Ю. М. Бардачов
Н. А. Соколова
В. Є. Ходаков

Дискретна Математика

«Вища школа»

Ю. М. Бардачов
Н. А. Соколова
В. Є. Ходаков

Дискретна математика

За редакцією доктора технічних наук,
професора *В. Є. Ходакова*

*Затверджено Міністерством освіти
і науки України*

Підручник для студентів вищих
технічних навчальних закладів

Київ
«Вища школа»
2002

УДК 519.854(075.8)
ББК 22.174 я73
Б24

Гриф надано Міністерством освіти і науки України (лист від 28 січня 2002 р. № 1/11-266)

Рецензенти:
д-р техн. наук, проф., завідувач кафедри системотехніки Харківського технічного університету радіоелектроніки *Е. Г. Петров;* д-р фіз.-мат. наук, проф., завідувач кафедри математичного моделювання Херсонського державного технічного університету *А. Н. Хомченко*

Редакція літератури з економіки і фундаментальних наук
Редактор *В. Ф. Хміль*

Бардачов Ю. М. та ін.
Б24 Дискретна математика: Підручник / Ю. М. Бардачов, Н. А. Соколова, В. Є. Ходаков; За ред. В. Є. Ходакова. — К.: Вища шк., 2002. — 287 с.: іл.
ISBN 966–642–090–2

Викладено основні поняття і наукові результати теорій множин, математичної логіки, відношень, формальних систем, алгоритмів, алгебр, комбінаторики, графів. Матеріал ілюстровано численними прикладами, кожний розділ містить контрольні запитання і перелік лабораторно-практичних занять.

Для студентів вищих технічних навчальних закладів.

УДК 519.854(075.8)
ББК 22.174 я73

ISBN 966-642-090-2

ЗМІСТ

Розділ 6. Формальні системи

Розділ 7. Алгебри

Розділ 8. Комбінаторика

Розділ 9. Графи

Присвячується 45-річчю
Херсонського державного
технічного університету

ПЕРЕДМОВА

Дискретна математика є розділом математики, що зародилася в давні часи. Її головною відмінністю є дискретність, тобто антипод неперервності. Дискретна математика включає традиційні розділи математики, які вже сформувалися (математичну логіку, алгебру, теорію чисел), і нові, що інтенсивно розвиваються.

У більш як двотисячорічній історії дискретної математики сучасний період є одним із найінтенсивніших періодів її розвитку: дуже швидко розширюється сфера застосування, інтенсивно зростають обсяги нової інформації та кількість нових результатів. Якщо ще порівняно недавно ця наука була сферою інтересів лише вузького кола фахівців, то тепер вона перетворюється на наукову дисципліну, дуже важливу і потрібну для багатьох, а у сфері сучасної освіти — для всіх.

Масове використання обчислювальної техніки (персональних комп'ютерів) значно розширює сферу прикладних досліджень, у яких все більше застосовується апарат дискретної математики.

У два останні десятиріччя в Україні література з дискретної математики практично не видавалася. Водночас інженери-математики та програмісти, які займаються прикладними дослідженнями, виявляють все більшу зацікавленість у використанні апарата дискретної математики, що пояснюється широким застосуванням комп'ютерної техніки й інформаційних технологій. Книга, що пропонується, — підручник, який включає не тільки основні поняття і теоретичні результати, а й методи та алгоритми розв'язання деяких прикладних задач. Він дає змогу скласти цілісне уявлення про весь комплекс можливостей цієї науки. Одна з основних цілей його створення якраз і полягає в тому, щоб навчити студентів та фахівців основам дискретної математики, сприяти глибшому розумінню і засвоєнню прикладних проблем, у світлі яких трактується основний зміст розділів дискретної математики.

У *першому* розділі розглядаються загальні відомості, що стосуються математики: основні поняття про інформацію, логічні й однорідні функції.

У *другому* розділі викладаються основи теорії множин: способи їх задання; поняття порожньої множини, операції над множинами, універсуму множин, множини підмножин, алгебри множин; методи доведення алгебри логіки, узагальнення операцій над множинами. Розгядаються нечіткі множини, основні поняття і визначення, нечіткі лінгвістичні змінні.

У *третьому* розділі викладаються основи математичної логіки. Досить детально та ґрунтовно розглядаються булеві функції. Виклад починається з визначення основних понять. Показуються способи задання булевих функцій; подаються булеві функції однієї і двох змінних, область визначення булевої функції, елементарні функції алгебри логіки та їхні властивості; розглядаються реалізація булевих функцій формулами, поняття про повноту, набори повних систем, канонічні форми перемикальних функцій, питання мінімізації булевих функцій, способи здобуття мінімальних форм, геометричне подання булевих функцій.

Четвертий розділ присвячений відношенням. Розкриваються основні поняття і властивості відношень еквівалентності, порядку, квазіпорядку, рівнопотужності.

П'ятий розділ висвітлює таке важливе питання, як теорія алгоритмів, яку, як і формальні системи, донедавна вважали «високою наукою». Розглядаються властивості алгоритмів, машини Тьюрінга (МТ) та операції над ними, рекурсивні функції, а також їх зв'язок із МТ.

У *шостому* розділі розглядаються основні поняття, аксіоматичний спосіб опису висловлення, властивості числення висловлень. На прикладі логіки висловлень читач знайомиться із суворою формалізацією математичної теорії — побудовою числення висловлень. Висвітлюються питання, пов'язані з ефективною обчислюваністю. Розглядаються система Поста і МТ. За допомогою останньої та частково-рекурсивних функцій уточнюються поняття алгоритму й обчислюваності.

У *сьомому* розділі наводяться основні поняття алгебри, типи алгебр та алгебричні системи.

Восьмий розділ присвячений комбінаториці, а також перелічувальній теорії Поя, в якій використовуються результати і методи теорії груп.

У *дев'ятому* розділі викладаються основи теорії графів (орієнтованих та неорієнтованих). Частина матеріалу досить добре відома фахівцям, частина ж є оригінальною авторською розробкою і публікується вперше.

В основу підручника покладено курс лекцій, що читаються протягом кількох років на факультеті кібернетики Херсонського державного технічного університету. Всі розділи написані спільно, за винятком дев'ятого, поданого Н. А. Соколовою.

РОЗДІЛ

1

ЗАГАЛЬНІ ВІДОМОСТІ

1.1. Основні поняття про інформацію

Неперервна і дискретна інформація. Інформація про різні природні явища та технологічні процеси сприймається людиною у вигляді тих або інших полів, які математично подаються за допомогою функцій вигляду

$$y = f(x, t),$$

де y – значення поля в цій точці; x — точка, в якій воно вимірюється; t — час. При вимірюванні поля у фіксованій точці $x = a$ функція $f(x, t)$ вироджується у функцію часу $y(t) = f(a, t)$.

Здебільшого всі скалярні* величини, що входять у співвідношення $y = f(x, t)$ (тобто t, y і координати точки x), можуть набувати неперервного ряду значень дійсних чисел. Під неперервністю розуміють те, що величини, які розглядаються, можуть змінюватися будь-якими дрібними кроками. Подану таким способом інформацію прийнято називати неперервною, або аналоговою (рис. 1, *а*).

Якщо стосовно тієї самої інформації про поле $y = f(x, t)$ зафіксувати деякі значення кроків усіх скалярних величин, які її характеризують, то дістанемо дискретне подання інформації (дискретну інформацію) (рис. 1, *б*). Оскільки точність вимірювань завжди обмежено, навіть маючи справу з неперервною інформацією, людина сприймає її у дискретному вигляді. Проте будь-яка неперервна інформація може бути апроксимована дискретною інформацією з будь-якою точністю, тому можна говорити про універсальність дискретної форми подання інформації.

Крім того, результати вимірювань будь-яких скалярних величин подаються, зрештою, в числовому вигляді, а оскільки при заданій точності вимірювань ці числа є кінцевими наборами цифр (із комою або без неї), дискретну форму подання інформації часто ототожнюють із цифровою.

Абстрактні алфавіти. Цифрова інформація насправді є окремим випадком алфавітного подання дискретної інформації. Його основою слугує довільний фіксований скінченний набір символів будь-якої природи — абстрактний алфавіт, або просто алфавіт.

---

* *Скаляр* – величина, кожне значення якої може бути виражено одним (дійсним) числом.

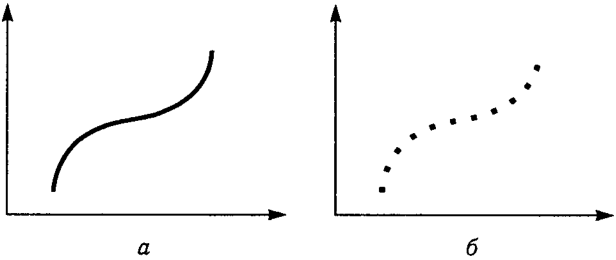


Рис. 1

Сукупність десяткових цифр разом із комою (для відокремлення дробової частини числа) можна розглядати як окремий випадок абстрактного алфавіту з 11 символами — його літерами.

Інший приклад — алфавіт природної людської мови, наприклад української. Мова математичних тестів (математичний алфавіт) може включати нарівні з літерами однієї мови літери інших мов, а також різні спеціальні символи (+, –, = та ін.).

Кодування. Під час оброблення інформації часто виникає необхідність подавати засобами одного алфавіту літери іншого. Таке подання в інформатиці називається кодуванням. Проблема вирішується просто, якщо потрібно закодувати літери алфавіту *X* із меншою кількістю літер, ніж у кодувального алфавіту *Y*. Якщо, наприклад, *X* — алфавіт десяткових цифр, а *Y* — звичайний український алфавіт, то для кодування *X* в *Y* досить позначити 0 = *а*, 1 = = *б*, 2 = *у*, ..., 9 = *к*. Зрозуміло, можливі й інші способи кодування, в тому числі такі, в яких літери алфавіту *X* кодуються кількома літерами алфавіту *Y*.

При кодуванні алфавіту з великою кількістю літер в алфавітах із меншою використовують послідовності літер (символів). Наприклад, *а* = 01, *б* = 02, ..., *и* = 10, *і* = 11, ...

Завдяки своїй простоті двійковий алфавіт поширений у технічних системах, ЕОМ. Для кодування алфавітів, якими користується людина, традиційно застосовують послідовності двійкових чисел.

Послідовностями з n двійкових чисел можна закодувати 2^n різних символів. При $n = 8$ їх число дорівнює 256, що цілком досить для кодування більшості алфавітів природних мов, які зустрічаються на практиці.

Послідовність із восьми двійкових цифр дістала назву «байт». Алфавіт, який складається з різних подібних послідовностей з 256 літер, називається байтовим. У практиці використання ЕОМ у міжнародному масштабі вкорінився єдиний стандарт кодування малих і великих літер латинського алфавіту, математичних символів, розділових знаків. «Потужність» (кількість літер) байтового алфавіту виявляється достатньою також для кодування літер українського алфавіту. Реальні алфавіти, що застосовуються для ЕОМ, обмежуються меншою кількістю символів.

Слова й абстрактні мови. Будь-яка скінченна послідовність літер в алфавіті *X* називається словом. При цьому розуміння слова не має ніякого значення.

Для того щоб з усієї множини слів вибрати правильні в якомусь значенні слова, над початковим алфавітом X будується формальна мова. Крім алфавіту X, формальна мова задається своєю особливою граматикою. Вона є нічим іншим, як скінченною сукупністю формальних правил, за допомогою яких породжуються всі слова цієї мови (тобто правильними є тільки такі слова).

Розроблено кілька способів задання граматики, що породжують формальні мови. Одним із найпростіших таких способів є використання нормальних форм Бекуса. Пояснимо це на прикладі.

Нехай над алфавітом, що складається з двійкових цифр $\langle 0,1\rangle$ та двійкової граматики <,>, треба побудувати мову, застосувавши всі раціональні двійкові (невід'ємні) числа (цілі та дробові). У методі Бекуса для цього використовують «заходи», тобто поняття < ціле число > і < двійкове число >.

Якщо застосувати спеціальне скорочення «:=» для виразу «ліва частина є будь-яким із значень форм, поданих у правій частині», то матимемо такий вираз:

<ціле число>::=⟨0⟩|⟨1⟩|⟨0⟩<ціле число>|⟨1⟩<ціле число>,

<двійкове число>::=<ціле число>, <ціле число>,

де | — знак «або».

За цим означенням правильно записаними двійковими числами будуть, наприклад, числа 01, 0010, 00 й узагалі числа з будь-якою кількістю незначущих нулів (як ліворуч, так і праворуч).

На перший погляд, наведена формалізація може здатися зайвою та надуманою. Справді, для означення добре відомих і досить простих понять без неї можна обійтися. Коли ж будуються більш незвичайні та складні поняття, подібні формальні означення стають не тільки необхідними, а й зручними.

Дані. Типи елементарних даних. Інформація, з якою мають справу ЕОМ, називається даними. Розрізнюють вхідні, проміжні та вихідні дані. Вони розбиваються на окремі складові, що мають назву елементарних даних різних типів. Тип даних залежить від значень, яких вони можуть набувати.

Найуживанішими є цілі та дійсні числа, слова і булеві величини.

Тип «булевий» присвоюється елементарним даним, що можуть набувати лише двох значень: істини та хибності. Для подання булевих величин використовується двійковий алфавіт, в якому істина позначається 1, а хибність — 0.

Моделі алгебри даних. Моделлю в математиці називається множина об'єктів, на яких визначено ті або інші предикати. Під предикатом тут розуміють функцію $y = f(x_1, ..., x_n)$, аргументи $(x_1, x_2, ..., x_n)$ якої належать множині M, а значення (y) може бути або істиною, або хибністю. При конкретній підстановці значень параметрів (аргументів) $x_1, x_2, ..., x_n$ предикат $y = f(x_1, x_2, ..., x_n)$ називається висловленням. Іншими словами, предикат є функцією, що залежить від параметрів $x_1, x_2, ..., x_n$ висловлення. Висловлення описує деяку властивість, яку може мати або не мати набір елементів $(x_1, x_2, ..., x_n)$ множини M.

Кількість елементів цього набору може бути будь-якою. При $n = 2$ виникає особливий тип предиката, який називається бінарним відношенням, або просто відношенням. Найуживанішими типами є відношення рівності (=) та нерівності ($\neq$), які природно вводяться для елементарних даних будь-якого типу. Таким чином, відповідний тип даних перетворюється на модель.

Стосовно чисел (цілих або дійсних) природно визначаються відношення $>, <, \geq, \leq$. Таким чином для відповідних типів даних вводяться багатші моделі.

Будь-яка множина M, як відомо, перетворюється на алгебру, якщо на ній задано деяку множину операцій. Під операцією розуміють функцію $y = f(x_1, x_2, ..., x_n)$, аргументи та значення якої є елементами множини M. При $n = 1$ операція називається унарною, а при $n = 2$ — бінарною. Найпоширенішими є бінарні операції.

Для цілих чисел природно визначаються бінарні операції додавання, віднімання, множення, а також унарна операція зміни знака числа. Стосовно дійсних чисел додаються бінарна операція ділення і, якщо необхідно, унарна операція взяття оберненої величини. Зрозуміло, за необхідності можуть бути введені й інші операції.

1.2. Логічні й однорідні функції

Логічні функції. *Функція* — математичне поняття, що виражає залежність одних змінних величин від інших. Якщо величини x та y пов'язані так, що кожному значенню x відповідає певне значення y, то y називають функцією аргументу x. Загалом записують $y = f(x)$ або $y = F(x)$.

Відмітною особливістю логічних функцій є те, що вони набувають значень у скінченних множинах, тобто область значень логічної функції завжди є скінченною сукупністю чисел, символів, понять, властивостей чи різних об'єктів. Якщо область значень функції містить k різних елементів, то вона називається k-значною функцією.

Для того щоб розрізняти область значень функції, їх потрібно якось позначати. Найзручніше елементи перенумерувати числами від 0 до $k - 1$ або позначити символами (наприклад, літерами). Перелік усіх символів, що відповідають області значень, називається алфавітом, а самі символи — його літерами.

Логічні функції можуть залежати від однієї, двох і, взагалі, будь-якого числа змінних (аргументів) $x_1, x_2, ..., x_n$. На відміну від самої функції, аргументи можуть набувати значень елементів як скінченних, так і нескінченних множин. У теоретико-множинному значенні логічна функція n змінних $y = f(x_1, x_2, ..., x_n)$ є відображенням множини наборів (n-вимірних векторів, кортежів, послідовностей) вигляду $(x_1, x_2, ..., x_n)$ області її визначення на множину значень.

Логічну функцію $f(x_1, x_2, ..., x_n)$ можна розглядати як операцію, задану законом композиції $X_1 \times X_2 \times ... \times X_n \rightarrow X$, де $x_1 \in X_1, x_2 \in X_2$,

..., $x_n \in X_n$, «$\times$» — знак декартового добутку, а $\in$ — знак, який вказує на належність.

Однорідні функції. Якщо аргументи функції $(x_1, x_2, ..., x_n)$ набувають значень із тієї самої множини, що й сама функція, то її називають однорідною. У цьому випадку $X_1 = X_2 = ... = X_n = X$, і тому однорідна функція, що розглядається як закон композиції $X^n \to X$, визначає деяку n-місцеву операцію на скінченній множині X.

Областю визначення однорідної функції $y = f(x_1, x_2, ..., x_n)$ є множина наборів $\{x_1, x_2, ..., x_n\}$, які називаються словами, де кожний з аргументів $x_1, x_2, ..., x_n$ заміняється літерами алфавіту $\{0, 1, \ldots, k-1\}$. Кількість n літер у слові позначає його довжину.

Очевидно, що кількість усіляких слів завдовжки n в алфавіті потужністю k дорівнює k^n (пізніше доведемо це). У свою чергу, оскільки існує можливість зіставити кожному такому слову одне з k значень множини X, загальна кількість однорідних функцій n змінних визначається виразом

$$X_ф = k^n.$$

Якщо літерами алфавіту слугують символи від 0 до $k - 1$, то кожне слово $x_1, x_2, ..., x_n$ символічно подається упорядкованою послідовністю n таких чисел і розглядається як запис n-розрядного числа в позиційній системі числення з основою k, тобто

$$x_1 k^{n-1} + x_2 k^{n-2} + ... + x_{(n-1)} k^1 + x_n k^0 = q.$$

Числа $q = 0, 1, \ldots, k^{n-1}$ є номерами слів, і таким чином на множині всіх слів визначається природна впорядкованість. Аналогічно номерами функції можна вважати k^n-розрядні числа в тій самій системі числення.

У тризначному алфавіті $\{0, 1, 2\}$ словами завдовжки 4 будуть усі чотирирозрядні числа з основою $k = 3$, тобто 0000, 0001, 0002, ..., 2221, 2222, які відповідатимуть десятковим числам від 0 до 80, де $80 = 2 \cdot 3^3 + 2 \cdot 3^2 + 2 \cdot 3^1 + 2 \cdot 3^0$.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке неперервна інформація?
2. Що таке дискретна інформація?
3. Що таке слово?
4. Які є типи елементарних даних?
5. Що таке логічна функція?
6. Які функції називаються однорідними?

СПИСОК ЛІТЕРАТУРИ

1. *Жевержеев В. Ф., Калицкий Л. А., Сапегов Н. А.* Специальный курс высшей математики для вузов. — М.: Высш. шк., 1970. — 416 с.
2. *Справочник* по теории вероятностей и математической статистике / Под ред. В. С. Королюка. — К.: Наук. думка, 1978. — 582 с.
3. *Лапа В. Г.* Математические основы кибернетики. — К.: Вища шк., 1971. — 420 с.

РОЗДІЛ 2

МНОЖИНИ

2.1. Основні поняття теорії множин

Відповісти на питання «Що таке множина?» не так просто, як це здається на перший погляд. У повсякденному житті та практичній діяльності часто доводиться говорити про деякі сукупності різних об'єктів: предметів, понять, чисел, символів тощо. Наприклад, сукупність деталей механізму, сукупність сторінок у книзі, сукупність книг у бібліотеці, стадо овець, група студентів, сукупність друкарських знаків тощо.

На підставі інтуїтивних уявлень про будь-які подібні чітко визначені сукупності об'єктів сформувалося математичне поняття множини як об'єднання об'єктів у єдине ціле. Саме такої точки зору дотримувався засновник теорії множин німецький математик Георг Кантор.

Множина належить до категорії найзагальніших, основоположних понять математики. Так, група математиків, які працювали під псевдонімом Н. Бурбаки, стверджувала: «Множина утворюється з елементів, що мають певні властивості знаходяться у певних відношеннях між собою чи з елементами інших множин» або ж «Логічно кажучи, майже всю сучасну математику можна вивести з єдиного джерела: теорії множин».

Математичне поняття множини пов'язане з абстракцією, яку називатимемо абстракцією множини. Суть її полягає в тому, що існуючі зв'язки предметів, які об'єднуються між собою та з іншими предметами, ігноруються, а замість них предметам, що об'єднуються, приписуються нові зв'язки один з одним, які виражають їх належність множині. При цьому вважається, що два предмети, які нічим не різняться, є одним і тим самим предметом.

Об'єкти, що утворюють множину, називаються її елементами, або членами. Прикладами множин можуть бути: множина сторінок книги (кожна сторінка є елементом цієї множини); множина всіх дійсних чисел, більших від 0 і менших від 1; множина студентів тощо.

Множина є визначеною, коли можна встановити, чи є будь-який об'єкт її членом або ні.

Для позначення конкретних множин використовують великі літери A, S, X, ... або великі літери з індексами A_1, A_2 і т. д.

Для позначення елементів множин загалом застосовують малі літери a, s, x, ... або малі літери з індексами a_1, a_2 і т. д.

Наведемо ще кілька прикладів множин:

- множина натуральних чисел;
- множина цифр десяткової системи;
- множина цифр двійкової системи;
- множина парних чисел.

Таким чином, ми дійшли проблеми задання множин. При цьому наведені вище приклади множин задають описи характеристичних властивостей, які повинні мати їхні елементи.

2.1.1. Способи задання множин

Є кілька способів задання множин.

1. Вербальний (словесний) за допомогою опису характеристичних властивостей, які повинні мати елементи множин.

2. Список (перелік) усіх елементів у фігурних дужках. Стосовно зазначених вище прикладів маємо:

- $\{1, 2, 3,...\}$;
- $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$;
- $\{0, 1\}$;
- $\{2, 4,...\}$.

3. Предикатний (висловлювальний, породжувальний) за допомогою предиката, тобто множина задається у вигляді $\{x : P(x)\}$ або $\{x \mid P(x)\}$, де $P(x)$ набуває значення «істина» для елементів множини. Приклади предикатів:

- $\{1, 2, 3, 4, ...\} = \{x \mid x$ — натуральне число$\}$;
- $\{0, 1, 2, 3, 4, ..., 9\} = \{x \mid x$ — цифра десяткової системи$\}$;
- $\{0, 1\} = \{x \mid x$ — цифра двійкової системи$\} = \{x \mid x^2 - x = 0\}$;
- $\{2, 4, 6, ...\} = \{x \mid x$ — парне число$\} = \{x \mid \exists n \in N\,(x = 2n)\}$.

4. Аналітичний.

Із наведених прикладів випливає, що множини бувають скінченними та нескінченними. Множини називають скінченними, якщо число їх елементів скінченне, тобто існує натуральне число n, яке є числом елементів множини. Множини називають нескінченними, якщо вони містять нескінченне число елементів.

Для позначення того, що x є елементом S (тобто x належить S), будемо застосовувати запис $x \in S$, а запис $x \notin S$ означатиме, що елемент x не належить множині S.

Записом $x_1, x_2, ..., x_n \in S$ користуються як скороченням для запису $x_1 \in S, x_2 \in S, ..., x_n \in S$.

Символ $\in$ називається символом належності.

Введені вище поняття теорії множин з успіхом можуть бути використані в основах аналізу, алгебрі, математичній логіці та ін. Однак при більш строгому розгляді такі інтуїтивні уявлення можуть виявитися незадовільними. Недосконалість інтуїтивних уявлень про множини, їх недостатність ілюструється, наприклад, відомим парадоксом Б. Рассела, який формулюється так. Розглянемо множину A всіх таких множин X, що X не є елементом X. Тоді, якщо A не є елементом A, то за означенням A також є елементом A. З іншого боку, якщо A є елементом A, то A — одна з тих множин X, які не є елементами самих себе, тобто A не є елементом A. У будь-якому випадку A є елементом A й A не є елементом A.

Цей парадокс свідчить про те, що теорія множин, яка широко використовується в її інтуїтивному, «наївному» викладі, є суперечливою. Формалізація теорії множин, пов'язана, зокрема, з усуненням парадоксів, сприяла розвитку не тільки методів теорії множин, а й математичної логіки.

2.1.2. Порожня множина

У теорії множин використовується поняття порожньої множини. Позначається вона символом $\varnothing$.

Множина може взагалі не містити елементів, наприклад

$$S = \{x \mid \text{x — непарне число, що ділиться на 2}\} = \varnothing;$$

$$K = \left\{x \in R \middle| x^2 + 1 = 0\right\} = \varnothing.$$

Для позначення цього факту вводиться поняття порожньої множини.

Це поняття відіграє дуже важливу роль при заданні множин за допомогою опису. Так, без поняття порожньої множини не можна говорити про множину відмінників групи або про множину дійсних коренів квадратного рівняння, не пересвідчившись заздалегідь, чи є взагалі в студентській групі відмінники або чи має задане рівняння дійсні корені. Поняття порожньої множини дає змогу оперувати множиною відмінників групи, не піклуючись про те, чи є відмінники в групі, яка розглядається. Порожню множину умовно будемо відносити до скінченних множин.

Таким чином, уведення порожньої множини дає можливість оперувати будь-якою множиною без попереднього застереження, існує вона чи ні.

2.1.3. Операції над множинами

Розглянемо дві множини A та B і введемо кілька операцій над ними. Для графічної ілюстрації будемо використовувати кола Ейлера. Для зображення множини на площині креслять замкнену лінію із заштрихованою внутрішньою областю (найчастіше — це коло, звідси й назва відповідного інструмента, що широко застосовується в теорії множин).

Зазначимо, що в подальшому викладі використовуватимемо символи логічних операцій кон'юнкції «$\wedge$», диз'юнкції «$\vee$», імплікації «$\rightarrow$», еквіваленції «$\Leftrightarrow$» (вони будуть повністю описані в розд. 3) для формалізованого запису означень і теорем.

1. *Об'єднання A і B* $(A \cup B)$ — множина, що складається з усіх елементів множини A, всіх елементів множини B і не містить ніяких інших елементів (рис. 2), тобто

$$A \cup B = \{x \mid x \in A \vee x \in B\},$$

де символ «$\vee$» позначає логічну операцію диз'юнкції (логічне «або»).

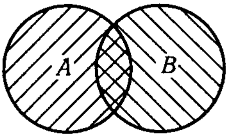


Рис. 2

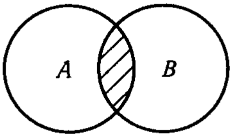


Рис. 3

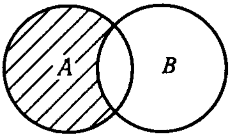


Рис. 4

2. *Переріз A і B* — множина, що складається з тих і тільки тих елементів, які належать одночасно множині A та множині B (рис. 3), тобто

$$A \cap B = \{x | x \in A \wedge x \in B\},$$

де символ «$\wedge$» позначає логічну операцію кон'юнкції (логічне «і»).

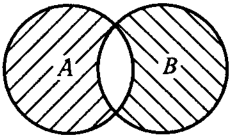


Рис. 5

3. *Різниця A і B* (відносне доповнення) — множина, що складається з тих і тільки тих елементів, які належать множині A й не належать B (рис. 4), тобто

$$A - B = \{x | x \in A \wedge x \notin B\}.$$

4. *Диз'юнктивна сума A і B* (симетрична різниця) — множина, що складається з усіх елементів A, які не належать множині B, й усіх елементів B, які не належать множині A, та яка не містить ніяких інших елементів (рис. 5), тобто

$$A \oplus B = \{x | x \in A \oplus x \in B\},$$

де для позначення операції диз'юнктивної суми двох множин використано той самий символ $\oplus$, що й для логічної операції додавання за mod 2, яка також позначається символом $\oplus$.

Очевидно, що

$$A \oplus B = (A - B) \cup (B - A).$$

2.1.4. Універсум *U*

Звичайно, вже в означенні конкретної множини явно або неявно обмежується сукупність об'єктів, що є допустимими. (Слони — серед тварин, натуральні числа — серед цілих або дійсних залежно від контексту).

Зручно сукупність допустимих об'єктів зафіксувати явно та вважати, що множини, які розглядаються, складаються з елементів цієї сукупності. Її називають основною множиною (універсумом) і позначають U.

Універсум U арифметики — числа, універсум U зоології — тварини і т. д.

Будь-яку множину розглядатимемо у зв'язку з універсумом, який на колах Ейлера асоціюватимемо з прямокутником на площині, всередині якого зображатимемо множини (рис. 6).

Нова операція $U - A = \overline{A}$ (абсолютне доповнення A) — це множина, що містить усі елементи універсуму, за винятком елементів A (рис. 7).

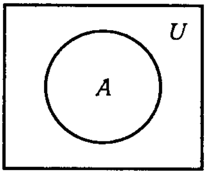


Рис. 6

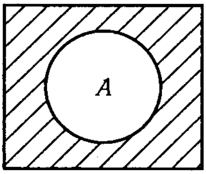


Рис. 7

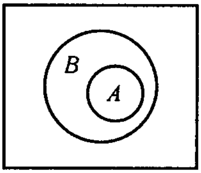


Рис. 8

Означення 2.1. *Множина A називається підмножиною множини B, якщо кожен елемент A є елементом B.*

Для позначення цього факту вводиться знак «$\subset$» — символ включення (або «$\subseteq$»); іншими словами, $A\subset B \Leftrightarrow \forall x(x\in A \Rightarrow x\in B)$ (рис. 8).

Якщо необхідно підкреслити, що множина A містить також інші елементи, крім елементів множини B, то використовують символ строгого включення: $У\subset A$. Зв'язок між символами $\subset$ і $\subseteq$ задається виразом $B\subset A \Leftrightarrow B\subseteq A \wedge B\neq A$. Загалом будемо використовувати символ «$\subset$».

Говорять, що множина B є істинною (або власною, від слова «власне») підмножиною A, якщо $B\subset A$ і $B\neq A$, на відміну від неістинних (або невласних) підмножин $\varnothing$ та U будь-якої множини A.

Порожня множина не містить елементів. Отже, додаючи до множини A порожню множину, ми фактично нічого не додаємо. Тому завжди можна вважати, що будь-яка множина A містить порожню множину як підмножину.

Нехай $A = \{x\,|\,x$ — людська істота$\}$ і $B = \{x\,|\,x$ — людська істота жіночої статі$\}$; тоді зрозуміло, що $B\subset A$, а B — істинна підмножина A.

Треба бути уважним, щоб розрізняти елементи множини та підмножини цієї множини. Наприклад, коли пишуть $a\in\{a, b, c\}$, це означає, що елемент a є членом множини, що складається з трьох елементів: a, b і c. Коли ж пишуть $\{a\}\subset\{a, b, c\}$, це означає, що множина, що складається з одного елемента a, є підмножиною множини, яка складається з трьох елементів: a, b, c.

Означення 2.2. *Дві множини рівні, якщо вони складаються з одних і тих самих елементів:*

$$A = B \Leftrightarrow \forall x(x\in A \Leftrightarrow x\in B).$$

Наприклад, $\{1, 2, 3\} = \{3, 2, 1\}$.

Твердження 2.1. *Справджується така рівність:*

$$A\subset B \wedge B\subset A \Leftrightarrow A = B.$$

Доведення необхідності. Розглянемо будь-який елемент $x\in A$. Множина A є підмножиною B, тому $x\in B$. З іншого боку, будь-який елемент $x\in B$ (оскільки $B\subset A$) належить також множині A, тобто $x\in A$. За означенням рівності множин маємо $A = B$. Необхідність доведено.

Доведення достатності. Розглянемо будь-який елемент $x \in A$. Оскільки $A = B$, маємо $x \in B$. Тоді за означенням включення множин $A \subset B$. Розглянемо будь-який елемент $x \in B$. Якщо $A = B$, то $x \in A$. За означенням включення $B \subset A$. Достатність доведено.

Аналогічно можна довести таке твердження про транзитивність включення множин.

***Твердження 2.2.** Справджується така рівність: $A \subset B \wedge B \subset C \Rightarrow A \subset C$.*

***Твердження 2.3.** Порожня множина єдина.*

Доведемо єдиність порожньої множини. Доведення проведемо від супротивного: припустимо, що існують дві порожні множини, тобто $\varnothing_1$ і $\varnothing_2$. Розглянемо два висловлення: $x \in \varnothing_1$ та $x \in \varnothing_2$. Обидва вони тотожно хибні, тому

$$x \in \varnothing_1 \Leftrightarrow x \in \varnothing_2.$$

За означенням рівності множин маємо $\varnothing_1 = \varnothing_2$. Одержана суперечність доводить єдиність порожньої множини. Приймемо без доведення такі твердження: $A \subset B \cup A$; $B \subset A \cup B$; $A \cap B \subset A$; $A \cap B \subset B$; $A - B \subset A$.

2.1.5. Множина підмножин

Слід розуміти, що елементи множини самі можуть бути деякими множинами. Наприклад, книга з множини книг у шафі може розглядатися як множина сторінок. Потрібно звернути увагу на те, що йдеться про елементи множини, а не про підмножини (ніяка сукупність сторінок не може розглядатися як підмножина множини книг).

***Означення 2.3.** Множину, елементами якої є всі підмножини множини A, називають множиною підмножин (множиною-степенем) множини A і позначають P(A).* Так, для триелементної множини $A = \{a, b, c\}$ маємо $P(A) = \{\varnothing, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$.

У разі кінцевої множини A, що складається з n елементів, множина підмножин $P(A)$ містить 2^n елементів. Доведення ґрунтується на підсумовуванні всіх коефіцієнтів розкладу бінома Н'ютона або на поданні підмножин n-розрядними двійковими числами, в яких 1 (або 0) відповідають елементам підмножин (доведення буде наведено в п. 5.4).

Слід підкреслити відмінності між відношенням належності $(\in)$ і відношенням включення $(\subset)$. Як уже зазначалося, множина A може бути своєю підмножиною $(A \subset A)$, але вона не може входити до складу своїх елементів $(A \notin A)$. Навіть у разі одноелементних підмножин потрібно відрізняти множину $A = \{a\}$ та її єдиний елемент a. Відношення включення має властивість транзитивності, відношення належності цієї властивості не має. Наприклад, множина $A = \{\{1\}, \{2, 3\}, \{4\}\}$ у числі своїх елементів містить множину $\{2, 3\}$, тоді можна записати:

$$2, 3 \in \{2, 3\} \text{ і } \{2, 3\} \in A.$$

Однак це не означає, що елементи 2 та 3 є в множині A (в наведеному прикладі немає 2 і 3 серед елементів множини A, тобто $2, 3 \notin A$).

2.1.6. Алгебра множин

Операції над множинами, як і операції над числами, мають деякі властивості (табл. 2.1). Останні виражаються сукупністю тотожностей незалежно від конкретного вмісту множин, що входять у них, і є підмножинами деякого універсуму *U*.

Таблиця 2.1

Комутативний закон	
1а) $A \cup B = B \cup A$	1б) $A \cap B = B \cap A$
Асоціативний закон	
2а) $A \cup (B \cup C) = (A \cup B) \cup C$	2б) $A \cap (B \cap C) = (A \cap B) \cap C$
Дистрибутивний закон	
3а) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	3б) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Властивості $\varnothing$ та U	
4а) $A \cup \varnothing = A$ 5а) $A \cup \overline{A} = U$ 6а) $A \cup U = U$ 7а) $\overline{\varnothing} = U$	4б) $A \cap U = A$ 5б) $A \cap \overline{A} = \varnothing$ 6б) $A \cap \varnothing = \varnothing$ 7б) $\overline{U} = \varnothing$
Закон ідемпотентності (самопоглинання)	
8а) $A \cup A = A$	8б) $A \cap A = A$
Закон поглинання	
9а) $A \cup (A \cap B) = A$	9б) $A \cap (A \cup B) = A$
Теорема де Моргана	
10а) $\overline{A \cup B} = \overline{A} \cap \overline{B}$	10б) $\overline{A \cap B} = \overline{A} \cup \overline{B}$
Властивості доповнення, різниці та рівності	
11) $A \cup B = U \wedge A \cap B = \varnothing \Rightarrow B = \overline{A}$ 12) $\overline{\overline{A}} = A$ 13) $A - B = A \cap \overline{B}$ 14) $A \oplus B = (A \cap \overline{B}) \cup (\overline{A} \cap B)$ 15) $A \oplus B = B \oplus A$ 16) $(A \oplus B) \oplus C = A \oplus (B \oplus C)$ 17) $A \oplus \varnothing = \varnothing \oplus A = A$ 18) $A \subset B \Leftrightarrow A \cap B = A \Leftrightarrow A \cup B = B \Leftrightarrow A \cap \overline{B} = \varnothing$ 19) $A = B \Leftrightarrow (A \cap \overline{B}) \cup (\overline{A} \cap B) = \varnothing$	

2.1.7. Методи доведення тотожностей алгебри логіки

Основний метод доведення тотожності в алгебрі множин ґрунтується на застосуванні означення 2.2 і твердження 2.1. Доведемо, наприклад, тотожність 3а) $A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$, використовуючи твердження 2.1.

Доведемо, що $A\cup(B\cap C)\subset(A\cup B)\cap(A\cup C)$. Для цього візьмемо будь-яке $x\in A\cup(B\cap C)$, тоді за означенням операцій «$\cup$» та «$\cap$» маємо $x\in A\vee(x\in B\wedge x\in C)$. За законом дистрибутивності диз'юнкції відносно кон'юнкції $(x\in A\vee x\in B)\wedge(x\in A\vee x\in C)$, тобто $x\in A\cup B\wedge x\in A\cup C$ або $x\in(A\cup B)\cap(A\cup C)$, що й треба було довести.

Доведемо, що $(A\cup B)\cap(A\cup C)\subset A\cup(B\cap C)$. Для цього візьмемо будь-яке $x\in(A\cup B)\cap(A\cup C)$. Звідси $(x\in A\vee x\in B)\wedge(x\in A\vee x\in C)$ або $x\in$ $\in A\vee(x\in B\wedge x\in C)$, тобто $x\in A\cup(B\cap C)$, що й потрібно було довести.

Згідно з твердженням 2.1 $A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$.

Будь-яка теорема алгебри множин виводиться з тотожностей табл. 2.1. Властивості 1 — 5 і 13, у свою чергу, можна довести тільки в термінах належності. Наприклад, доведемо властивість 8а) $A\cup A=A$, послідовно використовуючи властивості 4б), 5а), 3а), 5б), 4а):

$$A\cup A=(A\cup A)\cap U=(A\cup A)\cap\left(A\cup\overline{A}\right)=A\cup\left(A\cap\overline{A}\right)=A\cup\varnothing=A.$$

2.1.8. Узагальнення операцій над множинами

Із властивостей комутативності й асоціативності операцій об'єднання випливає, що об'єднання кількох множин можна виконати, послідовно об'єднуючи їх, причому порядок входження множин не впливає на результат, наприклад $A\cup B\cup C=(A\cup B)\cup C=A\cup(B\cup C)=(B\cup C)\cup A$. Отже, об'єднання сукупності множин можна подати співвідношенням

$$A_1\cup A_2\cup\ldots\cup A_n=\bigcup_{i=1}^{n}A_i.$$

Аналогічно на n множин узагальнюється операція перерізу:

$$A_1\cap A_2\cap\ldots\cap A_n=\bigcap_{i=1}^{n}A_i.$$

Використовуючи узагальнення операцій об'єднання та перерізу на n множин, можна узагальнити також інші співвідношення, наприклад закон де Моргана, який в узагальненому вигляді має вигляд

$$\overline{\bigcup_{i=1}^{n}A_i}=\bigcap_{i=1}^{n}\overline{A_i}\quad\text{і}\quad\overline{\bigcap_{i=1}^{n}A_i}=\bigcup_{i=1}^{n}\overline{A_i}\ .$$

Означення 2.4. *Сукупність множин* A_1, A_2, ..., A_n *називається розбиттям множини A, якщо:*

1. $\bigcup_{i=1}^{n} A_i = A$;
2. $A_i \cap A_j = \varnothing \ \ \forall i \neq j$.

Введемо ще одну операцію над множинами.

Означення 2.5. *Прямим (або декартовим) добутком множин A і B називається множина всіх упорядкованих пар елементів (a, b), з яких перший належить множині A, а другий — множині B:*

$$A \times B = \{(x, y) | x \in A \wedge y \in B\}.$$

Порядок уходження пар може бути будь-яким, але розташування елементів у кожній парі визначається порядком множин, що перемножуються. Тому $A \times B \neq B \times A$, тобто прямий добуток властивості комутативності не має.

Операція прямого добутку множин узагальнюється на будь-яку їх кількість і записується у вигляді

$$\prod_{i=1}^{n} A_i = A_1 \times A_2 \times ... \times A_n,$$

причому елементом прямого добутку n множин є впорядкована послідовність із n елементів $(a_1, a_2, ..., a_n)$, яка називається ще кортежем або вектором завдовжки n, а також n-кою.

Властивості асоціативності для прямого добутку також не виконуються, але виконується властивість дистрибутивності відносно об'єднання, перерізу і відносного доповнення (різниці):

$$(A_1 \cup A_2) \times B = (A_1 \times B) \cup (A_2 \times B);$$

$$(A_1 \cap A_2) \times B = (A_1 \times B) \cap (A_2 \times B);$$

$$(A_1 - A_2) \times B = (A_1 \times B) - (A_2 \times B).$$

Якщо як співмножник декартового добутку n-множин використовується одна множина A, то це записується так:

$$A \times A \times A \times ... \times A = A^n.$$

Операція декартового добутку відрізняється від операцій, введених раніше, тим, що елементи добутку множин суттєво відрізняються від елементів співмножників і є об'єктами іншої природи. Наприклад, якщо R — множина дійсних чисел, то декартовий добуток $R \times R$ — множина всіх точок площини.

2.2. Нечіткі множини та лінгвістичні змінні

Нечіткі множини широко використовуються в різних застосуваннях штучного інтелекту, теорії розпізнавання образів, теорії прийняття рішень тощо. Якщо до твердження «логічно кажучи, можна вивести майже всю сучасну математику з єдиного джерела — теорії множин» додати концепцію нечіткості, то це відкриє шлях до «подвоєння» математики: замінюючи звичайну множину нечіткою (розпливчастою), можна кожному об'єкту в математиці поставити у відповідність його нечіткий (розпливчастий) аналог.

Нехай множина $X = \{x\}$ є сукупністю об'єктів (точок), що позначаються через x. Тоді нечітка множина A в X є сукупністю упорядкованих пар $A = \{x;\ \mu_a(x)\}$, де $x \in X$, а $\mu_a(x)$ — ступінь належності x до A, тобто $\mu_a\colon X \to M$ — функція, що відображає X у простір M, який називається простором належності. Коли M містить тільки дві точки 0 та 1, множина A не є нечіткою, її функція належності збігається з характеристичною функцією нерозпливчастої множини (звичайної чіткої множини).

У подальшому вважатимемо, що M є відрізком [0, 1], причому 0 й 1 є відповідно нижчим і вищим ступенями належності. Основне припущення полягає в тому, що розпливчаста множина A може бути точно визначена зіставленням кожному об'єкту x числа, яке знаходиться в діапазоні від 0 до 1 і відображає ступінь його належності A.

Нехай {0, 1, 2,... } — сукупність натуральних чисел. У цьому просторі розпливчаста множина A «кілька об'єктів» може бути суб'єктивно визначена як набір упорядкованих пар $A = \{(3;\ 0{,}6), (4;\ 0{,}8), (5;\ 1{,}0), (6;\ 1{,}0), (7;\ 0{,}8)\}$, причому вважається, що перелічено тільки ті пари $(x;\ \mu_a(x))$, для яких функція $\mu_a(x)$ додатна.

У теорії нечітких множин так само, як і в теорії чітких множин, широко використовується поняття універсальної множини. При цьому універсальною множиною X нечіткої множини A називається область визначення функції належності μ_a.

