио, 1980. — 128 с.
29. *Манин Ю. И.* Доказуемое и недоказуемое. — М.: Сов. радио, 1979. — 168 с.
30. *Марков А. А., Нагорный Н. М.* Теория алгорифмов. — М.: Наука, 1984. — 432 с.
31. *Непейвода Н. Н.* Прикладная логика. — Новосибирск: НГУ, 2000. — 521 с.
32. *Нікітченко М. С., Шкільняк С. С.* Математична логіка. — К.: ВПЦ Київський університет. — 2003. — 120 с.
33. *Нікітченко М. С., Шкільняк С. С.* Математична логіка. Додаткові розділи. — К.: ВПЦ Київський університет. — 2004. — 77 с.
34. *Расева Е., Сикорский Р.* Математика метаматематики. — М.:, Наука, 1972. — 592 с.
35. *Редько В. Н.* Универсальные программные логики и их применение. — Системное и теоретическое программирование. Тез. докл. 4 Всес. симпозиума. — Кишинев, 1983.
36. *Справочная* книга по математической логике / Под ред. Дж. Барвайса. — Ч. 1–Ч. 4. — М.: Наука, 1982–1983.
37. *Хартманис Дж., Хопкрофт Дж.* Обзор теории сложности вычислений. — Кибернетический сборник. Новая серия. Вып.11. — М., Мир, 1974. — С. 131–176.
38. *Черч А.* Введение в математическую логику. — М.: ИЛ, 1960. — 486 с.
39. *Шенфилд Дж.* Математическая логика. — М.: Наука, 1975. — 528 с.
40. *Шкильняк С. С.* Исследование программных алгебр функций натуральных аргументов и значений. — Модели и системы обработки информации. Вып. 8. — К., 1989. — С. 9–16.
41. *Шкільняк С. С., Ткачук І. Ю.* Основи теорії алгоритмів. — К.: ВПЦ Київ. ун-т. — 2006. — 70 с.

СПИСОК СКОРОЧЕНЬ

АОФ — алгоритмічно обчислювана функція
ІАФ — істинна арифметична формула
МНР — машина з натуральнозначними регістрами
МНРО — МНР з оракулом
МТ — машина Тьюрінга
НА — нормальний алгоритм Маркова
ПРМ — примітивно рекурсивна множина
ПРП — примітивно рекурсивний предикат
ПРФ — примітивно рекурсивна функція
РМ — рекурсивна множина
РП — рекурсивний предикат
РПМ — рекурсивно перелічна множина
РФ — рекурсивна функція
СП — система Поста
ТЧ — теза Чорча
ЧРП — частково рекурсивний предикат
ЧРФ — частково рекурсивна функція
ФС — формальна система

ПОКАЖЧИК ТЕРМІНІВ

ЗМІСТ

МАТЕМАТИЧНА ЛОГІКА

ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ

This book concerns the definitions and results of the basics of mathematical logic and of the theory of algorithms. It covers propositional logic, first-order classical logic as well as intuitionistis and modal logics. Formal models of algorithms and computable functions, decidability and undecidability, relative computability, and reduction are concerned. The text is illustrated with examples; there are guestions. There are questions, tasks and exercises at the end of each chapter.

The book is meant for students of specialization “Informatics” and “Applied mathematics”.

Навчальне видання

Шкільняк Степан Степанович

МАТЕМАТИЧНА ЛОГІКА

ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ

Навчальний посібник

Educational edition

Shkilniak, Stepan S.

MATHEMATICAL LOGIC

THE BASICS OF ALGORITHM THEORY

Educational manual

Відповідальний редактор *В. Д. Бондар*
Редактори *Т. Д. Станішевська, Л. В. Логвиненко*
Коректор *Т. М. Федосенко*
Комп’ютерне верстання *М. І. Фадєєва*
Оформлення обкладинки *О. О. Стеценко*

Підп. до друку 20.10.08. Формат $60\times84/_{16}$. Папір офсетний.
Друк офсетний. Ум. друк. арк. 15,4. Обл.-вид. арк. 16,27. Наклад 1200 пр.

Міжрегіональна Академія управління персоналом (МАУП)
03039 Київ-39, вул. Фрометівська, 2, МАУП

ДП «Видавничий дім «Персонал»
03039 Київ-39, просп. Червонозоряний, 119, літ. ХХ

Свідоцтво про внесення до Державного реєстру суб’єктів видавничої справи ДК № 3262 від 26.08.2008