Нечітка множина $A = \{x;\ \mu_a(x)\}$, $x \in X$, визначається математично як сукупність упорядкованих пар, складених з елементів x універсальної множини X_{ai} і відповідних ступенів належності $\mu_a(x)$ або безпосередньо у вигляді функції

$$\mu_a : X \to [0,1].$$

У науковій літературі можливі такі записи нечітких множин:

$$A = \{(x_1;\ 0{,}2), (x_2;\ 0{,}6), (x_3;\ 0{,}3), (x_4;\ 0{,}8), (x_5;\ 1)\};$$

$$A = \{(0{,}2|x_1), (\ 0{,}6|x_2), (\ 0{,}3|x_3), (\ 0{,}8|x_4), (1|x_5)\}$$

або у вигляді табл. 2.2.

Можливі конфігурації функцій μ_a зображено на рис. 9.

Таблиця 2.2

X	x_1	x_2	x_3	x_4	x_5
$\mu_a(x)$	0,2	0,6	0,3	0,8	1

Щоб детальніше пояснити поняття нечіткої підмножини, розглянемо такий приклад. Передбачимо, що деяка кінцева множина U складається з п'яти елементів $U=\{x_1, x_2, x_3, x_4, x_5\}$, і нехай

$$A=\{(x_1;0,2),(x_2;0),(x_3;0,3),(x_4;0,8),(x_5;1)\}.$$

Отже, нечітка підмножина A: у невеликій мірі містить x_1; не містить x_2; містить x_3 трохи більше, ніж x_1; у значно більшій мірі містить x_4 і повністю містить x_5. Таким чином, можна створити математичну структуру, що дає змогу оперувати елементами, які неповно визначаються і належність яких цій підмножині лише в якійсь мірі ієрархічно впорядковано. До таких структур можна, наприклад, віднести: у заданій множині людей — деяку підмножину дуже високих людей; у множині основних кольорів — нечітку підмножину темно-зелених кольорів; у множині рішень — нечітку підмножину правильних рішень і т. д.

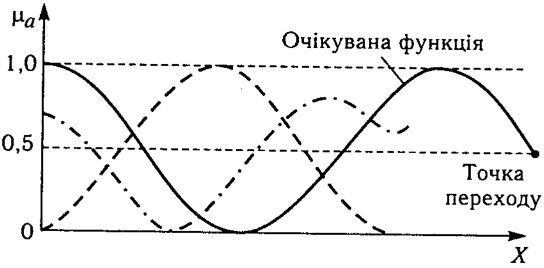


Рис. 9

Наведемо означення поняття нечіткої підмножини, введеного Л. Заде [3]: «Нечітка підмножина A універсальної множини U характеризується функцією належності $\mu_a : U \to [0,1]$, що ставить у відповідність кожному елементу $u \in U$ число $\mu_a(u)$ із множини $[0,1]$ і характеризує міру належності елемента u підмножині A».

Означення 2.6. *Множина, що містить єдиний елемент, називається синглетоном. Останній може визначатися як серед чітких, так і серед нечітких множин.*

Означення 2.7. *Носієм нечіткої множини A називається множина таких точок в U, для яких функція $\mu_a(u)$ — додатна.*

Означення 2.8. *Висотою нечіткої множини A називається величина* $\sup\limits_{u} \mu_a(u)$.

Означення 2.9. *Точкою переходу нечіткої множини A називається такий елемент множини U, ступінь належності якого множині A дорівнює 0,5 (див. рис. 9).*

Розгляд нечіткої множини як підмножини деякої універсальної множини випливає з таких міркувань. У теорії звичайних множин введено поняття характеристичного числа

$$X_A=\begin{cases}1, & \text{якщо} \quad x \in A;\\ 0, & \text{якщо} \quad x \notin A.\end{cases}$$

Використовуючи це поняття і форму запису нечіткої множини, наприклад

$$A=\{(0,2|x_1);(0,6|x_2);(0,3|x_3);(0,8|x_4);(1|x_5)\},$$

універсальну множину U можна записати у вигляді

$$U=\{(1|x_1),(1|x_2),(1|x_3),(1|x_4),(1|x_5)\}.$$

У цьому виразі міра належності μ_a всіх елементів x_i множині U дорівнює 1. З порівняння двох останніх виразів випливає, що множину A можна трактувати як підмножину U в сенсі виконання нерівності

$$\mu_a(x_i)\le\mu_a\le 1\ \forall x\in U$$

для всіх $x\in U$. Так, якщо функцію $\mu_a(x)$ визначено на всьому інтервалі [0,1], на відміну від характеристичного числа, яке набуває тільки двох значень, то нечітка множина є узагальненням поняття множини в теорії звичайних множин.

Із розглянутого випливає, що підмножина A універсальної множини U, з одного боку, характеризує нечіткі поняття або інші типи невизначеності, а з іншого — сама визначається функцією належності $\mu_a(x)$, яка ставить у відповідність кожному елементу $x\in U$ число $\mu_a(x)$ з інтервалу [0,1]. Це число визначає, в якій мірі елемент належить нечіткому поняттю або, що еквівалентно, підмножині A універсальної множини U.

Нечітку підмножину A універсальної множини U називають нормальною, якщо виконується умова

$$\max_{x\in X}\mu_a(x)=1.$$

У разі невиконання останньої рівності нечітка підмножина називається субнормальною.

Як правило, при формалізації первинних об'єктів, що дають якісну інформацію про об'єкт дослідження, формують нормальні нечіткі підмножини. Проте часто після виконання операцій об'єднання, перерізу та інших нечіткі підмножини можуть перейти з нормальної форми в субнормальну, тоді при розв'язуванні задач може виникнути потреба виконати нормалізацію нечітких підмножин. Це досягається діленням функцій належності нечіткої підмножини на її максимальне значення: $\mu_a'(x)=\dfrac{\mu_a(x)}{\max\limits_{x\in X}\mu_a(x)}$.

Нечіткі лінгвістичні змінні. Подальшим узагальненням поняття нечіткої множини є поняття лінгвістичної змінної.

Означення 2.10. *Лінгвістичною змінною називається кортеж*

$$(X,\ T(X),\ U,\ J,\ M),$$

де X — назва (ім'я) змінної; $T(X)$ (або просто T) —базова множина змінної X, тобто множина назв лінгвістичних значень змінної X зі значеннями з універсальної множини U; U — універсальна множина; J — синтаксичне правило (що має форму граматики), яке породжує назву T значень змінної X; M — семантичне правило, що ставить у відповідність кожній *нечіткій змінній X її значення M(X), тобто нечітку підмножину M(X) універсальної множини U.*

Конкретна назва X, породжена синтаксичним правилом, називається термом.

Терм, що складається з одного або кількох слів, які завжди фігурують разом, називають атомарним.

Терм, що складається з одного або більше атомарних термів, називається складеним.

Конкатенація деяких компонентів складеного терму (тобто результат приписування один до одного ланцюжків — компонентів складеного терму) є підтермом.

Пояснимо поняття лінгвістичної змінної на конкретному прикладі.

Нехай експерту треба оцінити вартість випуску продукції за допомогою понять «мала», «середня» і «висока». Максимальна вартість продукції становить 5000 грн. Для формалізації цього опису використовуємо поняття лінгвістичної змінної

$$\{X,\ T(X),\ U,\ J,\ M\},$$

де X — вартість (назва змінної); T — базова терм-множина, що складається з множини назв лінгвістичних значень змінної X; тут $T = \{T_1$ (мала), T_2(середня), T_3(висока)$\}$; U — універсальна множина, тут від 0 до 5000 грн; J — процедура перебору елементів множини T (мала, середня, висока); M — процедура експертного опиту.

На рис. 10 показано функції належності відповідних нечітких змінних T_1, T_2, T_3.

Існує кілька основних аспектів поняття лінгвістичної змінної, які потребують уточнення.

По-перше, важливо зрозуміти, що поняття функції належності відмінне від поняття ймовірності. Так, висловлення про те, що значення функції належності «середня вартість» дорівнює 0,5, не має ніякого відношення до ймовірності того, що значення змінної «вартість» дорівнює 2500 грн.

Правильна інтерпретація значення функції належності, яке дорівнює 0,5, полягає в тому, що воно є лише суб'єктивною мірою того, наскільки вартість суми 2500 грн відповідає в поданні суб'єкта слову «середня». Математичні операції, які застосовуються до значень функції належності, відмінні від операцій, що застосовуються до значень імовірностей, хоча між ними існує деяка аналогія.

По-друге, лінгвістична змінна має структуру в тому значенні, що вона пов'язана двома правилами:

- синтаксичним – визначає спосіб породження лінгвістичних значень, які належать терму — множині цієї змінної;
- семантичним — визначає спосіб обчислення значень лінгвістичної змінної.

Зазначимо у зв'язку з цим, що типове значення лінгвістичної змінної, наприклад «не дуже мала вартість» і «не дуже висока вартість», включає те, що можна було б назвати первинними термами, наприклад «мала» та «середня», значення яких суб'єктивне і залежить від контексту. Передбачається, що значення таких термів визначено заздалегідь.

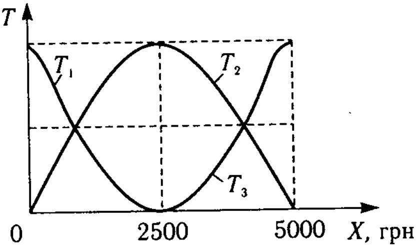


Рис. 10

Дамо означення понять «синтаксичне» та «семантичне» правила.

Синтаксичне правило (*процедура*)* — опис процесу утворення нових, осмислених для певної задачі управління, значень лінгвістичної змінної, виходячи з її терму — множини.

Семантичне правило (*процедура*)** — процес, що дає змогу перетворити кожне нове значення лінгвістичної змінної, яка утворюється синтаксичною процедурою, на нечітку змінну, тобто приписати йому деяку семантику формуванням відповідної нечіткої множини.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке множина? Наведіть приклади різних множин.
2. Як можна визначити множину?
3. В яких випадках можна говорити, що множини K_1, K_2 і K_3 є рівними?
4. Що таке скінченна та нескінченна множини?
5. До якої з цих множин належить порожня множина?
6. Для чого використовується порожня множина?
7. Що таке підмножина?
8. Чи завжди будь-яка множина містить порожню множину? Аргументуйте відповідь.
9. Чим відрізняється за значенням строге включення $\subset$ від включення $\subseteq$? Аргументуйте відповідь.
10. Чим відрізняється поняття включення ($\subset$ або $\subseteq$) від поняття належності ($\in$)?
11. Як розрізнюють елементи множини та її підмножини? Відповідь ілюструйте прикладами.
12. Що таке множина-степінь? Які використовуються при цьому позначення?
13. Чи може множина входити до складу своїх елементів?
14. Що таке розбиття із позиції теорії множин? Наведіть приклади різних класів.
15. Які є способи задання множини?
16. Що таке універсум?
17. Що таке простір і функція належності?
18. Що таке розпливчаста (нечітка) множина? Наведіть приклади таких множин.
19. Як визначається універсальна множина в теорії нечітких множин?
20. Які є типи функцій належності?
21. Що таке нечітка підмножина?
22. Що таке носій нечіткої множини?

**Синтаксис* — розділ граматики, що вивчає структуру речень і сполучень слів у реченні; сукупність правил утворення правильних (допустимих) конструкцій мови.

***Семантика* — смисловий аспект мови слів, частин слова, словосполучень; наука, що розглядає питання сполучення між елементами і їхніми смисловими значеннями.

23. Що таке висота нечіткої множини?
24. Що таке точка переходу нечіткої множини?
25. Чим різняться нормальна та субнормальна нечіткі підмножини?
26. Як досягається нормалізація субнормальних нечітких підмножин?
27. Що таке лінгвістична змінна?
28. Що таке базова терм-множина?
29. Що таке атомарний терм?
30. Що таке підтерм? Наведіть приклади різних підтермів.
31. Що таке синтаксичне правило?
32. Що таке семантичне правило?

ПЕРЕЛІК ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

1. Множини. Операції над ними.
2. Основні тотожності алгебри множин. Тотожні перетворення формул алгебри множин.
3. Доведення тотожності алгебри множин.
4. Нечіткі множини. Операції над ними.

СПИСОК ЛІТЕРАТУРИ

1. *Нефедов В. Н., Осипова В. А.* Курс дискретной математики. — М.: Изд-во МАИ, 1992. — 264 с.
2. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера. — М.: Энергоатомиздат, 1988. — 480 с.
3. *Заде Л. А.* Понятие лингвистической переменной и его применение к принятию приближенных решений. — М.: Мир, 1976. — 176 с.
4. *Кофман А.* Введение в теорию нечетких множеств. — М.: Радио и связь, 1982. — 276 с.
5. *Куратовский К., Мостовский Я.* Теория множеств. — М.: Мир, 1970. — 416 с.
6. *Лавров И. А., Максимова Л. Л.* Задачи по теории множеств, математической логике и теории алгоритмов. — М.: Наука, 1975. — 240 с.
7. *Слупецкий Е., Борковский Л.* Элементы математической логики и теории множеств. — М.: Прогресс, 1985. — 368 с.
8. *Сигорский В. П.* Математический аппарат инженера. — К.: Техніка, 1977. — 768 с.

РОЗДІЛ 3

МАТЕМАТИЧНА ЛОГІКА

Основи математичної логіки закладено в працях англійського математика Джорджа Буля (1815 — 1864), батька Етель Ліліан Войнич, яка є автором книги «Овід». Це такі праці, як «Математичний аналіз логіки» (1847) і «Закони мислення» (1854), де він уперше виклав алгебру логіки — алгебру Буля. Її формули застосовні незалежно від того, що мати на увазі під літерами, які вживаються в алгебрі. В алгебрі Буля літери позначають висловлення, а всі правила звичайної алгебри залишаються без зміни. Оскільки всі наші міркування складаються з висловлень або думок, булева алгебра є логікою, через що вона дістала назву алгебри логіки.

Праці Дж. Буля можна порівняти з працями М. І. Лобачевського. Дж. Буль запропонував у формулах літерами позначити не числа, а висловлення і показав, що можна так вибирати дії додавання та множення, щоб формули звичайної алгебри залишалися без зміни. В алгебрі логіки висловлення розглядаються не за їхнім змістом або значенням, а тільки відносно того, істинні вони чи хибні. Приймається, що кожне висловлення може бути тільки істинним або хибним. Істинність висловлення будемо позначати 1, а хибність — 0.

3.1. Булеві функції

3.1.1. Основні поняття та означення

Булеві функції належать до класу двозначних однорідних функцій. Це найпростіший і водночас найважливіший клас однорідних функцій, що використовуються для опису скінченних автоматів та ЕОМ. Останні, у свою чергу, призначаються для перероблення дискретної інформації. Як модель засобів перероблення застосовується поняття автомата. Для формального опису цифрового автомата слугує апарат алгебри логіки (булевої алгебри). Останню утворюють множини всіх булевих функцій разом з операціями заперечення, кон'юнкції, диз'юнкції, імплікації тощо.

Основним поняттям алгебри логіки є висловлення. Образно кажучи, висловлення — це деяке твердження, про яке можна сказати, що воно є істинним або хибним. Наприклад, «Херсон — місто на Дніпрі», «Сонце сходить уранці» — істинні висловлення, а «На вулиці йде дощ» може бути істинним або хибним залежно від додаткових відомостей. Будь-яке висловлення можна позначити символом x і вважати, що $x = 1$ при істинності, а $x = 0$ при хибності висловлення.

Будемо розглядати функції $f(x_1, x_2, ..., x_n)$, аргументи яких визначено на множині $E_2 = \{0,1\}$, такі, що $f(x_1, x_2, ..., x_n) \in E_2$, коли $x_i \in E_2$ $(i = 1, 2, ..., n)$. Ці функції називатимемо функціями алгебри логіки, або булевими функціями. Таким чином, запис $f(x_1, x_2, ..., x_n)$ є записом функції, яка залежить від довільних фіксованих аргументів.

Логічними (булевими) змінними в булевій алгебрі називають величини, які незалежно від їхньої конкретної суті можуть набувати лише двох значень. Ці значення будемо позначати нулем (0) й одиницею (1). Якщо змінна x має одиничне значення, то записуємо $x = 1$, а якщо нульове, то $x = 0$. *Булевою,* або *перемикальною,* функцією $f(x_1, x_2, ..., x_{n-1})$ називають функцію, яка, як і її n аргументів, може набувати лише двох значень: 0 або 1. В обчислювальній техніці булеві функції застосовуються для опису алгоритмів, засобів цієї техніки — дискретних пристроїв. Останні призначаються для перетворення дискретної інформації, що розкладається на елементарні одиниці — біти, які в пристроях реалізуються сигналами, що описуються двійковими змінними — булевими.

Сукупність значень аргументів є кортежем, точкою або набором. Функція, що залежить від n аргументів, називається n-місною і є повністю визначеною, якщо задано її значення для всіх наборів (кортежів, точок) значень аргументів. Кожному i-му кортежу можна поставити у відповідність терм — довільний елементарний добуток двійкових змінних $t_i = \tilde{x}_m \tilde{x}_{m-1} ... \tilde{x}_2 \tilde{x}_1 \tilde{x}_0$. Якщо в i-му кортежі $x_j = 1$, то в термі замість $\tilde{x}_j$ записується змінна x_j, а якщо $x_j = 0$, то $\overline{x}_j$.

3.1.2. Способи задання булевих функцій

Існує три способи задання перемикальної функції: вербальний (або словесний), аналітичний і табличний. Аналітичне задання функції — опис її аналітичним виразом (формулою). Наприклад, $f_1(x_1 x_2 x_3) = x_1 x_2 + x_2(\overline{x}_3 + x_1)$; $f_2(abc) = abc + \overline{a}b\overline{c}$. Одним із поширених способів задання булевої функції є її задання за допомогою таблиці відповідності (істинності). У табл. 3.1 у трьох перших колонках задано всі можливі кортежі (набори) значень трьох аргументів, тобто поєднання їхніх нульових та одиничних значень, а в четвертій колонці — значення функції цих аргументів.

Будь-яке ціле невід'ємне число можна записати у вигляді суми

$$N = g_1 r^{n-1} + g_2 r^{n-2} + ... + g_{n-1} r^1 + g_n r^0,$$

де r — основа системи числення; g — співмножник, що набуває значень від 0 до $r - 1$.

Кількість доданків визначається розрядністю чисел.

Таблиця 3.1

x_2	x_1	x_0	$f(x_2, x_1, x_0)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Кортеж значень аргументів можна розглядати як запис цілого додатного числа у двійковій системі числення ($r = 2$); тоді x_0 — розряд одиниць, x_1 — розряд двійок, x_2 — розряд четвірок. Наприклад, шостим набором у табл. 3.1 є $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$. Перший набір називається нульовим, а останній — одиничним.

Десятковий еквівалент двійкового подання числа називається номером кортежу.

3.1.3. Булеві функції однієї змінної

Загальна таблиця відповідності для булевих функцій однієї змінної має вигляд табл. 3.2.

Таблиця 3.2

x	$y_1 = f_1(x)$	$y_2 = f_2(x)$	$y_3 = f_3(x)$	$y_4 = f_4(x)$
0	1	0	1	0
1	1	0	0	1

Тут функції y_1 та y_2 є функціями-константами: $f_1(x)$ — абсолютно істинна (константа одиниці); $f_2(x)$ — абсолютно хибна (константа нуля), $f_3(x)$ — логічне заперечення, або НЕ, інверсія x (читається як «не x», зображається як $\overline{x}$), це єдина нетривіальна функція; $f_4(x)$ — змінна x (повторює значення змінної x і тому збігається з нею).

3.1.4. Область визначення булевої функції

Областю визначення булевої (перемикальної) функції n аргументів є сукупність 2^n булевих кортежів. Це положення очевидне з погляду інтерпретації набору двійковим поданням n-розрядного числа (кількість додатних n-розрядних двійкових чисел дорівнює 2^n).

Булева функція двох аргументів є повністю визначеною, якщо задано її значення в кожному з чотирьох можливих наборів $\left(2^2 = 4\right)$; булева функція трьох аргументів у восьми $\left(2^3 = 8\right)$ наборах також буде повністю визначеною.

Булева функція n аргументів є повністю визначеною, якщо задано всі її значення в кожному з 2^n наборів.

Теорема 3.1. *Число $f_2(n)$ усіх функцій, що залежать від n змінних x_1, $x_2, ..., x_n$, дорівнює 2^{2^n}.*

Дійсно, перемикальну функцію n аргументів визначено у 2^n наборах, у яких вона може набувати значення 0 або 1 із загальної кількості значень 2^n. Тому у відповідність кожній перемикальній функції можна поставити 2^n-розрядне двійкове число. Проте кількість різних 2^n-розрядних чисел дорівнює 2^{2^n}, а отже, кількість різних перемикальних функцій становить 2^{2^n}.

Від двох аргументів залежать 16 булевих функцій, від трьох — 256, від чотирьох — 65 500.

Функції двох змінних відіграють важливу роль, тому що з них може бути побудована будь-яка перемикальна функція.

3.1.5. Елементарні функції алгебри логіки

У математичній логіці часто вживаються елементарні функції, які відіграють таку саму важливу роль, як, наприклад, x^n або sin x у математичному аналізі.

Приклади елементарних функцій однієї змінної наведено в табл. 3.2.

У табл. 3.3. подано 16 функцій двох змінних, з яких шість $f_{16}(a,b)=0$, $f_{11}(a,b)=a$, $f_{13}(a,b)=b$, $f_{12}(a,b)=\overline{a}$, $f_{14}(a,b)=\overline{b}$, $f_{15}(a,b)=1$ є константами або функціями одного аргументу. Інші 10 функцій залежать від двох аргументів і мають свої загальноприйняті позначення та назви, зазначені в табл. 3.3. Розглянути всі функції, що залежать від трьох аргументів, важко, оскільки їх число становить $2^{2^3}=256$.

Функція $f_1(a,b)$ — кон'юнкція (логічне множення) істинна тоді, коли a і b — істинні. Кон'юнкцію називають також функцією І; умовно її позначають $f_1(a,b)=a*b=a\&b=a\wedge b$.

Функція $f_2(a,b)$ — диз'юнкція (логічне додавання) істинна тоді, коли істинними є або a, або b, або обидві змінні.

Диз'юнкцію часто називають також функцією АБО й умовно позначають так: $f_2(a,b)=a+b=a\vee b$.

Від диз'юнкції потрібно відрізняти функцію $f_6(a,b)$, яка називається функцією додавання за модулем 2 (функцією нерівнозначності або різноіменності) і є істинною, коли істинні або a, або b окремо. Умовне позначення цієї функції $f_6(a,b)=a\oplus b=a\overline{\sim}b$.

Приклад. Маємо два висловлення: «Завтра буде холодна погода», «Завтра піде сніг». Диз'юнкція цих висловлень — нове висловлення «Завтра буде холодна погода або піде сніг». З'єднувальний сполучник, що утворив нове висловлення, — АБО.

Кон'юнкція утвориться таким чином: «Завтра буде холодна погода і піде сніг». Це висловлення утворено за допомогою сполучника І.

Функція Шеффера (штрих Шеффера) — це функція $f_7(a,b)$, яка є хибною тільки тоді, коли a та b є істинними. Умовне позначення цієї функції $f_7(a,b)=a/b=a\wedge b$.

Німецький математик Д. Шеффер на основі цієї функції створив алгебру, названу алгеброю Шеффера. Функція $f_7(a, b)$ є універсальною.

Функція стрілка Пірса —Вебба — це функція $f_8(a, b)$, що є істинною тільки тоді, коли *a* і *b* є хибними. Умовне позначення цієї функції $f_8(a, b) = a \downarrow b = a \bar{\vee} b$.

Математики Ч. Пірс та Д. Вебб, які незалежно один від одного вивчали властивості цієї функції, створили алгебру, названу алгеброю Пірса — Вебба. Функція $f_8(a, b)$ також є універсальною.

Імплікація — це функція $f_3(a, b)$, яка є хибною тоді й тільки тоді, коли *a* є істинним, а *b* — хибним. Умовне позначення цієї функції $f_3(a, b) = a \rightarrow b$.

Таблиця 3.3

Позначення функції	Найменування функції	*a*				Примітка	Зберігає 0	Зберігає 1	Самодвоїста	Монотонна	Лінійна
		0	0	1	1						
		b									
		0	1	0	1						
$f_1 = a \wedge b = ab = a \,\&\, b$	Кон'юнкція (логічне множення)	0	0	0	1		×	×		×	
$f_2 = a \vee b = a + b$	Диз'юнкція (логічне додавання)	0	1	1	1		×	×		×	
$f_3 = a \rightarrow b$	Імплікація (від *a* до *b*)	1	1	0	1			×			
$f_4 = a \leftarrow b$	Обернена імплікація (від *b* до *a*)	1	0	1	1			×			
$f_5 = a \sim b$	Рівносильність	1	0	0	1			×			×
$f_6 = a \simeq b = a \oplus b$	Нерівносильність (сума за модулем 2)	0	1	1	0		×				×
$f_7 = a \bar{\wedge} b = a / b$	Функція Шеффера (інверсія кон'юнкції)	1	1	1	0	Універсальна					
$f_8 = a \bar{\vee} b = a \downarrow b$	Функція стрілка Пірса—Вебба (інверсія диз'юнкції)	1	0	0	0	Те саме					
$f_9 = a \overline{\rightarrow} b$	Інверсія імплікації (функція заборони за *b*)	0	0	1	0		×				
$f_{10} = a \overline{\leftarrow} b$	Інверсія, обернена імплікації (функція заборони за *a*)	0	1	0	0		×				
$f_{11} = a$	Повторення *a* (змінна *a*)	0	0	1	1	Тривіальна	×	×	×	×	×
$f_{12} = \bar{a}$	Інверсія *a* (функція НЕ *a*)	1	1	0	0				×		×
$f_{13} = b$	Повторення *b* (змінна *b*)	0	1	0	1	Те саме	×	×	×	×	×
$f_{14} = \bar{b}$	Інверсія *b* (функція НЕ *b*)	1	0	1	0				×		×
$f_{15} = 1$	Одинична функція (константа 1)	1	1	1	1	Тривіальна		×		×	×
$f_{16} = 0$	Нульова функція (константа 0)	0	0	0	0	Те саме	×			×	×

3.1.6. Булевий простір

Часто для спрощення запису перемикальної функції замість повного переліку термів використовують двійкові значення наборів, для яких функція набуває одиничних значень. Наприклад, запис

$$f(x_1, x_2, x_3) = \overset{3}{\underset{1}{\vee}} F(1, 4, 7)$$

означає, що функція набуває одиничних значень на наборах 1, 4 і 7. Таку форму запису називають числовою (табл.3.4).

Таблиця 3.4

x_3	x_2	x_1	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Булевим простором може бути названа множина всіх наборів булевих векторів: $M = \{X\}$. Поставимо у взаємно однозначну відповідність елементам $x_1, x_2, ..., x_n$ множини X двійкові змінні, що позначаються тими самими літерами $x_1, x_2, ..., x_n$, але набувають значень із множини $\{0,1\}$ $(0 \vee 1)$, а у відповідність елементам булевого простору M поставимо набори (кортежі) змінних, вважаючи, що змінна x_i $(i \in \{1, 2, ..., n\})$ набуває значення 1 в деякому кортежі, якщо елемент x_i множини X належить відповідному простору M і набуває значення 0 в іншому випадку.

Таким чином, упорядковану сукупність двійкових змінних $x_1, x_2, ..., x_n$ можна розглядати як деякий змінний вектор $X = (x_1, x_2, ..., x_n)$, що набуває значення з множини M усіх сталих n-компонентних булевих векторів. Сукупність значень вектора X, на яких булева функція набуває значення 1, позначимо через M^1, а сукупність значень, на яких функція перетворюється на 0, — через M^0. Очевидно, $M^1 \cup M^0 = M$ (для повністю визначеної булевої функції), тобто

$$f_1(x_1, x_2, ..., x_n) \in M^1, \ f_0(x_1, x_2, ..., x_n) \in M^0.$$

Безпосередній перелік цих елементів можна здійснити за допомогою булевої матриці, кожний рядок якої задає один з елементів множини M^1. Наприклад, функція $f_1(x_1, x_2, x_3, x_4) \overset{3}{\underset{1}{\vee}} F(3,\ 10,\ 6)$ набуває значення 1 на трьох кортежах. Булева матриця має вигляд

$$\left\| M^1 \in X \right\| = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{vmatrix}$$

– кортежі, на яких функція набуває значення 1.

3.2. Властивості функцій алгебри логіки

Означення 3.1. *Функція* $f(x_1, x_2, ..., x_n)$ *залежить від аргументу* x_i, *якщо існують такі значення* $\lambda_1, \lambda_2, ..., \lambda_n$ *змінних* $x_{i_1}, x_{i_2}, ..., x_{i_n}$, *що* $f(\lambda_1, \lambda_2, \lambda_{i-1}, 0, \lambda_{i+1}, ..., \lambda_n) \neq f(\lambda_1, \lambda_2, \lambda_{i-1}, 1, \lambda_{i+1}, ..., \lambda_n)$.

У цьому випадку змінна x_i називається суттєвою. Якщо x_i не є суттєвою змінною, то вона називається несуттєвою, або фіктивною. Отже, змінна x_i є фіктивною, якщо значення функції не змінюються при зміні x_i.

Нехай для функції $f_1(x_1, x_2, ..., x_n)$ змінна x_i є фіктивною. Розглянемо таблицю функцій (табл. 3.5) і за нею побудуємо нову таблицю, викреслюючи рядки вигляду $(\lambda_1, \lambda_2, \lambda_{i-1}, 1, \lambda_{i+1}, ..., \lambda_n)$ та стовпці аргументу x_i.

Таблиця 3.5

x_1	$\cdots$	x_i	$\cdots$	x_n	$f(x_1, ..., x_i, ..., x_n)$
x_1	$\cdots$	x_i	$\cdots$	x_n	F_1
x_1	$\cdots$	x_i	$\cdots$	x_n	F_2
x_1	$\cdots$	x_i	$\cdots$	x_n	$\cdots$
x_1	$\cdots$	x_i	$\cdots$	x_n	F_n

Одержана таблиця визначатиме деяку функцію $g(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)$. Будемо говорити, що функцію $g(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)$ здобуто з функції $f(x_1, x_2, ..., x_i, ..., x_n)$ вилученням фіктивної змінної x_i, а функцію $f(x_1, x_2, ..., x_i, ..., x_n)$ — із функції $g(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)$ введенням фіктивної змінної x_i.

Означення 3.2. *Функції* f_1 *та* f_2 *називаються рівними, якщо функцію* f_2 *можна одержати з* f_1 *додаванням і вилученням фіктивних аргументів.*

Можна вважати, що коли задано функцію f_1, то задано також функцію f_2.

Існують два типи функцій, які не мають суттєвих змінних:

- функція, тотожна 0 (константа 0);
- функція, тотожна 1 (константа 1).

Константи 1 і 0 можна розглядати як функції порожньої множини змінних.

Означення 3.3. *Булева функція* $f(x_1, \ldots, x_n)$ *називається симетричною відносно змінних* $x_1, x_2, \ldots, x_k$, *якщо для будь-якої підстановки*

$$\begin{pmatrix} 1 & 2 & \dots & k \\ & & \dots & \\ x_1 & x_2 & \dots & x_k \end{pmatrix}$$

стверджується рівність:

$$f(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) = f(x_{j_1}, \ldots, x_{j_k}, x_{j_{k+1}}, \ldots, x_{j_n}).$$

Функції, тотожно рівні константам 1 та 0, є симетричними відносно будь-якої сукупності змінних.

Зауваження. Якщо задано скінченну систему функцій $\{f_1,\dots,f_s\}$ при $s \geq 1$, то можна вважати, що всі ці функції залежать від одних і тих самих змінних $x_1,\dots, x_n$, тобто мають вигляд

$$\{f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n), \dots, f_s(x_1, \dots, x_n)\}.$$

3.3. Реалізація булевих функцій формулами

3.3.1. Поняття формули в алгебрі логіки

Як і в елементарній алгебрі, в алгебрі логіки, виходячи з елементарних функцій, можна будувати формули. Назвемо P множину всіх функцій.

Означення 3.4, а. *Нехай L — деяка (не обов'язково скінченна) підмножина функцій з P, $L \subset P$. Кожна функція $f(x_1, x_2, \dots, x_m)$ із $L\,(f \in L)$ називається формулою.*

Означення 3.4, б. *Нехай $f_0(x_1, x_2, \dots, x_m)$ — функція з $L\,(f_0 \in L)$ й A_1, $A_2, \dots, A_m$ — вирази, що є або формулами, або символами змінних із $U\,(A_i \in U)$. Тоді вираз $f_0(A_1, A_2, \dots, A_m)$ також називається формулою.*

Означення 3.5. *Усяке висловлення, що є складеним із деяких початкових висловлень за допомогою* 14 *логічних операцій з* 16, *крім* 0 *та* 1, *також називається формулою алгебри логіки.*

При утворенні (побудові) формул використовуються знаки (символи) трьох категорій:

- символи змінних $x, y, a, b, c, \dots$;
- символи логічних операцій $(\wedge), (\vee), (\oplus), (\rightarrow)$ і т. д.;
- пари символів ( ), [ ], { } — дужки.

Приклади формул. Нехай L — множина елементарних функцій. Такі вирази є формулами:

- $\{[(x_1x_2)+x_1]+x_2\}$;
- $[\overline{x}_1(x_1+x_3)]$;
- $\overline{\{x_1 \vee [(x_2 \rightarrow x_3)(x_3 \rightarrow x_2)]\}}$,

а ці вирази не є формулами:

- $(AB(xy$ — незакриті дужки;
- $\wedge B \wedge y$ — відсутній символ;
- $A \rightarrow B\, x \rightarrow y$ — операція імплікації обов'язково має бути взята в дужки.

На практиці дужки розділяють на внутрішні та зовнішні. Формула $F = A \wedge B$ без дужок не є формулою. Проте для скорочення запису зовнішні дужки часто пропускають, і тому цей вираз означає формулу.

3.3.2. Реалізація функцій формулами

Означення 3.6. *Нехай F — довільна формула. Тоді формули, що використовувалися для її побудови, називаються підформулами формули F.*

Нехай формула F є формулою для множини функцій $\{f_1(x_1, \dots, x_n)$, $f_2(x_1, \dots, x_n), \dots, f_s(x_1, \dots, x_n)\}$. Розглянемо множину функцій $\{g_1(x_1, \dots, x_n)$,

$g_2(x_1, ..., x_n), ..., g_s(x_1, ..., x_n)\}$, де функція g_i має ті самі змінні, тобто залежить від тих самих змінних, що і функція f_i $(i = 1, ..., s)$.

Розглянемо формулу $Fg = F(g_1, ..., g_s)$, яка випливає з F заміною $(f_1, ..., f_s)$ на $(g_1, ..., g_s)$. У цьому випадку формула Fg має ту саму структуру, що й формула F.

Означення 3.7. *Якщо формула* $F(x_1, x_2, ..., x_n)$ *описує функцію* $f(x_1, x_2, ..., x_n)$, *тобто формула F є формулою для змінних* $x_1, x_2, ..., x_n$, *де* $f \in L$, *то кажуть, що формулі* $F(x_1, x_2, ..., x_n)$ *відповідає функція* $f(x_1, x_2, ..., x_n)$, *або формулі F зіставлена функція f.* Якщо функція f відповідає формулі F, то кажуть також, що формула F реалізує функцію f. Оскільки функції розглядаються з точністю до фіктивних змінних, вважаємо, що формула F реалізує будь-яку функцію f.

Якщо функція $f(x_1, x_2, ..., x_n)$, що реалізується формулою $f(x_1, x_2, ..., x_n)$, має несуттєву змінну x_i, то змінну x_i можна вилучити, замінивши функцію f рівною їй функцією f', а формулу F — формулою F', яка випливає з F завдяки ототожненню змінної x_i з будь-якою із змінних, що залишилися. Очевидно, формула F' є формулою, яка реалізує функцію f'.

3.3.3. Принцип суперпозиції

Функцію f, що відповідає формулі F, називають суперпозицією функцій з множини функцій, а процес здобуття функції з множини функцій — операцією суперпозиції.

Приклад. Функція $F_1 = (((x_1 x_2) + x_1) + x_2)$ будується за три кроки (табл. 3.6):

- $(x_1 x_2)$;
- $((x_1 x_2) + x_1)$;
- $(((x_1 x_2) + x_1) + x_2) = F_1$.

Таблиця 3.6

x_1	x_2	$x_1 x_2$	$(x_1 x_2) + x_1$	$F_1 = ((x_1 x_2) + x_1) + x_2$
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

Функція $F_2 = \overline{x}_1(x_2 + x_3)$ будується також за три кроки (табл. 3.7):

- $(x_2 + x_3)$;
- $(\overline{x}_1)$;
- $((x_2 + x_3)\overline{x}_1) = F_2$.

Таблиця 3.7

x_1	x_2	x_3	x_2+x_3	$\bar{x}_1$	$F_2=\bar{x}_1(x_2+x_3)$
0	0	0	0	1	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

При складанні логічного висловлення із простих використовується принцип суперпозиції, тобто підстановка у функцію замість її аргументу інших функцій. Замість будь-якої змінної використовується як власне незалежна змінна, аргумент, так і змінна, що є функцією інших змінних. Цей принцип є правильним також у звичайній алгебрі. За допомогою принципу суперпозиції з двомісних булевих функцій можна побудувати будь-яку булеву функцію.

Принцип суперпозиції дає змогу на основі трьох основних елементарних функцій (заперечення, кон'юнкція, диз'юнкція) здобути складене логічне висловлення, що описує функціонування цифрових систем й автоматів. Покажемо це стосовно операцій кон'юнкції, диз'юнкції та інверсії.

Якщо F_1 й F_2 — формули, то $\bar{F}_1$, $(F_1 \& F_2), (F_1 \vee F_2), (F_1 \to F_2)$, $(F_1 \sim F_2)$ — також формули.

При перетворенні формул використовуються такі операції підстановки змінних і безповторної підстановки функцій:

- операція підстановки змінних

$$\begin{pmatrix} x_1 & \dots & x_n \\ x_{i_1} & \dots & x_{i_n} \end{pmatrix},$$

що дає змогу виконати підстановку змінних у функцію $f\left(x_1, ..., x_{i_n}\right)$ та здобути в результаті функцію $f\left(x_{i_1}, ..., x_{i_n}\right)$. Очевидно, підстановка змінних включає їх перейменування, перестановку й ототожнення;

- операція безповторної підстановки функцій дає можливість будувати вирази $f(A_1, ..., A_m)$, де A_i — або формула, або змінна з U, $A_i \in U$, причому хоча б одне з A відмінне від змінної, а множини змінних, що входять в A_i й A_j, не перетинаються $(i \neq j)$.

Очевидно, кожна формула може бути здобута з функцій, що належать їх множині, застосуванням спочатку операції безповторної підстановки функцій, а потім операції підстановки змінних. Уведена мова формул зручна для запису функцій алгебри логіки, які описують різні умови для висловлень.

3.3.4. Рівносильність формул

Означення 3.8. *Формули F_1 та F_2 називаються рівносильними, якщо при будь-яких значеннях змінних $x_1, x_2, ..., x_n$, що входять у ці формули, вони набувають однакових значень.*

Приклади:

- $\overline{\overline{x}}$ рівносильне x;
- $x \vee x$ рівносильне x;
- $(x \& y) \vee y$ рівносильне y;
- $x \vee y$ рівносильне $y \vee x$.

Між поняттями рівносильності й еквівалентності існує зв'язок: якщо формули F_1 та F_2 — рівносильні, то формула $(F_1 \sim F_2)$ (еквівалентність) набуває значення істини при всіх значеннях змінних, і навпаки: якщо формула $(F_1 \sim F_2)$ набуває значення істини при всіх значеннях змінних, то формули F_1 й F_2 — рівносильні.

При визначенні рівносильності двох формул не обов'язково передбачати, що вони містять одні й ті самі значення змінних.

Приклади важливих рівносильних формул:

- $\overline{\overline{x}}$ рівносильне x;
- xy рівносильне yx;
- $(xy)z$ рівносильне $x(yz)$;
- $x + y$ рівносильне $y + x$;
- $(x + y) + z$ рівносильне $x + (y + z)$;
- $x + x$ рівносильне x;
- xx рівносильне x;
- $x + (xy)$ рівносильне x.

3.4. Основні тотожності

Кожній формулі відповідає функція алгебри логіки, а різним формулам — різні функції.

Означення 3.9. *Формули F_1 й F_2 називаються тотожними, або еквівалентними, якщо відповідні функції f_1 та f_2 — рівні, тобто $f_1 = f_2$. Тоді запис $F_1 = F_2$ означає, що F_1 й F_2 — тотожні формули.*

Приклад. $x\overline{x} = 0$, $x \to y = \overline{y} \to \overline{x}$ і т. д.

Означення 3.10. *Дві формули F_1 й F_2 називаються еквівалентними, якщо вони задають одну і ту саму булеву функцію всіх змінних, що входять у ці формули.*

Із табл. 3.3 випливає, що такі елементарні функції, як заперечення, диз'юнкція, кон'юнкція, Шеффера, стрілка Пірса–Вебба, імплікація і т. д., знаходяться в певному зв'язку одна з одною. Розглянемо ці зв'язки та властивості зазначених функцій.

Кон'юнкція, диз'юнкція, заперечення (функції І, АБО, НЕ). Використовуючи основні положення алгебри логіки, неважко переконатися у правильності таких восьми аксіом. Нехай x — деяка логічна змінна. Тоді:

1. $\overline{\overline{x}} = x$, що означає можливість вилучення з логічного виразу всіх членів, які мають подвійне заперечення, замінивши їх початковою величиною;
2. $x + x = x$; $xx = x$ — правила подібних перетворень, що дають змогу скорочувати довжину логічних виразів;
3. $x + 0 = x$;
4. $x + 1 = 1$;
5. $x \cdot 0 = 0$;
6. $x \cdot 1 = x$;
7. $x\overline{x} = 0$;
8. $x + \overline{x} = 1$.

Диз'юнкція та кон'юнкція мають властивості, аналогічні властивостям арифметичних операцій додавання і множення:

- властивість асоціативності (сполучний закон)

$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3, \quad x_1(x_2x_3) = (x_1x_2)x_3;$$

- властивість комутативності (переставний закон)

$$x_1 + x_2 = x_2 + x_1, \quad x_1x_2 = x_2x_1;$$

- властивість дистрибутивності (розподільний закон):

для кон'юнкції відносно диз'юнкції

$$x_1 \& (x_2 + x_3) = (x_1 \& x_2) + (x_1 \& x_3);$$

для диз'юнкції відносно кон'юнкції

$$x_1 + x_2x_3 = (x_1 + x_2) \& (x_1 + x_3).$$

Властивість дистрибутивності фактично означає правила розкриття дужок і взяття в них логічних виразів.

Правильність наведених властивостей легко доводиться за допомогою викладених вище аксіом.

Доведемо, наприклад, що

$$x_1 + x_2x_3 = (x_1 + x_2) \& (x_1 + x_3).$$

Справді,

$$(x_1 + x_2)(x_1 + x_3) = x_1x_1 + x_1x_3 + x_1x_2 + x_2x_3 = x_1 + x_1x_2 + x_1x_3 + x_2x_3 =$$
$$= x_1(1 + x_2 + x_3) + x_2x_3 = x_1 + x_2x_3.$$

Аналогічно можна довести також інші закони.

Нескладно встановити правильність співвідношень, відомих як *закони де Моргана:*

$$\overline{x_1x_2} = \overline{x}_1 + \overline{x}_2;$$
$$\overline{x_1 + x_2} = \overline{x}_1\overline{x}_2.$$

Із законів де Моргана маємо

$$x_1x_2 = \overline{\bar{x}_1 + \bar{x}_2};$$
$$x_1 + x_2 = \overline{\bar{x}_1\bar{x}_2}.$$

За допомогою цих співвідношень можна виражати кон'юнкцію через диз'юнкцію та заперечення або диз'юнкцію через кон'юнкцію і заперечення.

Закони де Моргана та наслідки їх є дійсними для будь-якої кількості змінних:

$$\overline{x_1 + x_2 + \ldots + x_n} = \bar{x}_1 \& \ldots \& \bar{x}_n;$$
$$\overline{x_1x_2 \ldots x_n} = \bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_n.$$

Для логічних функцій встановлюються співвідношення, відомі як *закони поглинання:*

$$x_1 + x_1x_2 = x_1;$$
$$x_1(x_1 + x_2) = x_1.$$

У табл. 3.8 показано правильність законів поглинання.

Таблиця 3.8

x_1	x_2	$x_1 + x_2$	x_1x_2	$x_1 + x_1x_2$	$x_1 + (x_1 + x_2)$
0	0	0	0	0	0
0	1	1	0	0	0
1	0	1	0	1	1
1	1	1	1	1	1

Функція додавання за модулем 2 — це функція

$$x_1 \oplus x_2 = x_1\bar{x}_2 + \bar{x}_1x_2 = (x_1 + x_2)(\bar{x}_1 + \bar{x}_2).$$

Вона має такі властивості:

- комутативність (сполучний закон)

$$x_1 \oplus x_2 = x_2 \oplus x_1;$$

- асоціативність (переставний закон)

$$x_1 \oplus (x_2 \oplus x_3) = (x_1 \oplus x_2) \oplus x_3;$$

- дистрибутивність (розподільний закон)

$$x_1(x_2 \oplus x_3) = (x_1x_2) \oplus (x_1x_3).$$

Для цієї функції справджуються такі аксіоми:

$$x \oplus x = 0; \qquad x \oplus 1 = \bar{x};$$
$$x \oplus \bar{x} = 1; \qquad x \oplus 0 = x.$$

На основі аксіом і властивостей можна вивести правила переведення функцій I, АБО, НЕ через функцію додавання за модулем 2 і навпаки:

$$\bar{x}_1 = x_1 \oplus 1;$$
$$x_1 + x_2 = x_1 \oplus x_2 \oplus x_1 x_2;$$
$$x_1 x_2 = (x_1 \oplus x_2) \oplus (x_1 \oplus x_2).$$

Функція імплікації (→) — це функція, що виражається як $x_1 \to x_2 = \bar{x}_1 + x_2$.

Для цієї функції справджуються такі аксіоми:

$$x \to x = 1; \qquad x \to 1 = 1; \qquad 0 \to x = 1;$$
$$x \to \bar{x} = \bar{x}; \qquad x \to 0 = \bar{x}; \qquad 1 \to \bar{x} = x.$$

Звідси випливає, що імплікація має тільки властивість комутативності (переставний закон) у зміненому вигляді $x_1 \to x_2 = \bar{x}_2 \to \bar{x}_1$.

Властивість асоціативності для цієї функції не справджується (табл. 3.9).

Таблиця 3.9

x_1	x_2	x_3	$x_1 \to (x_2 \to x_3)$	$(x_1 \to x_2) \to x_3$
0	0	0	1	0
0	1	0	1	0
1	0	0	1	1
1	1	0	0	0
0	0	1	1	1
0	1	1	1	1
1	0	1	1	1
1	1	1	1	1

Функції I, АБО, НЕ через імплікацію виражаються так:

$$x_1 + x_2 = \bar{x}_1 \to x_2;$$
$$x_1 x_2 = \overline{x_1 \to \bar{x}_2};$$
$$\bar{x}_1 = x_1 \to 0.$$

Функція Шеффера (/) — це функція, що може бути виражена співвідношенням $x_1 / x_2 = \overline{x_1 x_2}$.

Для цієї функції справджуються такі аксіоми:

$$x/x = \bar{x}; \qquad x/\bar{x} = 1; \qquad x/0 = 1;$$
$$x/1 = \bar{x}; \qquad \bar{x}/0 = 1; \qquad \bar{x}/1 = x.$$

Для функції Шеффера справджується тільки властивість комутативності

$$x_1/x_2 = x_2/x_1,$$

однак для трьох і більше змінних

$$x_1/(x_2/x_3) \neq (x_1/x_2)/x_3.$$

Для перевірки виконання властивості асоціативності функції Шеффера розглянемо функцію трьох змінних.

Застосовуючи правила перетворення до правої частини останнього виразу, знаходимо $(x_1/x_2)/x_3 = \overline{\overline{(x_1x_2)}x_3} = x_1x_2 + \overline{x}_3$.

Відповідно для лівої частини виразу маємо $(x_1/x_3)/x_2 = \overline{\overline{(x_1x_3)}x_2} = x_1x_3 + \overline{x}_2$.

Отже, властивість асоціативності не справджується для функції трьох змінних, а це означає, що вона не поширюється також на функцію змінних.

При розгляді властивості дистрибутивності функції Шеффера матимемо на увазі, що:

$$x_1(x_2 + x_3) = (x_1x_2) + (x_1x_3);$$

$$x_1 + (x_2x_3) = (x_1 + x_2)(x_1 + x_3).$$

Наведемо два приклади перетворення цих виразів.

Перетворимо вираз $x_1/(x_2/x_3)$

$$\overline{x_1\overline{(x_2x_3)}} = \overline{x}_1 + x_2x_3 = (\overline{\overline{\overline{x}_1 + x_2}})(\overline{\overline{\overline{x}_1 + x_3}}) = x_1/x_2 \,\&\, x_1/\overline{x}_3 =$$

$$= \overline{\overline{x_1/(x_2/1) \,\&\, x_1/(x_3/1)}} = [x_1/(x_2/1)/(x_3/1)].$$

Перетворюємо вираз $(x_1/x_2)/(x_1/x_3)$:

$$\overline{\overline{(x_1x_2)}\,\overline{(\mathrm{x}_1\mathrm{x}_3)}} = x_1x_2 + x_1x_3 = x_1(\overline{\overline{x_2 + x_3}}) = x_1(\overline{\overline{x}_2\overline{x}_3}) =$$

$$= \overline{\overline{x_1[(x_2/1)/(x_3/1)]}} = \overline{x_1/[(x_2/1)/(x_3/1)]} = [x_1/(x_2/1)/x_1/(x_3/1)]/1.$$

На підставі цих прикладів можна зробити висновок, що властивість дистрибутивності не справджується для функції трьох змінних, а отже, і для функції *n* змінних.

З основних властивостей функції Шеффера можна дістати такі формули перетворення:

$$\begin{cases} x_1x_2 = \overline{x_1/x_2} = x_1/x_2/x_1/x_2; \\ \overline{x} = x/x; \\ x_1 + x_2 = \overline{\overline{x}_1\overline{x}_2} = \overline{x}_1/\overline{x}_2 = x_1/x_1/x_2/x_2. \end{cases}$$

Функція стрілка Пірса — Вебба $(\downarrow)$ — це функція, що описується виразом $x_1 \downarrow x_2 = \overline{x_1 + x_2} = \overline{x}_1\overline{x}_2$.

Для цієї функції легко показати справедливість таких аксіом:

$$x \downarrow x = \overline{x};$$

$$x \downarrow 0 = x;$$

$$x \downarrow \overline{x} = 0;$$

$$x \downarrow 1 = 0\ .$$

На підставі цих аксіом можна встановити, що для функції стрілка Пірса — Вебба справджується тільки властивість комутативності: $x_1 \downarrow x_2 = x_2 \downarrow x_1$.

Функції І, АБО, НЕ виражаються через функцію стрілка Пірса — Вебба так:

$$\begin{cases} x_1x_2 = (x_1 \downarrow x_1) \downarrow (x_2 \downarrow x_2); \\ x + x_2 = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2); \\ \overline{x} = x \downarrow x. \end{cases}$$

Основні тотожності для функцій наведено в табл. 3.10.

Таблиця 3.10

№ пор.	Тотожність	Найменування законів	Примітка
1	а) $x_1 \wedge x_2 = x_2 \wedge x_1$ або $x_1x_2 = x_2x_1$ б) $x_1 \vee x_2 = x_2 \vee x_1$ або $x_1 + x_2 = x_2 + x_1$	Переставні (комутативні) закони	Дистрибутивні закони (б) не мають аналога у звичайній алгебрі
2	а) $(x_1 + x_2)x_3 = x_1x_3 + x_2x_3$ б) $x_1 + x_2x_3 = (x_1 + x_2)(x_1 + x_3)$	Розподільні закони: а) кон'юнкція відносно диз'юнкції б) диз'юнкція відносно кон'юнкції	
3	а) $(x_1x_2)x_3 = x_1(x_2x_3)$ б) $(x_1 + x_2) + x_3 = x_1 + (x_2 + x_3)$	Сполучні закони (асоціативні)	
4	а) $xx = x$ б) $x + x = x$	Закони повторення (тавтології)	Закони заперечення
5	а) $x_1(x_1 + x_2) = x_1$ б) $x_1 + x_1x_2 = x_1$	Закони поглинання (абсорбції)	
6	$x\overline{x} = 0;\ x + \overline{x} = 1$	Закон доповнення	
7	а) $\overline{x_1x_2} = \overline{x}_1 + \overline{x}_2$ б) $\overline{x_1 + x_2} = \overline{x}_1\overline{x}_2$	Правила де Моргана	
8	$\overline{\overline{x}} = x$	Закон подвійного заперечення	
9	а) $x_1x_2 + x_1\overline{x}_2 = x_1$ б) $(x_1 + \overline{x}_2)(x_1 + x_2) = x_1$	Закони склеювання	
10	а) $x \cdot 1 = x$ б) $x + 1 = 1$	Закони універсальної множини	
11	а) $x \cdot 0 = 0$ б) $x + 0 = x$	Закони нульової множини	

Сформулюємо такі правила.

1. Якщо в логічному добутку один із співмножників дорівнює нулю, то логічний добуток також дорівнює нулю.

2. Якщо в логічному добутку, що містить не менше двох співмножників, є співмножник, який дорівнює одиниці, то цей співмножник можна вилучити.

3. Якщо в логічній сумі, що містить не менше двох доданків, є доданок, який дорівнює нулю, то цей доданок можна вилучити.

4. Якщо в логічній сумі один із доданків дорівнює одиниці, то ця сума дорівнює одиниці.

Очевидно, якщо F' — підформула формули F, то коли замінити будь-яке з її входжень на еквівалентну формулу B', формула F перейде у формулу B, еквівалентну формулі F.

Цей принцип разом із тотожностями для елементарних функцій, до яких приєднуються всі тотожності, що випливають після підстановки замість змінних $x_1, x_2, ..., x_n$ будь-яких форм, дає змогу здійснювати еквівалентні перетворення і таким чином одержувати нову тотожність.

Приклади: 1. Довести тотожність 5 б) з табл. 3.10. Маємо

$$x_1 + x_1x_2 = x_1 \cdot 1 + x_1x_2 = x_1(x_2 + \bar{x}_2) + x_1x_2 = x_1x_2 + x_1\bar{x}_2 + x_1\mathrm{x}_2 =$$
$$= x_1x_2 + x_1\bar{x}_2 = x_1(x_2 + \bar{x}_2) = x_1 \cdot 1 = x_1, \quad \text{або} \quad x_1 + x_1x_2 = x_1(1 + x_2) = x_1 \cdot 1 = x_1.$$

Таким чином, $x_1 + x_1x_2 = x_1$, тобто дістаємо правило поглинання добутку x_1x_2 співмножником x_1.

2. Спростити вираз $y = xyz + \bar{x}yz + \overline{xyz} + \bar{x}y\bar{z}$. Маємо

$$y = xyz + \bar{x}yz + \overline{xyz} + \bar{x}y\bar{z} = xyz + \bar{x}yz + \overline{xyz} +$$
$$+ \bar{x}y\bar{z} + \bar{x}y\bar{z} = xyz + \bar{x}y(z + \bar{z}) + \overline{xz}(\bar{y} + y) = xyz + \bar{x}y + \overline{xz}.$$

3. Спростити вираз $F = x + xy\bar{z} + \bar{y} + x\bar{y}z + \bar{z}(\bar{z} + \overline{xy})$. Маємо

$$F = x + xy\bar{z} + \bar{y} + x\bar{y}z + \bar{z}(\bar{z} + \overline{xy}) = x(1 + y\bar{z}) + \bar{y}(1 +$$
$$+ xz) + \bar{z} + \overline{zxy} = x + \bar{y} + \bar{z}(1 + \overline{xy}) = x + \bar{y} + \bar{z}.$$

4. Довести, що $x(x + y) = x$. Маємо

$$xx + xy = x + xy = x(1 + y) = x \cdot 1 = x.$$

5. Довести, що $x + \bar{x}y = x + y$. Маємо

$$x + \bar{x}y = (x + \bar{x})(x + y) = 1(x + y) = x + y.$$

6. Довести, що $x(\bar{x} + y) = xy$.

$$x(\bar{x} + y) = x\bar{x} + xy = 0 + xy = xy.$$

7. Довести, що $(x + y)(\bar{x} + z) = xz + \bar{x}y$. Маємо

$$(x + y)(\bar{x} + z) = (x + y)\bar{x} + (x + y)z = x\bar{x} + y\bar{x} + xz + yz =$$
$$= 0 + xz + y\bar{x} + yz = xz + y\bar{x} + yz(x + \bar{x}) = xz + y\bar{x} +$$
$$+ yzx + yz\bar{x} = xz + xzy + \bar{x}y + \bar{x}yz = xz(1 + \mathrm{y}) + \bar{x}y(1 + z) = xz + \bar{x}y.$$

Уперше теорія Дж. Буля була застосована Еренфестом у 1916 р. до аналізу перемикальних схем. Використання логічних функцій виявилося особливо корисним для опису роботи логічних елементів ЕОМ, теорії алгоритмів.

3.5. Принцип двоїстості

Існує ще один спосіб здобуття тотожності. Він ґрунтується на використанні принципу двоїстості з теорії булевої алгебри.

Означення 3.11. *Функція* $[f(x_1, x_2, ..., x_n)]$*, *що дорівнює функції* $\overline{f}(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$, *називається двоїстою функцією до функції* $f(x_1, x_2, ..., x_n)$.

Таблиця для двоїстої функції (табл. 3.11) випливає з таблиці для функції $f(x_1, x_2, ..., x_n)$ інвертуванням (тобто заміною 0 на 1 та 1 на 0) значень функції й заміною порядку проходження інвертованих значень функції на протилежний (перевертанням стовпця інвертованих значень функції):

- функція 0 двоїста до 1;
- функція 1 двоїста до 0;
- функція x двоїста до $\overline{x}$;
- функція $\overline{x}$ двоїста до x;
- функція x_1x_2 двоїста до x_1+x_2 } — взаємно двоїсті функції:
- функція x_1+x_2 двоїста до x_1x_2 } кон'юнкція та диз'юнкція.

Таблиця 3.11

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	$[f(x_1, x_2, x_3)]^*$
0	0	0	1	1
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	0
1	0	1	0	0
1	1	0	0	1
1	1	1	0	0

З означення двоїстості функцій випливає, що

$$f^{**} = \left(f^*\right)^* = f.$$

Для запису формули f^*, двоїстої до формули f, треба у формулі f всюди 0 замінити на 1, 1 — на 0, знак & — на знак «+», а знак «+» — на &, тобто якщо

$$f = c\,[0, 1, \overline{x}, x_1 \wedge x_2, x_1 \vee x_2],$$

то

$$f^* = \overline{c}\ [1, 0, x, x_1 + x_2, x_1 \wedge x_2].$$

На підставі законів де Моргана виводиться таке положення: якщо $F(x_1, x_2, ..., x_n)$ й $F^*(x_1, x_2, ..., x_n)$ — двоїсті формули, то формула $\overline{F}^*(x_1, x_2, ..., x_n)$ є рівносильною формулі $F(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$.

Звідси випливає, що $F^*(x_1, x_2, ..., x_n) = \overline{F}(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$, тобто двоїста формула виражається як заперечення формули, знайденої з початкової

---

* — знак інвертування.

заміщенням кожної змінної її запереченням. Таблиця відповідності двоїстої функції формується заміною значень аргументів у початковій функції на протилежні, тобто 0 замінюється на 1, а 1 — на 0.

Формула або функція, рівносильна своїй двоїстій, називається самодвоїстою.

Якщо формули $F_1(x_1, x_2, ..., x_n)$ й $F_2(x_1, x_2, ..., x_n)$ є рівносильними, то й двоїсті до них формули $F_1^*(x_1, x_2, ..., x_n)$ та $F_2^*(x_1, x_2, ..., x_n)$ — також рівносильні.

Принцип двоїстості дає змогу майже вдвічі зменшити трудомісткість виведення при розгляді властивостей елементарних функцій.

3.6. Набори повних функцій

3.6.1. Розвинення булевих функцій за змінними

Уведемо позначення

$$x^{\sigma} = x\sigma + \overline{x\sigma}, \text{ де } \sigma = 0 \vee 1;$$

тоді $x^{\sigma} = \begin{cases} \bar{x}, & \text{при} \quad \sigma = 0; \\ x, & \text{при} \quad \sigma = 1. \end{cases}$

Якщо $x = \sigma$, то $x^{\sigma} = 1$.

Твердження 3.1. *Кожна функція алгебри логіки $f(x_1, x_2, ..., x_n)$ може бути подана в такій формі:*

$$f(x_1, x_2, ..., x_m, x_{m+1}, ..., x_n) = \bigvee_{\substack{\sigma_1, ..., \sigma_m \\ (1 \le m \le n)}} x_1^{\sigma_1}, ..., x_m^{\sigma_m} f(\sigma_1, ..., \sigma_m, x_{m+1}, ..., x_n),$$

де $\vee$ — символ узагальненої диз'юнкції.

Диз'юнкція береться за всіма можливими наборами значень змінних $x_1,\ x_2, ..., x_m$. Це подання називається розвиненням функції за m змінними: $x_1,\ x_2, ..., x_m$.

Теорема 3.2. *Кожна функція алгебри логіки може бути виражена у вигляді формули через заперечення, кон'юнкцію та диз'юнкцію.*

Доведення. 1. Нехай $f(x_1, x_2, ..., x_n) \equiv 0$. Тоді очевидно, що $f(x_1, x_2, ..., x_n) = x_i \bar{x}_i$.

2. Нехай $f(x_1, x_2, ..., x_n) \equiv 0$ й $\overline{f}(x_1, x_2, ..., x_n) = 1$. Тоді відповідно до твердження 3.1

$$f(\tilde{x}_1^{\sigma_1}, \tilde{x}_2^{\sigma_2}, ..., \tilde{x}_n^{\sigma_n}) = \bigvee_{\substack{(\sigma_1, \sigma_2, ..., \sigma_n) \\ f(\sigma_1, \sigma_2, ..., \sigma_n) = 1}} \tilde{x}_1^{\sigma_1} \tilde{x}_2^{\sigma_2}, ..., \tilde{x}_n^{\sigma_n},$$

тобто функція виражається у вигляді формули через заперечення, кон'юнкцію і диз'юнкцію.

3.6.2. Набори повних систем

Означення 3.12. *Система функцій* $(f_1, f_2, ..., f_s)$ *з* $P(f \in P)$ *називається функціонально повною, якщо будь-яка булева функція може бути записана у вигляді формули через функції цієї системи.*

Наведемо приклади повних систем.

1. Система P — множина всіх булевих функцій. Кількість функцій — 2^{2^n}. Так, усі 16 функцій двох змінних утворюють повну систему.

2. Система функцій $\{\overline{x}; (x_1 x_2); (x_1 + x_2)\}$. Виходячи з попередньої теореми й означення 3.12, випливає, що не кожна система є повною, наприклад, система $\{0, 1\}$. Звідси маємо таку теорему.

Теорема 3.3. *Нехай задано дві системи функцій:*

$$A = \{f_1, f_2, ..., f_m\}; \tag{3.1}$$

$$B = \{g_1, g_2, ..., g_s\}, \tag{3.2}$$

відносно яких відомо, що система (3.1) *повна і кожна її функція виражається у вигляді формули через функції системи* (3.2). *Тоді система* (3.2) *також є повною.*

Доведення. Нехай h — довільна система функцій, $h \in P$. З урахуванням повноти системи (3.1) h можна виразити як

$$h = C(f_1, f_2, ..., f_m).$$

За умовою $f_1 = C_1(g_1, g_2, ..., g_s)$;

$$f_2 = C_2(g_1, g_2, ..., g_s);$$

....

$$f_m = C_m(g_1, g_2, ..., g_s).$$

Тому у формулі $h = C(f_1, f_2, ..., f_m)$ можна вилучити f_i, тобто записати

$$C(f_1, f_2, ..., f_m) = C(C_1(g_1, g_2, ...), C_2(g_1, g_2, ...), ..., C_m(g_1, g_2, ..., g_s)).$$

Цей вираз визначає формулу (3.2) з будовою C':

$$C(C_1(g_1, g_2, ...), C_2(g_1, g_2, ...), ..., C_m(g_1, g_2, ..., g_s)) = C'(g_1, g_2, ..., g_s),$$

або остаточно

$$h = C'(g_1, g_2, ..., g_s).$$

Теорему доведено: B належить до повних систем.

Спираючись на цю теорему, можна встановити повноту ще кількох систем і таким чином розширити список повних систем.

3. Система функцій $\{\overline{x}, (x_1 x_2)\}$. Для доведення візьмемо за систему (3.1) систему з прикладу 2, а за систему (3.2) — систему, що розглядається. Використовуємо тотожність $x_1 + x_2 = \overline{\overline{x}_1 \overline{x}_2}$, що випливає з тотожності еле-

ментарних функцій, функцію (x_1+x_2) можна завжди виразити через логічний добуток x_1x_2. З цього випливає, що функцію (x_1+x_2) можна вилучити з переліку повних систем.

4. Система $\{\bar{x},(x_1+x_2)\}$, що доводиться або аналогічно попередньому, тобто $x_1x_2=\overline{\bar{x}_1+\bar{x}_2}$, або через принцип двоїстості.

5. Система $\{0, 1, x_1x_2, x_1+x_2\}$.

Доведення. За систему (3.1) візьмемо систему з прикладу 3, а за (3.2) — систему з прикладу, що розглядається. Маємо $x_1+1=\bar{x}_1$.

6. Функція Шеффера $\overline{x_1x_2}$.

7. Функція стрілка Пірса — Вебба $\overline{x_1+x_2}$.

Доведення двох останніх прикладів читачеві пропонуємо виконати самостійно.

Означення 3.13. *Система функцій $f(x_1,x_2,...,x_n)\in P$, суперпозицією яких може бути подана будь-яка функція з деякої множини булевих функцій, називається функціонально повною. Якщо в такій системі допускаються константи* 0 *і* 1, *то її називають ослаблено функціонально повною. Кажуть, що функціонально повна система функцій утворює базис у логічному просторі. Система функцій називається мінімально повним базисом, якщо вилучення з неї будь-якої функції перетворює цю систему на неповну.*

3.6.3. Теорема Жегалкіна

Теорема 3.4. *Будь-яка перемикальна функція $f\in P$ (кожна функція з P) може бути зображена за допомогою полінома за* mod 2 *(полінома Жегалкіна).*

Поліном Жегалкіна має вигляд

$$f(x_1,x_2,\ldots,x_n)=\sum_{i_1,\ldots,i_s} a_{i_1},\ldots,a_{i_s}x_{i_1},\ldots,x_{i_s}$$

і є канонічним многочленом

$$f(x_1,x_2,...,x_n)=k_0+k_1x_1+k_2x_2+k_3x_1x_2+\ldots+k_nx_1x_2,\ldots,x_n,$$

де $k_1,...,k_s$ — коефіцієнти, що набувають значення 0 або 1. Число членів-наборів $(x_{i_1},...,x_{i_s})=2^n$. Оскільки f дорівнює 0 або 1, загальне число членів (поліномів) становить 2^{2^n}.

Теорема Жегалкіна дає можливість подати логічну функцію у вигляді поліномів різного степеня. Існує кілька класів перемикальних функцій, важливих для логічного аналізу.

Приклад. Виразити x_1+x_2 у вигляді композиції полінома Жегалкіна.

Шукаємо вираз x_1+x_2 у вигляді полінома з невизначеними коефіцієнтами:

$$x_1+x_2=ax_1x_2+bx_1+cx_2+d.$$

При $x_1=x_2=0$ маємо $d=0$; при $x_1=0, x_2=1$ дістаємо $c=1$; при $x_1=1$, $x_2=0$ маємо $b=1$; при $x_1=x_2=1$ дістаємо $1=a+b+c$, тобто $a=1$.

Тоді остаточно

$$x_1 + x_2 = x_1 x_2 + x_1 + x_2.$$

Існує кілька повних систем. Із поняттям повноти тісно пов'язане поняття замикання і замкненого класу.

Означення 3.14. *Нехай A — деяка підмножина функцій з $P(A \in P)$. Замиканням A називається множина всіх булевих функцій, зображуваних у вигляді формул через функції множини A. Замикання множини A позначатимемо $[A]$.*

Приклади: 1. $A = P$; очевидно, що $[A] = P$.

2. $A = \{1, x_1 + x_2\}$. Замиканням цієї множини буде клас L усіх лінійних функцій, тобто функцій, що мають вигляд

$$f(x_1, x_2, ..., x_n) = c_0 + c_1 x_1 + ... + c_n x_n,$$

де $c_i = 0 \vee 1 (i = 0, ..., n)$.

Означення 3.15. *Клас (множина) A називається функціонально замкненим, якщо $[A] = A$.*

Приклади: 1. Клас $A = P$ — замкнений.

2. Клас $A = \{1, x_1 + x_2\}$ не є замкненим.

3. Усякий клас $[A]$ буде замкненим.

У термінах замкненого класу можна дати таке означення повноти: A — повна система, якщо $[A] = P$.

3.6.4. Типи булевих функцій

В алгебрі логіки у множині різних булевих функцій розрізнюють п'ять типів булевих функцій.

1. Позначимо через T_0 клас усіх булевих функцій $f(x_1, x_2, ..., x_n)$, що зберігають константу 0, тобто функцій, для яких

$$f(0, 0, ..., 0) = 0.$$

Якщо $f \in T_0$, а f^1 — функція, що дорівнює функції f, то й $f^1 \in T_0$. Число функцій класу T_0 становлять $N_0 = \frac{1}{2} \cdot 2^{2^n}$. Це функції f_1, f_2, f_6, f_9, f_{10}, f_{11}, f_{13}, f_{16} для $n = 2$.

2. Нехай T_1 — клас функцій $f(x_1, x_2, ..., x_n)$, що зберігають константу 1, тобто для яких

$$f(1, 1, ..., 1) = 1.$$

Клас T_1 складається з функцій, двоїстих до функцій з класу T_0 (клас T_0 двоїстий до T_1).

Клас T_1 містить $N_1 = \frac{1}{2} \cdot 2^{2^n}$ функцій. Це функції $f_1, f_2, f_3, f_4, f_5, f_{11}, f_{13}$, f_{15} для $n = 2$.

3. Нехай S — клас самодвоїстих функцій f з P таких, що $f^* = f$.

Для самодвоїстої функції справджується тотожність

$$\overline{f}(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n) = f(x_1, x_2, \ldots, x_n),$$

тобто в наборах $(\alpha_1, \alpha_2, ..., \alpha_n)$ та $(\bar{\alpha}_1, \bar{\alpha}_2, ... \bar{\alpha}_n)$, які називатимемо протилежними, самодвоїста функція набуває протилежних значень.

4. Нехай M — клас монотонних функцій і нехай $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$, $\beta = (\beta_1, \beta_2, ..., \beta_n)$ — будь-які набори.

Означення 3.16. *Для двох наборів* $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ *та* $\beta = (\beta_1, \beta_2, ..., \beta_n)$ *виконується відношення передування, якщо* $\alpha_1 \le \beta_1$, $\alpha_2 \le \beta_2, ..., \alpha_n \le \beta_n$.

Наприклад, набори $\alpha = (0, 1, 0, 1)$ й $\beta = (1, 1, 0, 1)$ знаходяться у відношенні передування, тобто значення набору не зменшується. Набори (0, 1) та (1, 0) не знаходяться у відношенні передування.

Означення 3.17. *Функція* $f(x_1, x_2, ..., x_n)$ *називається монотонною, якщо для будь-яких двох наборів* α *і* β, *що знаходяться у відношенні передування (тобто значення наборів не зменшується), справджується нерівність*

$$f(\alpha) \le f(\beta).$$

Функція, рівносильна монотонній, також є монотонною. Тут M — множина монотонних функцій.

5. Розглянемо L-клас лінійних функцій.

Перемикальна функція двох змінних називається лінійною, якщо вона може бути подана поліномом першого степеня

$$f(x_0, x_1) = k_0 \oplus k_1 x_0 \oplus k_2 x_1 = k_0 + k_1 x_0 + k_2 x_1, (k_0, k_1, k_2 = 0 \vee 1),$$

тобто канонічним многочленом, що не містить добутку змінних. Так, якщо кількість коефіцієнтів дорівнює $n + 1$, то число лінійних многочленів становить 2^{n+1} (для двох змінних $(2^3 = 8)$ існує вісім лінійних функцій).

Легко довести (пропонуємо читачеві виконати це самостійно), що класи T_0, T_1, S, M, L є замкненими.

3.6.5. Теорема про повноту. Теорема Поста

Теорема 3.5 (про функціональну повноту). *Для того щоб система функцій* $(f \in A)$ *була повною, необхідно й достатньо, щоб вона не містилася в жодному з п'яти замкнених класів* T_0, T_1, S, M, L:

$$f \not\subset T_0, f \not\subset T_1, f \not\subset S, f \not\subset M, f \not\subset L.$$

Для доведення треба розглянути п'ять функцій:

$$f_1 \notin T_0, f_2 \notin T_1, f_3 \notin S, f_4 \notin M, f_5 \notin L$$

і покласти $A' = \{f_1, f_2, f_3, f_4, f_5\}$.

Теорема 3.6. *З усякої повної системи функцій можна вилучити підсистему, що містить не більш як чотири функції.*

Доведення. Дійсно, яка-небудь одна функція $f_i \notin T_0$ або не самодвоїста, тому що $f_i(0, 0, ..., 0) = f_i(1, 1, ..., 1)$, або не зберігає 1 і не монотонна, буде повною системою чотирьох функцій.

Отже, розглянуті елементарні логічні функції можуть мати або не мати такі властивості:

- збереження нуля $(k_0 = 0)$;
- збереження одиниці;
- самодвоїстість (непарність) $f(x_0, x_1) = \overline{f}(\overline{x}_0\overline{x}_1)$ для функцій двох змінних й $f(x_1, x_2, ..., x_n) = \overline{f(x_1, x_2, ..., x_n)}$ у загальному випадку;
- монотонність $f(x_0, x_1) \le f(x_0', x_1')$ при $x_0 \le x_0'$ та $x_1 \le x_1'$;
- лінійність

$$f(x_0, x_1) = k_0 + k_1x_0 + k_2x_1;$$

$$f(x_1, ..., x_n) = k_0 + k_1x_1 + k_2x_2 + ... + k_nx_n,$$

де k_0, k_1, k_2 — двійкові константи (0 або 1).

Система (набір) елементарних логічних функцій називається функціонально повною, якщо довільну перемикальну функцію можна подати у вигляді суперпозиції функцій цієї системи.

Теорема 3.7 (теорема Поста). *Щоб система перемикальних функцій була повною, необхідно й достатньо, щоб вона містила хоча б одну функцію, яка не зберігає нуль, одиницю, тобто не є лінійною, монотонною, самодвоїстою.*

Іншими словами, для задоволення критерію повноти необхідно й достатньо, щоб серед функцій системи були:

- функція, що не зберігає константу 0;
- функція, що не зберігає константу 1;
- функція, що не є самодвоїстою;
- функція, що не є монотонною;
- функція, що не має властивості лінійності.

Якщо кожна з узятих функцій має лише одну властивість (але відмінну від інших), то для функціональної повноти потрібна система з п’яти функцій.

Повна система називається такою, що не скорочується, якщо вилучення будь-якої функції системи порушує її повноту. У зв’язку з тим, що кожна з функцій має кілька властивостей, функціонально повні системи можуть бути побудовані за допомогою однієї, двох, трьох і чотирьох функцій. Найпоширенішою є система з трьох функцій: І, АБО, НЕ. За допомогою цих функцій можуть бути описані процеси управління будь-якими виробництвами, будь-яка функція, що характеризує роботу будь-якого пристрою обчислювальної системи, подібно до того, як у музиці за допомогою семи нот можна записати симфонію, оперу тощо.

3.7. Канонічні форми перемикальних функцій

3.7.1. Проблема розв'язуваності

Означення 3.18. *Формула називається тотожно істинною, якщо вона при всіх значеннях змінних, що входять у неї, набуває значення* 1.

Приклади тотожно істинних формул:

$$x \vee \overline{x} = 1; x \to (y \to x); x(x \to y) \to y.$$

Означення 3.19. *Формула називається тотожно хибною, якщо вона при всіх значеннях змінних, що входять у неї, набуває значення* 0, *наприклад*

$$x\overline{x} = 0.$$

Означення 3.20. *Формула називається здійсненною (нейтральною), якщо вона не є тотожною* 0 *або* $1(0 \vee 1)$, *тобто набуває значення* 1 *при деяких значеннях змінних, що входять у неї.*

Можна поставити таке завдання: задати єдиний спосіб (алгоритм), який дає змогу для кожної формули з'ясувати, чи є вона здійсненною, тобто чи не є вона тотожною 0 або 1. Таке завдання має назву проблеми *розв'язуваності.*

Нехай $F(x_1, x_2, ..., x_n)$ — формула, що виражає деяку функцію n змінних: $x_1, x_2, ..., x_n$. При цьому як змінні $x_1, ..., x_n$, так і функція F можуть набувати лише двох значень, число ж можливих комбінацій значень змінних $x_1, ..., x_n$ є скінченним і дорівнює 2^n. Для кожної такої комбінації можна визначити значення формули F. Знаючи ці її значення для кожної комбінації змінних $x_1, x_2, ..., x_n$, можна встановити, здійсненна чи ні функція.

Викладений спосіб, звичайно, дає принципове вирішення проблеми розв'язуваності, але при великій кількості змінних він практично нездійсненний через величезне число можливих наборів значень змінних.

Існує інший спосіб, що ґрунтується на зведенні формул до нормальної форми. Якщо у процесі такого зведення формула не перетворюється на тотожні 0 або 1, то це свідчить про її здійсненність.

Синтез комбінаційної схеми зводиться до визначення булевого виразу заданої перемикальної функції. Подальший перехід від булевого виразу до системи є однозначним і не викликає особливих труднощів. Методи, які дають змогу для будь-якої перемикальної функції записати булевий вираз, ґрунтуються на тому, що вводяться вирази певного типу — канонічні форми, а потім формуються досить прості правила запису будь-якої функції у цих формах. Як канонічні звичайно використовуються досконалі диз'юнктивна та кон'юктивна нормальні форми (ДДНФ і ДКНФ).

3.7.2. Нормальні та досконалі диз'юнктивні нормальні форми перемикальних функцій

При розгляді перемикальних функцій можна, наприклад, зустріти вираз вигляду $x_1x_2\overline{x}_3$, який називається елементарною кон'юнкцією.

Означення 3.21. *Логічний добуток будь-якої кількості різних незалежних змінних (літер), що входять із запереченням або без нього, називається елементарною кон'юнкцією.*

Звідси випливає, що $(x_1 + x_0)x_2$ й $x_1x_2 + \overline{x}_3$ — не є елементарною кон'юнкцією.

Функція $F' = x_{i_1} \wedge x_{i_2} \wedge ... \wedge x_{i_r}$ — елементарна кон'юнкція, r — її ранг, а $i_v \neq i_k$ при $v \neq k$.

Означення 3.22. *Якщо функцію задано формулою у вигляді диз'юнкції елементарних кон'юнкцій, то її задано диз'юнктивною нормальною формою (ДНФ).*

Наприклад:

$$f(x)=x_1x_2x_3+\bar{x}_1x_2x_3+\bar{x}_1x_2.$$

Конституентою одиниці (мінтермом) називають перемикальну функцію n аргументів, яка набуває значення 1 тільки на одному їх кортежі. Кількість різних конституент одиниці для функцій n аргументів дорівнює числу різних кортежів наборів, тобто 2^n. Згідно з табл. 3.3 конституентами одиниці є f_1, f_8, f_9, f_{10} для $n = 2$.

Твердження 3.2. *Будь-яка таблично задана функція алгебри логіки може бути подана у вигляді*

$$f(x_1,x_2,\ldots,x_n)=F_1^1\vee F_2^1\vee\ldots\vee F_n^1=\bigvee_{i=1}^{n}F_i^1,$$

де F_i^1 — елементарна кон'юнкція рангу n; i — номери наборів, на яких функція F_i^1 дорівнює 1; $\vee$ — символ узагальненої диз'юнкції.

Означення 3.23. *ДДНФ перемикальної функції називається диз'юнкція тих конституент одиниці, які перетворюються в одиницю на тих самих наборах, що й задана функція.*

Будь-яка перемикальна функція має одну ДДНФ (кількість її членів дорівнює кількості одиничних значень функції) і кілька ДНФ. Будь-яка ДНФ утворюється внаслідок більшого або меншого скорочення ДДНФ, причому від будь-якої ДНФ можна перейти до ДДНФ. Такий перехід називається розгортанням.

Наприклад,

$$f(x_0,x_1,x_2,x_3)=\bar{x}_1x_2+x_0x_2+\bar{x}_0x_1\bar{x}_2+x_3=$$
$$=\bar{x}_1x_2(x_0+\bar{x}_0)(x_3+\bar{x}_3)+x_0x_2(x_1+\bar{x}_1)(x_3+\bar{x}_3)+\bar{x}_0x_1\bar{x}_2(x_3+\bar{x}_3)+$$
$$+x_3(x_0+\bar{x}_0)(x_1+\bar{x}_1)(x_2+\bar{x}_2)=\ldots.$$

Можна дати також інші означення ДДНФ.

Означення 3.24. *ДДНФ формули $A(x_1,x_2,\ldots,x_n)$, що містить n різних змінних, називається ДНФ, яка має такі властивості:*

- *в ній немає однакових доданків;*
- *жоден із доданків не містить двох однакових співмножників;*
- *жоден із доданків не містить змінну разом з її запереченням;*
- *в кожному окремому доданку є як співмножник або змінна x_i, або її заперечення для будь-якого $i = 1, 2, \ldots, n$.*

3.7.3. Нормальні та досконалі кон'юнктивні нормальні форми перемикальних функцій

Означення 3.25. *Логічна сума будь-якої кількості різних незалежних змінних, що входять із запереченням або без нього, називається елементарною диз'юнкцією.*

Звідси випливає, що $(x_0+x_2+\bar{x}_3)$, $(x_0+\bar{x}_1+x_2+\bar{x}_3)$ й $(x_0+x_1+\bar{x}_0)$ — елементарні диз'юнкції. Аналогічно до попереднього $F^0=x_{i_1}\vee x_{i_2}\vee\ldots\vee x_{i_r}$ — елементарна диз'юнкція рангу r.

Означення 3.26. *Якщо яку-небудь функцію задано формулою у вигляді кон'юнкції елементарних диз'юнкцій, то функцію задано її кон'юнктивною нормальною формою (КНФ).*

Наприклад:

$$f(x) = \left(x_0 + x_2 + \overline{x}_3\right)\left(x_0 + \overline{x}_1 + \overline{x}_2 + \overline{x}_3\right)\left(x_0 + x_3\right).$$

Будь-яка перемикальна функція може бути задана і своєю досконалою кон'юнктивною нормальною формою (ДКНФ). Для цього використовується поняття конституенти нуля — макстерму.

Означення 3.27. *Конституентою нуля (макстермом) називають перемикальну функцію n аргументів, яка набуває значення* 0 *тільки в одному наборі. Існує* 2^n *конституент нуля. Згідно табл. 3.3 конституентами нуля є* f_2, f_3, f_4, f_7 для $n = 2$.

Конституенти нуля можна виразити у вигляді диз'юнкцій аргументів, частина з яких береться від'ємними.

Твердження 3.3. *Будь-яка таблично задана функція алгебри логіки може бути подана в аналітичній формі*

$$f(x_1, x_2, \ldots, x_n) = F_1^0 \wedge F_2^0 \wedge \ldots \wedge F_\ell^0 = \bigwedge_{i=0}^{\ell} \left(\tilde{x}_1 + \tilde{x}_2 + \ldots + \tilde{x}_i\right),$$

де $\wedge$ *— символ узагальненої диз'юнкції, а* $\tilde{x}_i$ *— це* x_i *або* $\overline{x}_i$ $(i = 1, \ldots, n)$.

Досконалою КНФ перемикальної функції називається кон'юнкція тих конституент нуля, які перетворюються на нуль в тих самих кортежах (наборах) значень аргументів, що й задана функція.

Іншим означенням ДКНФ є таке.

Означення 3.28. *ДКНФ перемикальної функції називається її КНФ, що задовольняє такі умови:*

- *в ній немає двох однакових співмножників;*
- *жоден зі співмножників не містить двох однакових доданків;*
- *жоден зі співмножників не містить якої-небудь змінної разом з її запереченням;*
- *кожен співмножник містить як складову* x_i *або* $\overline{x}_i$ *для будь-якого* $i = 1, 2, \ldots, n$.

3.7.4. Властивості досконалих форм

Це такі властивості:

- будь-яка кон'юнктивна або диз'юнктивна нормальна форма не дає однозначного подання функції, яке буде тільки при досконалих нормальних формах (ДДНФ і ДКНФ);
- у ДДНФ (ДКНФ) немає двох однакових мінтермів (макстермів);
- у ДДНФ (ДКНФ) жоден із мінтермів (макстермів) не містить двох однакових співмножників (змінних);
- у ДДНФ (ДКНФ) жоден із мінтермів (макстермів) не містить разом зі змінною її заперечення.

Теорема 3.8. *Будь-яка функція алгебри логіки, крім абсолютно істинної й абсолютно хибної функцій, може бути подана в ДКНФ і ДДНФ:*

$$f(x_1, x_2, ..., x_n) = \underset{0}{\wedge}\left(\tilde{x}_1 \vee \tilde{x}_2 \vee ... \vee \tilde{x}_n\right) \text{ — ДКНФ;}$$

$$f(x_1, x_2, ..., x_n) = \underset{1}{\vee}(\tilde{x}_1 \wedge \tilde{x}_2 \wedge ... \wedge \tilde{x}_n) \text{— ДДНФ,}$$

де $\underset{0}{\wedge}$ *і* $\underset{1}{\vee}$ *— символи узагальненої кон'юнкції та диз'юнкції конституент нуля й одиниці відповідно, а* $\tilde{x}_i$ *— це* x_i *або* $\overline{x}_i$ $(i = 1, 2,..., n)$.

Наприклад,

$$f(x_1, x_2, x_3) = \underset{1}{\vee}(0, 2, 4, 7) = \overline{x}_1\overline{x}_2\overline{x}_3 \vee \overline{x}_1 x_2\overline{x}_3 \vee x_1\overline{x}_2\overline{x}_3 \vee x_1 x_2 x_3.$$

3.7.5. Перехід від табличного подання перемикальної функції до алгебричного

Мінтерми та макстерми використовуються для переходу від табличного подання функції до алгебричного. Щоб здійснити такий перехід, кожному кортежу — набору змінних ставиться у відповідність мінтерм (конституента одиниці) — кон'юнкція всіх змінних, які входять у прямому вигляді, якщо значення заданої змінної в наборі дорівнює одиниці, або в інверсному вигляді, якщо воно дорівнює нулю. Для *n* змінних додаються $q = 2^n$ мінтермів вигляду $m_0, m_1, \ldots, m_{2^n - 1}$.

Усі мінтерми функції двох змінних наведено в табл. 3.12. Значення функції *F*, що відповідає заданому *i*-му набору змінних, будемо позначати f_i. Як випливає з табл. 3.12, алгебричне подання функції *F* є сумою мінтермів, що відповідають наборам змінних, для яких $f_i = 1$.

Таблиця 3.12

A	*B*	Мінтерм	Макстерм	Значення функції *F*(*A*, *B*)
0	0	$m_0 = \overline{A}\overline{B}$	$M_0 = A + B$	$f_0 = 1$
0	1	$m_1 = \overline{A}B$	$M_1 = A + \overline{B}$	$f_1 = 0$
1	0	$m_2 = A\overline{B}$	$M_2 = \overline{A} + B$	$f_2 = 0$
1	1	$m_3 = AB$ «за 1»	$M_3 = \overline{A} + \overline{B}$ «за 0»	$f_3 = 1$

Наприклад, $F_5^1(A, B) = f_0 m_0 + f_1 m_1 + f_2 m_2 + f_3 m_3 = 1(\overline{A}\overline{B}) + 0(\overline{A}B) + 0(A\overline{B}) + {}$ $+1(AB) = \overline{A}\overline{B} + AB.$

Загалом алгебричний вираз будь-якої логічної функції можна подати в такій формі:

$$F^1 = \sum_{i=0}^{q-1} f_i m_i,$$

де f_i — значення функції (0 або 1), а m_i — мінтерм, що відповідає i-му кортежу змінних. Таке подання функції називається її ДДНФ.

Інша алгебрична форма подання функції буде при використанні макстермів. Макстермом (конституентою нуля) називається диз'юнкція всіх змінних, які входять у прямому вигляді, якщо значення заданої змінної дорівнює 0, або в інверсному вигляді, якщо воно дорівнює 1 (див. табл. 3.12). Число макстермів, як і мінтермів, для функції n змінних $q = 2^n$. Алгебричний вираз функції записується у вигляді добутку

$$F^0 = \prod_{i=0}^{q-1} (f_i + M_i),$$

де f_i та M_i — значення функції і макстерм, що відповідають i-му набору змінних. Таке подання функції називається її ДКНФ.

Наприклад, для функції f_5, використовуючи табл. 3.12, ДКНФ має вигляд

$$F_5^0 (A, B) = (f_0 + M_0)(f_1 + M_1)(f_2 + M_2)(f_3 + M_3) =$$
$$= \left(1 + \bar{A} + \bar{B}\right)\left(0 + A + \bar{B}\right)\left(0 + \bar{A} + B\right)\left(1 + \bar{A} + \bar{B}\right) = \left(A + \bar{B}\right)\left(\bar{A} + B\right).$$

Використовуючи закони алгебри логіки, неважко довести еквівалентність ДКНФ і ДДНФ функції f_5.

Якщо у виразах F_1 та F_0 замість f_i записати інверсії значення функції, то дістанемо ДДНФ і ДКНФ для функції, що є інверсією заданої. Наприклад, на підставі табл. 3.12 маємо

$$\bar{f}_5 = \overline{A \sim B} = \bar{A}B + A\bar{B} = \left(\bar{A} + \bar{B}\right)\left(A + B\right) = f_6 = A \oplus B.$$

Таким чином здійснюється перехід від таблиці істинності до алгебричного подання логічної функції, завдяки чому будь-яка логічна функція може бути подана у вигляді ДДНФ або ДКНФ. Однак ДДНФ або ДКНФ функцій часто не є найпростішими їх виразами. Використовуючи логічну тотожність і закони, можна здебільшого добути простіші форми подання функції, які називаються мінімізованими.

Зворотний перехід від алгебричного до табличного подання функції виконується послідовною підстановкою в алгебричний вираз усіх q можливих наборів змінних, визначенням відповідних значень $F = f_i$ для кожного i-го набору $\left(0 \le i \le q - 1\right)$ та заповненням таблиць істинності.

3.7.6. Поняття про індекс (коефіцієнт) простоти

Якщо всім можливим наборам значень аргументів якої-небудь функції поставити у відповідність такі самі поєднання значень вхідних сигналів якої-небудь системи з невідомою внутрішньою структурою («чорного ящика») і якщо при цьому значення функції збігаються зі значеннями вихідних сигналів, то така система реалізує задану логічну функцію або ця функція є законом функціонування системи. Система складається з окремих компонентів — логічних елементів.

Основною метою синтезу є пошук структури системи за заданим описом її роботи. Одним з основних питань є таке: як для довільної функції алгебри логіки $f(x_1, x_2, ..., x_n)$ побудувати її мінімальну ДНФ або КНФ. Ця задача називається проблемою мінімізації булевих функцій. Найзрозумілішим є алгоритм, що ґрунтується на переборі, тобто перегляді всіх можливих ДНФ і КНФ функції. Проте ним не можна скористатися практично вже при $n = 3$, а при $n = 1$ та $n = 2$ проблема є тривіальною.

Спочатку знаходять конкретну функцію алгебри логіки, визначену для всіляких наборів значень аргументів. Після цього функцію подають у будь-якій з двох досконалих нормальних форм. Потім здійснюють низку спрощень, що досягається за допомогою різних тотожних перетворень з метою здобуття формули, еквівалентної початковій, але яка реалізується простіше.

При оцінюванні складності приймають коефіцієнт простоти.

Розглянемо функцію

$$A_1 = \overline{x}_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 + x_1\overline{x}_2 x_3 + x_1 x_2\overline{x}_3 + x_1 x_2 x_3, \qquad (3.3)$$

еквівалентну формулі

$$A_2 = A_1 = \overline{x}_2\overline{x}_3 + x_1. \qquad (3.4)$$

При цьому постає завдання вибору форми, найприйнятнішої для практичної реалізації. З цією метою вводиться індекс (коефіцієнт) простоти $L(A)$, що характеризує складність ДНФ (КНФ).

Найчастіше зустрічаються такі типи коефіцієнтів простоти:

- $L'(A)$ — число символів змінних, які зустрічаються в запису ДНФ. Якщо проаналізувати функції (3.3) і (3.4), то можна встановити, що $L'(A_1) = 15$, а $L'(A_2) = 3$, тобто функція (3.4) є простішою;
- $L''(A)$ — число елементарних кон'юнкцій, що входять у функцію A. Для ДНФ A_1 й A_2 очевидно, що $L''(A_1) = 5$, а $L''(A_2) = 2$, тобто функція (3.4) теж є простішою;
- $L'''(A)$ — число символів інверсій, які зустрічаються в запису ДНФ. Для ДНФ A_1 та A_2 $L'''(A_1) = 7$, а $L'''(A_2) = 3$, тобто функція (3.4) знову-таки є простішою.

3.7.7. Мінімальні диз'юнктивні та кон'юнктивні нормальні форми

Досконалі ДНФ та КНФ використовуються для первинного подання заданої перемикальної функції. Однак ці форми є незручними для опису і побудови логічних схем, тому що схеми, що реалізують їх, часто виявляються складними, тобто містять елементи, які можна виключити при синтезі схем.

Наприклад, функція «змінна у» $f_{13}(x, y) = \underbrace{\overline{x}y \vee yx}_{\text{ДДНФ}} = y$, виходячи з ДДНФ, реалізується на чотирьох елементах (НЕ, І, АБО, І), як показано на рис. 11.

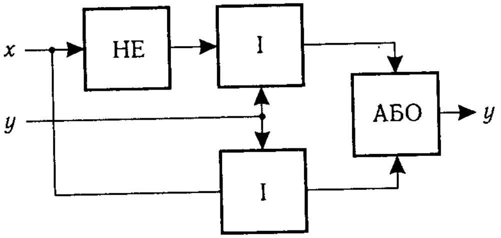


Рис. 11

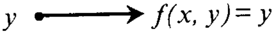


Рис. 12

Ця функція записується як $f_{11}(x,y)=y$, і при побудові схеми за цією формою не потрібно жодного елемента, оскільки вона реалізується відрізком провідника (рис. 12).

Це свідчить про те, наскільки важливо мати ефективні методи пошуку найраціональніших щодо технічної реалізації форм подання перемикальних функцій.

Тому чергова задача синтезу схем полягає у спрощенні виразів для перемикальних функцій. Цей етап називається їх мінімізацією.

Уведемо деякі означення.

Означення 3.29. *ДНФ (КНФ) функції називається мінімальною, якщо кількість змінних, які вона містить, буде не більшою, ніж у будь-якої іншої ДНФ (КНФ) тієї самої функції.*

Означення 3.30. *ДНФ (КНФ), що реалізує функцію* $f(x_1, x_2, ..., x_n)$ *і має мінімальний індекс L, називається мінімальною відносно L. Ідеться про коефіцієнт простоти з мінімальним числом символів, а не змінних.*

Наприклад, $xy \vee x\bar{y}z \vee yz$ містить сім літер, але три змінні.

Деякі перемикальні функції мають кілька мінімальних ДНФ.

Означення 3.31. *ДНФ функції називається мінімальною, якщо кількість символів, які вона містить, буде не більшою, ніж у будь-якій іншій ДНФ тієї самої функції.*

Мінімізацією називається перетворення функції, яке веде до зменшення числа символів, а отже, числа змінних. Мінімізація веде до спрощення алгебричного виразу, тобто до спрощення автомата, що описується заданим виразом. Найпростішим є алгебричний метод мінімізації, алгоритм якого відображено на рис. 13.

Мінімальні форми перемикальних функцій можуть бути знайдені аналітично або за допомогою мінімізаційних карт. Аналітично їх здобувають у такій послідовності:

- функція має бути подана в одній з двох досконалих форм (ДДНФ або ДКНФ);
- знаходять скорочену ДНФ (КНФ), будь-яка функція має одну таку форму;
- знаходять можливі тупикові ДНФ (КНФ);
- з одержаних тупикових форм вибирають мінімальні ДНФ (КНФ).

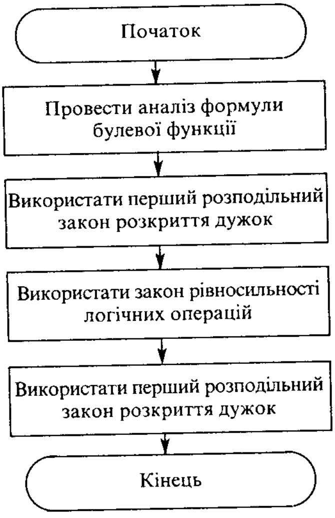


Рис. 13

3.8. Скорочена диз'юнктивна нормальна форма перемикальних функцій

3.8.1. Поняття скороченої форми

Уведемо поняття накриття для перемикальних функцій.

Нехай в якому-небудь наборі аргументів $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ функція f набуває значення a_1, а функція φ в цьому самому наборі — значення a_2. Це означає, що функція f у заданому наборі своїм значенням a_1 накриває значення a_2.

Досконала ДНФ будується так, що кожна одиниця перемикальної функції накривається одиницею тільки одного добутку, що є конституентою одиниці. Тому кількість конституент одиниці, які входять у ДДНФ, дорівнює числу наборів, у яких функція дорівнює 1.

Ідея побудови скороченої ДНФ полягає в тому, що в неї включаються елементарні добутки, які накривають своїми одиницями не одну, а кілька одиниць заданої функції.

Наприклад, ДДНФ функції $f_3(x, y)$ (імплікація) має вигляд

$$f_3(x, y) = \overline{x}\overline{y} \vee x\overline{y} \vee xy.$$

Кожен із добутків $\overline{x}\,\overline{y}, x\,\overline{y}, x\,y$ накриває тільки одну одиницю функції. Проте елементарний добуток x накриває одиницями дві одиниці функції у двох наборах: 1,0 та 1,1; елементарний добуток $\overline{y}$ — також у двох наборах: 0,0 і 1,0; x та $\overline{y}$ спільно накривають одиницями всі одиниці функції $f_3(x, y)$, яка внаслідок цього може бути подана як

$$f_3(x, y) = x \vee \overline{y}.$$

Якщо деяка перемикальна функція φ (в окремому випадку — елементарний добуток) дорівнює нулю в тих наборах, у яких дорівнює нулю інша функція f, то вважають, що функція φ входить у функцію f, тобто функція φ входить у функцію f тоді, коли вона накриває нулями всі нулі функції f, а одиниці функції можуть бути накриті як нулями, так і одиницями функції φ. Отже, функція φ має не меншу кількість нулів, ніж функція f.

Умова входження записується як $\varphi \in f$ (φ належить f). Наприклад, у функцію $f_6(x, y)$ входитимуть всі функції з нульовими значеннями в наборах 0,0 та 1,1, тобто функції $f_9(x, y)$, $f_{10}(x, y)$, $f_{16}(x, y)$. Тому можна записати:

$$\overline{x}y \in f_6(x, y);\ x\overline{y} \in f_6(x, y);\ 0 \in f_6(x, y);$$

$$\overline{a}b \in f_6(a, b);\ a\overline{b} \in f_6(a, b);\ 0 \in f_6(a, b).$$

Константа 0 входить у будь-яку перемикальну функцію, а в константу 1 входять усі функції.

Функцію φ, що входить у задану функцію f, називають її імплікантою.

Застосування терміна «імпліканта» пов'язано з перемикальною функцією

$$f_3(x, y) = x \to y,$$

яка називається імплікацією. Користуючись табл. 3.3, можна пересвідчитися, що для f_3 вираз $\varphi \to f$ тотожно дорівнює 1, тобто завжди є істинним тоді, коли функція φ входить у функцію f.

Простими імплікантами перемикальної функції називаються терми — елементарні добутки, що самі входять у задану функцію, але ніяка власна частина їх у функцію f не входить.

Прості імпліканти є найкоротшими термами — елементарними добутками, що входять у задану перемикальну функцію.

Власною частиною імпліканти називають терм-добуток, здобутий вилученням із нього одного або кількох співмножників. Наприклад, добуток $x\bar{y}z$ має власні частини $x\bar{y}$, $\bar{y}z$, xz, x, $\bar{y}$, z.

Якщо який-небудь елементарний добуток-терм входить у задану функцію, то при доданні до нього будь-яких співмножників новий добуток також входитиме в цю функцію, тому що він стає нулем разом із початковим добутком.

Наприклад, для функції

$$f(x, y, z) = \bar{x}yz + x\overline{\bar{y}z} + x\bar{y}z + x\bar{y}$$

простими імплікантами будуть добутки $\bar{x}yz$ й $x\bar{y}$, а $x\overline{\bar{y}z}$ та $x\bar{y}z$ не є ними, оскільки їхня власна частина $x\bar{y}$ входить у задану функцію.

Теорема 3.9. *Будь-яка перемикальна функція дорівнює диз'юнкції всіх своїх простих імплікант.*

Доведення. У нульових наборах (кортежах) за означенням усі прості імпліканти входять у перемикальну функцію, якщо вони також дорівнюватимуть 0 у цих наборах і, отже, дорівнюватиме 0 їх диз'юнкція.

В одиничних наборах для кожного такого набору-кортежу знайдеться хоча б одна імпліканта, що дорівнює 1. Прості імпліканти вибираються серед усіх елементарних добутків, які входять у перемикальну функцію. У число цих добутків входять усі конституенти одиниці заданої функції, а будь-яка проста імпліканта є власною частиною деяких конституент одиниці. Якщо деяка конституента одиниці не входить у набір усіх простих імплікант, то це означає, що вони замінюються коротшими елементарними добутками — простими імплікантами. Проста імпліканта дорівнює 1 у тому самому наборі, на якому дорівнюють 1 конституенти, що входять у неї.

Отже, серед усіх простих імплікант завжди знайдуться такі, які разом із заданою функцією перетворюються на 1 у цьому наборі. Таким чином, диз'юнкція всіх простих імплікант накриває всі нулі й одиниці заданої функції, тобто збігається з нею.

Диз'юнкція всіх простих імплікант називається скороченою ДНФ перемикальної функції.

3.8.2. Утворення скороченої диз'юнктивної нормальної форми методом Квайна

При мінімізації за методом Квайна передбачається, що початкова функція задається в ДДНФ. Використовується перетворення ДДНФ за допомогою операцій неповного склеювання і поглинання.

В операції повного склеювання

$$xy + x\overline{y} = x(y + \overline{y}) = x$$

два члени xy та $x\overline{y}$ склеюються за змінною y.

Операція неповного склеювання записується так:

$$x + xy + x\overline{y} = xy + x\overline{y}.$$

В операції поглинання

$$x + xy = x(1 + y) = x$$

член xy поглинається членом x.

Теорема 3.10 (теорема Квайна). *Якщо в ДНФ перемикальної функції виконати всі операції неповного склеювання, а потім усі операції поглинання, то дістанемо скорочену ДНФ, тобто диз'юнкцію всіх простих імплікант.*

Доведення. Розглянемо операцію розгортання, що є оберненою до операції склеювання. Розгортання — це множення деяких членів на вираз типу $(x + \overline{x})$, що, природно, не змінює значення цих членів. За допомогою операції розгортання проста імпліканта може бути подана у вигляді диз'юнкцій мінтермів-конституент одиниці.

Наприклад, $x\overline{y}$ — імпліканта аргументів x, y, z, u:

$$x\overline{y} = x\overline{y}(z + \overline{z}) = x\overline{y}z + x\overline{y}z = x\overline{y}z(u + \overline{u}) + \overline{x\overline{y}}(u + \overline{u}) =$$
$$= x\overline{y}zu + x\overline{y}z\overline{u} + \overline{x\overline{y}}zu + xy\overline{z}u.$$

Якщо утворений вираз міститиме кілька однакових мінтермів-конституент одиниць, то, замінивши диз'юнкцію мінтермів одним мінтермом, дістанемо в підсумку ДДНФ. Оскільки операція розгортання є оберненою до операції склеювання, то, застосувавши операції склеювання до ДДНФ, можна знайти просту імпліканту.

Виконавши спочатку операції неповного склеювання, здобудемо всі прості імпліканти. Тому при проведенні операцій склеювання кожний член потрібно залишати у виразі для використання його при інших склеюваннях.

Після виконання всіх операцій неповного склеювання диз'юнктивна форма міститиме крім простих імплікант інші члени. Провівши далі операції поглинання, дістанемо тільки прості імпліканти.

Підкреслимо, що перетворення треба починати, виходячи з ДДНФ.

За методом Квайна необхідно виконати такі дії:

- провести в ДДНФ функції $f(x_1, x_2, ..., x_n)$ усі можливі операції неповного склеювання конституент одиниці, внаслідок чого утворяться добутки — мінтерми $(n-1)$-го рангу (кон'юнкція $(n-1)$-го рангу). Після цього виконати операцію поглинання;

- провести можливі склеювання членів $(n-1)$-го рангу, потім поглинання і дістати кон'юнкцію $(n-2)$-го рангу;
- виконати операції склеювання членів із числом літер $(n-2)$-го рангу, одержати кон'юнкцію $(n-3)$-го рангу і т. д.

Приклад. Розглянемо функцію

$$F(x_1x_2x_3x_4)=\bar{x}_1\bar{x}_2x_3x_4+\bar{x}_1x_2\bar{x}_3\bar{x}_4+\bar{x}_1x_2\bar{x}_3x_4+\bar{x}_1x_2x_3x_4+$$
$$+x_1\bar{x}_2\bar{x}_3x_4+x_1\bar{x}_2x_3x_4+x_1x_2\bar{x}_3\bar{x}_4+x_1x_2\bar{x}_3x_4.$$

Мінтермами четвертого рангу тут є (табл. 3.13)

$$\bar{x}_1\bar{x}_2x_3x_4,\ \ \bar{x}_1x_2\bar{x}_3\bar{x}_4,\ \ \bar{x}_1x_2\bar{x}_3x_4,\ \ \bar{x}_1x_2x_3x_4,$$
$$x_1\bar{x}_2\bar{x}_3x_4,\ \ x_1\bar{x}_2x_3x_4,\ \ x_1x_2\bar{x}_3\bar{x}_4,\ \ x_1x_2\bar{x}_3x_4.$$

Мінтермами третього рангу слугують

$$\bar{x}_1x_3x_4,\ \ \bar{x}_1x_2\bar{x}_3,\ \ x_2\bar{x}_3x_4,\ \ \bar{x}_1x_2x_4\ \ x_1\bar{x}_2x_4,$$
$$\bar{x}_2x_3x_4,\ \ x_2\bar{x}_3\bar{x}_4,\ \ x_1x_2\bar{x}_3,\ \ x_1\bar{x}_3x_4.$$

Табл. 3.13 заповнюють для здобуття мінтермів третього рангу. Аналогічно знаходять мінтерми другого рангу.

Оскільки подальше склеювання неможливе, етап визначення імплікант закінчено. Простими імплікантами є такі мінтерми:

$$\bar{x}_1x_3x_4,\ \ \bar{x}_1x_2x_4,\ \ x_1\bar{x}_2x_4,\ \ \bar{x}_2x_3x_4,\ \ x_1\bar{x}_3x_4,\ \ x_2\bar{x}_3.$$

Таблиця 3.13

	$\bar{x}_1\bar{x}_2x_3x_4$	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	$\bar{x}_1x_2\bar{x}_3x_4$	$\bar{x}_1x_2x_3x_4$	$x_1\bar{x}_2\bar{x}_3x_4$	$x_1\bar{x}_2x_3x_4$	$x_1x_2\bar{x}_3\bar{x}_4$	$x_1x_2\bar{x}_3x_4$
$\bar{x}_1\bar{x}_2x_3x_4$	1			$\bar{x}_1x_3x_4$		$\bar{x}_2x_3x_4$		
$\bar{x}_1x_2\bar{x}_3\bar{x}_4$		1	$\bar{x}_1x_2\bar{x}_3$				$x_2\bar{x}_3\bar{x}_4$	$x_2\bar{x}_3$
$\bar{x}_1x_2\bar{x}_3x_4$		$\bar{x}_1x_2\bar{x}_3$	1	$\bar{x}_1x_2x_4$				$x_2\bar{x}_3x_4$
$\bar{x}_1x_2x_3x_4$	$\bar{x}_1x_3x_4$		$\bar{x}_1x_3x_4$	1				
$x_1\bar{x}_2\bar{x}_3x_4$					1	$x_1\bar{x}_2x_4$		$x_1\bar{x}_3x_4$
$x_1\bar{x}_2x_3x_4$	$\bar{x}_2x_3x_4$				$x_1\bar{x}_2x_4$	1		
$x_1x_2\bar{x}_3\bar{x}_4$		$x_2\bar{x}_3\bar{x}_4$					1	$x_1x_2\bar{x}_3$
$x_1x_2\bar{x}_3x_4$		$x_2\bar{x}_3$	$x_2\bar{x}_3x_4$		$x_1\bar{x}_3x_4$		$x_1x_2\bar{x}_3$	1
$\bar{x}_1x_3x_4$	$\vee$			$\vee$				
$\bar{x}_1x_2x_4$			$\vee$	$\vee$				
$x_1\bar{x}_2x_4$					$\vee$	$\vee$		
$\bar{x}_2x_3x_4$	✓					$\vee$		
$x_1\bar{x}_3x_4$					$\vee$			$\vee$
$x_2\bar{x}_3$		$\vee$	$\vee$				$\vee$	$\vee$

Розставлення міток. На етапі розставлення міток складають таблицю, число рядків якої дорівнює числу знайдених первинних імплікант функції, що мінімізується. Число стовпців збігається з числом мінтермів ДДНФ. Якщо в деякий мінтерм ДДНФ входить яка-небудь із первинних імплікант, то на перетині відповідних стовпця та рядка ставлять мітку (див. продовження табл. 3.13).

Визначення суттєвих імплікант. Якщо в якому-небудь із стовпців складеної таблиці існує тільки одна мітка, то первинна імпліканта, що є у відповідному рядку, називається суттєвою. Вона не може бути вилучена з правої частини початкового рівняння, оскільки без неї не буде покриття всієї множини мінтермів заданої функції. Тому з таблиці міток виключають рядки, які відповідають суттєвим імплікантам, і стовпці мінтермів, що покриваються ними.

Для функції, що розглядається, суттєвою імплікантою є $x_2\bar{x}_3$. Вона покриває мінтерми $\bar{x}_1x_2\bar{x}_3\bar{x}_4$, $\bar{x}_1x_2\bar{x}_3x_4$, $x_1x_2\bar{x}_3\bar{x}_4$, $x_1x_2\bar{x}_3x_4$.

При переході до наступного етапу ці мінтерми можуть бути закреслені.

Закреслювання зайвих стовпців. Якщо в одержаній після попереднього етапу табл. 3.14 є два стовпці з мітками в однакових рядках, то один із них закреслюють. Це можна зробити завдяки тому, що покриття стовпця, який залишився, здійснюється покриттям виключеного мінтерма. У табл. 3.14 однакових стовпців немає.

Таблиця 3.14

	$\bar{x}_1\bar{x}_2x_3x_4$	$\bar{x}_1x_2x_3x_4$	$x_1\bar{x}_2\bar{x}_3x_4$	$x_1\bar{x}_2x_3x_4$
$\bar{x}_1x_3x_4$	V	V		
$\bar{x}_2x_3x_4$	V			V
$\bar{x}_1x_2x_4$		V		
$x_1\bar{x}_2x_4$			V	V
$x_1\bar{x}_3x_4$			V	

Закреслювання зайвих первинних імплікант. Якщо після закреслювання деяких зайвих стовпців на попередньому етапі в таблиці з'являються рядки, в яких немає жодної мітки, то первинні імпліканти, які відповідають цим рядкам, виключаються, оскільки вони не покривають ті мінтерми, що залишилися.

Вибір мінімального покриття максимальними інтервалами. В одержаній таблиці вибирають таку сукупність первинних імплікант, яка виключає мітки у всіх стовпцях (принаймні одну мітку в кожному стовпці). При кількох можливих варіантах такого вибору перевага віддається варіанту покриття з мінімальним сумарним числом літер у простих імплікантах, які створюють покриття.

Для функції, що розглядається, вибирають покриття з імплікант $\bar{x}_1x_3x_4$ та $x_1\bar{x}_2x_4$, оскільки вони спільно покривають усі ті мінтерми, які залишилися після закреслювання зайвих стовпців. Мінімальна ДНФ для цієї функції має такий вигляд:

$$f(x_1x_2x_3x_4) = \bar{x}_1x_3x_4 + x_1\bar{x}_2x_4 + x_2\bar{x}_3.$$

3.8.3. Метод Квайна — Маккласкі

У методі Квайна є істотна незручність, пов'язана з необхідністю повного попарного порівняння імплікант на етапі визначення простих імплікант. Із зростанням числа мінтермів, що визначають ДДНФ заданої функції, збільшується кількість цих порівнянь. Це збільшення характеризується факторіальною функцією. Тому при досить великому числі мінтермів застосування методу Квайна стає важким. У 1956 р. Маккласкі запропонував модернізацію першого етапу методу Квайна, яка істотно зменшує кількість порівнянь мінтермів.

Ідея її полягає ось у чому. Якщо записати мінтерми у вигляді їхніх двійкових номерів, то всі номери можна розбити за числом одиниць у цих номерах на непересічні групи. При цьому в *i*-ту групу ввійдуть усі номери, що мають у своєму двійковому запису *i* одиниць. Попарно порівнювати можна тільки сусідні за номером групи, оскільки саме вони різні для тих мінтермів, що входять у них, в одному розряді. При створенні мінтермів з рангом, вищим від нульового, в розряди, які відповідають вилученим змінним, записують знак «тире». Така модифікація на практиці дуже зручна, оскільки дає змогу уникнути виписування громіздких мінтермів та імплікант, замінюючи виписування їхнім двійковим номером.

Приклад. Функцію в ДДНФ задано мінтермами з номерами 0,1,2,3,4,7,8,9,11,15. Записати ці мінтерми по групах у двійковому коді:

Нульова група 0000
Перша — “ — 0001, 0010, 0100, 1000
Друга — “ — 0011, 0110, 1001
Третя — “ — 0111, 1011
Четверта — “ — 1111

Порівнюючи сусідні групи, визначаємо по групах мінтерми першого рангу:

Нульова група 000–, 00–0, 0–00, –000
Перша — “ — 00–1, –001, 001–, 0–10, 01–0, 100–
Друга — “ — 0–11, –011, 011–, 10–1
Третя — “ — –111, 1–11

Тепер знаходимо мінтерми другого рангу:

Нульова група 00—, –00–, 0—0
Перша — “ — –0–1, 0–1–
Друга — “ — —11

Складаємо таблицю міток (табл. 3.15).

Таблиця 3.15

	0000	0001	0010	0011	0100	0110	0111	1000	1001	1011	1111
00—	v	v	v	v							
–00–	v	v						v	v		
0—0	v		v		v	v					
–0–1		v		v					v	v	
0–1–			v	v		v	v				
—11				v			v			v	v

Подальша мінімізація функції за методом Маккласкі збігається з мінімізацією за методом Квайна. Для ДНФ, що розглядається, мінімальна форма має такий вигляд:

$$f(x_1x_2x_3x_4)=\bar{x}_2\bar{x}_3+\bar{x}_1\bar{x}_4+x_3x_4$$

3.9. Тупикові нормальні форми

3.9.1. Поняття про тупикову диз'юнктивну нормальну форму

Нехай функція $f(x_1, x_2, ..., x_n)$ має довільну ДНФ вигляду

$$A = A' + K \text{ або } A = A' + \alpha K',$$

де A — ДНФ, утворена з інших кон'юнкцій, що входять в A; K — елементарна кон'юнкція; α — деякий множник із K; K' — добуток інших множників із K.

Існують два основних типи перетворень (операцій) ДНФ.

1. Операція вилучення елементарних кон'юнкцій. Перехід від A до A' — перетворення, яке здійснюється виключенням елементарних кон'юнкцій K, що можливо тільки при $A = A'$.

2. Операція вилучення множника, тобто перехід від A-ДНФ до ДНФ вигляду $(A' + K')$ вилученням множника. Це перетворення можливо тільки при $A = A' + K'$.

Означення 3.32. *ДНФ, яку не можна спростити за допомогою перетворень* 1 *і* 2 *(їх не можна застосувати), називають тупиковою ДНФ.*

Приклад. ДНФ $A = x_1 + \bar{x}_2 + \bar{x}_3$ є тупиковою.

Теорема 3.11. *Нехай* $f(x_1, x_2, ...x_n)$ *— довільна перемикальна функція, для якої* $A = A' = \bigvee_{i=1}^{r} K_i'$ *— її довільна тупикова ДНФ (після застосування операцій* 1 *і* 2*). Тоді існує тільки одна впорядкована ДДНФ, з якої за допомогою алгоритму перебору (перегляду) найшвидшого спуску утворюється тупикова ДНФ.*

Доведення. Нехай функція $f(x_1, x_2, ...x_n)$ має таку ДДНФ:

$A_{\text{ДДНФ}} = \bigvee_{i=1}^{r} (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n) = \bigvee_{i=1}^{r} K_i'$, де $(\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_n)$ — її довільний член-конституента 1, а $\tilde{x}_i = x_i' \vee \bar{x}_i$.

Для функції, що не дорівнює 0, завжди існує принаймні хоча б одна кон'юнкція $K_j' = (\tilde{x}_{j_1}, \tilde{x}_{j_2}, ...\tilde{x}_{j_n}) = 1$. Цей множник можна подати у вигляді $K_j' = K_{j_0} K$. Тоді

$$A = A' + K_j = A' + K_{j_0} K = A'.$$

Наслідок 3.1. *Внаслідок того, що серед тупикових ДНФ є обов'язково й мінімальні відносно* $L(A)$*, алгоритм спрощення дає змогу дістати мінімальну ДНФ, де коефіцієнти простоти* $L(A') \leq L(A)$ *та* $L(A' + K') \leq L(A)$.

Можна сформулювати один з алгоритмів утворення тупикових ДНФ (алгоритм спрощення, тобто алгоритм найшвидшого спуску):

- функцію подають у ДДНФ;
- упорядковують ДДНФ (записують співмножники так, щоб було зручно здійснювати перетворення);

- до ДДНФ застосовують операцію виключення елементарних кон'юнкцій, а потім операцію вилучення співмножників.

У підсумку здобувають початкову ДНФ, але спрощену або мінімізовану.

Теорема 3.12. *ДНФ, утворена після застосування алгоритму найшвидшого спуску, є тупиковою.*

Результат використання алгоритму спрощення залежить від вибору впорядкування початкової ДДНФ і може мати різну складність, тобто різний коефіцієнт простоти.

3.9.2. Утворення тупикових і мінімальних диз'юнктивних нормальних форм

Черговий етап спрощення перемикальної функції полягає в пошуку тупикових, а потім мінімальних форм. Можна навести ще одне означення тупикової ДНФ.

Означення 3.33. *Диз'юнкція простих імплікант, жодну з яких виключити не можна, є тупиковою ДНФ заданої перемикальної функції. Деякі функції мають кілька тупикових форм. Тупикові форми, що містять найменшу кількість символів, будуть мінімальними.*

Теорема 3.13. *Будь-яка мінімальна ДНФ перемикальної функції є тупиковою.*

Доведення (від супротивного). Нехай мінімальна ДНФ функції $f(x_1, x_2, ..., x_n) = K_1 + K_2 + ... + K_r$, де $K_1, K_2, ..., K_r$ — елементарні кон'юнкції (прості імпліканти).

Припустимо, що K_1 — непроста імпліканта. Тоді $K_1 = \alpha K_1'$, де K_1' — залишкові елементарні добутки, а α — проста імпліканта.

Тому $A = \alpha + K_1 + K_2 + ... + K_r = \alpha + \alpha K_1' + K_2 + ... + K_r$. Оскільки проста імпліканта поглинає елементарні твори K_1, дістаємо $A = \alpha + K_2 + ... + K_r$.

Цей вираз містить менше число символів, ніж початковий, що суперечить мінімальності ДНФ. Отже, припущення, що K_1 — непроста імпліканта, є неправильним. Тому всі члени мінімальної форми перемикальної функції — прості імпліканти.

Означення 3.34. *Для утворення мінімальної ДНФ досить знайти всі тупикові форми заданої функції і вибрати з них мінімальні.*

Існує кілька методів пошуку тупикових форм перемикальної функції (найшвидшого спуску, імплікантних матриць тощо). Розглянемо метод імплікантних матриць на двох таких прикладах.

Приклад. Знайти мінімальну ДНФ, якщо ДДНФ має вигляд

$$f(a,b,c,d) = \bar{a}bcd + \bar{a}\bar{b}cd + a\bar{b}cd + a\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}\bar{d}.$$

Будуємо імплікантну матрицю у вигляді таблиці, у вертикальні й горизонтальні входи якої записуємо всі конституенти одиниці та всі прості імпліканти заданої функції.

Для кожної імпліканти знаходимо конституенти одиниці, що поглинаються, тобто ті конституенти, власною частиною яких є задана імпліканта. Наприклад, $\bar{a}cd$ поглинає дві конституенти: $\bar{a}bcd$ й $\bar{a}\bar{b}cd$. Клітинки імплікантної матриці, утворені перетином рядків з імплікантами і колонок із поглинанням, будемо позначати косим хрестиком (x).

Таблиця 3.16

№ пор.	Прості імпліканти	Конституенти одиниці					
		$\bar{a}bcd$	$\bar{a}\bar{b}cd$	$a\bar{b}cd$	$a\bar{b}\bar{c}d$	$a\bar{b}\bar{c}\bar{d}$	$\bar{a}\bar{b}\bar{c}\bar{d}$
		1	2	3	4	5	6
1	$\bar{a}cd$	x	x				
2	$\bar{b}cd$		x	x			
3	$a\bar{b}d$			x	x		
4	$a\bar{b}\bar{c}$				x	x	
5	$\bar{b}\bar{c}\bar{d}$				x	x	

Щоб знайти мінімальну ДНФ функції, досить визначити мінімальне число імплікант, які спільно накривають усі конституенти, тобто всі колонки імплікантної матриці (табл. 3.16):

$\bar{a}cd$ — накриває 1 і 2; $a\bar{b}d$ — накриває 3 та 4; $\bar{b}\bar{c}\bar{d}$ — накриває 5 і 6. Тоді форма є $f = \bar{a}cd + a\bar{b}d + \bar{b}\bar{c}\bar{d}$ тупиковою. Вона ж — мінімальна.

Приклад. Знайти мінімальну ДНФ, якщо ДДНФ має вигляд

$$f(a, b, c) = \underset{1}{\bar{a}bc} + \underset{2}{\bar{a}\bar{b}c} + \underset{3}{abc} + \underset{4}{ab\bar{c}} + \underset{5}{a\bar{b}\bar{c}} + \underset{6}{\bar{a}\bar{b}\bar{c}}.$$

Виконавши всі операції склеювання та поглинання, дістанемо:

$$1-2 = \bar{a}c \quad (\text{по } b);\ 1-3 = bc\ (\text{по } a);$$

$$2-6 = \bar{a}\bar{b}\ (\text{по } c);.\ 3-4 = ab\ (\text{по } c);$$

$$4-5 = a\bar{c}\ (\text{по } b);\ 5-6 = \bar{b}\bar{c}\ (\text{по } a).$$

Отже, $f(a, b, c) = \bar{a}c + bc + \bar{a}\bar{b} + ab + a\bar{c} + \bar{b}\bar{c}$ — скорочена ДНФ.
Складаємо імплікантну матрицю (табл. 3.17). З неї

Таблиця 3.17

№ пор.	Прості імпліканти	Конституенти одиниці					
		$\bar{a}bc$	$\bar{a}\bar{b}c$	$a\bar{b}c$	$a\bar{b}\bar{c}$	$a\bar{b}\bar{c}$	$\bar{a}\bar{b}\bar{c}$
		1	2	3	4	5	6
1	$\bar{a}c$	x	x				
2	bc	x		x			
3	$\bar{a}\bar{b}$		x				
4	ab			x	x		
5	$a\bar{c}$				x	x	
6	$\bar{b}\bar{c}$					x	x

випливає, що:

- перша тупикова ДНФ $f(a, b, c) = \bar{a}c + ab + \bar{b}\bar{c}$;
- друга тупикова ДНФ $f(a, b, c) = bc + \bar{a}\bar{b} + a\bar{c} + \bar{b}\bar{c}$.

Отже, перша тупикова ДНФ є мінімальною.

Таким чином, алгоритм утворення мінімальної ДНФ зводиться ди виконання таких трьох дій:

1. Перемикальну функцію подають у ДДНФ.
2. У ДДНФ проводять усі операції неповного склеювання і поглинання. Одержують скорочену ДНФ.
3. Знаходять мінімальні ДНФ за імплікантною матрицею.

3.10. Утворення мінімальних кон'юнктивних нормальних форм

Щоб знайти вираз заданої перемикальної функції, найзручніший для синтезу автомата, потрібно мати також мінімальні КНФ і вибрати з них ту, реалізація якої найпростіша.

Аналогічно з викладеним вище обмежимося формулюванням алгоритму утворення мінімальної КНФ:

1. Перемикальну функцію подають у ДКНФ, тобто у вигляді добутку конституент нуля:

- якщо функцію задано таблицею, то її записують «за нулями» ДКНФ;
- якщо функцію задано довільною формою, то застосовують формулу розгортання $x = (x+y)(x+\overline{y})$, $(x+y) = (x+y+z)(x+y+\overline{z})$. Виконують спочатку всі операції неповного склеювання, а потім операції поглинання, які визначаються виразами $(x+y)(x+\overline{y}) = x(x+\overline{y})(x+y)$ та $x(x+y) = x$ відповідно.

2. Знаходять мінімальні КНФ, використовуючи імплікантні матриці, тобто таблиці, у верхній рядок яких записують конституенти нуля, що входять у ДКНФ, а в ліву частину — члени, що входять у скорочену КНФ. Клітинки імплікантної матриці на перетині колонки з конституентою нуля і рядка з членом, який її поглинає, позначають косим хрестиком. Мінімальні КНФ записують у вигляді добутків, що містять найменшу кількість членів, які перекривають хрестиками всі колонки імплікантної матриці.

Приклад. Знайти мінімальну КНФ функції

$$f(a,b,c) = \underset{1}{(a+\overline{c})}\underset{2}{(\overline{a}+\overline{b}+c)}\underset{3}{(b+c)}\underset{4}{(\overline{a}+\overline{b}+\overline{c})}\ .$$

1. Утворюємо ДКНФ, для чого застосовуємо операцію розгортання до членів 1 і 3:

$$(a+\overline{c}) = (a+b+\overline{c})(a+\overline{b}+\overline{c});$$

$$(b+c) = (a+b+c)(\overline{a}+b+c);$$

$$f(a,b,c) = \underset{1}{(a+b+\overline{c})}\underset{2}{(a+\overline{b}+\overline{c})}\underset{3}{(\overline{a}+\overline{b}+c)}\underset{4}{(a+b+c)}\underset{5}{(\overline{a}+b+c)}\underset{6}{(\overline{a}+\overline{b}+\overline{c})}.$$

2. Виконуємо всі операції склеювання та поглинання:

$$1-2 = a+\overline{c} \text{ (по } b); \ 1-4 = a+b \text{ (по } c); \ 2-6 = \overline{b}+\overline{c} \text{ (по } a);$$

$$3-5 = \overline{a}+c \text{ (по } b); \ 3-6 = \overline{a}+\overline{b} \text{ (по } c); \ 4-5 = b+c \text{ (по } a).$$

Одержані члени ($n = 2$) між собою не склеюються, тому скорочена КНФ має вигляд

$$f(a,b,c) = (a+\bar{c})(a+b)(\bar{b}+\bar{c})(\bar{a}+c)(\bar{a}+\bar{b})(b+c) .$$

3. Будуємо імплікантну матрицю (табл. 3.18).

Таблиця 3.18

Прості імпліканти	Конституенти нуля					
	$a+b+\bar{c}$	$a+\bar{b}+\bar{c}$	$\bar{a}+\bar{b}+c$	$a+b+c$	$\bar{a}+b+c$	$\bar{a}+\bar{b}+\bar{c}$
	1	2	3	4	5	6
$a+\bar{c}$	x	x				
$a+c$				x		
$\bar{b}+\bar{c}$		x				x
$\bar{a}+c$			x		x	
$\bar{a}+\bar{b}$			x			x
$b+c$				x	x	

Маємо дві мінімальні КНФ: $f(a,b,c) = (\bar{a}+c)(\bar{a}+\bar{b})(b+c)$; $f(a, b, c) = (a+b)(\bar{b}+\bar{c})\times$ $\times(\bar{a}+c)$.

3.11. Метод мінімізаційних карт (діаграми Карно — Вейча)

Діаграми Карно — Вейча — це спеціальні таблиці, що використовуються для задання перемикальних функцій і дають змогу спростити процес пошуку мінімальних форм.

Процес пошуку (склеювання термів) реалізується таким чином, що визначають терми, відмінні тільки одним символом, який в один із термів входить із запереченням, а в інший — без, і далі виконують спрощення згідно з тотожністю

$$Ax \vee A\bar{x} = A(x+\bar{x}) = A .$$

Можна склеювати чотири терми:

$$A(\bar{x}_1\bar{x}_2 + \bar{x}_1x_2 + x_1\bar{x}_2 + x_1x_2) = A ,$$

а також будь-яке число 2^k термів, відмінних один від одного тільки k символами, що входять у ці терми. Практичне застосування склеювання ускладнюється тим, що не завжди вдається помітити всі можливості подібного склеювання безпосередньо з виразів мінімізаційних функцій. Тому дуже корисними при формальному проектуванні виявляються діаграми Карно — Вейча, наочність яких можна порівняти з наочністю графіків для об'єктних функцій. Діаграми Карно — Вейча мають вигляд розгортки кубів на площині.

Усупереч поширеній думці ми переконані, а практика це підтверджує, що діаграми Карно — Вейча можуть застосовуватись при мінімізації функцій, які залежать не тільки від малого числа змінних. Є досвід успішного використання діаграм для 10 змінних і, мабуть, це не межа. Нижче наведено ці діаграми для двох, трьох, чотирьох, п'яти та семи змінних. Аналогічно можуть бути побудовані діаграми для більшого числа змінних.

На діаграмах члени, що склеюються, знаходяться в сусідніх клітинках. При цьому дві сусідні клітинки утворюють 1-куб, чотири — 2-куб, вісім клітинок — 3-куб, 16 — 4-куб і т. д.

Діаграма для чотирьох змінних показує приклад утворення 4- та 2-куба.

Спочатку функцію подають в одній з двох досконалих форм: ДДНФ або ДКНФ. Далі шукають 1-, 2-, 3-куби тощо, починаючи зі старшого, тобто спочатку 3-куб, потім 2-куб і т. д. Чим вищий номер куба, тим простіший терм, оскільки він містить менше змінних (більше змінних виключається).

Діаграма Карно — Вейча для двох змінних має вигляд

$\bar{x}_1\bar{x}_2$	$\bar{x}_1 x_2$	$x_1 x_2$	$x_1\bar{x}_2$
00_0	01_1	11_3	10_2

, або

	x_1	$\bar{x}_1$
x_2	11_3	01_1
$\bar{x}_2$	10_2	00_0

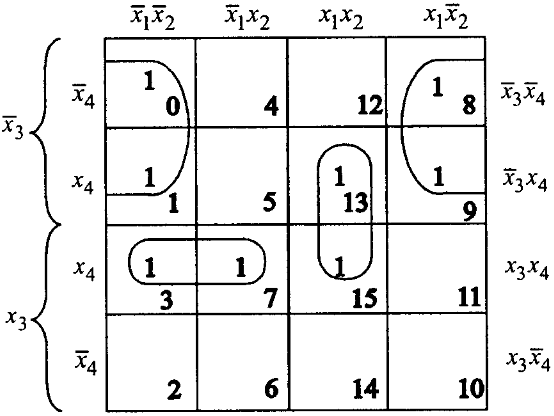
.

Діаграма Карно — Вейча для трьох змінних набуває

	$\bar{x}_2\bar{x}_3$	$\bar{x}_2 x_3$	$x_2 x_3$	$x_2\bar{x}_3$
$\bar{x}_1$	000_0	001_1	011_3	010_2
x_1	100_4	101_5	111_7	110_6

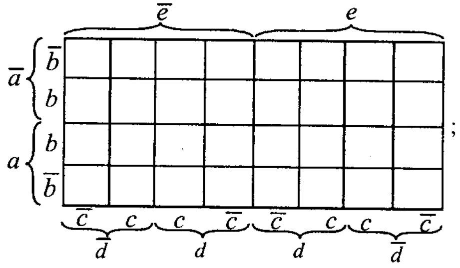
;

для чотирьох змінних — вигляду



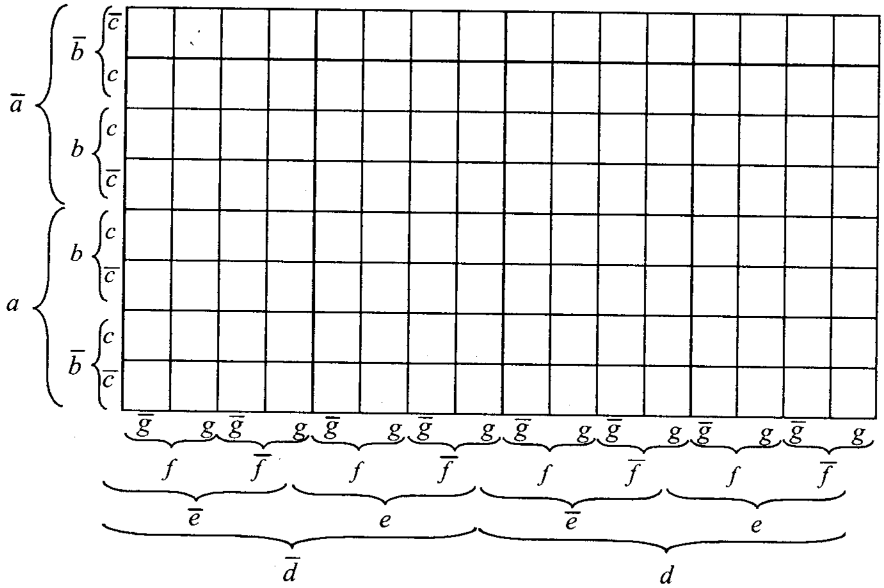
;

для п'яти змінних — вигляду



;

для семи змінних — вигляду



Наприклад, на діаграмі для чотирьох змінних початкова ДДНФ має вигляд

$$f_{ДДНФ} = \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3 x_4 + \bar{x}_1\bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 x_3 x_4 + x_1 x_2 \bar{x}_3 x_4 + \\ + x_1 x_2 x_3 x_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3 x_4$$

і містить вісім конституент одиниці.

На діаграмі для цієї функції маємо два 1-куби й один 2-куб. Дістаємо мінімальну ДНФ

$$f_{\min} = \bar{x}_1 x_3 x_4 + x_1 x_2 x_4 + \bar{x}_2\bar{x}_3 ,$$

що містить три терми. Як бачимо, метод досить ефективний. Він дає змогу швидко знайти мінімальну ДНФ.

3.12. Метод мінімізації Блейка — Порецького

Недоліком розглянутих вище методів мінімізації логічних функцій є те, що для їх застосування необхідно, щоб функція, яка мінімізується, була спочатку подана в ДДНФ. Процес такого подання функції з великим числом аргументів часто є надто трудомістким. Наприклад, якщо функцію f задано у вигляді ДНФ $f(x_1, x_2, x_3, x_4, x_5) = x_1x_2x_3x_5 + x_2x_3x_4x_5 + \overline{x}_1$ і потрібно мінімізувати її, то спочатку треба утворити ДДНФ цієї функції, що складається з 19 мінтермів.

Бажано знайти можливість побудови скороченої ДНФ не за ДДНФ, а за довільною ДНФ функції. Ідею такої побудови розглянуто у працях А. Блейка і П. С. Порецького. Вона випливає з такої теореми.

Теорема 3.14. *Якщо в ДНФ функції $f(x_1, x_2, ..., x_n)$ входять дві кон'юнкції Ax_i та $B\overline{x}_i$, то справджується рівність*

$$F = P + AB, \tag{3.5}$$

де P — ДНФ, еквівалентна функції f.

Доведення. Треба показати, що ліва і права частини рівності (3.5) набувають однакових значень у всіх можливих наборах значень аргументів. Розглянемо довільний набір $(x_1, x_2, ..., x_n)$, для якого $f(x_1, x_2, ..., x_n) = 1$.

Внаслідок еквівалентності між f та P справджується рівність, що випливає з (3.5):

$$1 = 1 + AB.$$

Ця рівність тотожна, оскільки в диз'юнкції, записаній праворуч, є одиниця.

Розглянемо тепер довільний набір значень аргументів, для яких $f(x_1, x_2, ..., x_n) = 0$. На підставі (3.5) маємо

$$0 = 0 + AB.$$

Ця рівність може виконуватися лише за умови, що $AB = 0$. Таким чином, треба довести, що в наборах значень аргументів, у яких функція перетворюється на нуль, обов'язково перетворюється на нуль A або B чи одночасно A і B. ДНФ P за твердженням теореми має вигляд

$$P = S + Ax_i + B\overline{x}_i.$$

Оскільки на наборах, що розглядаються, $P = 0$, мають одночасно виконуватися такі три рівності:

$$S = 0,\ Ax_i = 0;\ B\overline{x}_i = 0.$$

Із двох останніх рівностей випливає, що або A, або B, або одночасно A і B перетворюються на нуль. Оскільки A та B одночасно на нуль не перетворюються, теорему доведено.

Легко перевірити, що доповнення ДНФ відповідно до доведеної теореми після елементарного поглинання приводить до скороченої ДНФ.

Після побудови скороченої ДНФ можна використати метод Квайна, починаючи з другого етапу (тобто з побудови таблиці міток).

Приклад. Знайти скорочену ДДНФ функції

$$f(x_1, x_2, x_3) = x_1x_2 + x_1x_2\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 + \overline{x}_2\overline{x}_3 + x_1\overline{x}_3$$

Для цієї функції існує дві пари кон'юнкцій, що задовольняють умови теореми: $(x_1x_2\overline{x}_3; x_1\overline{x}_2\overline{x}_3)$ й $(x_1x_2; \overline{x}_2\overline{x}_3)$. Тому

$$f(x_1,x_2,x_3) = x_1x_2 + x_1x_2\overline{x}_3 + x_1\overline{x}_2\overline{x}_3 + \overline{x}_2\overline{x}_3 + x_1\overline{x}_2 + x_1\overline{x}_3.$$

Після елементарного поглинання дістаємо скорочену ДНФ

$$f(x_1,x_2,x_3) = x_1x_2 + \overline{x}_2\overline{x}_3 + x_1\overline{x}_2 + x_1\overline{x}_3.$$

3.13. Геометричне подання функцій алгебри логіки

3.13.1. Загальні положення і постановка задачі в геометричній формі

Багато перетворень, що виконуються над перемикальними функціями, зручно інтерпретуються за допомогою геометричного подання функцій. Нехай E^n — множина всіх можливих наборів $(x_1, x_2, ..., x_n) : E^n = \{(x_1, x_2, ..., x_n)\}$, де $x_i = 0 \vee 1$.

Цю множину можна розглядати як множину всіх вершин одиничного *n-вимірного* куба. Оскільки ніяких інших, крім згаданих, точок не розглядаємо, множину E^n будемо називати *n-вимірним* кубом, а набори $(x_1, x_2, ..., x_n)$ — його вершинами. Так, функцію двох змінних можна інтерпретувати за допомогою квадрата, заданого в системі координат x_1, x_2.

Відклавши на кожній осі одиничні відрізки x_1 та x_2, дістанемо квадрат, вершини якого відповідають комбінаціям змінних, — геометричне подання функції двох змінних (рис. 14).

Із такого геометричного подання функції двох змінних випливає, що дві вершини, які належать одному і тому самому ребру, називаються сусідніми, причому вони «склеюються»за змінною, що міняється вздовж цього ребра:

$$x_1x_2 \vee \overline{x}_1x_2 = x_2(x_1 + \overline{x}_1) = x_2.$$

Для функції трьох змінних геометричне подання — це куб (рис. 15), вершини якого позначено десятковими цифрами, двійковими цифрами та довільними змінними x. Ребра куба поглинають вершини, грані — свої ребра, а отже, вершини.

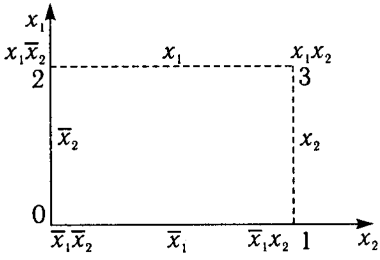


Рис. 14

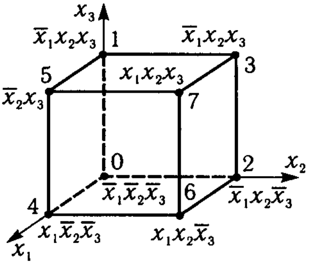


Рис. 15

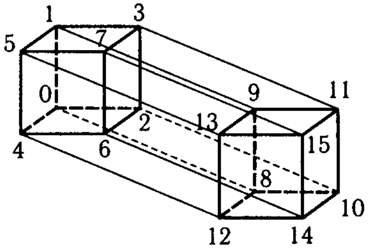


Рис. 16

Загалом сукупність векторів $(x_1, x_2, ..., x_n)$ відображається на множину вершин n-вимірного куба. Всі такі вершини утворюють логічний простір.

Булева функція відображається на *n-вимірному* кубі вилученням вершин, що відповідають векторам $(x_1, x_2, ..., x_n)$, на яких функція $y = f(x_1, x_2, \ldots, x_n)$ набуває значення 1. Такі значення позначаються точками.

У геометричному сенсі кожний набір $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$, тобто кожна вершина, може розглядатися як n-вимірний вектор, що визначає точку n-вимірного простору. Тому вся множина наборів, на яких визначено функцію n змінних, подається у вигляді вершин n-вимірного куба.

Координати вершин куба мають бути зазначені в порядку, що відповідає порядку переліку змінних у запису функції. Геометричне подання може використовуватися під час розробки методів мінімізації із застосуванням мінімізаційних карт.

Функція чотирьох змінних геометрично подається у вигляді чотиривимірного куба (рис. 16), функція п'яти змінних — у вигляді п'ятивимірного куба (рис. 17) і т. д.

Означення 3.35. *Нехай* $\sigma_{i_1}, \sigma_{i_2}, ..., \sigma_{i_n}$ *— фіксований набір чисел з* 0 *та* 1 *такий, що* $1 < i_1 < i_2 < ... < i_r < n$. *Множина всіх вершин* $(\alpha_1, \alpha_2, ..., \alpha_n)$ *куба* E^n *таких, що* $(\alpha_1, \alpha_2, ..., \alpha_n) \subset E^n$ *й* $\alpha_{i_1} = \sigma_{i_1}$, $\alpha_{i_2} = \sigma_{i_2}, ..., \alpha_{i_r} = \sigma_{i_r}$, *називається* $(n - r)$*-вимірною гранню.*

Очевидно, що $(n - r)$-вимірна грань є $(n - r)$-вимірним підкубом куба E^n (частиною куба E^n).

Нехай $y = f(x_1, x_2, ..., x_n)$ — функція алгебри логіки. Складемо підмножину N_i вершин куба E^n так, що

$$(\alpha_1, \alpha_2, ..., \alpha_n) \in N_1$$

тоді й тільки тоді, коли

$$f(\alpha_1, \alpha_2, ..., \alpha_n) = 1.$$

Зрозуміло, що за підмножиною N_i початкова функція відновлюється однозначно.

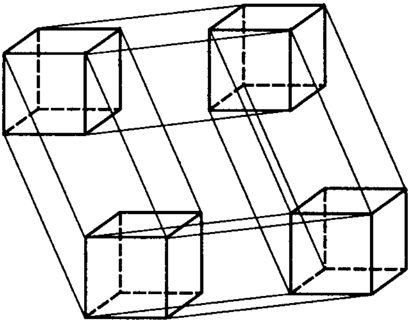


Рис. 17

Приклад. Задано функцію $f(x_1, x_2,...,x_n)$ (табл. 3.19). Цій функції відповідає множина мінтермів $N = \{(000), (100), (101), (110), (111)\}$. Геометрично це може бути подано у вигляді, показаному на рис. 14.

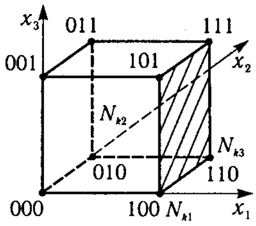


Рис. 18

Таблиця 3.19

$x_1x_2x_3$	$f(x_1x_2x_3)$	$x_1x_2x_3$	$f(x_1x_2x_3)$
000	1	100	1
001	0	101	1
010	0	110	1
011	0	111	1

Функції відповідають грані:

$$N_{k1} = \{(0, 0, 0), (1, 0, 0)\};$$

$$N_{k2} = \{(1, 0, 0), (1, 0, 1), (1, 1, 1), (1, 1, 0)\},$$

які мають ранги 1 і 2. Ці грані (рис. 18) є відповідно одновимірною гранню (ребром) та двовимірною гранню (площиною). Грані N_{k1} й N_{k2} розташовуються всередині множини

$$N = N_{k1} \cup N_{k2}.$$

3.13.2. Деякі властивості геометричного подання функцій

Якщо функція $f(x_1, x_2, ..., x_n) = g(x_1, ..., x_n) \vee h(x_1, x_2, ..., x_n)$ має дві складові, то:

1. $N_g \subset N_f$ і $N_h \subset N_f$ (кожна складова належить функції).
2. $N_f = N_g \cup N_n$ (об'єднання множин).

Якщо ДНФ функції $f(x_1, x_2, ..., x_n)$ має вигляд $A_{ДНФ} = K_1 \vee K_2 \vee ... \vee K_S$, де K_i конституенти 1, то

$$N_{k_i} \subset N_f (i = 1, 2, ..., s), N_{k_1} \subset N_f, ..., N_{k_s} \subset N_f,$$

тобто кон'юнкція є гранню, розташованою всередині множини N_f, причому $N_f = N_{k_1} \cup N_{k_2} ... \cup N_{k_s}$ — об'єднання всіх множин.

Отже, ДНФ функції відповідає покриття множини N_f гранями $N_{k1}, N_{k2}, ..., N_{ks}$.

Правильним є й обернене твердження: усякому покриттю множини N_f гранями N_{k_i}, розташованими всередині множини N_f, відповідає ДНФ функції $f(x_1, x_2, ..., x_n)$.

Нехай r_i — ранг грані N_{k_i} (він дорівнює рангу кон'юнкції K_i). Тоді рангом покриття називатимемо суму

$$r = \sum_{i=1}^{s} r_i \cdot$$

Тепер можна дати формулювання геометричної задачі (задачі про покриття): для заданої множини N знайти таке покриття гранями, які належать $N_f = N_{k_1} \cup N_{k_2} ... \cup N_{k_s}$, щоб ранг був найменшим.

Ця задача еквівалентна задачі про мінімізацію булевої функції в геометричній формі.

Отже, задача мінімізації булевих функцій має дві постановки:

- в аналітичній формі;
- в геометричній формі (задача про покриття).

3.13.3. Геометричне подання скороченої диз'юнктивної нормальної форми

Означення 3.36. *Грань N_k, що міститься в N_f, називається максимальною відносно N_f, якщо не існує грані N'_k такої, що:*

1) $N_k \subset N_k \subset N_f$;

2) *вимірність грані N'_k більша від вимірності грані N_k.*

Приклад. Функції, задані табл. 3.19 і показані на рис. 18, мають вигляд

$$K_1(x_1, x_2, x_3) = x_2 x_3, \quad N_{k1} = \{(0,0,0),(1,0,0)\};$$
$$K_2(x_1, x_2, x_3) = x_1 x_2, \quad N_{k2} = \{(1,0,0),(1,0,1)\};$$
$$K_3(x_1, x_2, x_3) = x_1, \quad N_{k3} = \{(1,0,0),(1,0,1),(1,1,0),(1,1,1)\}.$$

Тоді грані N_{k1} та N_{k3} є максимальними, а грань N_{k2} — не максимальна для N_f, тому що $N_{k2} \subset N_{k3}$ і вимірність N_{k3} більша за вимірність N_{k2}.

Означення 3.37. *Кон'юнкція K, яка відповідає максимальній грані N_k множини N_f, називається простою імплікантою функції f.*

Оскільки умова $N_k \subset N'_k$ еквівалентна тому, що всі співмножники з K' містяться в K, із простої імпліканти K функції f не можна вилучити жодного співмножника, інакше дістанемо (після вилучення співмножника) кон'юнкцію K', для якої $N'_k \not\subset N_f$. Кожну грань N_k, що входить в N_f і $N'_k \subset N_f$, можна розширити до максимальної грані.

Нехай максимальними гранями функції $f(x_1, x_2, ..., x_n)$ є N_{k_1}, $N_{k_r}, ..., N_{k_m}$. Тоді $N_f = N_{k_1} \cup N_{k_r} \cup ... \cup N_{k_m}$, тому що $N_{k_i} \subset N_f \ (i = 1, ..., m)$, і кожна точка з N_f належить деякій максимальній грані. Остання рівність еквівалентна такій:

$$f = K_1 \vee K_2 \vee ... \vee K_m.$$

Означення 3.38. *ДНФ, що є диз'юнкцією всіх простих імплікант функції, називається скороченою ДНФ: $A_{\text{скор.ДНФ}} = K_1 \vee K_2 \vee ... \vee K_m$. Ця скорочена ДНФ однозначно визначає та реалізує функцію f.*

Для функції $f(x_1, x_2, x_3)$, заданої табл. 3.16 і показаної на рис. 18, маємо $K_1 = \overline{x}_2 \overline{x}_3$, $K_2 = x_1 \overline{x}_2$, $K_3 = x_1$.

Тоді для цієї функції характерним є покриття $N_f = N_{k_1} \cup N_{k_3}$. Йому відповідає скорочена ДНФ $A_{\text{скор}} = \overline{x}_2 \overline{x}_3 + x_1$.

Бачимо, що геометричний підхід водночас дає спосіб побудови скороченої ДНФ. Однак бажано мати також аналітичну форму функції.

3.13.4. Тупиковість на основі геометричних подань

Означення 3.39. *Покриття множини N_f, що складається з максимальних (відносно N_f) граней, називається незвідним, якщо сукупність граней, утворена з початкової викиданням будь-якої грані, не буде покриттям N_f.*

Означення 3.40. *ДНФ, що відповідає незвідному покриттю множини, називається тупиковою (в геометричному сенсі).*

Приклад. Задано функцію (табл. 3.20), для якої

$$N_f = N_{\overline{X}_3} \cup N_{\overline{X}_1} \cup N_{X_1X_2}$$

є незвідним покриттям, а $A_{\text{скор.ДНФ}} = \overline{x}_3 + \overline{x}_1 + x_1x_2$ — тупиковою ДНФ в геометричному сенсі (рис. 19).

Таблиця 3.20

$x_1x_2x_3$	$f(x_1, x_2, x_3)$	$x_1x_2x_3$	$f(x_1, x_2, x_3)$
000	1	100	1
001	1	101	0
010	1	110	1
011	1	111	1

Поняття тупикової ДНФ щодо перетворень 1 і 2 (див. п. 3.9.1) і тупикової ДНФ в геометричному сенсі є еквівалентними.

Тупикова ДНФ утворюється зі скороченої вилученням деяких членів. Мінімальна ДНФ (відносно L) є тупиковою. Серед тупикових ДНФ є також мінімальна ДНФ (відносно L).

Приклад. Розглянемо утворення тупикової форми для функції, заданої у вигляді табл. 3.21.

Таблиця 3.21

$x_1x_2x_3x_4$	$f(x_1x_2x_3x_4)$	$x_1x_2x_3x_4$	$f(x_1x_2x_3x_4)$
0100	0	1110	0
0110	0	На інших	
1001	0	ребрах	1
1110	0		

Ця функція геометрично подається у вигляді чотиривимірного куба (рис. 20). Вершини, позначені точками, відповідають конституентам одиниці.

Множина N_f має такі максимальні грані: N_1, N_2, N_3, N_4 — двовимірні; N_5, N_6, N_7 — ребра (рис. 21).

Скорочена ДНФ відповідає покриттю

$$N_f = N_1 \cup N_2 \cup N_3 \cup N_4 \cup N_5 \cup N_6 \cup N_7.$$

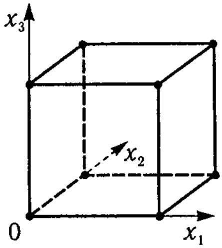


Рис. 19

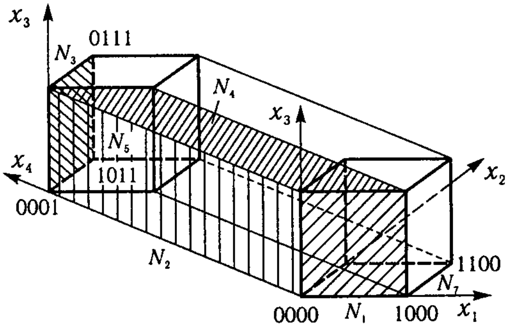


Рис. 20

Грані N_3 і N_4 завжди входять у будь-яке покриття, тобто в будь-яку мінімальну ДНФ, оскільки тільки вони відповідають точкам 0111 та 1011.

Розглянемо варіанти тупикових ДНФ для заданої функції.

Грань N_1 є доповненням до вже вибраних граней N_3 й N_4. Залишається покрити дві точки (вершини): 1100 та 1101. Для цього можна взяти ребро N_6 (з'єднує точки 1101 і 1100) або два ребра N_5 й N_7. Отже, маємо дві тупикові ДНФ:

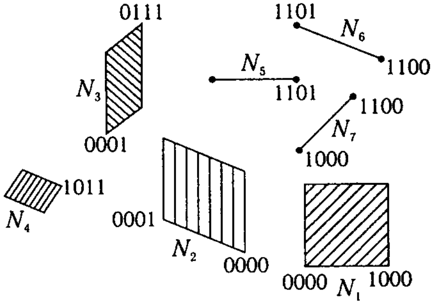


Рис. 21

а) $N_1 \cup N_3 \cup N_4 \cup N_5 \cup N_7$;

б) $N_1 \cup N_3 \cup N_4 \cup N_6$.

Візьмемо грань N_2. Залишаються непокритими точки 1000, 1100, 1101. Для покриття можна взяти або ребра N_5 та N_7, або N_6 й N_7. Маємо ще дві тупикові ДНФ:

в) $N_2 \cup N_3 \cup N_4 \cup N_5 \cup N_7$;

г) $N_2 \cup N_3 \cup N_4 \cup N_6 \cup N_7$.

Отже, всього є чотири тупикові ДНФ. Кожній грані відповідають прості імпліканти:

$$(N_1)K_1 = x_2\bar{x}_4;\ (N_2)\,K_2 = x_1x_2;\ (N_3)\,K_3 = x_1x_4;$$

$$(N_4)\,K_4 = x_2x_3;\ (N_5)K_5 = x_2x_3x_4;\ \left(N_6\right)K_6 = x_1x_2x_3;$$

$$(N_7\ )K_7 = x_1x_3\bar{x}_4.$$

Використовуючи коефіцієнти простоти, переконуємося, що варіант б) дає мінімальну ДНФ.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке висловлення?
2. Що таке булеві змінні?
3. Які є способи задання перемикальних функцій?
4. Яка сфера застосування булевих функцій?
5. Які основні властивості булевих функцій?
6. Що таке формула в алгебрі логіки?
7. Що таке рівносильність формул?

8. Як формулюється основна тотожність алгебри логіки?
9. Як формулюється принцип двоїстості?
10. Які приклади ілюструють повні системи?
11. Які є типи булевих функцій?
12. Як формулюється теорема про повноту системи функцій?
13. Для чого використовуються диз'юнктивні нормальні форми перемикальних функцій?
14. Коли застосовуються кон'юнктивні нормальні форми перемикальних функцій?
15. Що таке ДДНФ і ДКНФ?
16. Що таке мінімальні диз'юнктивні та кон'юнктивні форми?
17. Що таке скорочена диз'юнктивна нормальна форма перемикальних функцій?
18. Що таке скорочена кон'юнктивна нормальна форма перемикальних функцій?
19. Як використовується метод Квайна?
20. Як використовується метод Квайна — Маккласкі?
21. Що таке тупикові форми перемикальних функцій?
22. Як утворюють мінімальні форми за допомогою діаграм Карно — Вейча?
23. У чому суть методу Блейка — Порецького?
24. Які особливості геометричного подання перемикальних функцій?
25. Які особливості задачі мінімізації в геометричній формі?

ПЕРЕЛІК ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

1. Висловлення. Просте й складне висловлення.
2. Булеві функції. Таблиці істинності.
3. Закони логіки. Тотожно-істинне висловлення. Проблема розв'язуваності.
4. Алгебра Жегалкіна.
5. Нормальні форми булевих функцій. Досконалі ДНФ і КНФ булевих функцій.
6. Мінімізація булевої функції методом Квайна.
7. Мінімізація булевої функції методом Карно — Вейча.
8. Мінімізація булевої функції методом Маккласкі.
9. Мінімізація булевої функції методом Блейка — Порецького.
10. Мінімізація булевої функції геометричним методом.

СПИСОК ЛІТЕРАТУРИ

1. *Яблонский С. В.* Введение в дискретную математику. — М.: Наука, 1989. — 384 с.
2. *Нефедов В. Н., Осипова В. А.* Курс дискретной математики. — М.: Изд-во МАИ, 1992. — 264 с.
3. *Глушков В. М.* Синтез цифровых автоматов. — М.: Наука, 1962. — 476 с.
4. *Глушков В. М., Капитонова Ю. В., Мищенко А. Т.* Логическое проектирование дискретных устройств. — К.: Наук. думка, 1987. — 264 с.
5. *Савельев А. Я.* Прикладная теория цифровых автоматов. — М.: Высш. шк., 1987. — 272 с.
6. *Вавилов Е. Н., Портной Т. П.* Синтез схем электронных цифровых машин. — М.: Сов.радио, 1965. — 316 с.
7. *Подлипенский В. С.* Бесконтактные логические схемы автоматики. — К.: Наук. думка, 1965. — 238 с.
8. *Новиков П. С.* Элементы математической логики. — М.:Наука, 1973. — 399 с.
9. *Гиндикин С. Г.* Алгебра логики в задачах. — М.: Наука, 1972. — 288 с.

РОЗДІЛ 4

ВІДНОШЕННЯ

4.1. Поняття та основні властивості відношень

У наведеному вище означенні множини Н. Бурбакі йшлося про те, що елементи множини можуть знаходитися в деяких відношеннях між собою або з елементами інших множин, але означення відношення не давалося. Загалом відношення означає який-небудь зв'язок між предметами або поняттями. Відношення між парами об'єктів називаються бінарними (двомісними).

Приклади бінарних відношень:

- відношення належності;
- включення множин;
- рівність дійсних чисел;
- нерівності (натуральних, дійсних чисел);
- бути братом;
- ділитися на яке-небудь натуральне число;
- входити до складу якого-небудь колективу.

Для будь-якого бінарного відношення можна записати відповідне йому співвідношення. Для наведених відношень це, наприклад:

- $a \in A$;
- $A \subset B$;
- $a = b$; a, $b \in R$;
- $a > b$; a, $b \in N$;
- Іван — брат Петра;
- 4 ділиться на 2;
- Чернов — староста групи 1ПР.

Загалом відношення записується у вигляді співвідношень xAy (зверніть увагу, що строгого означення відношення ми ще не дали), де A — відношення, яке встановлює зв'язок між елементом $x \in X$ і елементом $y \in Y$. Зрозуміло, що відношення повністю визначається множиною всіх пар елементів (x, y), для яких воно справджується. Тому будь-яке бінарне відношення можна розглядати як множину впорядкованих пар (x, y), тобто можна дати таке означення бінарного відношення.

Означення 4.1. *Бінарним відношенням A, що діє з множини X у множину Y, називається деяка підмножина* $X \times Y \left(A \subset X \times Y\right)$.

Отже, бінарне відношення встановлює відповідність елементів множини X елементам множини Y. Зрозуміло, що співвідношення xAy можна записати у вигляді $(x, y) \in A$, де $A \subset X \times Y$. Наприклад, $3 < 7$ еквівалентно $(3, 7) \in$ « < » (але не можна записати $(7, 3) \in$ « < »).

Елемент x називають першою координатою, а елемент y — другою координатою впорядкованої пари. D_0 (A) — множина перших координат — називається областю значень (лівою областю) відношення A. $D_3(A)$ — множина других координат — називається областю значень (правою областю) відношення A. Якщо $x \in X$, $y \in Y$, тоді $D_0(A) \subset X$, $D_3(A) \subset Y$. У таких випадках кажуть, що A є відношенням від X до Y. Його називають також відповідністю та позначають $X \to Y$. Якщо $Y = X$, то будь-яке відношення $A : X \to Y$ є підмножиною $X \times X$ і називається відношенням в X.

Приклад. $X = \{2, 3\}$, $Y = \{3, 4, 5, 6\}$.

Нехай A — відношення «бути дільником» від X до Y. Тоді $A = \{(2, 4), (2, 6), (3, 3), (3, 6)\}$. Відношення B — « = » від X до Y: $B = \{(3, 3)\}$; відношення C — « > » від X до Y: $C = \varnothing$. $D_0(A) = \{2, 3\} = X$, $D_3(A) = \{3, 4,6\} \subset Y$.

Якщо D_0 $(A) = X$, то кажуть, що відношення A задано на X.

Очевидно, що для відношення включення підмножин деякого універсуму U: D_0 $(U) = D_3$ $(U) = P$ (U), де P (U) — множина всіх підмножин універсуму U.

Цікавими є такі окремі випадки відношень в X:

1. Повне (універсальне) відношення $P = X \times X$, яке справджується для будь-якої пари (x_i, x_j) елементів з X. Наприклад, P — відношення «вчитися в одній групі» у множині X, де X — множина студентів групи 1ПР1.

2. Тотожне (діагональне) відношення E, що виконується тільки між елементом і ним самим. Наприклад, рівність на множині дійсних чисел.

3. Порожнє відношення, яке не задовольняє жодна пара елементів з X. Наприклад, A — відношення «бути братом» у множині X, де X — множина жінок.

Означення 4.2. *Розглянемо відношення $A \subset X \times Y$. Нехай елемент $x_i \in X$. Перерізом відношення A за елементом x_i називається множина елементів y з Y, для яких пара $(x_i, y) \in A$:*

$$A(x_i) = \{y \in Y \mid (x_i, y) \in A\}.$$

Множину всіх перерізів відношення A називають фактором-множиною множини Y за відношенням A і позначають Y/A. Вона повністю визначає відношення A.

Приклад. $X = \{x_1, x_2, x_3, x_4, x_5\}$, $Y = \{y_1, y_2, y_3, y_4\}$. Відношення $A = \{(x_1, y_1), (x_1, y_3), (x_2, y_1), (x_2, y_3), (x_2, y_4), (x_3, y_1), (x_3, y_2), (x_3, y_4), (x_4, y_3), (x_5, y_2), (x_5, y_4)\}$.

Очевидно, A $(x_1) = \{y_1, y_3\}$. Випишемо перерізи за всіма елементами множини X у такому вигляді:

x_1	x_2	x_3	x_4	x_5
$\{y_1, y_3\}$	$\{y_1, y_3, y_4\}$	$\{y_1, y_2, y_4\}$	$\{y_3\}$	$\{y_2, y_4\}$

Об'єднання множин другого рядка утворять фактор-множину Y/A.

Об'єднання перерізів за елементами деякої підмножини $B \subset X$ є перерізом A (B) відношення A за підмножиною B, тобто

$$A(B) = \bigcup_{x \in B} A(x).$$

Так, для $B = \{x_2, x_3\}$ маємо A $(B) = \{y_1, y_2, y_3, y_4\} = A$ $(x_2) \cup A$ (x_3).

Подання бінарних відношень за допомогою матриці та графа. З попереднього зрозуміло, що відношення може бути подане за допомогою фактора-множини. Розглянемо ще два способи подання скінченного бінарного відношення: за допомогою матриці й графа.

Матричний спосіб ґрунтується на поданні відношення $A \subset X \times Y$ відповідною йому прямокутною таблицею (матрицею), що складається з нулів та одиниць, де стовпці — перші координати, а рядки — другі, причому на перетині i-го стовпця і j-го рядка буде 1, якщо виконується співвідношення x_i A y_j, або 0 — якщо воно не виконується.

Приклад. Для наведеного у попередньому прикладі відношення матриця має такий вигляд:

	x_1	x_2	x_3	x_4	x_5
y_1	1	1	1	0	0
y_2	0	0	1	0	1
y_3	1	1	0	1	0
y_4	0	1	1	0	1

Матриця повного відношення — це квадратна матриця, що складається лише з одиниць; матриця тотожного (діагонального) відношення — це квадратна матриця, яка складається з нулів та одиниць по головній діагоналі; матриця порожнього відношення — це квадратна матриця, що складається лише з нулів.

Відношення можна також зображати за допомогою орієнтованого графа. Його вершини відповідають елементам множин X та Y, а дуга, спрямована від вершини x_i до y_j, означає, що співвідношення x_i A y_j виконується.

Приклад. Граф відношення, наведеного у другому прикладі на с. 82, має вигляд, показаний на рис. 22.

Граф бінарного відношення — це дводольний граф. Відношення в X зображується графом із вершинами, що відповідають елементам цієї множини. Якщо x_i A x_i й x_i A x_j, то вершини зв'язуються двома протилежно спрямованими дугами, які умовно можна замінювати однією неспрямованою дугою (ребром). Співвідношенню x_j A x_j відповідає петля.

Приклад. Бінарне відношення $A = \{(x_1, x_2), (x_1, x_5), (x_2, x_1), (x_2, x_3), (x_3, x_1), (x_3, x_4), (x_4, x_5), (x_6, x_2)\}$ подається графом, зображеним на рис. 23.

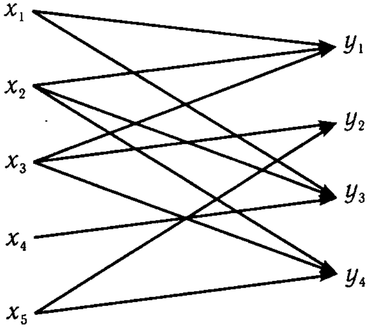


Рис. 22

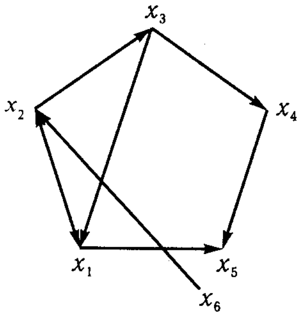


Рис. 23

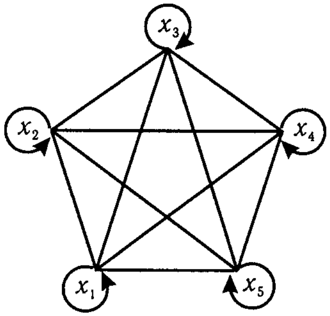


Рис. 24

Граф повного відношення $A = X \times X$, де $X = \{x_1, x_2, x_3, x_4, x_5\}$, показано на рис. 24; граф діагонального відношення — на рис. 25, а граф порожнього відношення — на рис. 26.

Симетричне відношення. Оскільки відношення — це множина, над ним можуть виконуватися всі теоретико-множинні операції. Крім того, виділяються специфічні для відношень операції: обернення (симетризація) і композиція.

Означення 4.3. *Відношення, симетричне (обернене) деякому відношенню $A \subset X \times Y$, позначається A^{-1} і є підмножиною множини $Y \times X$, утвореною тими парами $(y, x) \in X \times Y$, для яких $(x, y) \in A$.*

Перехід від A до A^{-1} здійснюється взаємною перестановкою координат кожної впорядкованої пари. Наприклад, відношення A — «x дільник y» має обернене до нього A^{-1} — «y кратне x». Для наведеного вище відношення A обернене відношення $A^{-1} = \{(4, 2), (6, 2), (3, 3), (6, 3)\}$.

При переході від A до A^{-1} область визначення стає областю значень і навпаки. Матриця A^{-1} — це транспонована матриця відношення A.

Граф оберненого відношення A^{-1} утворюється із графа відношення A заміною всіх дуг на протилежні.

Композиція відношень. Нехай дано три множини X, Y, Z та два відношення $A \subset X \times Y$, $B \subset Y \times Z$.

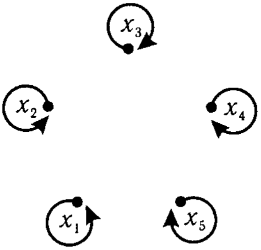


Рис. 25

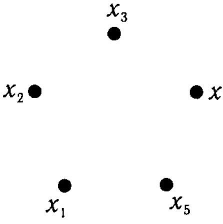


Рис. 26

Означення 4.4. *Композиція відношень A і B є відношенням C, що складається з усіх тих пар $(y, x) \in X \times Z$, для яких існує таке $y \in Y$, що $(x, y) \in A$ й $(y, z) \in B$. Будемо позначати композицію відношень символом* о.

Тоді

$$B\text{o}A = C = \{(x, z) \in X \times Z :\ \exists\, y \in Y\,((x, y) \in A \wedge (y, z) \in B)\}.$$

Можна легко показати, що переріз відношення C за x збігається з перерізом відношення B за підмножиною $A(x) \subset Y$, тобто $C(x) = B(A(x))$.

Приклад. Розглянемо композицію відношення A з другого прикладу на с. 82 з відношенням $B = \{(y_1, z_2), (y_2, z_1), (y_2, z_2), (y_3, z_3), (y_4, z_3)\}$. Це $C = B \text{ o } A = \{(x_1, z_2), (x_1, z_3), (x_2, z_2), (x_2, z_3), (x_3, z_1), (x_3, z_2), (x_3, z_3), (x_4, z_3), (x_5, z_1), (x_5, z_2), (x_5, z_3)\}$. Візьмемо переріз відношення C за x_3: $C(x_3) = \{z_1, z_2, z_3\}$. З іншого боку, маємо $B(A(x_3)) = B(\{y_1, y_2, y_4\}) = \{z_2\} \cup \{z_1, z_2\} \cup \{z_3\} = \{z_1, z_2, z_3\}$.

Щодо властивостей композиції можна зазначити таке:

- $B\text{o}A \neq A\text{o}B$, тобто не виконується закон комутативності;
- $D\text{o}(B\text{o}A) = (D\text{o}B)\text{o}A = D\text{o}B\text{o}A$, тобто виконується закон асоціативності;
- $(B\text{o}A)^{-1} = A^{-1}\text{o}B^{-1}$.

Перші дві властивості очевидні, третю читачеві пропонуємо довести самостійно.

Подання композиції відношень матрицями та графами.

Твердження 4.1. *Матриця композиції відношень $C = B \text{ o } A$ утворюється як добуток матриць відношень B й A з подальшою заміною відмінних від нуля елементів одиницями.*

Справді, елемент c_{ik} матриці композиції знайдемо як суму добутків відповідних елементів матриць B та A (відповідно до правила множення матриць):

$$c_{ik} = b_{i1}a_{1k} + b_{i2}a_{2k} + \ldots + b_{in}a_{nk} = \sum_{j=1}^{n} b_{ij}a_{jk}$$

Очевидно, така сума відмінна від нуля тоді й тільки тоді, коли хоча б один доданок відмінний від нуля, тобто дорівнює одиниці:

$$b_{ij}a_{jk} = 1 \Leftrightarrow b_{ij} = 1 \wedge a_{jk} = 1 \Leftrightarrow y_j B z_i \wedge x_k A y_j \Leftrightarrow x_k B\text{o}A z_i.$$

Якщо у виразі c_{ik} не один, а кілька одиничних доданків, то кожен з них відповідає одному й тому самому співвідношенню $x_k\ B\text{o}Az_i$, через що їх сума має бути замінена одиницею.

Приклад. Для композиції відношень A та B із попереднього прикладу матриця утворюється так:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 2 & 0 & 1 \\ 1 & 2 & 1 & 1 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Нехай $A \subset X \times Y, B \subset Y \times Z$. Щоб побудувати граф $C = B\text{o}A$, потрібно до графа відношення A добудувати граф відношення B. Граф відношення

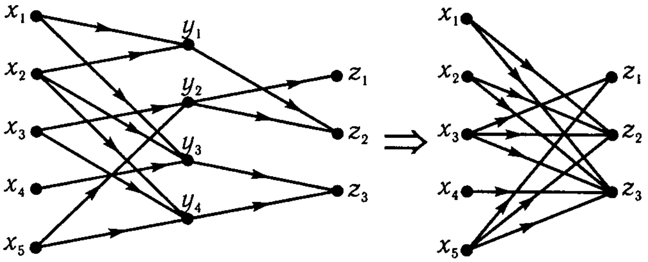


Рис. 27

дістанемо, якщо вилучимо вершини, які відповідають елементам множини Y. При вилученні вершини y_i кожний шлях, що проходить через неї від вершин множини X до вершин множини Z, замінюється однією дугою з тим самим напрямком.

Побудуємо граф композиції для першого прикладу, наведеного на с. 85 (рис. 27).

Якщо A і B — відношення в X, то графи цих відношень та граф композиції $C = B \circ A$ будуються на множині X за загальним правилом.

Приклади: Побудуємо, наприклад, граф композиції для відношення $A = \{(x_1, x_1), (x_1, x_5), (x_2, x_2), (x_2, x_4), (x_2, x_5), (x_5, x_1), (x_5, x_4)\}$ (на рис. 28, *а* йому відповідають суцільні лінії) і відношення $B = \{(x_1, x_2), (x_2, x_3), (x_2, x_4), (x_4, x_3), (x_4, x_5), (x_5, x_2)\}$ (на рис. 28, *а* йому відповідають штрихові лінії). Граф композиції зображено на рис. 28, *б*.

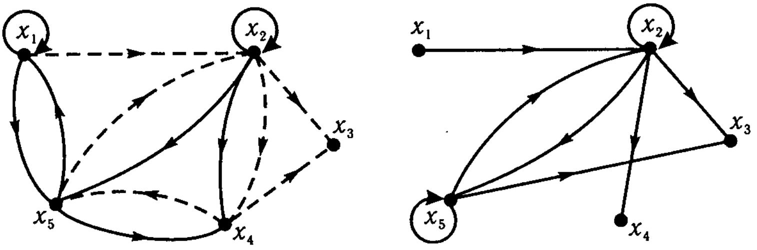
2. Знайдемо матрицю композиції відношень A та B із попереднього прикладу:

$$A = \begin{array}{|c|ccccc|} \hline & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline x_1 & 1 & 0 & 0 & 0 & 1 \\ x_2 & 0 & 1 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 0 & 0 & 0 \\ x_4 & 0 & 1 & 0 & 0 & 1 \\ x_5 & 1 & 1 & 0 & 0 & 0 \\ \hline \end{array}; \quad B = \begin{array}{|c|ccccc|} \hline & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline x_1 & 0 & 0 & 0 & 0 & 0 \\ x_2 & 1 & 0 & 0 & 0 & 1 \\ x_3 & 0 & 1 & 0 & 1 & 0 \\ x_4 & 0 & 1 & 0 & 0 & 0 \\ x_5 & 0 & 0 & 0 & 1 & 0 \\ \hline \end{array};$$

$$B \times A = \begin{array}{|c|ccccc|} \hline & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline x_1 & 0 & 0 & 0 & 0 & 0 \\ x_2 & 1 & 1 & 0 & 0 & 1 \\ x_3 & 0 & 1 & 0 & 0 & 1 \\ x_4 & 0 & 1 & 0 & 0 & 0 \\ x_5 & 0 & 1 & 0 & 0 & 1 \\ \hline \end{array}.$$

Властивості відношень. Нехай A — бінарне відношення у множині $X (A \subset X \times X)$. Тоді відношення A є:

- рефлексивним, якщо $E \subset A$, тобто, іншими словами, воно завжди виконується між елементом і ним самим $(\forall x \in X (xAx))$. Як приклад такого відношення можна навести відношення нестрогої нерівності $(\leq, \geq)$ на N, Z й R;



а б

Рис. 28

- антирефлексивним, якщо $A \cap E = \varnothing$, тобто якщо співвідношення x_iAx_j виконується, то $x_i \neq x_j$. Це, наприклад, відношення строгої нерівності на N, Z та R, відношення «бути старшим» у множині людей;
- симетричним, якщо $A = A^{-1}$, тобто при виконанні співвідношення x_iAx_j виконується співвідношення x_jAx_i. Як приклад такого відношення можна навести відстань між двома точками на площині, відношення «бути братом» у множині людей;
- асиметричним, якщо $A \cap A^{-1} = \varnothing$, тобто із двох співвідношень x_iAx_j й x_jAx_i щонайменше одне не виконується. Як приклад такого відношення можна навести відношення «бути батьком» у множині людей, відношення строгого включення в множині всіх підмножин деякого універсуму. Очевидно, якщо відношення асиметричне, то воно й антирефлексивне;
- антисиметричним, якщо $A \cap A^{-1} \subset E$, тобто обидва співвідношення x_iAx_j та x_jAx_i одночасно виконуються тоді й тільки тоді, коли $x_i = x_j$. Як приклад можна навести нестрогу нерівність $(\leq, \geq)$ на N, Z та R;
- транзитивним, якщо $A \circ A \subset A$, тобто з виконання співвідношень x_iAx_j й x_jAx_k випливає виконання співвідношення x_iAx_k. Як приклад можна навести відношення «бути дільником» у Z, «бути старшим» у множині людей.

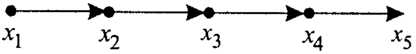
Матриця рефлексивного відношення характеризується тим, що всі елементи її головної діагоналі — одиниці, а матриця антирефлексивного відношення — тим, що зазначені елементи є нулями. Зауважимо, що симетричність відношення спричинює також симетричність матриці. Матриця асиметричного відношення характеризується тим, що всі елементи її головної діагоналі — нулі й немає жодної пари одиниць на місцях, симетричних відносно головної діагоналі. Матриця антисиметричного відношення має ті самі властивості, що й асиметричного, за винятком вимоги рівності нулю елементів головної діагоналі. Матриця транзитивного відношення характеризується тим, що коли $a_{ji} = 1$ й $a_{kj} = 1$, то $a_{ki} = 1$, причому наявність одиничних елементів на головній діагоналі не порушує транзитивності матриці.

Граф рефлексивного відношення характеризується тим, що петлі є у всіх вершинах, граф антирефлексивного відношення не має жодної петлі.

Для симетричного відношення вершини графа можуть бути пов'язані тільки парами протилежно спрямованих дуг (тобто ребрами). У графа асиметричного відношення петлі відсутні, а вершини можуть бути пов'язані тільки однією спрямованою дугою.

У разі антисиметричного відношення можуть бути петлі, але зв'язок між вершинами, якщо він є, також відображається тільки однією спрямованою дугою.

Граф транзитивного відношення характеризується тим, що коли через деяку сукупність вершин графа проходить шлях, то існують дуги, які з'єднують будь-яку пару вершин з цієї сукупності в напрямку шляху. Як правило, на графі транзитивного відношення зображують тільки цей шлях, а зумовлені транзитивністю дуги опускають. Такий граф називають графом редукції (або кістяковим графом). Наприклад, граф редукції на п'яти вершинах має такий вигляд:



Бінарні (двомісні) відношення, що розглядаються, є окремим випадком n-місних відношень при $n = 2$.

Багатомісні відношення. n-місним є відношення $\{(x_1, x_2, ..., x_n) | (x_1, x_2, ..., x_n) \subset X_1 \times X_2 \times ... \times X_n\}$, де $(x_1, x_2, ..., x_n)$ називається n-вимірним вектором, кортежем або просто впорядкованою n-кою. Можна також визначити n-місне відношення за індукцією. За означенням бінарне відношення — це множина впорядкованих пар (x_1, x_2), де $x_1 \in X_1$, $x_2 \in X_2$. Впорядкована трійка (x_1, x_2, x_3) може розглядатися як упорядкована пара $((x_1, x_2), x_3)$, де перша координата (x_1, x_2) є впорядкованою парою, причому $(x_1, x_2) \in X_1 \times X_2$, і т. д. І нарешті, n-вимірний вектор $(x_1, x_2, ..., x_n)$ може розглядатися як упорядкована пара $((x_1, x_2, ..., x_{n-1}), x_n)$.

Як приклади тримісного (тернарного) відношення можна навести всі арифметичні операції над числами (в них виділяється перший операнд, другий операнд і результат операції), а також відношення між батьками й дітьми, впорядкована трійка в яких — це батько, мати, дитина.

Пропорція $\frac{x}{y} = \frac{z}{u}$ ілюструє чотиримісне відношення.

Функціональне відношення. Відношення $f \subset X \times Y$ називається функціональним, якщо його елементи (впорядковані пари) мають різні перші координати: $\forall x \in D_0(f)\ \exists! y((x, y) \in f)$. Іншими словами, кожному $x \in X : (x, y) \in f$ відповідає один і тільки один елемент $y \in Y$. Очевидно, для функціонального відношення A кожний переріз за будь-яким $x \in X$ містить не більш як один елемент. Якщо $x \notin D_0(f)$, то переріз за x — порожній.

Якщо $D_0(f) = X$, то функціональне відношення f називається всюди визначеним. Матриця функціонального відношення містить у кожному стовпці не більш як один одиничний елемент, а його граф характеризується тим, що з кожної вершини може виходити тільки одна дуга (враховуючи й петлі).

Приклад. Розглянемо множини $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ та $Y = \{y_1, y_2, y_3\}$. Відношення $A = \{(x_1, y_1), (x_2, y_2), (x_3, y_1), (x_5, y_3), (x_6, y_1)\}$, очевидно, є функціональним. Для множин $X = \{1, 2, 3, 4\}$ й $Y = \{1, 4, 9, 16, 25\}$ відношення $B = \{(1, 1)\ (2, 4)\ (3, 9)\ (4, 16)\}$ також є функціональним.

Функції та відображення. Усяке функціональне відношення можна розглядати як функцію. При цьому перша координата x упорядкованої пари $(x, y) \in A$ є прообразом (аргументом, змінною), а друга y — образом (значенням). Це можна записати як $y = f(x)$ чи xfy або $(x, y) \in f$. (Усі три записи є еквівалентними.)

Потрібно розрізняти функцію f як множину впорядкованих пар (відношення) і значення функції $y = f(x)$ як другу координату однієї з таких пар.

Слід зазначити, що відношення, обернене до функціонального, загалом не є функціональним. У розглянутому вище прикладі відношення A є функціональним, але симетричне йому відношення $A^{-1} = \{(y_1, x_1), (y_1, x_3), (y_1, x_6), (y_2, x_2), (y_3, x_5)\}$ не є функціональним.

Якщо функціональне відношення $f \subset X \times Y$ всюди визначено на X $(D_0(f) = X)$, то його називають відображенням множини X в Y і записують $X \xrightarrow{f} Y$.

Відображення можна також розглядати як функцію f, визначену на множині X, але яка набуває значень на множині Y.

Очевидно, що відмінність між відображенням та функцією зводиться до способу означення цих відношень на множині X, причому відображення потрібно розглядати як окремий випадок функції. Більшість математиків не розрізняють поняття відображення і функції. Вони записують $f : X \to Y$ або $x \to f(x)$.

Типи відображень. При відображенні X в Y кожний елемент x з X має один і тільки один образ $(\forall x \in X \;\; \exists! \;\; y \in Y (y = f(x))$.

Однак зовсім не обов'язково, щоб кожний елемент з Y був образом деякого елемента з X. Графічно цю ситуацію можна зобразити так, як показано на рис. 29.

Якщо ж будь-який елемент з Y є образом принаймні одного елемента з X, то кажуть, що множина X відображується на Y (явище сюр'єкції, або накриття). Цю ситуацію зображено на рис. 30.

Якщо для будь-яких двох різних елементів x_1 й x_2 з X їх образи $y_1 = f(x_1)$, $y_2 = f(x_2)$ також різні, то відображення f називається ін'єкцією,

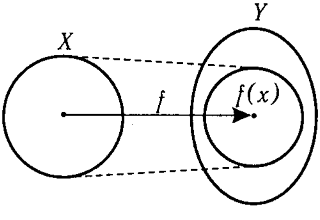


Рис. 29

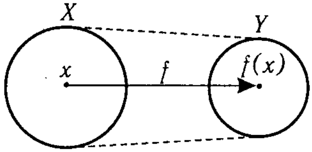


Рис. 30

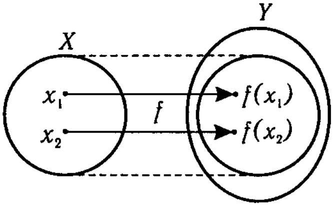


Рис. 31

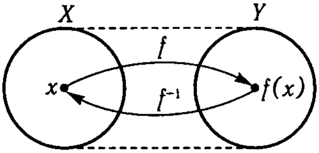


Рис. 32

або взаємно однозначним відображенням. Цю ситуацію показано на рис. 31.

Відображення, яке є одночасно сюр'єктивним та ін'єктивним, називається бієкцією (накладанням). У цьому випадку кажуть, що між елементами X й Y існує взаємно однозначна відповідність. При цьому (рис. 32) виникає питання, чи є обернене відношення f також взаємно однозначним.

Будь-яке відображення f із X в Y є елементом множини $P(X\times Y)$.

Якщо f — взаємно однозначне відображення, а $X = Y$, то $f : X \to X$ називається відображенням множини X на себе. Елементи $(x, x)\in X\times X$ утворюють тотожне відображення E, причому $f\text{o}f^{-1} = f^{-1}\text{o}f = E$.

Образи і прообрази. Загалом при відображенні $f : X \to Y$ елемент із Y може бути образом не одного, а кількох елементів із X. Так, для розглянутого у прикладі на с. 89 відношення елемент y_1 є образом для елементів x_1, x_3, x_6.

Сукупність усіх елементів, образом яких є заданий елемент у, називається повним прообразом елемента y і позначається $f^{-1}(y)$. У наведеному прикладі $f^{-1}(y_1)=\{x_1, x_3, x_6\}$.

Сукупність елементів $f(x)$, які є образами всіх елементів множини X, називається образом цієї множини та позначається $f(X)$.

Нехай $R \subset Y$. Сукупність усіх елементів із X, образи яких належать R, називається повним прообразом множини R і позначається (f^{-1} (R)).

Основні властивості відображень. Їх чотири.

1. Повний прообраз об'єднання дорівнює об'єднанню повних прообразів, тобто

$$f^{-1}(A\cup B)=f^{-1}(A)\cup f^{-1}(B).$$

2. Повний прообраз перерізу дорівнює перерізу повних прообразів, тобто

$$f^{-1}(A\cap B)=f^{-1}(A)\cap f^{-1}(B).$$

3. Образ об'єднання дорівнює об'єднанню образів, тобто

$$f(A\cup B)=f(A)\cup f(B).$$

4. Образ перерізу є підмножиною перерізу образів, тобто

$$F(A\cap B)\subset f(A)\cap f(B).$$

Звуження та продовження функції. Нехай функцію $f : X \to Y$ задано на X, f_1 — на множині $Q \subset X$, причому для кожного $x \in Q$ виконується $f(x) = f_1(x)$. Тоді f_1 називається обмеженням (звуженням) функції f на Q, а f — продовженням функції f_1 на X.

Приклад. Функція $f(x) = x^3$, задана на множині R, відображає цю множину на себе. Якщо ввести обмеження $D_0(f) = Z$ (множина цілих чисел), то дістанемо звуження $f_1(x)$ функції $f(x)$ на Z, причому $f_1(x)$ відображає множину Z, але не на Z, оскільки не кожне ціле число є кубом цілого числа.

Композиція відображень. Якщо $f : X \to Y$, $q : Y \to Z$, то їх композиція $(q \circ f) : X \to Z$, причому $(q \circ f)(x) = q(f(x))$. Наприклад, якщо $f = \sin$, $q = \ln$, то $(q \circ f)(x) = (\ln \circ \sin)(x) = \ln(\sin(x)) = \ln \sin x$.

Теорема 4.1. *Функція f є взаємно однозначним функціональним відношенням тоді й тільки тоді, коли f^{-1} — взаємно однозначне функціональне відношення.*

Доведення. Доведемо, що f^{-1} — функція. Нехай $(y, x_1) \in f^{-1}$, $(y, x_2) \in f^{-1}$. За означенням оберненого відношення маємо $(x_1, y) \in f$, $(x_2, y) \in f$. Оскільки f за умовою є взаємно однозначною функцією, дістаємо $x_1 = x_2$, а це означає, що f^{-1} — функціональне відношення. Покажемо, що f^{-1} — взаємно однозначне функціональне відношення. Нехай $(y_1, x) \in f^{-1}$ й $(y_2, x) \in f^{-1}$. Це означає, що $(x, y_1) \in f$, $(x, y_2) \in f$. Оскільки f — функція, маємо $y_1 = y_2$, а це означає, що f^{-1} є взаємно однозначним функціональним відношенням. Таким чином, необхідну умову теореми доведено. Читачеві пропонуємо показати, що таким чином доведено також її достатню умову.

Теорема 4.2. *Композиція двох функціональних відношень є функціональним відношенням.*

Доведення. Нехай $f \exists b \to B$, а $q : B \to C$. За означенням композиції відношень $h = q \circ f = \{(a, c) \mid ((a, b) \in f \wedge (b, c) \in q)\}$. Отже, це за означенням — підмножина декартового добутку $A \times C$.

Доведемо, що h — функціональне відношення. Нехай задано дві пари, які належать h:

$$\begin{cases} (a, c_1) \in h \Rightarrow \exists b_1 \in B((a, b_1) \in f \wedge (b_1, c_1) \in q); \\ (a, c_2) \in h \Rightarrow \exists b_2 \in B((a, b_2) \in f \wedge (b_2, c_2) \in q). \end{cases}$$

Оскільки f — функціональне відношення, маємо $b_1 = b_2$, а оскільки q — функціональне відношення, дістаємо $c_1 = c_2$; отже, h — функціональне відношення.

Для багатомісних функцій $f : A^m \to B, g : B^n \to C$ можливими є різні варіанти підстановки f у g, які дають функції різних типів. Наприклад, при $m = 3, n = 4$ функція $h_1 = g(x_1, f(y_1, y_2, y_3), x_3, x_4)$ має шість аргументів і діє з $B \times A^3 \times B^2 \to C$, а функція $h_2 = g(f(y_1, y_2, y_3), f(z_1, z_2, z_3), x_3, x_4)$ має вісім аргументів та діє з $A^6 \times B^2 \to C$. Особливо цікавим є випадок, коли задано множину функцій типу $f_1 : A^{m_1} \to A, A^{m_2} \to A, \ldots, A^{m_n} \to A$. У цьому разі може бути виконане будь-яке перейменування аргументів, наприклад перейменування x_3 в x_2, що породжує з функції $f(x_1, x_2, x_3, x_4)$ функцію трьох аргументів $f(x_1, x_2, x_2, x_4)$.

Означення 4.5. *Функція, що утворюється з функцій $f_1, f_2, ..., f_n$ деякою підстановкою їх одна в одну і перейменуванням аргументів, називається суперпозицією $f_1, f_2, ..., f_n$.*

Приклади: 1. У функції $f_1(x_1, x_2, x_3) = x_1 + 2x_2 + 7x_3$ перейменування x_3 в x_2 приводить до функції $f_1(x_1, x_2, x_3) = x_1 + 2x_2 + 7x_2 = f_2(x_1, x_2) = x_1 + 9x_2$. Перейменування x_1 та x_3 в x_2 приводить до одномісної функції $f_3(x_2) = 10x_2$ [1].

2. Елементарною функцією в математичному аналізі називається кожна функція f, що є суперпозицією фіксованого (тобто незалежного від значень аргументів f) числа арифметичних функцій, а також функцій e^x, $\log x$, $\sin x$, $\arcsin x$. Наприклад, функція $\log^2(x_1 + x_2) + 3\sin\sin x_1 + x_3$ — елементарна, оскільки є результатом кількох послідовних суперпозицій $x_1 + x_2$, x^2, $\log x$, $3x$, $\sin x$ [1].

3. Коло одиничного радіуса з центром у точці G (3, 2) (рис. 33), тобто множина пар дійсних чисел (x, y), які задовольняють співвідношення $(x - 3)^2 + (y - 2)^2 \leq 1$, задає відношення між віссю абсцис і віссю ординат (тобто R та R). Образом числа 4 при цьому є однина 2, образом відрізка [2, 3] — відрізок [1, 3] на осі ординат, цей же відрізок [1, 3] є образом відрізка [3, 4] на осі абсцис. Задане відношення (або відповідність) не є функціональним. Прикладом функціональних відношень між дійсними числами на рис. 33 слугує дуга ABC [1].

Ще раз нагадаємо, що для задання відповідності треба зазначити не тільки множину G, а й множини A та B, тобто вказати, підмножиною якого прямого добутку є G. У цьому прикладі коло G задає також інша відповідність: між відрізком [2, 4] і відрізком [1, 3]. При цьому деякі властивості відповідності $G \subset R^2$ та $G \subset [2, 4] \times [1, 3]$ різняться: наприклад, друга відповідність на відміну від першої всюди визначена і сюр'єктивна.

Ураховуючи ці співвідношення, слід би задавати відповідність як трійку множин (G, A, B). Тоді не довелося б обмовлятися, що одне коло може задавати дві відповідності, це й так було б зрозуміло з відмінності трійок (G, R, R) та $(G, [1, 4], [1, 3])$. Проте такі застереження доводиться робити рідко: або множини A і B є зрозумілими з контексту, або відмінності в їх виборі не впливають на властивості відповідності, які досліджуються.

4. Англо-російський словник установлює відповідність між множиною англійських та російських слів. Ця відповідність не є функціональною (оскільки одному англійському слову, як правило, ставляться у відповідність кілька російських слів); крім того, вона практично ніколи не є повністю визначеною: завжди можна знайти англійське слово, що міститься в цьому словнику [1].

5. Позиція на шахівниці є взаємно однозначною відповідністю між множиною фігур, які залишилися на дошці, та множиною зайнятих ними полів [1].

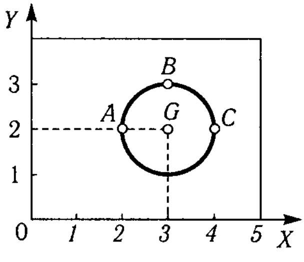


Рис. 33

6. Різні види кодування (кодування літер Морзе, подання чисел у різних системах числення, секретні шифри, вхідні й вихідні номери в діловій переписці тощо) є відповідністю між об'єктами, що кодуються, і кодами, що присвоюються їм. Ця відповідність, як правило, має всі властивості взаємно однозначної відповідності, крім, може бути, однієї — сюр'єктивності. Єдність образу та прообразу в кодуванні гарантує однозначність шифрування і дешифрування. Відсутність сюр'єктивності означає, що не кожний код має значення, тобто відповідає якому-небудь об'єкту. Наприклад, кодування телефонів м. Києва семизначними номерами не є сюр'єктивним, оскільки деякі семизначні номери не відповідають жодним телефонам [1].

7. Усяка нумерація зліченної множини є її відображенням на N.

8. Функція $f(x) = \sqrt{x}$ є не повністю визначеною, якщо її тип $N \to N$, і повністю визначеною, якщо її тип $N \to R$ або $R_+ \to R$ (R_+ — додатна підмножина R).

9. Функція sin x має тип $R \to R$. Відрізок [$-\pi/2$, $\pi/2$] вона взаємно однозначно відображає на відрізок [-1, 1]. Тому на відрізку [-1, 1] для неї існує обернена функція arc sin x.

10. Вище наводилися приклади кодувальних функцій, які кожному об'єкту зі своєї області значень ставлять у відповідність деякий код. Для кодувальної функції оберненою буде декодувальна функція, яка кожному коду ставить у відповідність закодований ним об'єкт. Якщо кодувальна функція не є сюр'єктивною, то декодувальна функція не всюди є визначеною [1].

11. Функції sin x та $\sqrt{x}$ мають тип $R \to R$, тобто відображають одну й ту саму множину в себе. Тому їх композиція можлива в довільному порядку і дає функції $\sin\sqrt{x}$ та $\sqrt{\sin\ x}$. Зазначимо, що області визначення їх різні: першу функцію визначено на додатній півосі, другу — на множині відрізків [$2k\pi$, $(2k + 1)\pi$], де $k = 0, \pm 1, \pm 2, \ldots$. Таким чином, область визначення композиції може бути вужчою від областей визначення обох початкових функцій і навіть виявитися порожньою.

12. Множина $K = \{k_1, \ldots, k_m\}$ команд ЕОМ відображається в машинні коди цієї ЕОМ, тобто в натуральні числа. Кодувальна функція φ має тип $K \to N$. За допомогою суперпозиції цієї функції та арифметичних функцій стають можливими арифметичні дії над командами (які самі по собі числами не є!), тобто $\varphi(k_1) + \varphi(k_2)$, $\varphi(k_1) + 4$ і т. д.

13. У функції $f_1(x_1, x_2, x_3) = x_1 + 2x_2 + 7x_3$ перейменування x_3 в x_1 приводить до функції $f_1(x_1, x_2, x_3) = x_1 + 2x_2 + 7x_1$, що є функцією двох аргументів $f_2(x_1, x_2) = 8x_1 + 2x_2$. Перейменування x_2 й x_3 в x_1 приводить до одномісної функції $f_3(x_1) = 10x_1$.

14. Кожне натуральне число n єдиним способом розкладається на добуток простих чисел (простих дільників цього числа). Тому якщо домовитися розташовувати прості дільники n у певному порядку (наприклад, у порядку неспадання), то матимемо функцію $q(n)$ типу $N \to \bigcup_{i=1}^{\infty} N^i$, яка відображає N у множину векторів довільної довжини. Наприклад, $q(42) = (2, 3, 7)$, $q(23) = (23)$, $q(100) = (2, 2, 5, 5)$. Це відображення не є сюр'єктивним, оскільки в область значень q не входять вектори, для компонент яких не виконується умова неспадання, а також вектори з непростими компонентами.

4.2. Відношення еквівалентності

Відношення еквівалентності є експлікацією (перекладом інтуїтивних уявлень у ранг строгих математичних понять) таких слів, як «подібність», «нерозрізненість», «взаємозамінність», «рівносильність».

Бінарне відношення в множині X називається відношенням еквівалентності (позначається « ~ »), якщо виконуються такі властивості:

- рефлексивність ($x \sim x$);
- симетричність $(x \sim y \Rightarrow y \sim x)$;
- транзитивність $(x \sim y \wedge y \sim z \Rightarrow x \sim z)$.

Найважливіше значення відношення еквівалентності полягає в тому, що воно задає ознаку для розбиття множини X на непересічні підмножини. Наведемо приклади відношень еквівалентності.

1. «Проживати в одному будинку» у множині людей.

2. «Подібність трикутників» у множині всіх трикутників на площині.

3. «Паралельність прямих» у множині всіх прямих на площині.

Називатимемо класом еквівалентності елемента a множину всіх елементів множини X, які еквівалентні елементу $a : [a]_\sim = \{x \in X \mid a \sim x\}$.

Твердження 4.2. $a \in [a]_\sim$.

Це твердження природно випливає із рефлексивності відношення еквівалентності.

Твердження 4.3. $a \sim b \Leftrightarrow [a]_\sim = [b]_\sim$.

Необхідність доведемо від супротивного: нехай $A = [a]_\sim \neq B = [b]_\sim$. Оскільки множини A і B не є рівними, існує елемент c такий, що:

1) $c \in A \wedge c \notin B$ або 2) $c \in B \wedge c \notin A$.

Розглянемо перший випадок: $c \in A \Rightarrow a \sim c$. За умови $a \sim b$ та за властивістю симетричності $b \sim a$. Оскільки відношення еквівалентності транзитивне, маємо $b \sim a \wedge a \sim c \Rightarrow b \sim c$, тобто $c \in [b]_\sim$, що суперечить умові. Другий випадок доводиться аналогічно.

Доведемо достатність від супротивного: нехай a не є еквівалентним b, тобто ці елементи не можуть належати одному класу еквівалентності, що суперечить умові $[a]_\sim = [b]_\sim$. Одержана суперечність доводить твердження.

Теорема 4.3. *Якщо на множині X задано відношення еквівалентності, то воно задає розбиття множини і це розбиття — єдине.*

Доведення. Нехай на множині X задано відношення еквівалентності. Розглянемо сукупність усіх класів еквівалентності $\{[x]_\sim\}$, які можуть бути утворені. Доведемо, що:

1. $\bigcup\limits_{x\in X} [x]_\sim = X$.
2. $\forall x, y \in X : [x]_\sim \neq [y]_\sim \left([x]_\sim \cap [y]_\sim = \varnothing\right)$.

Спочатку доведемо п. 1. Розглянемо будь-який елемент $x_1 \in \bigcup\limits_{x\in X} [x]_\sim$.

Якщо x_1 належить об'єднанню класів еквівалентності, то цей елемент є хоча б в одному класі з цього об'єднання, тобто існує $y \in X$ таке, що $x_1 \in [y]$, а за означенням класу еквівалентності $x_1 \in X$.

Отже, $\bigcup\limits_{x\in X} [x]_\sim \subset X$.

Розглянемо будь-який елемент $x_1 \in X$. За твердженням 4.1 $x_1 \in [x_1]_\sim$. Це означає, що він належить також об'єднанню класів еквівалентності $x_1 \in \bigcup\limits_{x\in X} [x]_\sim$, тобто $X \subset \bigcup\limits_{x\in X} [x]_\sim$.

За теоремою про рівність множин $X = \bigcup\limits_{x\in X} [x]_\sim$.

П. 2 доведемо від супротивного: нехай є два класи еквівалентності $[x]_\sim$ і $[y]_\sim$, які не збігаються та переріз яких непорожній: $[x]_\sim \cap [y]_\sim = C \neq \varnothing$. Тоді знайдеться елемент $z \in C$. Це означає, що $z \in [x]_\sim$ і $z \in [y]_\sim$. За означенням класів еквівалентності $z \sim x$ та $z \sim y$. Із симетричності й транзитивності відношення еквівалентності випливає, що $x \sim y$, а за твердженням 4.2 це означає, що $[x]_\sim = [y]_\sim$, тобто суперечить припущенню. Одержана суперечність доводить п. 2.

Таким чином, доведено, що класи еквівалентності утворюють розбиття множини X. Залишилось довести єдиність такого розбиття. Припустимо супротивне: розбиття не єдине. Нехай за заданим відношенням еквівалентності існують два різних розбиття R_1 й R_2 $(R_1 \neq R_2)$. Це означає, що є така точка $x \in X$, яка належить деякому класу розбиття $[y_1]_\sim$ в R_1 та $[y_2]_\sim$ в R_2 ($[y_1]_\sim \neq [y_2]_\sim$). Оскільки $x \in [y_1]_\sim$, маємо $y_1 \sim x$, а оскільки $x \in [y_2]_\sim$, дістаємо $y_2 \sim x$. Внаслідок симетричності й транзитивності відношення еквівалентності $y_1 \sim y_2$; за твердженням 4.2 це означає, що $[y_1]_\sim = [y_2]_\sim$, а це суперечить умові $[y_1]_\sim \neq [y_2]_\sim$. Одержана суперечність доводить єдиність розбиття.

Аналогічно можна довести твердження, обернене до теореми 4.3, про те, що коли на множині означено розбиття, то воно задає деяке відношення еквівалентності, внаслідок чого класи (елементи) розбиття є класами еквівалентності.

Доведемо це. Нехай існує розбиття R множини X на підмножини X_i $(i = \overline{1,n})$, тобто

$$\bigcup_{i=1}^{n} X_i = X;$$

$$\forall i \neq j\left(X_i \cap X_j = \varnothing\right).$$

Введемо відношення

$$x\rho y \Leftrightarrow \exists X_i\left(x, y \in X_i\right).$$

Тоді:

- рефлексивність $(x\rho x)$ очевидна;
- симетричність $(x\rho y \Leftrightarrow y\rho x)$ очевидна, тому що

$$x, y \in X_i \Leftrightarrow y, x \in X_i;$$

- транзитивність $(x\rho y \wedge y\rho z \Rightarrow x\rho z)$ доведемо від супротивного.

Нехай $x\rho y \wedge y\rho z \Rightarrow \exists X_i\left(x, y \in X_i\right) \wedge \exists X_j\left(y, z \in X_j\right)$ й $X_i \neq X_j\,(i \neq j)$; тоді $X_i \cap X_j \neq \varnothing$, що суперечить умові.

Усі елементи, які належать деякому класу X_i розбиття $\{X_1, X_2, ..., X_n\}$ множини X, пов'язані між собою відношенням еквівалентності. Вони взаємозамінні в тому сенсі, що будь-який з цих елементів задає клас, тобто може слугувати його представником (еталоном).

Означення 4.6. *Підмножина $\tilde{X}$ множини X, що містить один і тільки один елемент із кожного класу деякого розбиття, називають системою представників відповідного відношення еквівалентності.*

Наведемо деякі приклади відношення еквівалентності.

Приклади: 1. Відношення «проживати в одному будинку» на множині жителів міста, очевидно, є відношенням еквівалентності й розбиває цю множину на неперерізні підмножини людей, які є сусідами в будинку.

2. Розглянемо відношення рівності за mod m на множині цілих чисел Z. Говорять, що x дорівнює y за mod m $(0 \leq y < m)$, якщо $(x - y)$ ділиться на m без залишку. Записують це так: $x = y$ (mod m). Усі цілі числа, які дорівнюють у за mod m, утворюють

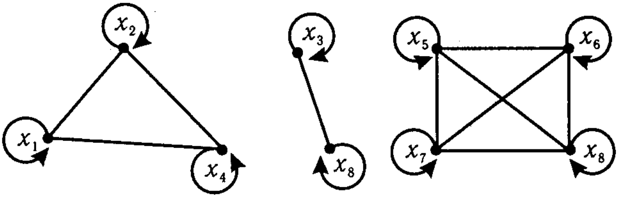


Рис. 34

підмножину цілих чисел, що мають однаковий залишок y при поділі на m. Очевидно, такі підмножини є класами еквівалентності, а як представника кожної з них природно вибрати залишок $y = 0, 1, 2, ..., m - 1$. Таким чином, відношення рівності за mod m означає розбиття множини цілих чисел на m класів $Z_0, Z_1, Z_2, ..., Z_{m-1}$, де $Z_j = \{j, j + m, j + 2m, ...\}$ — множина, яка називається класом лишків за mod m.

Наприклад, при $m = 4$ маємо $M_0 = \{0, 4, 8, 12, ...\}$, $M_1 = \{1, 5, 9, 13, 17, ...\}$, $M_2 = \{2, 6, 10, 14, ...\}$, $M_3 = \{3, 7, 11, 15, ...\}$.

Матриця і граф відношення еквівалентності. Нехай відношення еквівалентності задано в множині X.

Елементи, що належать одному класу еквівалентності, попарно еквівалентні між собою. Отже, стовпці матриці відношення еквівалентності для елементів одного класу еквівалентності однакові та містять одиниці у всіх рядках, які відповідають цим елементам. Оскільки класи еквівалентності не перерізаються, у стовпцях, які відповідають елементам різних класів, не буде одиниць в одних і тих самих рядках.

При побудові матриці відношення розташуємо елементи множини так, щоб ті елементи, які належать одному класу еквівалентності, були поруч. Тоді одиничні елементи матриці відношення еквівалентності утворять непересічні квадрати, діагоналі яких розташовуються на головній діагоналі матриці.

Приклад. Для відношення еквівалентності, заданого класами еквівалентності

$$X_1 = \{x_1, x_2, x_4\};\ X_2 = \{x_3, x_8\};\ X_3 = \{x_5, x_6, x_7, x_9\},$$

матриця матиме такий вигляд:

	x_1	x_2	x_4	x_3	x_8	x_5	x_6	x_7	x_9
x_1	1	1	1	0	0	0	0	0	0
x_2	1	1	1	0	0	0	0	0	0
x_4	1	1	1	0	0	0	0	0	0
x_3	0	0	0	1	1	0	0	0	0
x_8	0	0	0	1	1	0	0	0	0
x_5	0	0	0	0	0	1	1	1	1
x_6	0	0	0	0	0	1	1	1	1
x_7	0	0	0	0	0	1	1	1	1
x_9	0	0	0	0	0	1	1	1	1

Граф відношення еквівалентності також має характерний вигляд. Це граф, кожна компонента з'єднання якого, що відповідає класу еквівалентності, є повним графом із петлями на кожній вершині.

Для цього прикладу граф має вигляд, зображений на рис. 34.

4.3. Відношення порядку

Означення 4.7. *Бінарне відношення в матриці X називається відношенням нестрогого порядку, якщо воно:*

- *рефлексивне* $(\forall x\in D_0(\sigma)(x\sigma x))$;
- *асиметричне* $(\forall x, y\in D_0(\sigma)(x\sigma y \wedge y\sigma x \Rightarrow x = y))$;
- *транзитивне* $(\forall x, y, z\in D_0(\sigma)(x\sigma y \wedge y\sigma z \Rightarrow x\sigma z))$.

Часто відношення σ позначають «$\leq$», оскільки нестрога нерівність є прикладом відношення нестрогого порядку в множині Z або R, що найчастіше використовується (як і «$\geq$»), але ототожнювати їх усе ж не варто. Як приклад відношення нестрогого порядку в множині людей можна назвати відношення «бути не старшим» або «бути не молодшим».

Означення 4.8. *Бінарне відношення* ρ *в множині X називається відношенням строгого порядку, якщо воно:*

- *асиметричне* $(\forall x, y\in D_0(\rho)(x\rho y \Rightarrow y\rho x))$;
- *транзитивне* $(\forall x, y, z\in D_0(\rho)(x\rho y \wedge y\rho z \Rightarrow x\rho z))$.

Як приклад відношення строгого порядку можна навести відношення «$>$» або «$<$» у множинах N, Z чи R, а також відношення «бути молодшим» або «бути старшим» у множині людей.

Якщо виконується співвідношення $x\rho y$ (або $x\sigma y$), то кажуть, що елемент x передує y, а y іде за x.

Множина, в якій визначено відношення порядку (строгого або нестрогого), називається упорядкованою, і кажуть, що порядок уведено цим відношенням.

Означення 4.9. *Множина М називається абсолютно (лінійно) впорядкованою, якщо для будь-яких двох її елементів x та y виконується* $x\sigma y$ *або* $y\sigma x$ *(*$x\rho y$ *або* $y\rho x$*).* Наприклад, множина дійсних чисел R з відношенням порядку «$\leq$» (або «$\geq$», «$<$», «$>$») є абсолютно впорядкованою.

Може виявитися, що для деяких пар (x, y) жодне зі співвідношень $x\sigma y$ або $y\sigma x$ ($x\rho y$ або $y\rho x$) не виконується. Такі елементи x й y називаються незрівнянними. У цьому випадку кажуть, що множина є частково впорядкованою.

Приклад. Розглянемо відношення включення множин на множині всіх підмножин деякого універсуму $P(U)$. Оскільки для нього виконуються властивості:

- рефлексивності, тому що $(\forall X\in P(U)(X\subset X))$;
- асиметричності $(X\subset Y \wedge Y\subset X \Rightarrow X = Y)$ (за теоремою рівності множин);
- транзитивності $(X\subset Y \wedge Y\subset Z \Rightarrow X\subset Z)$, яку доведено вище, відношення включення множин є відношенням нестрогого порядку і $P(U)$ є впорядкованою множиною.

Проте очевидно, що серед усіляких підмножин U знайдуться такі множини X й Y, що ні $X\subset Y$, ні $Y\subset X$ не виконуються. Отже, $P(U)$ з відношенням нестрогого порядку «$\subset$» є частково впорядкованою множиною.

Як ілюстрацію відношення лінійного порядку можна навести також відношення старшинства на множині офіцерських звань: лейтенант, старший лейтенант, капітан, майор, підполковник, полковник, генерал, маршал. Очевидно, що на заданій множині виконується відношення «бути молодшим за званням», яке описується співвідношенням «х є молодшим за званням, ніж у». Зазначимо, що співвідношення «бути молодшим за віком» на цій множині, взагалі кажучи, не збігається із заданим відношенням (наприклад, генерал може бути молодшим за полковника або одного з ним віку). Отже, оскільки побудоване відношення є транзитивним і симетричним, це відношення строгого порядку. Крім того, воно виконується для будь-яких елементів множини, які розглядаються. Отже, цей порядок є лінійним [7].

Вагові функції. Нехай $f: X \to R$ є відображенням, заданим на множині X. Це означає, що кожному елементу $x \in X$ відповідає деяке дійсне число $y = f(x)$, яке називається вагою. Відображення f при цьому має назву вагової функції.

Іноді поняття ваги збігається з буквальним значенням цього слова (наприклад, маса деталі), а іноді ні (це може бути будь-яка числова характеристика об'єкта, наприклад опір резистора, об'єм тіла, площа геометричної фігури).

Теорема 4.4. *Якщо відображення $f: X \to R$ взаємно однозначне (ін'єктивне), то на множині X можна встановити абсолютно строгий порядок.*

Доведення. Задамо на множині X відношення $\rho: x\rho y$, якщо в R справджується $f(x) < f(y)$. Покажемо, що це відношення є асиметричним і транзитивним.

Для будь-яких двох x й y, для яких виконується співвідношення $x\rho y$, маємо $f(x) < f(y)$, тобто нерівності $f(y) > f(x)$ не може бути; тому маємо $\overline{y\rho x}$. Отже, введене відношення ρ — асиметричне.

Для будь-яких трьох елементів $x, y, z \in X$, для яких $x\rho y \wedge y\rho z$, це означає $f(x) < f(y) \wedge f(y) < f(z)$ внаслідок транзитивності відношення « $<$ » $f(x) < f(z)$, тобто $x\rho z$.Отже, введене відношення ρ — транзитивне.

Таким чином, доведено, що введене відношення ρ є відношенням строгого порядку. Покажемо, що множина X з уведеним відношенням строгого порядку ρ є абсолютно впорядкованою. Розглянемо будь-які два елементи $x, y \in X$ $(x \neq y)$. Внаслідок того, що $x \neq y$, а відображення f взаємно однозначне, виконується нерівність $f(x) < f(y)$ або $f(y) < f(x)$. Це означає, що справджується відношення $x\rho y$ або $y\rho x$. Отже, будь-які два елементи множини X порівнянні, а тому множина X є абсолютно (лінійно) впорядкованою.

Прикладом абсолютно впорядкованої множини з відношенням строгого порядку, заданим ваговою функцією, може бути множина елементів періодичної системи Менделєєва.

Квазіпорядок. Якщо відображення $f: X \to R$ не взаємно однозначне (не ін'єктивне), то для двох різних елементів x, $y \in X$ може виконуватись рівність $f(x) = f(y)$. Тому абсолютно строгий порядок задати на множині X не можна. Водночас якщо об'єднати в окремі класи X_i, i = 1, 2,..., елементи, вага яких однакова, то матимемо розбиття множини X на класи еквівалентності.

Тепер можна говорити про впорядкування сукупності класів еквівалентності $\{X_1, X_2, ...\}$ за їхніми представниками $\alpha_1, \alpha_2, \alpha_3, ...$, де $\alpha_i \in X_i$. Оскільки система представників не містить однакових елементів, у цій системі можна задати абсолютно строгий порядок: $\alpha_i \rho \alpha_j \Leftrightarrow f(\alpha_i) < < f(\alpha_j)$.

Таке впорядкування ототожнює елементи множини X, які належать одному й тому самому класу еквівалентності, і задає на цій множині квазіпорядок (майже порядок). Також кажуть, що строгий порядок на множині класів еквівалентності $\{X_1, X_2, ...\}$ множини X індукує квазіпорядок на цій множині.

Якщо на множині X введений квазіпорядок, то класи еквівалентності множини X, на яких вагова функція набуває фіксованих значень, називаються областями рівня.

Приклад. Для порівняння комплексних чисел $z = a + bi$ не підходять звичні відношення порядку ($<$, $\leq$, $>$, $\geq$). Однак можна ввести квазіпорядок A за правилом $z_k A z_n$, якщо $a_k = a_n$.

При цьому різні комплексні числа з однаковими дійсними частинами об'єднуються в класи еквівалентності, множина яких може бути впорядкована за їхніми представниками.

Структура впорядкованих множин. Нехай A — відношення порядку на множині X, а $X_1 \subset X$ (під A розуміють або σ, або ρ).

Означення 4.10. *Мажорантою (верхньою межею, верхньою гранню) підмножини X_1 називають такий елемент $m \in X$, що для будь-якого елемента $q \in X_1$ справджується відношення qAm.*

Мінорантою (нижньою межею, нижньою гранню) підмножини X_1 називають такий елемент $n \in X$, що для будь-якого елемента $q \in X_1$ справджується відношення nAq.

Окремо розглянемо випадок, коли $X_1 = X$ й $A = \rho$. Тоді мажорантою множини X називається такий елемент $m \in X$, що для будь-якого елемента $q \in X \setminus \{m\}$ справджується відношення $q\rho m$, а мінорантою множини X — такий елемент $n \in X$, що для будь-якого елемента $q \in X \setminus \{n\}$ справджується відношення $n\rho q$.

Якщо мажоранта $m \in X_1$, то m називають максимальним елементом X_1 (позначають max (X_1)).

Якщо міноранта $n \in X_1$, то n називають мінімальним елементом X_1 (позначають min (X_1)).

Твердження 4.3. *Якщо максимальний елемент існує, то він єдиний.*

Доведемо твердження від супротивного: нехай існує два максимальних елементи m_1 = max (X_1) й m_2 = max (X_1). Однак тоді m_2Am_1 суперечить тому, що m_2 — максимальний елемент, а m_1Am_2 — тому, що m_1 — максимальний елемент. Одержані суперечності доводять твердження.

Аналогічно доводиться єдиність мінімального елемента підмножини X_1, якщо він існує.

Підмножина $X_1 \subset X$ може мати кілька мажорант і мінорант. Якщо множина мажорант M_1 підмножини $X_1 \subset X$ має мінімальний елемент min (M_1), то він називається точною верхньою межею підмножини X_1 та позначається sup (X_1) (скорочено від supremum). Якщо множина мінорант M_2 підмножини X_1 має максимальний елемент max (M_2), то він називається точною нижньою межею підмножини X_1 і позначається inf (X_1) (скорочено від infimum).

Матриця відношення нестрогого порядку. Оскільки відношення нестрогого порядку є рефлексивним, головна діагональ матриці цього відношення містить одиниці. Через те що воно є асиметричним, жоден одиничний елемент не має симетричного собі відносно головної діагоналі. Оскільки це відношення є транзитивним, наявність одиниць на перетині *i*-го стовпця та *j*-го рядка й одиниці на перетині j-го стовпця і *k*-го рядка спричинює наявність одиниці на перетині *i*-го стовпця та *k*-го рядка.

Приклад. Для відношення нестрогого порядку A = «бути дільником» на множині $X = \{1, 2, 3, 4\}$ й $Y = \{1, 2, 3, 4, 6, 7, 12, 14, 21, 28, 42, 84\}$ матриці мають такий вигляд [1]:

$$A \subset X \times X = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 \\ 4 & 1 & 1 & 0 & 1 \end{array};$$

$$A \subset Y \times Y = \begin{array}{c|cccccccccccc} & 1 & 2 & 3 & 4 & 6 & 7 & 12 & 14 & 21 & 28 & 42 & 84 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 12 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 14 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 21 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 28 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 42 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 84 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}.$$

Матриця відношення строгого порядку. Матриця відношення строгого порядку будується аналогічно матриці нестрогого порядку і різниться тільки наявністю нулів на головній діагоналі (внаслідок властивості антирефлексивності).

Приклад. Для відношення строгого порядку « > » на множині $X = \{1, 2, 3, 4, 5\}$ матриця має вигляд

	1	2	3	4	5
1	0	1	1	1	1
2	0	0	1	1	1
3	0	0	0	1	1
4	0	0	0	0	1
5	0	0	0	0	0

Граф відношення нестрогого порядку. Внаслідок виконання властивостей рефлексивності, асиметричності та транзитивності граф відношення нестрогого порядку характеризується тим, що в кожній його вершині існує петля, жодна пара вершин не пов'язана дугами протилежного напрямку, а всі його вершини будь-якого шляху попарно пов'язані між собою дугами в напрямку цього шляху. Як звичайно, граф транзитивного відношення будемо зображати графом редукції.

Приклад. Як приклад для побудови графа відношення нестрогого порядку розглянемо відношення з прикладу на с. 100. На рис. 35 зображено його граф редукції з петлями.

Граф відношення строгого порядку. Граф відношення строгого порядку будується аналогічно графу нестрогого порядку і різнитися тільки відсутністю петель.

Приклади: 1. Для побудови графа строгого порядку скористаємося прикладом наведеної вище матриці. Граф відношення строгого порядку « > » на множині $X = \{1, 2, 3, 4, 5\}$ має вигляд, показаний на рис. 36, *а*. Його граф редукції зображено на рис. 36, *б*.

2. Розглянемо ще раз відношення нестрогого порядку A з прикладу на с. 100, граф якого показано на рис. 35. Нехай $X_1 = \{4, 6, 14, 28, 42\}$ — підмножина множини X. Тоді множина мажорант — $\{84\}$, множина мінорант — $\{1, 2\}$. Мінімального й максимального елементів у множині X_1 немає; тому $\sup(X_1) = 84$, а $\inf(X_1) = 2$.

Для $X_1 = X$ множина мажорант — $\{84\}$, множина мінорант — $\{1\}$, $\max(X) = 84$, $\min(X) = 1$, $\sup(X) = 84$, $\inf(X) = 1$.

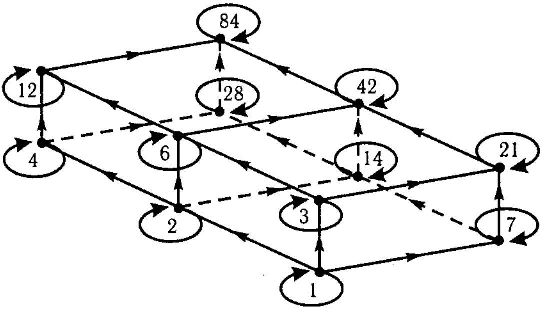


Рис. 35

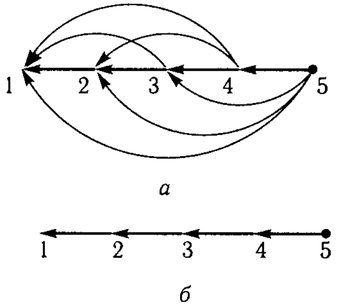


Рис. 36

Означення 4.11. *Відношення τ на множині X, що задовольняє властивості рефлексивності $\forall x \in X(x\tau x)$ та симетричності $\forall x, y(x\tau y \Leftrightarrow y\tau x)$, називається відношенням толерантності.*

Приклади: **1.** Як приклад відношення толерантності можна навести відношення «відстань між двома точками не перевищує деякого заданого числа a». Це означає, що толерантними є будь-які дві точки, відстань між якими не перевищує a (очевидно, що це — відношення толерантності).

Як застосування цього відношення можна запропонувати моделювання зорового органа, для якого в межах гостроти зору точки є невиразними між собою.

2. Між чотирилітерними словами можна встановити відношення толерантності, якщо вони різняться не більш як однією літерою. Як розважальний приклад у цьому випадку можна навести такий ланцюжок толерантних російських слів:

муха → мура → тура → тара → кара → каре → кафе → кафр →
→ каюр → каюк → крюк → крок → срок → сток → стон → слон.

3. На множині кортежів (векторів) $X = \{(x_1, x_2, ..., x_n)\}$ толерантність можна задати різними способами, наприклад обумовити наявність у парі кортежів хоча б однієї загальної компоненти. Компонентами кортежу можуть бути будь-які об'єкти. Якщо вони набувають цілочислового значення від 0 до m–1, то кортеж можна розглядати як n-розрядне число, записане в системі числення з основою m.

Наприклад, кортеж (9, 3, 0, 4, 5, 8) — це десяткове число 930458. Кількість усіх таких кортежів дорівнює m^n. При $m = 2$ маємо двійковий кортеж, його компоненти набувають значення 0 і 1. Для кожного кортежу $(x_1, x_2, x_3, ..., x_n)$ існує тільки один нетолерантний йому кортеж $(1–x_1, 1–x_2, 1–x_3, ..., 1–x_n)$.

4.4. Відношення рівнопотужності. Потужність множин

Означення 4.12. *Множина A рівнопотужна множині B (потужність множини A дорівнює потужності множини B), якщо існує взаємно однозначна відповідність (бієкція) множини A на множину B. Цей факт записують як $A \sim B$ або $|A| = |B|$.*

Розглянемо множину натуральних чисел N та її підмножини $N_k = \{1, 2, 3, ..., k\}$. Множина A, рівнопотужна підмножині натурального ряду N_k, називається скінченною, а k — потужністю множини, або її кардинальним числом. Для скінченної множини частіше вживається термін «кількість елементів». Записується цей факт як $|A| = k$.

Відношення рівнопотужності має такі властивості:

- рефлексивності ($A \sim A$), що очевидно;
- симетричності $(A \sim B \Leftrightarrow B \sim A)$ — випливає з того, що відображення, обернене до взаємно однозначного, також є взаємно однозначним;
- транзитивності $(A \sim B \wedge B \sim C \Leftrightarrow A \sim C)$ — випливає з того, що композиція двох взаємно однозначних відображень також взаємно однозначна.

Таким чином, відношення рівнопотужності є відношенням еквівалентності (часто називається відношенням кардинальної еквівалентності) й індукує розбиття множини всіх множин на неперерізні класи однакових за потужністю множин.

Розглянемо сім'ю скінченних множин A_i, $i = \overline{1, n}$.

Твердження 4.4. *Якщо* $|A_1| = m_1$, $|A_2| = m_2$, ..., $|A_n| = m_n$, *то* $|A_1 \times A_2 \times \ldots \times A_n| = m_1 \times m_2 \times \ldots \times m_n$.

Доводиться твердження за методом індукції. Справді, при $n = 1$ воно є правильним. Припустимо, що це твердження є правильним при $n = k$ і доведемо його при $n = k + 1$. Для цього кожному кортежу $\tilde{a}_k$ з $A_1 \times A_2 \times \ldots \times A_k$ припишемо праворуч елемент з A_{k+1}. Оскільки $|A_{k+1}| = m_{k+1}$, для кожного кортежу $\tilde{a}_k$ це можна зробити m_{k+1} способом. Кортеж $\tilde{a}_k$ з $A_1 \times A_2 \times \ldots \times A_k$ можна вибрати $m_1 \times m_2 \times \ldots \times m_k$ способами. Таким чином, всього є $m_1 \times m_2 \times \ldots \times m_{k+1}$ кортежів, що й треба було довести.

Наслідок 4.1. *Якщо* $A_i = A\ \forall\, i = 1, n$, *то* $|A^n| = |A|^n$.

Теорема 4.5. *Якщо* $|A| = n$, *то* $|P(A)| = 2^n$.

Доведення. Занумеруємо елементи множини A натуральними числами 1, 2, ..., n, тобто $A = \{a_1, a_2, ..., a_n\}$, і розглянемо множину B_n усіх двійкових кортежів завдовжки n. Кожній підмножині $A^* \subset A$ поставимо у відповідність кортеж $(v_1, v_2, ..., v_n) \in B_n$ таким чином:

$$v_i = \begin{cases} 0, & \text{якщо} \quad a_i \notin A^*; \\ 1, & \text{якщо} \quad a_i \in A^*. \end{cases}$$

При цьому порожній підмножині $\varnothing \subset A$ відповідатиме кортеж (0, 0, ..., 0), а множині $A \subset A$ — кортеж (1, 1, ..., 1). Очевидно, що ми дістали відображення і воно є бієктивним. Отже, $|P(A)| = |B_n|$.

Оскільки $B_n = B^n$, де $B = \{0, 1\}$ — двоелементна множина, $|B_n| = |B|^n = 2^n$, що й треба було довести.

Властивості скінченних множин. Їх три:

1. *Переріз скінченного числа скінченних множин є скінченним.*
2. *Об'єднання скінченного числа скінченних множин — скінченне, причому* $\left|\bigcup_{i=1}^{m} A_i\right| \le \sum_{i=1}^{m} |A_i|$.
3. *Декартів добуток скінченного числа скінченних множин є скінченним* (див. твердження 4.4).

Множини, що не є скінченними, називаються нескінченними.

Нескінченна множина, рівнопотужна множині натуральних чисел N, називається зчисленною. Її потужність (або кардинальне число) дорівнює $\aleph_0$ (алеф-нуль).

Властивості зчисленних множин. Їх чотири:

1. *Будь-яка підмножина зчисленної множини — скінченна або зчисленна.*

Справді, будь-яка підмножина зчисленної множини — або скінченна, або нескінченна. Нехай нескінченною є множина $N' \subset N$.

Виберемо в N' найменший елемент і позначимо його n_1; в $N' - \{n_1\}$ такий елемент позначимо n_2; найменший елемент в $N' - \{n_1, n_2\}$ позначимо n_3 тощо. Оскільки для кожного натурального числа маємо тільки скінченну множину менших натуральних чисел, будь-який елемент N' рано чи пізно матиме свій номер. Ця нумерація, тобто відображення (n_i, i), і є взаємно однозначним відображенням між N' та N.

Тому часто зчисленною називають множину, рівнопотужну натуральному ряду або будь-якій його нескінченній підмножині.

2. *Об'єднання скінченного числа зчисленних множин є зчисленним.*

Справді, перенумеруємо спочатку всі перші елементи множин A_1, A_2, ..., A_k, потім — усі другі тощо.

3. *Об'єднання зчисленного числа скінченних множин є зчисленним.* Справді, спочатку нумеруються всі елементи першої множини, потім — усі елементи другої множини тощо.

З останнього твердження випливає, що множина всіх слів у будь-якому скінченному алфавіті — зчисленна. Менш очевидно, що зчисленним є об'єднання зчисленної множини зчисленних множин. Прикладом такого об'єднання може бути множина всіх векторів із натуральними компонентами.

4. *Об'єднання зчисленного числа зчисленних множин є зчисленним.*

Справді, нехай існує деяке зчисленне число зчисленних множин.

Оскільки множин — зчисленне число, занумеруємо їх A_1, A_2, A_3, ... Через те що кожна множина є зчисленною, занумеруємо елементи цих множин і розташуємо у порядку зростання їхніх номерів для множини

$$A_i\left(a_1^i, a_2^i, a_3^i, \ldots\right).$$

У підсумку матимемо елементи всіх зчисленних множин у своєму розпорядженні і застосуємо канторівську нумерацію:

$$\begin{tikzcd}
A_1 & a_1^1 \arrow[d] & a_2^1 \arrow[r] & a_3^1 \arrow[dl] & a_4^1 & \cdots \\
A_2 & a_1^2 \arrow[ur] & a_2^2 \arrow[dl] & a_3^2 \arrow[ur] & a_4^2 & \cdots \\
A_3 & a_1^3\ 1 \arrow[d] & a_2^3 \arrow[ur] & a_3^3 & a_4^3 & \cdots \\
A_4 & a_1^4 \arrow[ur] & a_2^4 & a_3^4 & a_4^4 & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{tikzcd}$$

Очевидно, ця нумерація забезпечує бієкцію об'єднання зчисленного числа зчисленних множин на N. Отже, воно є зчисленне, що й треба було довести.

Очевидно, для доведення властивостей 2 та 3 також можна застосувати канторівську нумерацію або відразу вважати ці властивості наслідками властивості 4.

Із властивості 4 випливають такі наслідки.

Наслідок 4.2. *Об'єднання всіх скінченних підмножин зчисленної множини — зчисленна множина.*

Наслідок 4.3. *Декартів добуток скінченного числа зчисленних множин — зчисленний.*

Теорема 4.6 (теорема Кантора). *Множина всіх дійсних чисел відрізка* [0,1] *не є зчисленною.*

Доведемо теорему від супротивного. Припустимо, що задана множина є зчисленною й існує її нумерація. Розташуємо всі числа, зображені нескінченними десятковими дробами, у порядку цієї нумерації:

$$\begin{array}{lllll} 0, & a_{11} & a_{12} & a_{13} & \dots \\ 0, & a_{21} & a_{22} & a_{23} & \dots \\ 0, & a_{31} & a_{32} & a_{33} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{array}$$

Розглянемо будь-який нескінченний дріб 0, b_1 b_2 b_3 ... такий, що $b_1 \neq a_{11}, b_2 \neq a_{22}, b_3 \neq a_{33}$ тощо. Цей дріб не може ввійти у вказану послідовність, оскільки від першого числа він відрізняється першою цифрою, від другого — другою тощо. Отже, всі числа з відрізка [0, 1] не можуть бути пронумеровані, так що множина всіх дійсних чисел відрізка [0, 1] є незчисленною. Її потужність називається континуумом; множини такої потужності називаються континуальними. Метод, використаний при доведенні цієї теореми, називається діагональним методом Кантора. Потужність незчисленної множини дорівнює $\aleph_1$ (алеф-один).

Наслідок 4.4. *Будь-який відрізок дійсної осі має потужність континуум.*

Теорема 4.7. *Множина правильних дробів має потужність континуум.*

Доведення цієї теореми, як показано в [2], ґрунтується на поданні дійсних чисел відрізка [0, 1] у вигляді двійкових дробів. Будь-яке дійсне число α, що лежить на відрізку [0, 1], можна записати у вигляді

$$\alpha = \frac{\alpha_1}{2} + \frac{\alpha_2}{2^2} + \dots + \frac{\alpha_k}{2^k} + \dots, \qquad (4.1)$$

де $\alpha_i = 0$ або 1, тобто подати у двійковій системі числення аналогічно тому, як подають його в більш звичній десятковій системі.

Запишемо скорочено

$$\alpha = 0, \alpha_1 \alpha_2 \dots \alpha_k \dots . \qquad (4.2)$$

Слід зазначити, що раціональне число вигляду $\frac{P}{2^k}$ можна записати у вигляді двійкового дробу двома способами: або з нулем, або з одиницею в періоді.

Наприклад, для дробу 3/4 маємо:

$$\frac{3}{4} = \frac{1}{2} + \frac{1}{2^2} + \frac{0}{2^3} + \frac{0}{2^4} + \dots = 0{,}1100\dots;$$

$$\frac{3}{4} = \frac{1}{2} + \frac{0}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \dots = 0{,}10111\dots .$$

Якщо не користуватися записом, що містить одиницю в періоді, то кожне число $\alpha \in [0,1]$ однозначно подається у вигляді двійкового дробу (4.2), тобто множина правильних двійкових дробів рівнопотужна множині дійсних чисел на відрізку [0, 1] і має потужність континуум, що й треба було довести.

Теорема 4.8. *Множина всіх підмножин зчисленної множини є незчисленною.*

Для доведення скористаємося тим самим методом, що й при доведенні властивості 3 зчисленних множин.

Нехай A — зчисленна множина. Занумеруємо елементи множини $A = \{a_1, a_2, a_3, ...\}$і поставимо у відповідність кожній підмножині $A^* \subset A$ двійковий дріб $\{0, v_1\ v_2 ...\}$, де

$$v_i = \begin{cases} 1, & \textit{якщо елемент} \quad a_i \in A^*; \\ 0, & \textit{якщо елемент} \quad a_i \notin A^*. \end{cases}$$

Маємо взаємно однозначну відповідність (бієкцію) між множиною всіх підмножин зчисленної множини A та множиною правильних двійкових дробів, яка є незчисленною.

Теорема 4.9. *Об'єднання нескінченної множини зі зчисленною — рівнопотужне початковій множині.*

Справді, якщо початкова нескінченна множина є зчисленною, то об'єднання її з іншою зчисленною множиною — зчисленне (властивість 2 зчисленних множин). Якщо початкова нескінченна множина є незчисленною, то об'єднання її зі зчисленною множиною, очевидно, не змінить його потужності.

Наслідок 4.5. *Множина, утворена внаслідок вилучення з нескінченної множини елементів зчисленної — рівнопотужна початковій.*

Парадокс Кантора. Одна з основних труднощів полягає в тому, що навіть із множин, точність опису яких не викликає сумнівів, за допомогою начебто цілком законних засобів можна сконструювати описи множин, що приводять до суперечностей, «парадоксів теорії множин». Як приклад можна назвати «множину всіх множин». За означенням вона повинна містити всі мислимі множини і мати максимальну потужність. Однак вона сама міститься в множині своїх підмножин як елемент.

За допомогою методу, аналогічного діагональному методу Кантора, можна довести, що для множини будь-якої потужності множина всіх її підмножин має вищу потужність (як було показано вище для скінченних і зчисленних множин), тобто не існує множини максимальної потужності. Парадокс Кантора якраз і полягає в тому, що «множина всіх множин» повинна мати максимальну потужність, а це суперечить результатам теорії множин.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке бінарне відношення? Наведіть приклади.
2. Що таке фактор-множина?
3. Що таке повне, тотожне і порожнє відношення?

4. Як можна довести тотожність $(BoA)^{-1} = A^{-1}oB^{-1}$?
5. Що таке функціональне відношення? Наведіть приклади.
6. Що таке сюр'єкція, ін'єкція, бієкція?
7. Який приклад ілюструє ін'єктивне функціональне відношення, що не є сюр'єктивним?
8. Що таке відношення еквівалентності? Наведіть приклади.
9. Яке розбиття індукує відношення еквівалентності «вчитися в одній групі», задане на множині всіх студентів вищого навчального закладу?
10. Яке розбиття індукує відношення рівності на множині цілих чисел?
11. Чи можна за графом відношення зробити висновок про те, що це граф саме відношення еквівалентності?
12. Чи можна за матрицею відношення дійти висновку про те, що це матриця саме відношення еквівалентності?
13. Які приклади ілюструють відношення нестрогого порядку?
14. Які приклади ілюструють відношення строгого порядку?
15. Що таке абсолютно впорядкована множина? Наведіть приклади.
16. Які приклади ілюструють частково впорядковані множини?
17. Які приклади ілюструють вагові функції?
18. Що таке мажоранта, міноранта, мінімальний і максимальний елементи підмножин упорядкованої множини?
19. Чи завжди впорядкована множина має точну верхню (нижню) межу?
20. Які приклади ілюструють відношення толерантності?
21. Що таке рівнопотужні множини?
22. Які властивості має потужність скінченних множин?
23. Яка множина називається зчисленною? Наведіть приклади.
24. Які приклади ілюструють множини потужності континуум?

ПЕРЕЛІК ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

1. Властивості бінарних відношень. Побудова фактора-множини. Побудова матриць і графів відношень.
2. Відношення еквівалентності. Розбиття. Побудова матриці та графа відношення еквівалентності.
3. Відношення нестрогого і строгого порядків. Визначення типу впорядкованості множин.
4. Структура впорядкованих множин.
5. Потужність скінченних, зчисленних, континуальних множин.

СПИСОК ЛІТЕРАТУРИ

1. *Сигорский В. П.* Математический аппарат инженера. — К.: Техніка, 1977. — 768 с.
2. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера. — М.: Энергоатомиздат, 1988. — 480 с.
3. *Лапа В. Т.* Математические основы кибернетики. — К.: Вища шк., 1971. — 420 с.
4. *Шиханович Ю. А.* Введение в современную математику. — М.: Наука, 1965.
5. *Шрейдер Ю. А.* Равенство, сходство, порядок. — М.: Наука, 1965.
6. *Яблонский С. В.* Введение в дискретную математику. — М.: Наука, 1986. — 384с.
7. *Нефедов В. Н., Осипова В. А.* Курс дискретной математики. — М.: Изд-во МАИ, 1992. — 264 с.
8. *Шабанов-Кушнаренко Ю. П., Бондаренко М. Ф.* Математическая логика для инженера. — Харьков.: Изд-во ХТУРЭ, 1995.

РОЗДІЛ 5

АЛГОРИТМИ

5.1. Основні поняття і властивості алгоритмів

З давніх часів у математиці склалося інтуїтивне уявлення про алгоритм як формальне розпорядження, що визначає сукупність операцій і порядок їх виконання для розв'язання задач деякого типу. Термін походить від латинізованої вимови (Algorithmi) імені середньовічного узбецького математика аль-Хорезмі, який ще в IX ст. сформулював правила, що дають змогу складати та розв'язувати квадратні рівняння.

З алгоритмами, тобто ефективними процедурами, які однозначно приводять до результату, математика мала справу завжди й у своєму розвитку накопичила безліч різних алгоритмів. Дістаючи відповідну інтерпретацію в конкретних додатках, вони становлять значну і найсуттєвішу частину математичного апарату, який використовується в техніці. Відомі зі шкільної програми методи множення «стовпчиком» та ділення «кутом», алгоритм Евкліда знаходження найбільшого спільного дільника двох додатних натуральних чисел, метод Гаусса розв'язання системи лінійних рівнянь, правило диференціювання складної функції, добування квадратного кореня з раціонального числа із заданою мірою точності, обчислення рангу цілочислової матриці, означення тотожної істинності формули логіки висловлень, спосіб побудови трикутника за трьома заданими сторонами — все це алгоритми.

Поки математика мала справу в основному з числами й обчисленнями, а поняття алгоритму ототожнювалося з поняттям методу обчислення, необхідності у вивченні самого поняття алгоритму не виникало. Традиції організації обчислень складалися віками і стали складовою частиною загальної наукової культури в тій самій мірі, що й елементарні навички логічного мислення. У своєму розвитку математика накопичила величезну кількість алгоритмів, але все різноманіття обчислень комбінувалося з невеликого числа чітко визначених операцій арифметики, тригонометрії та аналізу. Тому поняття методу обчислення вважалося спочатку інтуїтивно зрозумілим і не потребувало спеціальних досліджень.

Подальший процес розвитку обчислювальних методів сприяв утвердженню думки про те, що розв'язок будь-якої математичної проблеми має бути алгоритмічним. Такої думки, наприклад, дотримувались визначні математики Р. Декарт, Г. В. Лейбніц, Д. Гільберт. Останній активно стимулював алгоритмічні дослідження. Серед його знаменитих 23 проблем, сформульованих у 1900 р. на Міжнародному математичному конгресі, є десята проблема Гільберта: дано алгебричне рівняння $f(x_1, ..., x_n)=0$ з цілими коефіцієнтами. Потрібно знайти метод, згідно з яким за допомогою скінченного числа операцій можна було б встановити, чи має це рівняння розв'язки в цілих числах. Метод, про який ідеться в цій проблемі, розуміють нині як алгоритм.

У той час теорія алгоритмів ще не була створена і мова могла йти лише про позитивне вирішення цієї проблеми. Проте у 1970 р. радянський математик Ю. В. Матіясевич довів, що такого алгоритму не існує, хоча вперше віра в універсальність алгоритмічних методів була підірвана в 1931 р. працею К. Геделя «Про формально нерозв'язувані висловлення principia mathematica і споріднених систем». У ній було показано, що є такі математичні проблеми, які не можуть бути розв'язані за допомогою алгоритмів із деякого точно визначеного класу алгоритмів. Таким чином, у теорії алгоритмів виникла зовсім нова ситуація.

До того часу, поки математики вірили в можливість алгоритмічного розв'язання математичних задач, не виникала потреба уточнювати поняття алгоритму: якщо для розв'язання деякого класу задач пропонувався конкретний алгоритм, то вчені доходили згоди вважати його дійсно шуканим алгоритмом. І тільки доведення того, що не існує алгоритму для розв'язання певного класу задач, яке є висловлюванням про всі можливі алгоритми, потребує уточнення цього важливого поняття.

Таким чином, до уточнення поняття «алгоритм» у математиці були поширені дві точки зору:

1. *Усі проблеми є алгоритмічно розв'язуваними.* Проте алгоритм для вирішення деяких з них не знайдено, тому що не вистачає засобів у сучасній математиці для його побудови. Прихильники цієї думки вважали, що для вирішення проблем, які називаються алгоритмічно нерозв'язуваними, просто не вистачає засобів сучасної математики, побудова шуканих алгоритмів — справа майбутнього.

2. Існують *класи задач, для розв'язання яких узагалі не існує алгоритму,* тобто деякі проблеми не можна вирішувати механічно за допомогою формальних міркувань й обчислень. Ці проблеми потребують творчого мислення. Це дуже сильне твердження, адже воно поширюється на всі майбутні часи і засоби.

Поки визначення алгоритму ґрунтувалось на інтуїтивному уявленні, про математичне доведення правоти другої думки не могло бути й мови. Завдяки появі гіпотез про існування стандартних форм, у яких можуть бути виражені будь-які алгоритми, стало можливим сформулювати поняття «алгоритм» та «алгоритмічно нерозв'язувана проблема» у точних математичних термінах.

Починаючи з 1935 р., було запропоновано кілька уточнень поняття алгоритму. Математики погодились із тим, що уточнені поняття адекватно виражають інтуїтивне уявлення про алгоритм. Для цього є такі підстави:

- доведено, що всі відомі на цей час уточнені поняття алгоритму еквівалентні між собою;

- усі алгоритми в точному розумінні є алгоритмами в інтуїтивному розумінні;
- усі відомі алгоритми в інтуїтивному розумінні можна «промоделювати» алгоритмами в точному розумінні.

Довести адекватність інтуїтивного і точного понять алгоритмів неможливо тому, що не існує точного означення алгоритму в інтуїтивному розумінні. Натомість ця адекватність широко використовується як гіпотеза. Першим таку гіпотезу в 1936 р. висловив А. Черч. Нині вона відома як теза Черча. Отже, ця гіпотеза не може бути доведена математичними методами, оскільки її сформульовано на основі практичного досвіду і вона не має внутріматематичного характеру. Її правильність підтверджується практикою. Аналогічна ситуація є, наприклад, у фізиці: закони Ньютона підтверджуються практикою, але не мають математичного доведення.

Досвід парадоксів теорії множин навчив математику обережно оперувати нескінченністю та, якщо це можливо, навіть про нескінченність міркувати за допомогою методів, в яких використовуються тільки скінченні множини скінченних об'єктів. З'ясування того, які об'єкти, а також які дії над ними можна вважати точно визначеними, які властивості та можливості мають комбінації елементарних дій, що можна і чого не можна робити за їх допомогою — все це стало предметом теорії алгоритмів та формальних систем, яка спочатку виникла в межах метаматематики і стала найважливішою її частиною. Основним внутрішнім математичним додатком теорії алгоритмів виявилися докази неможливості алгоритмічного (тобто точного й однозначного) розв'язання деяких математичних проблем. Такі докази (як і точні формулювання тверджень, що доводяться) нездійсненні без точного поняття алгоритму.

У техніку термін «алгоритм» введено разом із терміном «кібернетика». Якщо поняття методу обчислення не потребувало пояснень, то поняття процесу управління довелося формулювати практично заново. Знадобилося усвідомити, які вимоги має задовольняти послідовність дій (або її опис), щоб вважатися конструктивно заданою, тобто мати право називатися алгоритмом. У цьому усвідомленні величезну допомогу інженерній інтуїції надала практика використання обчислювальних машин, що зробила поняття алгоритму відчутною реальністю. Традиційно процес розв'язування задачі на ЕОМ складається із таких етапів:

- постановка задачі, тобто чітке її формулювання із зазначенням мети, яку потрібно досягти. Тут виділяються початкові дані, результати і визначається зв'язок між початковими даними та результатами;
- створення математичної моделі, тобто визначення математичних співвідношень (формул, рівнянь тощо), що пов'язують результати з початковими даними;
- розробка алгоритму розв'язання задачі, тобто побудова схеми (набору інструкцій) цього процесу;
- написання програми, тобто запис алгоритму розв'язання задачі мовою, зрозумілою для ЕОМ;

- налагодження програми, тобто пошук помилок у програмі й усунення їх;
- розв’язання задачі на комп’ютері, тобто розрахунки за готовою програмою та аналіз одержаних результатів.

У різних задачах деякі з перерахованих етапів можуть бути відсутні, деякі етапи можуть бути поділені на більше число, але схема залишається такою, як викладено вище.

Щодо сучасної практики алгоритм — це програма, а критерієм алгоритмічності процесу є можливість його запрограмувати. Саме завдяки цій реальності алгоритму, а також тому що підхід інженера до математичних методів завжди був конструктивним, поняття алгоритму в техніці за короткий проміжок часу стало надзвичайно популярним (можливо, навіть більше, ніж у самій математиці).

Однак у кожної популярності є свої втрати. У повсякденній практиці слово «алгоритм» вживається дуже широко, втрачаючи часто своє точне значення. Приблизні описи поняття «алгоритм» (на зразок того, яке наведено на початку розділу) часто приймаються як точні означення. У результаті як алгоритм часто вважається будь-яка інструкція, розбита на кроки.

Чітке уявлення про те, що таке алгоритм, важливо не тільки для правильного слововикористання. Воно потрібне також при розробленні конкретних алгоритмів, особливо коли мається на увазі їх подальше програмування. Проте це уявлення ще важливіше для наведення порядку в бурхливо зростаючому алгоритмічному господарстві. Щоб орієнтуватися в морі алгоритмів, яке захлеснуло технічну кібернетику, треба вміти порівнювати різні алгоритми розв’язання одних і тих самих задач, причому не тільки за якістю розв’язання, а й за характеристиками самих алгоритмів (кількістю дій, ємністю пам’яті і т. д.). Таке порівняння неможливе без уведення точної мови для обговорення всіх цих питань; іншими словами, самі алгоритми мають стати такими самими предметами точного дослідження, як і ті об’єкти, для роботи з якими вони призначені.

Таким чином, можна сформулювати основні вимоги до алгоритмів.

1. Будь-який алгоритм застосовується для початкових даних і видає результати. У технічних термінах це означає, що алгоритм має входи й виходи. Крім того, під час роботи алгоритму з’являються проміжні результати, що використовуються далі. Таким чином, кожний алгоритм має справу з даними початковими, проміжними і вихідними. Оскільки маємо намір уточнювати поняття алгоритму, треба уточнити також поняття даних, тобто зазначити, які вимоги мають задовольняти об’єкти, щоб з ними могли працювати алгоритми. Зрозуміло, що ці об’єкти мусять бути чітко визначені і відрізнятися як один від одного, так і від «не об’єктів». У багатьох важливих випадках добре зрозуміло, що це означає: до таких алгоритмічних об’єктів належать числа, вектори, матриці суміжності графів, формули. Замість того, щоб намагатися дати загальне словесне означення чіткої визначеності об’єкта, в теорії алгоритмів фіксують конкретні скінченні набори початкових об’єктів (їх називають елементарни-

ми) та скінченний набір засобів побудови інших об'єктів з елементарних. Набір елементарних об'єктів утворюють скінченний алфавіт початкових символів (цифр, літер тощо), з яких будуються інші об'єкти; типовим засобом побудови є індуктивні означення, які показують, як будувати нові об'єкти з уже побудованих. Найпростіше індуктивне означення — це означення деякої множини слів, класичним прикладом якого є означення ідентифікатора на мові Паскаль: ідентифікатор — це або літера, або ідентифікатор, до якого приписано праворуч літеру чи цифру. Слова скінченної довжини в скінченних алфавітах (зокрема, числа) — звичайний тип алгоритмічних даних, а число символів у слові (довжина слова) — природна одиниця обсягу інформації, що обробляється. Складнішим випадком алгоритмічних об'єктів є формули. Вони також визначаються індуктивно і також є словами в скінченному алфавіті, але не кожне слово в цьому алфавіті є формулою. У цьому випадку, як правило, основним алгоритмам передують допоміжні, які перевіряють, чи задовольняють початкові дані потрібні вимоги. Така перевірка називається синтаксичним аналізом.

2. Дані для їх розміщення потребують пам'яті. Пам'ять вважається однорідною й дискретною, тобто складається з однакових комірок, причому кожна комірка може містити один символ алфавіту даних. Таким чином, одиниці обсягу даних і пам'яті узгоджено. При цьому пам'ять може бути нескінченною.

3. Як виконавець алгоритму можуть виступати людина, різні технічні пристрої, наприклад робот або ЕОМ. У будь-якому випадку має бути досягнута визначена мета, інакше вся виконувана послідовність дій втрачає зміст.

Однак навіть при виконанні цих вимог не всяка послідовність дій є алгоритмом. Для цього треба виконати кілька умов, що визначають властивості алгоритму:

1. Детермінованість — визначення кроків алгоритму, тобто після кожного кроку або зазначається, який крок слід робити далі, або дається команда на зупин, після чого робота алгоритму вважається закінченою. Ця властивість означає, що застосування алгоритму до тих самих даних має привести до одного і того самого результату.

2. Масовість — алгоритм може бути використаний для розв'язання цілого класу задач одного типу, наприклад квадратного рівняння з різними коефіцієнтами.

3. Результативність (скінченність, збіжність) — виконання алгоритму має або закінчуватися результатом, або інформацією про те, чому не може бути одержаний результат, причому множина різних кроків, з яких складено алгоритм, є скінченною. Наприклад, після розв'язання квадратного рівняння дає значення його коренів або інформацію про їх відсутність при від'ємному дискримінанті. Проте перевірити результативність (збіжність) набагато важче, ніж вимоги, викладені вище. На відміну від них збіжність, як правило, не вдається встановити простим переглядом

опису алгоритму; загального ж методу перевірки збіжності, придатного для будь-якого алгоритму і будь-яких даних, як буде показано далі, взагалі не існує.

4. Зрозумілість — алгоритм має бути зрозумілим конкретному виконавцю, який повинен виконати кожну команду алгоритму у строгій відповідності з її призначенням.

5. Дискретність — можливість розбиття алгоритму на скінченну кількість етапів, причому результати попереднього етапу є вхідними для наступного.

Таким чином, кожен алгоритм будується з розрахунку на конкретного виконавця і відповідним чином подається.

Потрібно розрізняти: опис алгоритму (інструкцію або програму); механізм його реалізації (наприклад, ЕОМ), що включає засоби пуску, зупину, реалізації елементарних кроків, видачі результатів та забезпечення детермінованості, тобто керування ходом обчислення; процес реалізації алгоритму, тобто послідовність кроків, яка буде породжена при застосуванні алгоритму до конкретних даних. Будемо припускати, що опис алгоритму і механізм його реалізації — скінченні.

Приклад. Розглянемо як приклад алгоритму таку класичну задачу: дано два натуральних числа X та Y; треба знайти їх найбільший спільний дільник (GCD). У теорію алгоритмів розв'язання цієї задачі ввійшло під назвою алгоритму Евкліда. Можна стверджувати, що метод її розв'язання є таким:

Крок 1. Початок.
Крок 2. Ввести в розгляд числа X й Y.
Крок 3. Якщо $X = Y$, то перейти до кроку 7, інакше — до кроку 4.
Крок 4. Якщо $X < Y$, то перейти до кроку 5, інакше — до кроку 6.
Крок 5. Зменшити Y на значення X, перейти до кроку 3.
Крок 6. Зменшити X на значення Y, перейти до кроку 3.
Крок 7. Прийняти GCD таким, що дорівнює X.
Крок 8. Кінець.

Це був приклад числового алгоритму, який зводить розв'язання наведеної задачі до арифметичних дій над числами. Існує інший тип алгоритмів, що містять розпорядження, яке стосується не цифр, а об'єктів будь-якої природи. Це логічні алгоритми.

Приклад. Типовим прикладом логічних алгоритмів може бути алгоритм пошуку шляху в скінченному лабіринті. Задачу цю розв'язано ще до нашої ери за допомогою відомої Аріадни. Лабіринт зручно зобразити у вигляді графа, де вершини — це майданчики лабіринту, дуги — його коридори. Нехай треба з'ясувати, чи досяжний майданчик f із майданчика a, і якщо це так, то знайти шлях з a в f, а якщо ні — повернутися в a. Припускається, що заздалегідь нічого не відомо про будову цього лабіринту.

Нехай людина, яка шукає шлях у лабіринті, має у своєму розпорядженні нитку, кінець якої закріплено на майданчику a, та, рухаючись по лабіринту, може розмотувати клубок (Р) або, навпаки, намотувати (Н) на нього нитку.

Можна робити позначки на прохідних коридорах і розрізняти потім коридори: ще жодного разу не пройдені (зелені — З), пройдені один раз (жовті — Ж) та пройдені двічі (червоні — Ч). Метод пошуку може бути заданий такою схемою:

1) майданчик f — зупинка (мету досягнуто);

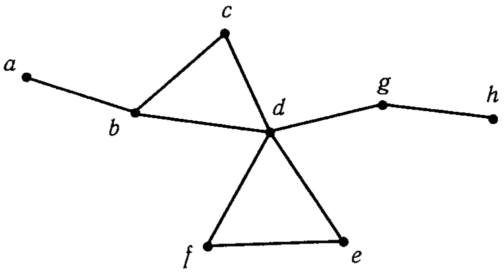


Рис. 37

2) петля (П) — намотування нитки (від майданчика розходяться принаймні два жовтих коридори);

3) зелена вулиця (ЗВ) — розмотування нитки (від майданчика виходить принаймні один зелений коридор);

4) майданчик *a* — зупинка (початковий пункт);

5) відсутність усіх попередніх ознак (ВО) — намотування нитки.

Потрапивши на який-небудь майданчик, потрібно звіритися зі схемою ознак у зазначеному порядку і зробити черговий хід відповідно до першої виявленої ознаки, не перевіряючи інших ознак доти, доки не настане зупинка. Один із можливих варіантів пошуку шляху в лабіринті (рис. 37) містить ходи, наведені в табл. 5.1.

Якщо перерахувати коридори, які залишилися жовтими (*ab*, *bc*, *cd*, *df*), то дістанемо шлях від *a* до *f*.

Таблиця 5.1

Номер ходу	Ознака	Хід	Пройдений коридор	Колір коридору після його проходження
1	ЗВ	Р	*ab*	Ж
2	Те саме	Те саме	*bc*	Те саме
3	— “ —	— “ —	*cd*	— “ —
4	— “ —	— “ —	*dg*	— “ —
5	— “ —	— “ —	*gh*	— “ —
6	ВО	Н	*hg*	Ч
7	Те саме	Те саме	*gd*	Те саме
8	ЗВ	Р	*db*	Ж
9	П	Н	*bd*	Ч
10	ЗВ	Р	*df*	Ж

Існують різні способи опису алгоритмів (словесний, табличний, псевдокод, графічний тощо). Алгоритм для ЕОМ найзручніше зображати графічно у вигляді схем.

Схеми алгоритмів складаються з символів, що мають задане значення, короткого пояснювального тексту і сполучних ліній. Схема програми складається з:

- символів процесу, які показують фактичні операції оброблення даних (включаючи символи, що визначають шлях, якого варто дотримуватись з урахуванням логічних умов);
- лінійних символів, що показують потік керування;
- спеціальних символів, використовуваних для полегшення написання та читання програми.

Існують три основних типи процесів оброблення інформації: лінійний, розгалужувальний і циклічний. При лінійному процесі оброблення інформації здійснюється послідовно, одна дія за іншою, і кожен етап алгоритму виконується тільки один раз. При розгалужуваному процесі інформація обробляється по одному із двох можливих шляхів, тобто ті чи інші дії виконуються залежно від деякої умови. При циклічному процесі ті самі дії з оброблення інформації потрібно виконати багаторазово. Їм відповідають базові структури (конструкції) алгоритмів: проходження (лінійна структура), розгалуження (структура, що розгалужується), повторення (циклічна структура). Реальний алгоритм будь-якого ступеня складності можна задати комбінацією зазначених базових структур. Для першого прикладу на с. 113 граф зображено на рис. 38.

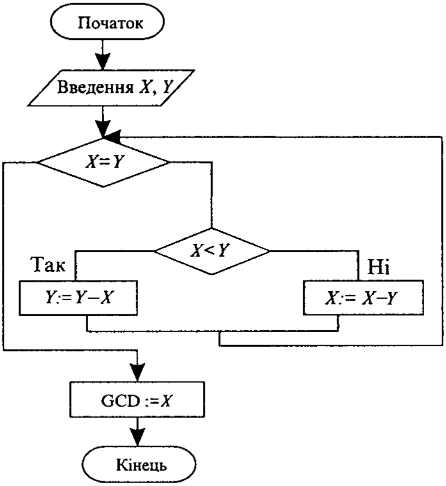


Рис. 38

Такий граф, в якому вершинам відповідають кроки, а ребрам — переходи між ними, називається схемою алгоритму. Його вершини можуть бути двох типів: вершини, з яких виходить одне ребро (їх називають операторами), і вершини, з яких виходять два ребра (їх називають логічними умовами, або предикатами) Крім того, існують єдиний оператор кінця, з якого не виходить жодного ребра, та єдиний оператор початку. Важливою особливістю схеми є те, що зв'язки, які вона описує, не залежать від того, чи є кроки елементарними або самостійними алгоритми, як кажуть у програмуванні, блоками. Можливість «розблокувати» алгоритм добре відома у програмуванні й вона там широко використовується: великий алгоритм поділяється на блоки, які можна роздати для програмування різним особам. Для певного блока неважливо, яку будову мають інші блоки; для програмування блока досить знати, де знаходиться вся початкова інформація, яка форма її подання, що має робити блок і куди записати результат.

За допомогою схем можна, навпаки, кілька алгоритмів, що розглядаються як блоки, зв'язати в один великий алгоритм. Зокрема, якщо алгоритм A_1, який обчислює функцію $f_1(x)$, сполучений з алгоритмом A_2, що обчислює функцію $f_2(x)$ (рис. 39), і при цьому початковими даними для A_2 є результат A_1, то утворена схема задає алгоритм, який обчислює $f_2(f_1(x))$, тобто композицію f_1 й f_2. Таке сполучення алгоритмів називається їх композицією.

На схемі добре видно різницю між описом алгоритму та процесом його реалізації. Опис — це граф; процес реалізації — шлях у ньому. Різні шля-

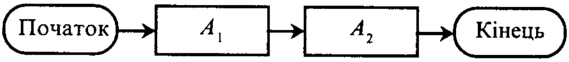


Рис. 39

хи в одному і тому самому графі виникають при різних даних, які створюють різні логічні умови в точках розгалуження. Відсутність збіжності означає, що в процесі обчислення не виникають умови, які ведуть до кінця, і процес іде по нескінченному шляху (зациклюється).

При всій наочності мови схем не треба, однак, переоцінювати її можливості. Вона досить груба та відображає зв'язки лише за управлінням (що робити в наступний момент, тобто якому блоку передати управління), а не за інформацією. Наприклад, рис. 39 при застереженні відносно даних зображає обчислення композиції $f_2(f_1(x))$, однак він же міг відображати послідовне обчислення двох незалежних функцій $f_1(x)$ й $f_2(y)$, порядок яких неістотний.

Схеми не містять відомостей ні про дані, ні про пам'ять, ні про набір елементарних кроків, який використовується. Зокрема, треба мати на увазі, що коли в схемі немає циклів, це ще не означає, що немає циклів в алгоритмі (вони можуть бути в якому-небудь неелементарному блоці). По суті схеми — це не мова, а засіб, хоча дуже зручний для однієї мети — опису детермінізму алгоритму. Корисно ввести одне уточнення: умови, тобто точки розгалуження, можуть бути не тільки двозначними, а й багатозначними; важливо, щоб правильною була лише одна з можливих відповідей (наприклад, оператор CASE в Паскалі).

Виявляється, що будь-який логічний алгоритм можна досить простими методами звести до числового. Тому теорію числових алгоритмів можна вважати універсальним апаратом для дослідження всіх алгоритмічних проблем.

Твердження 5.1. *Будь-яку алгоритмічну проблему можна звести до обчислення значень деякої цілочислової функції цілочислових аргументів.*

Для доведення позначимо всі початкові умови задачі F, які переробляються алгоритмом A, у вигляді послідовності з цілими невід'ємними індексами:

$$a_0, a_1, a_2, ..., a_n, ...$$

Розв'язки теж можна подати занумерованою послідовністю

$$\beta_0, \beta_1, \beta_2, ..., \beta_m, ...$$

Після введення нумерації оперуватимемо не записами умов і розв'язків, а їхніми номерами. Тепер можна подати алгоритм, що переробляє номер запису умов на номер запису розв'язку. Цей алгоритм здійснює обчислення значень числової функції

$$m = \varphi(n),$$

тобто він є числовим алгоритмом.

Якщо існує алгоритм розв'язання початкової задачі F, то є алгоритм, який визначає значення відповідної функції φ. Справді, для знаходження значення $\varphi(n)$ при $n = n^*$ можна вибрати запис умови для n^*, далі за допомогою наявного алгоритму знайти запис розв'язку й за ним визначити відповідний номер m^*. Таким чином,

$$\varphi(n^*) = m^*.$$

Правильним є також твердження, обернене до твердження 5.1: *якщо існує алгоритм визначення функції* $\varphi(n)$, то є *алгоритм розв'язку початкової задачі* F.

Розглянемо метод Геделя, що часто застосовується для нумерації. Подамо деяке число n у вигляді

$$n = 2^{\alpha 1} \cdot 3^{\alpha 2} \cdot 5^{\alpha 3} \cdot 7^{\alpha 4} ... p_n^{\alpha_n},$$

де $p_0 = 2$; $p_1 = 3$; $p_2 = 5$ тощо, тобто p_n — просте число.

Зважаючи на те, що будь-яке число можна розкласти на прості співмножники єдиним чином, можна стверджувати, що кожному числу однозначно відповідає набір $a_1, a_2, ..., a_n$ і, навпаки, кожному набору $a_1, a_2, ..., a_n$ — число n.

Цим способом можна нумерувати будь-які впорядковані послідовності k чисел.

Приклад. Кожній парі чисел a_1 й a_2, для якої y — шуканий найбільший спільний дільник, можна поставити у відповідність геделів номер цієї пари

$$n = 2^{\alpha_1} 3^{\alpha_2}.$$

Тоді алгоритм Евкліда зводиться до обчислення функції $y = \varphi(n)$.

Вище були сформульовані основні властивості алгоритмів. Однак поняття, використані в цих формулюваннях (такі, як детермінованість, результативність тощо), самі потребують уточнення. Очевидно, їх словесні означення міститимуть нові поняття, які знову потрібно уточнювати, тощо. Отже, означення алгоритму не є математично коректним. Тому в теорії алгоритмів користуються іншим підходом: вибирають скінченний набір початкових об'єктів, які оголошуються елементарними, і скінченний набір способів побудови з них нових об'єктів.

Цей підхід був уже використаний при розгляді питання про дані: уточненням поняття «дані» надалі вважатимемо множини слів у скінченних алфавітах. Для уточнення детермінізму будемо використовувати або схеми та словесні описи, еквівалентні їм (наприклад, як наведений у першому прикладі на с. 113, або опис механізму реалізації алгоритму. Крім того, треба зафіксувати набір елементарних кроків і домовитися про організацію пам'яті. Після того, як це буде зроблено, матимемо конкретну алгоритмічну модель.

Алгоритмічні моделі, що розглядаються в цьому розділі, претендують на право вважатися формалізацією поняття «алгоритм». Це означає, що вони мають бути універсальними, тобто допускати опис будь-яких алгоритмів. Тому може виникнути природне заперечення підходу, який пропонується: чи не приведе вибір конкретних засобів до втрати загальності формалізації? Якщо мати на увазі основну мету при створенні теорії алгоритмів — універсальність і пов'язану з нею можливість говорити в межах якої-небудь моделі про властивості алгоритмів узагалі, то ця суперечність знімається так. По-перше, доводиться звідність одних моделей до інших, тобто показується, що кожний алгоритм, описаний засобом однієї моделі, може бути описаний засобами іншої. По-друге, завдяки взаємній звідності моделей в теорії алгоритмів удалося виробити інваріантну відносно моделей систему понять, що дає змогу говорити про властивості алгоритмів незалежно від того, яку формалізацію алгоритму вибрано. Ця система понять ґрунтується на понятті обчислюваної функції, тобто функції, для обчислення якої існує алгоритм.

Однак хоча загальність формалізації у конкретній моделі не втрачається, різний вибір початкових засобів приводить до моделей різного типу. Можна виділити три основних типи універсальних алгоритмічних моделей, які різняться початковими евристичними міркуваннями щодо того, що таке алгоритм. Перший тип зв'язує поняття алгоритму з найтрадиційнішими поняттями математики — обчисленнями та числовими функціями. Найбільш розвинена й вивчена модель цього типу — рекурсивні функції — є історично першою формалізацією поняття алгоритму. Другий тип ґрунтується на уявленні про алгоритм як про деякий детермінований пристрій, здатний виконувати в кожний окремий момент лише дуже примітивні операції. Таке уявлення не залишає сумнівів в однозначності алгоритму й елементарності його кроків. Крім того, евристика цих моделей близька до ЕОМ, а отже, до інженерної інтуїції. Основною теоретичною моделлю цього типу є машина Тьюрінга. Нарешті, третій тип алгоритмічних моделей — це перетворення слів у довільних алфавітах, в яких елементарними операціями є підстановки, тобто заміни частини слова (підслова) іншим словом. Переваги цього типу — в його максимальній абстрактності та можливості застосувати поняття алгоритму до об'єктів довільної (не обов'язково числової) природи. Проте, як буде зрозуміло з подальшого, моделі другого й третього типів досить близькі (їх взаємна звідність доводиться просто) і різняться в основному евристичними акцентами. Приклади моделей цього типу – нормальні алгоритми Маркова.

5.2. Рекурсивні функції

5.2.1. Примітивно-рекурсивні функції

Подамо історично першу формалізацію поняття алгоритму — рекурсивні функції, використовуючи теоретичний матеріал, викладений в [1] і [3]. В теорії рекурсивних функцій, як і загалом у теорії алгоритмів, прийнято конструктивний, фінітний підхід, основною рисою якого є те, що вся множина об'єктів (тут — функцій), що досліджуються, будується зі скінченного числа початкових об'єктів (базису) за допомогою простих операцій, ефективна виконуваність яких досить очевидна. Операції над функціями надалі називатимемо операторами.

Для алгоритмічних проблем типовою є ситуація, коли потрібно знайти алгоритм для визначення функції $f(x_1, ..., x_n)$, яка набуває цілочислових значень та залежить від цілочислових аргументів. У теорії рекурсивних функцій визначають множину натуральних чисел $N = 0, 1, 2, 3, \dots$ і розглядають тільки числові функції, розуміючи під ними функції n змінних, аргументи і значення яких належать N.

Означення 5.1. *Функція* $f(x_1, ..., x_n)$ *називається частково числовою n-місною функцією, якщо* $D_0(f) \subset N^n$, *а* $D_3(f) \subset N^n$, *де* $D_0(f)$ *— область визначення функції* $f(x_1, ..., x_n)$ *а* $D_3(f)$ *— область її значень.*

Якщо $D_0(f) = N^n$, то функція f є всюди визначеною.

Означення 5.2. *Обчислюваною називається числова функція* $f(x_1, ..., x_n)$, *значення якої можна обчислювати за допомогою деякого алгоритму на підставі відомих значень аргументу.*

Найпростіші функції. Зосередимося тепер на конкретному виборі засобів, за допомогою яких будуватимуться обчислювані функції. Опишемо клас числових функцій, що мають вагоме значення в теорії рекурсивних функцій.

Означення 5.3. *Такі всюди визначені числові функції називаються найпростішими:*

1. $O(x) = 0$ (*нуль-функція*);
2. $S(x) = x + 1$ (*функція наступності, або додавання одиниці*);
3. $I_m^n(x_1, ..., x_n) = x_m$, де $m = 1, 2, \dots, n$ (*функція проекції, або функція тотожності, або введення фіктивних змінних*).

Оператори суперпозиції, примітивної рекурсії та мінімізації. Надто могутнім засобом утворення нових функцій з тих, що вже існують, є суперпозиція, знайома з розд. 3 і 4.

Означення 5.4. *Оператором суперпозиції* S_m^n *називається підстановка у функцію m змінних* $f(x_1, ..., x_m)$ *m функцій* $f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)$ *n одних і тих самих змінних. Вона задає нову функцію n змінних.*

Наприклад, для функцій

$$f(x_1, ..., x_m), f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n) \qquad (5.1)$$

маємо

$$S_m^n(f, f_1, ..., f_m) = f(f_1(x_1 ... x_n), ..., f_m(x_1, ..., x_n)) = \varphi(x_1, ..., x_n).$$

Це означення породжує сім'ю операторів суперпозиції $\{S_m^n\}$. Завдяки функціям проекції будь-яку підстановку функцій у функції можна виразити через оператор S_m^n.

Приклад. Функція $f(x_1, x_2) = \varphi(f_1(x_1), f_2(x_1, x_2))$ може бути записана як

$$f(x_1, x_2) = S_2^2\left(\varphi(x_1, x_2), S_2^1(I_1(x_1, x_2), f_1(x_1), f_3(x_1)\right), f_2(x_1, x_2)),$$

де f_3 — будь-яка функція аргументу x_1.

Якщо задано функції I_m^n й оператори S_m^n, то можна вважати заданими також перейменування, перестановки та ототожнення змінних:

$$f(x_2, x_1, x_3, ..., x_n) = \varphi\left(I_2^2(x_1, x_2), I_1^2(x_1, x_2), x_3, ..., x_n\right);$$
$$f(x_1, x_1, x_3, ..., x_n) = \varphi\left(I_1^2(x_1, x_2), I_1^2(x_1, x_2), x_3, ..., x_n\right).$$

Означення 5.5. *Кажуть, що оператор примітивної рекурсії R^n визначає $(n + 1)$-місну функцію $f(x_1, ..., x_n, y)$ через n-місну функцію $\varphi(x_1, ..., x_n)$ й $(n + 2)$-місну функцію $\psi(x_1, ..., x_n, y, z)$ таким чином (за схемою примітивної рекурсії):*

$$\left.\begin{array}{l} f(x_1, ..., x_n, 0)=\varphi(x_1, ..., x_n), \\ f(x_1, ..., x_n, y+1)=\psi(x_1, ..., x_n, y, f(x_1, ..., y_n, y)). \end{array}\right\} \qquad (5.2)$$

Той факт, що функцію визначено схемою (5.2), виражається рівністю $f(x_1, ..., x_n, y) = R^n(\varphi, \psi)$.

У випадку, коли $n = 0$, тобто функція f, що визначається, є одномісною, схема (5.2) набуває простішого вигляду:

$$f(0) = a; \quad f(y+1) = \psi(y, f(y)), \qquad (5.3)$$

де a — константа.

Схеми примітивної рекурсії визначають f рекурсивно не тільки через інші функції φ і ψ, а й через значення f у попередніх точках: значення f у точці $y + 1$ залежить від значення f у точці y.

Означення 5.6. *Функція називається примітивно-рекурсивною, якщо вона може бути утворена з найпростіших функцій за допомогою скінченного числа застосувань операторів суперпозиції та примітивної рекурсії.*

Цьому означенню можна надати більш формального індуктивного вигляду:

1. Функції $O(x)$, $S(x)$ й I_m^n (для всіх натуральних n, m, де $m = 1, 2, ..., n$) є примітивно-рекурсивними.

2. Якщо $f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n), \varphi(x_1, ..., x_m)$ — примітивно-рекурсивні функції, то $S_m^n(\varphi, f_1, ..., f_m)$ — примітивно-рекурсивні функції для будь-яких натуральних n, m.

3. Якщо $f(x_1, ..., x_n)$ і $\varphi(x_1, ..., x_n, y, z)$ — примітивно-рекурсивні функції, то $R^n(f, \varphi)$ — примітивно-рекурсивна функція.

Інших примітивно-рекурсивних функцій немає.

5.2.2. Частково-рекурсивні функції

Означення 5.7. *Розглянемо обчислювану n-місну частково числову функцію $f(x_1, ..., x_n)$. Зафіксуємо деякі значення $x_1, ..., x_{n-1}$ її перших $(n - 1)$ аргументів та розглянемо рівняння $f(x_1, ..., x_{n-1}, y) = x_n$. Будемо обчислювати послідовно значення $f(x_1, ..., x_{n-1}, y)$ для $y = 0, 1, 2, ...$. Найменше a, для якого $f(x_1, ..., x_{n-1}, a) = x_n$, позначимо так:*

$$\mu_y(f(x_1, ..., x_{n-1}, a)) = x_n. \qquad (5.4)$$

Зазначений процес є нескінченним у таких випадках:

1. *Значення* $f\left(x_1, ..., x_{n-1}, 0\right)$ *не визначено.*

2. *Значення* $f\left(x_1, ..., x_{n-1}, y\right)$ *для* $y = 0, 1, 2, \ldots, a-1$ *визначено, але вони відмінні від* x_n, *а значення* $f\left(x_1, ..., x_{n-1}, a\right)$ *не визначено.*

3. *Значення* $f\left(x_1, ..., x_{n-1}, y\right)$ *визначено для всіх* $y = 0, 1, 2, \ldots$, *причому вони відмінні від* x_n.

В усіх цих випадках значення виразу (5.4) вважається невизначеним. У решті випадків зазначений процес зупиняється і дає найменший розв'язок $y = a$ рівняння $f\left(x_1, ..., x_{n-1}, y\right) = x_n$, який і є значенням виразу (5.4). Значення виразу (5.4) для заданої функції f залежить від значень параметрів $x_1, ..., x_{n-1}, x_n$; тому вираз (5.4) задає часткову функцію змінних $x_1, ..., x_n$, що визначає оператор мінімізації (μ — оператор).

Означення 5.8. *Функція* $f\left(x_1, ..., x_n\right)$ *називається частково-рекурсивною, якщо вона може бути утворена з найпростіших функцій за допомогою скінченного числа застосувань операторів суперпозиції, примітивної рекурсії та мінімізації. Всюди визначені частково-рекурсивні функції називаються загальнорекурсивними.*

Приклади: 1. Двомісна функція додавання $f_+(x, y) = x + y$ задовольняє схему примітивної рекурсії:

$$x + 0 = x = I_1^1(x);$$

$$x + (y + 1) = (x + y) + 1 = S(x + y).$$

Таким чином, $x + y = R^1\left(I_1^1(x), g(x, y, z)\right)$, де $g(x, y, z) = S(z) = z + 1$.

2. Двомісна функція множення $f_x(x, y) = xy$ задовольняє схему примітивної рекурсії:

$$x \cdot 0 = 0 = O(x);$$

$$f_x(x, y+1) = x(y+1) = xy + x = f_x(x, y) + x = f_+\left(x, f_x(x, y)\right).$$

3. Двомісна функція піднесення до степеня $f_s(x, y) = x^y$ задовольняє схему примітивної рекурсії:

$$f_s(x, 0) = x^0 = 1;$$

$$f_s(x, y+1) = x^{y+1} = x^y x = f_x\left(x, f_s(x, y)\right).$$

4. Розглянемо функції:

$$x \dot{-} y = \begin{cases} x - y, & \text{якщо} \quad x \geq y; \\ 0 & \text{у протилежному випадку;} \end{cases}$$

$$x \dot{-} 1 = \begin{cases} x - 1, & \text{якщо} \quad x \geq 1; \\ 0 & \text{у протилежному випадку.} \end{cases}$$

Маємо:

1) $f(x) = x \dot{-} 1$ визначається за схемою:

$$f(0) = 0 \dot{-} 1 = 0; f(y + 1) = y;$$

2) $f(x, y) = x \dot{-} y$ визначається за схемою:

$$f(x, 0) = x; \quad f(x, y + 1) = x \dot{-} (y + 1) = (x \dot{-} y) \dot{-} 1 = f(x, y) \dot{-} 1.$$

Отже, обидві функції задовольняють схему примітивної рекурсії.

5. Двомісна функція $f(x, y) = |x - y| = (x \dot{-} y) + (y \dot{-} x)$ є примітивно-рекурсивною внаслідок примітивної рекурсивності функцій $\dot{-}$ й f_+.

6. Розглянемо функції $sg(x) = \begin{cases} 0, & x = 0; \\ 1, & x \neq 0; \end{cases}$ і $\overline{sg}(x) = \begin{cases} 0, & x \neq 0; \\ 1, & x = 0; \end{cases}$

Їх примітивна рекурсивність випливає з того, що

$$sg(0) = 0;\ \overline{sg}(0) = 1$$

та

$$sg(x + 1) = 1;\ \overline{sg}(x+1) = 0.$$

За допомогою функції $sg(x)$ і її заперечення $\overline{sg}(x) = 1 \dot{-} sgx$ можна побудувати примітивно-рекурсивний опис функцій, пов'язаних з діленням.

Із наведених прикладів зрозуміло, що багато числових функцій є примітивно-рекурсивними. Виникає запитання: який зв'язок існує між примітивно-рекурсивними та частково-рекурсивними функціями?

Приклади: 1. Розглянемо функцію:

$$f(x, y) = \begin{cases} x - y, & \text{якщо} \quad x \geq y, \\ \text{є невизначеною у протилежному випадку.} \end{cases}$$

Завдяки примітивній рекурсивності функцій

1) $x \dot{-} y = \begin{cases} x - y, & \text{якщо} \quad x \geq y; \\ 0 \text{ у протилежному випадку.} \end{cases}$

2) $x \dot{-} 1 = \begin{cases} x - 1, & \text{якщо} \quad x \geq 1; \\ 0 & \text{у протилежному випадку;} \end{cases}$

3) $| x - y | = (x \dot{-} y) + (y \dot{-} x)$

і застосуванню оператора мінімізації $f(x, y) = \mu_z\left(|x - (z + y)| = 0\right)$ функція є частково-рекурсивною.

2. Двомісні функції $\min(x, y) = x \dot{-} (x \dot{-} y)$ та $\max(x, y) = y + (x \dot{-} y)$ є примітивно-рекурсивними.

Із прикладів 4,6 на с. 121, 122 і 2 на с. 122 випливає примітивна рекурсивність числових функцій, які на множині $\{0, 1\}$ поводяться як логічні. Справді, якщо $x, y \in \{0, 1\}$, то

$$\overline{x} = 1 \dot{-} x;$$
$$x \vee y = \max(x, y); \tag{5.5}$$
$$x \,\&\, y = \min(x, y).$$

Із функціональної повноти (див. п. 3.6) цієї множини функцій і того, що суперпозиція є примітивно-рекурсивним оператором, установлюємо примітивну рекурсивність усіх логічних функцій.

Означення 5.9. *Розглянемо множину A, що є підмножиною N. Характеристичною функцією χ_A множини A називається функція*

$$\chi_A(x) = \begin{cases} 0, \textit{ якщо } x \in A; \\ 1 \textit{ у протилежному випадку.} \end{cases}$$

Множина A називається примітивно-рекурсивною, якщо її характеристична функція — теж примітивно-рекурсивна.

Означення 5.10. *Частково-характеристичною функцією* $\chi_A^{ч}$ *множини* A *називається функція*

$$\chi_{\text{A}}^{ч}(x)=\begin{cases}0, \ \textit{якщо} \ \ x\in A;\\ \textit{є не} \ \ \textit{визначеною у протилежному випадку.}\end{cases}$$

Множина A називається частково-рекурсивною, якщо її частково-характеристична функція — теж частково-рекурсивна.

Очевидно, характеристична й частково-характеристична функції збігаються лише для множини всіх натуральних чисел N.

Покажемо, що кожна примітивно-рекурсивна множина є частково-рекурсивною. Справді, нехай χ_A — характеристична функція множини A. Тоді функція $\chi^{x}(x)$, яка визначається рівністю $\chi^{x}(x)=0-\chi_A(x)$, буде частково-характеристичною функцією множини A. Із того, що оператор віднімання є частково-рекурсивним, випливає, що функція $\chi^{x}(x)$ — також частково-рекурсивна.

Теорема 5.1. *Нехай* $f(x)$ *— примітивно-рекурсивна функція, а* A *— деяка примітивно-рекурсивна множина. Тоді часткова функція, визначена за схемою*

$$f^{x}(x)=\begin{cases}f(x), \textit{ якщо } x\notin A;\\ \textit{є невизначеною, якщо } x\notin A,\end{cases} \tag{5.6}$$

є частково-рекурсивною.

Уже доведено, що частково-характеристична функція $\chi^{x}(x)$ множини A теж частково-рекурсивна. Із схеми (5.6) випливає, що для всіх значень x

$$f^{x}(x)=f(x)+f^{x}(x),$$

тобто функція $f^{x}(x)$ також є частково-рекурсивною.

5.2.3. Теза Черча

Теорема 5.1 надає можливість побудови численних прикладів частково-рекурсивних функцій. Взагалі поняття частково-рекурсивної функції — одне з основних понять теорії алгоритмів. Його значення є таким. З одного боку, кожна стандартно задана частково-рекурсивна функція є обчислюваною за певною процедурою, яка відповідає інтуїтивному уявленню алгоритму, а з іншого — які б досі ні будувалися класи точно визначених алгоритмів, завжди з'ясовувалось, що числові функції, які обчислювались за алгоритмами цих класів, були частково-рекурсивними.

Тому загальноприйнятою є така наукова гіпотеза.

Теза Черча. *Клас алгоритмічно обчислюваних часткових числових функцій збігається з класом усіх частково-рекурсивних функцій.*

У формулювання цієї тези входить інтуїтивне поняття обчислюваності, тому його не можна ні довести, ні спростувати в загальноматематичному значенні. Це факт, на користь якого свідчить багаторічна математична практика. З означення частково-рекурсивних функцій випливає, що система функцій $\left\{O(x), S(x), I_1^1(x)\right\}$ є повною системою відносно операцій суперпозиції, примітивної рекурсії та мінімізації у множині алгоритмічно обчислюваних часткових числових функцій. А чи є вона мінімальною?

Теорема 5.2. Система функцій $\{O(x), S(x)\}$ *є повною системою відносно операцій суперпозиції, примітивної рекурсії та мінімізації у множині алгоритмічно обчислюваних часткових числових функцій.*

Для доведення цієї теореми досить навести схему, що визначає $I_1^1(x)$ за $O(x)$ й $S(x)$. Ця схема є такою:

$$I_1^1(0) = O(x) = 0;$$
$$I_1^1(x+1) = S\left(I_1^1(x)\right).$$

Часткова рекурсивність — це уточнення поняття обчислюваної функції. За допомогою цього означення можна доводити чи спростовувати обчислюваність.

5.3. Машини Тьюрінга

5.3.1. Основні приклади й означення

Розглянемо ще одну стандартну форму подання будь-якого алгоритму. У 1935 р. Е. Л. Пост розробив найпростішу з можливих систем оброблення інформації та показав, що запропонована ним система має дуже важливу властивість, а саме: властивість алгоритмічної повноти. Він запропонував усяку інформацію, яка підлягає обробленню за суттю, спочатку перетворювати за формою, а саме: переводити зі звичайного алфавіту в алфавіт двійковий, а потім кожне двійкове слово обробляти літера за літерою. Після того, як певну інформацію переписано у двійковому алфавіті, вона літера за літерою заноситься на інформаційну стрічку машини. Цю стрічку поділено на однакові секції (комірки), число яких — нескінченне. Для реалізації двійкового кодування, як правило, використовується один символ (наприклад, $\vee$ або 1), тому що як другий символ застосовується порожня комірка.

На відміну від машини Поста, МТ може знаходитися у процесі роботи в різних станах, змінюючи їх стрибкоподібно. Стани МТ, які називаються внутрішніми, становлять її внутрішній алфавіт $Q = \{q_1, q_2, ..., q_n\}$. МТ можуть різнитися між собою за внутрішніми алфавітами, а це значить, що число їхніх станів є різним. Серед внутрішніх станів МТ окреме місце посідають такі два: початковий (позначається q_0 або q_1) та заключний (позначається ! або q_z, де z не є числовим індексом, а є мнемонічною ознакою кінця).

Інша важлива відмінність МТ від машини Поста — це те, що для кожної конкретної МТ проектують її власний зовнішній алфавіт $A = \{a_1, a_2, ..., a_m\}$. Символи цього алфавіту можуть розпізнаватися і виписуватися кареткою. У МТ для кожної машини можна визначати алфавіт заново. Слова, що переробляються в цьому алфавіті, зображуються в комірках нескінченної стрічки, причому в кожній з них може зберігатися тільки один символ.

Робота машини відбувається в дискретному часі. На кожному кроці кареткою (за різними джерелами головкою або пристроєм звернення до стрічки) розглядаються єдина комірка, а символ a_i, який зчитується, замінюється іншим a_j ($i = j$ означає, що символ не змінюється) залежно від стану машини q_k у заданий тактовий момент із множини її можливих станів. Крім того, виробляється наступний стан машини, і команда, яка керує переміщенням по стрічці, підготовлює чергову комірку для розгляду на наступному кроці. Таких команд у МТ три: R — розглядати сусідню праворуч комірку; L — розглядати сусідню ліворуч комірку та E — продовжувати розглядати колишню комірку. Очевидно, що є неістотним, чи рухається каретка вздовж стрічки, чи каретка є нерухомою, а рухається кожного разу на один крок саме стрічка.

Нехай МТ може знаходитися тільки в скінченному числі різних внутрішніх станів і в будь-який момент часу чергова операція є функцією поточного внутрішнього стану машини та скінченного виразу, зафіксованого на стрічці. Відповідно до наведеного вище означення, можна формально визначити МТ в такий спосіб:

$$Q(t+1) = F_1(Q(t), A(t));\ R(t+1) = F_2(Q(t), A(t));$$
$$D(t+1) = F_3(Q(t), A(t)),$$

де F_1, F_2, F_3 — деякі функції, що пов'язують попередній стан і попередній вхідний (який розглядається) символ. Функція F_1 визначає зміну стану, а функція F_2 вихідне значення записує в комірку, що розглядається; операція запису може містити вилучення символу, який знаходився раніше в комірці . Функція F_3 просто повідомляє, куди саме рухається стрічка.

Описувана машина є скінченною, але її можна вважати актуально нескінченною у тому розумінні, що після досягнення будь-якого кінця стрічки завжди може бути додана нова комірка. При цьому, звичайно, у будь-який конкретний момент часу довжина стрічки залишається скінченною.

Формальний математичний опис МТ дано у працях А. М. Тьюрінга, Е. Л. Поста, С. К. Клині [7], Дейвіса, Арбиба [11] та М. Мінського [4]. Використовувані в цих працях різні системи позначень еквівалентні одна одній. Дейвіс пише про четвірки

$$q_i a_j a_k q_t,\ q_i a_j R q_t,\ q_i a_j L q_t,\ q_i a_j E q_t,$$

які означають, що при заданих внутрішньому стані q_i і поточному символі a_j може виникнути одна з таких ситуацій: запис a_k замість a_j та перехід у стан q_t; рух праворуч і перехід у стан q_t; рух ліворуч та перехід у стан q_t; перехід у стан q_t без руху. У позначеннях М. Мінського ці четвірки замінюються такими п'ятірками:

$$\left(q_i, a_j, q_{ij}, a_{ij}, d_{ij}\right),$$

де q_i — заданий внутрішній стан; a_j — поточний (що розглядається) символ; q_{ij} — новий стан машини, a_{ij} — символ, який записується, а d_{ij} показує напрямок руху стрічки.

У [2] використовується система правил (команд), які мають вигляд

$$q_i a_j \to q_i' a_j' d_r. \tag{5.7}$$

Це означає, шо машина, яка має внутрішній стан q_i й розглядає комірку із символом a_j, записує в цю комірку символ a_j', переходить у стан q_i' і здійснює крок d_r. Також у [2] застосовується блок-схема, яка називається діаграмою переходів. У ній станам відповідають вершини, а правилу вигляду (5.7) — ребро, що веде з q_i у q_i', на якому написано: $a_j \to a_j' d_r$. Умова однозначності потребує, щоб для будь-яких j та $i \neq z$, де q_z — заключний стан МТ, у системі команд була команда, аналогічна (5.7), з лівою частиною $q_i a_j$; стан q_z у лівій частині команд не зустрічається. На діаграмі переходів це виражається умовою, за якою з кожної вершини, крім q_z, вийдуть точно m ребер, причому на різних ребрах ліві частини будуть різними; у вершини q_z немає вихідних ребер.

У [3] використовується таблиця, стовпцям якої відповідають стани, рядкам — символи зовнішнього алфавіту, а на перетині j-го рядка та i-го стовпця записано трійку символів $q_i a_j d_r$ (така таблиця називається функціональною схемою), де q_i — новий стан машини; a_j — символ, який записується, а d_r показує напрямок руху стрічки. Функціональна схема, що відповідає деякому алгоритму, задається подібно до загальної таблиці переходів скінченного автомата.

Домовимося у подальшому застосовувати всі наведені схеми, крім діаграм переходів, причому у функціональних схемах будемо користуватися трійками вигляду $a_j d_r q_i$, маючи на увазі, що при певному символі та стані в комірку, яку розглядає каретка, спочатку записується символ a_j, а потім здійснюються крок d_r і перехід до стану q_i. Якщо символ у комірці або внутрішній стан не змінюється, то відповідні символи у трійці можна не зазначати.

Повний стан МТ, за яким однозначно можна визначити її подальшу поведінку, визначається внутрішнім станом машини, станом стрічки (тобто словом, записаним на ній) та положенням каретки на стрічці.

Означення 5.11. *Конфігурацією на стрічці (або машинним словом) називається сукупність, утворена:*

- *послідовністю $a_{i_1}, a_{i_2}, \ldots, a_{i_n}$ символів, записаних у комірках стрічки, де a_{i_1} — символ, записаний у першій комірці ліворуч; a_{i_2} — символ, записа-*

ний у другій комірці теж ліворуч, тощо (така послідовність називається словом в алфавіті А);

- *станом внутрішньої пам'яті q_i;*
- *номером k комірки.*

Конфігурацію машини будемо записувати так:

$$\alpha_1 q_i \alpha_2,$$

де q_i — поточний внутрішній стан; α_1 — слово ліворуч від каретки, а α_2 — слово, утворене символом, який розглядається кареткою, і символами праворуч від нього, причому ліворуч від α_1 та праворуч від α_2 немає непорожніх комірок. Наприклад, конфігурація з внутрішнім станом q_i, в якій на стрічці записано *abcde*, а каретка розглядає *c*, запишеться як abq_icde.

Стандартною початковою конфігурацією назвемо конфігурацію вигляду $q_0\alpha$, тобто конфігурацію, що містить початковий стан, в якому каретка розглядає крайній лівий символ слова, написаного на стрічці. Аналогічно стандартною заключною конфігурацією назвемо конфігурацію вигляду $q_z\alpha$. До кожної конфігурації T машини Z, яка не є заключною, застосовна лише одна команда вигляду (5.7), що переводить T у конфігурацію T'. Ці відношення між конфігураціями позначимо $T \to T'$. Якщо для T_1 і T_n існує послідовність конфігурацій $T_1, T_2, ..., T_n$ така, що $T_1 \to T_2 \to ... \to \to T_n$, позначимо це $T_1 \longrightarrow T_n$. Послідовність конфігурацій $T_1 \to T_2 \to T_3 \to ...$ однозначно визначається початковою конфігурацією T_1; вона повністю описує роботу машини Z, починаючи з T_1. Ця послідовність є скінченною, якщо в ній зустрічається заключна конфігурація, і нескінченною в іншому випадку.

Означення 5.12. *Якщо машина, почавши роботу з деяким словом, записаним на стрічці, прийде в заключний стан, то вона називається застосовною до цього слова. Результатом її роботи вважається слово, записане на стрічці в заключному стані. Якщо ж машина в жодний момент часу не прийде в заключний стан, то вона називається не застосовною до цього слова, і результат її роботи невизначений.*

Приклади: 1. Наведемо простий приклад роботи МТ. Нехай задано два слова a_0a_1 й $a_2a_3a_4a_5$ (рис. 40). Завдання полягає в тому, щоб, починаючи з лівого кінця стрічки, вилучити символи a_3 та a_5, замінити їх символами a_2 й a_4, після чого повернутися до лівого кінця стрічки. Порожню комірку позначимо λ.

Відповідні четвірки мають такий вигляд:

$$\begin{array}{ccc}
q_0a_0Rq_0 & q_2a_3a_2q_2 & q_3a_4Lq_3 \\
q_0a_1Rq_0 & q_2a_4Rq_2 & q_3a_2Lq_3 \\
q_0\lambda Rq_1 & q_2a_5a_4q_3 & q_3\lambda Lq_3 \\
q_1a_2Rq_2 & q_3\lambda Lq_3 & q_3a_1Lq_z
\end{array}$$

	λ	a_0	a_1	λ	a_2	a_3	a_4	a_5	λ	

Рис. 40

Послідовні конфігурації розв'язування сформульованої задачі мають такий вигляд:

$$q_0a_0a_1\lambda a_2a_3a_4a_5\lambda \to a_0q_0a_1\lambda a_2a_3a_4a_5\lambda \to$$

$$a_0a_1q_0\lambda a_2a_3a_4a_5\lambda \to a_0a_1\lambda q_1a_2a_3a_4a_5\lambda \to$$

$$a_0a_1\lambda a_2q_2a_3a_4a_5\lambda \to a_0a_1\lambda a_2q_2a_2a_4a_5\lambda \to a_0a_1\lambda a_2a_2q_2a_4a_5\lambda \to$$

$$a_0a_1\lambda a_2a_2a_4q_2a_5\lambda \to a_0a_1\lambda a_2a_2a_4q_2a_5\lambda \to a_0a_1\lambda a_2a_2q_3a_4a_5\lambda \to$$

$$a_0a_1\lambda a_2q_3a_2a_4a_4\lambda \to a_0a_1\lambda q_3a_2a_2a_4a_4\lambda \to$$

$$a_0a_1q_3\lambda a_2a_2a_4a_4\lambda \to a_0q_3a_1\lambda a_2a_2a_4a_4\lambda \to q_3a_0a_1\lambda a_2a_2a_4a_4\lambda.$$

Функціональну схему МТ для цього прикладу наведено в табл. 5.2.

Таблиця 5.2

	q_0	q_1	q_2	q_3
a_0	R			
a_1	R			Lq_z
a_2		Rq_2		L
a_3			a_2R	
a_4			R	L
a_5			a_4Rq_3	
λ	Rq_1			L

У наведеній вище обчислювальній процедурі послідовні позиції каретки, що зчитує, зазначаються переміщеннями символу стану в кожному поточному описі.

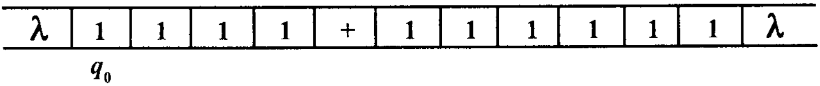
2. Складемо функціональну схему, яка відповідає алгоритму додавання двох чисел. Числа подаються сукупністю одиниць, загальна кількість яких дорівнює заданому числу. Числа відокремлюються знаком +. Порожню комірку позначимо λ (рис. 41).

Нехай, наприклад, потрібно додати числа 4 і 6. Початкове слово на стрічці має вигляд 1111 + 111111. Функціональну схему наведено в табл. 5.3.

Таблиця 5.3

	q_0	q_1	q_2
1	λRq_2	$1Lq_1$	$1Rq_2$
λ	Rq_2	λRq_0	$1Eq_1$
+	λRq_z	$+Lq_1$	$+Rq_2$

Систему команд МТ можна інтерпретувати як програму, тобто сукупність розпоряджень, що однозначно приводять до результату, та як опис роботи конкретного механізму. У [3] пропонується логічна схема такого



λ	1	1	1	1	+	1	1	1	1	1	1	λ
	q_0											

Рис. 41

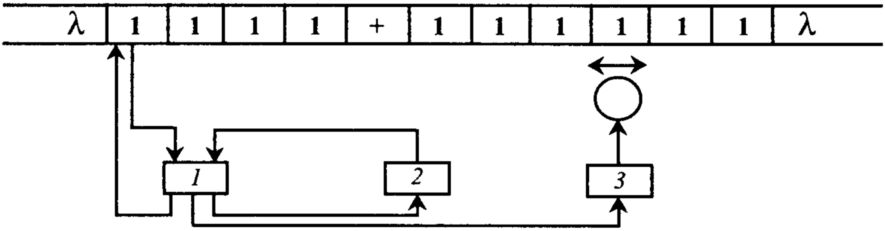


Рис. 42

конкретного механізму (рис. 42). Функціональна таблиця цілком визначає функціонування машини і реалізується в ній логічним блоком *1*. На два його входи подаються символи, які зчитуються й над якими виконуються операції (заміна іншими символами), та стани, що відіграють роль команд, які визначають ці операції. На одному з виходів логічного блока утворюється символ, який у цьому такті заміщає на стрічці символ, що розглядується, а на інших двох виходах — команди, які визначають функціонування машини на наступному такті (переміщення по стрічці і новий стан). Для запам'ятовування цих команд уводяться затримки *2* та *3*, що є внутрішньою пам'яттю машини.

Перед початком роботи на стрічку наносять початкове слово і задають початкові умови, тобто зазначать першу комірку, яка розглядається, та початковий стан. Після пуску машини процес перетворення інформації відбувається автоматично.

Відповідно до наведеної вище схеми додавання в останньому прикладі початкові умови визначаються коміркою із крайньою лівою одиницею і станом q_0. На першому такті одиниця стирається, видається команда руху праворуч та переходу у стан q_2 $(\lambda R q_2)$. Наступні такти зводяться до рухів праворуч через усі одиниці $(1 R q_2)$ і знак « + » $(+R q_2)$ доти, доки не буде досягнута перша порожня комірка. Тоді в неї вписується одиниця, а машина переходить у стан q_1 $(1 E q_1)$. Після цього відбувається рух у зворотному напрямку через усі символи 1 та « + » до першої порожньої комірки ліворуч. Далі проходить рух праворуч і машина переходить у стан q_0 $(\lambda R q_0)$.

Завдяки цьому циклу одиниця лівого доданка переноситься у правий доданок, що відповідає слову 111 + 1111111. Очевидно, через чотири таких цикли початкове слово перетвориться до вигляду 1111111111. До початку п'ятого циклу у стані q_0 розглядається символ « + », який стирається, відбуваються рух праворуч і зупин $(\lambda E q_z)$. У результаті матимемо слово 1111111111, що відповідає числу 10.

5.3.2 Обчислюваність за Тьюрінгом

Постає питання про функції, для яких існує МТ, що їх обчислює. Щодо МТ можна сказати, що ця функція визначається поведінкою машини. Аргумент з'являється на стрічці до початку обчислення, а значення функції для цього аргументу — те, яке залишається на стрічці, коли обчислення закінчено. Скористуємось спеціальним кодуванням натуральних чисел в алфавіті, що складається з одного символу — одиниці: кожне натуральне число n подається $(n + 1)$ символом 1, тобто числа 0, 1, 2,.... кодуємо словами 1, 11, 111, ...

Означення 5.13. *Часткова числова n-місна функція* $f(x_1, ..., x_n)$ *називається обчислюваною за Тьюрінгом, якщо існує машина М, яка обчислює її таким чином:*

1. *Якщо набір значень аргументів* $(x_1, ..., x_n)$ *належить області визначення функції f, то машина X, починаючи роботу в конфігурації*

$$q_0 1^{x_1+1} \lambda 1^{x_2+1} ... \lambda 1^{x_n+1}, \tag{5.8}$$

де q_0 *— початковий стан, в якому каретка МТ розглядає крайній лівий символ, зупиняється, закінчуючи роботу в конфігурації*

$$q_z 1^{y+1},$$

де $y = f(x_1, ..., x_n)$.

2. *Якщо набір значень аргументів* $(x_1, ..., x_n)$ *не належить області визначення функції f, то машина М, починаючи роботу в конфігурації* (5.8), *працює нескінченно довго, тобто не переходить у заключний стан.*

Приклади: 1. Розглянемо функцію наступності (або додавання одиниці) $S(x) = x + 1$. Ця функція відіграє важливу роль у теорії алгоритмів, для неї за самою її природою легко довести її обчислюваність. Для цього побудуємо машину М із зовнішнім алфавітом $\{\lambda, 1\}$, двома внутрішніми станами $\{q_0, q_z\}$ і програмою:

$$q_0 1 \to q_0 1R$$
$$q_0 \lambda \to q_z 1E.$$

2. Розглянемо ще один приклад обчислення на МТ — обчислення тотожної функції. Нехай $I(x) = x$; покажемо, що функція $I(x)$ є обчислюваною. Машина M із зовнішнім $\{1, \lambda\}$ та внутрішнім $\{q_0, q_z\}$ алфавітами обчислює функцію $I(x)$ за допомогою програми:

$$q_0 1 \to q_0 1R$$
$$q_0 \lambda \to q_z \lambda E.$$

3. Розглянемо останню із найпростіших функцій $O(x) = 0$. Покажемо, що вона є обчислюваною за Тьюрінгом. Машина M із зовнішнім $\{1, \lambda\}$ та внутрішнім $\{q_0, q_z\}$ алфавітами обчислює функцію $O(x)$ за допомогою програми:

$$q_0 1 \to q_0 \lambda R$$
$$q_0 \lambda \to q_z \lambda E.$$

4. Машина із зовнішнім $A = \{1, \lambda\}$ та внутрішнім $\{q_1, q_z\}$ алфавітами і системою команд

$$q_1 1 \to q_1 1R$$
$$q_1 \lambda \to q_1 1R$$

із будь-якої початкової конфігурації працюватиме нескінченно довго, заповнюючи одиницями всю стрічку праворуч від початкової точки.

Якщо машина M переводить конфігурацію $\alpha_1 q_1 \alpha_2$ у конфігурацію $\beta_1 q_z \beta_2$, то говоритимемо, що машина M переробляє слово $\alpha_1\alpha_2$ у слово $\beta_1\beta_2$, та позначати це як $M(\alpha_1\alpha_2)=\beta_1\beta_2$. Запис $M(\alpha)$ іноді будемо вживати також в іншому значенні: як позначення машини M з початковими значеннями α.

5. Подвоювання (перезапис) слова, тобто перероблення слова α в алфавіті $A=\{1\}$ у $\alpha\square\alpha$. Для чисел цю задачу розв'язує машина M, систему команд для якої наведено в табл. 5.4.

Таблиця 5.4

	q_0	q_1	q_2	q_3
1	$0Rq_1$	$1Rq_1$	$1Rq_2$	$1Lq_3$
λ	λRq_z	$\square Rq_2$	$1Lq_3$	
$\square$	$\square Lq_0$	$\square Rq_2$		$\square Lq_3$
0	$1Lq_0$			$0Rq_0$

Машина M при кожному проході початкового числа 1^α замінює ліву з його одиниць нулем і заносить (у стані q_3) одну одиницю праворуч від 1^α у найближчу порожню комірку. Коли всі одиниці початкового числа вичерпано, каретка повертається в початковий стан, замінюючи всі нулі одиницями.

6. Сконструюємо МТ для визначення суми чисел, заданих не в унарному коді, а в різних системах числення. Наприклад, знайдемо суму числа, заданого у трійковій системі, та числа, заданого у вісімковій системі. Очевидно, зовнішній алфавіт МТ A = {0, 1, 2, 3, 4, 5, 6, 7, +, λ }, а внутрішній, наприклад, $Q=\{q_1, q_2, q_3, q_4, q_5\}$.

Нехай доданки відокремлюються знаком « + », порожні секції, як і раніше, позначимо λ, і нехай на стрічці ліворуч від знака « + » розташовано доданок у вісімковому коді, а праворуч — доданок у трійковому коді. Ідея розв'язання задачі така: від числа, розташованого праворуч, віднімаємо одиницю, діючи за правилами трійкової арифметики, потім просуваємося ліворуч та до вісімкового числа додаємо одиницю, діючи за правилами вісімкової арифметики. Після цього повертаємося до правого числа і повторюємо всі операції доти, доки доданок, розташований праворуч від знака « + », не буде вичерпано.

Нарешті, останній раз зсуваємося праворуч, стираємо знак « + » та зупиняємося. Функціональною схемою цієї МТ є табл. 5.5.

Таблиця 5.5

	q_1	q_2	q_3	q_4	q_s
0	$2Lq_1$	L	$1Rq_4$	R	
1	$0Lq_2$	L	$2Rq_4$	R	
2	$1Lq_2$	L	$3Rq_4$	R	λRq_5
3			$4Rq_4$	R	
4			$5Rq_4$	R	
5			$6Rq_4$	R	
6			$7Rq_4$	R	
7			$0Lq_3$	R	
+	λRq_5	Lq_3		R	
λ			$1Rq_4$	Lq_1	q_z

Побудовану МТ можна використати як трійково-вісімковий дешифратор.

5.3.3. Деякі операції над машинами Тьюрінга

Робота МТ повністю визначається початковими даними і системою команд. Однак для розуміння того, як конкретна машина розв'язує задану задачу, як правило, виникає потреба у змістовних поясненнях на зразок тих, які наводилися для машини у прикладі 2 на с. 128. Ці пояснення часто можна зробити більш формальними та точними, якщо використати логічні схеми і деякі операції над МТ.

Нагадаємо, що композицією двох функцій $f_1(x)$ та $f_2(y)$ називається функція $g(x)=f_2(f_1(x))$, що утворюється при застосуванні f_2 до результату обчислення f_1. Для того щоб функція $g(x)$ була визначена при заданому x, необхідно та достатньо, щоб функція f_1 була визначена на x, а функція f_2 — на $f_1(x)$.

Означення 5.14. *Нехай M_1 й M_2 — дві МТ із спільним зовнішнім алфавітом $A=\{a_1, a_2, ..., a_m\}$ і внутрішніми алфавітами $Q_1=\{q_0, q_1 ..., q_m\}$ та $Q_2=\{g_0, g_1, ..., g_s\}$. Композицією M машин M_1 та M_2 називається машина із зовнішнім алфавітом $A=\{a_1, a_2, ..., a_m\}$, внутрішніми станами $\{q_0, q_1, ..., q_{m-1}, q_{m+1}, ..., q_{m+s+1}\}$ і програмою, що створюється так. У всіх командах машини M_1 заключний символ q_m змінюється на символ q_{m+1}, а інші символи залишаються без змін. У всіх командах машини M_2 символи g_i $(i=0,1,...,s)$ змінюються відповідно символами q_{m+i+1}. Сукупність усіх команд машин M_1 й M_2, змінена в такий спосіб, є програмою композиції M машин M_1 і M_2. «Робота» машини M рівносильна послідовній «роботі» машин M_1 та M_2.*

Теорема 5.3. *Якщо $f_1(x)$ і $f_2(y)$ — обчислювані за Тьюрінгом функції, то їх композиція $f_2(f_1(x))$ також є обчислюваною за Тьюрінгом.*

Нехай M_1 — машина, що обчислює функцію f_1, а M_2 — машина, яка обчислює функцію f_2, і множини їхніх станів відповідно $Q_1=\{q_0, q_1, ..., q_m\}$ та $Q_2=\{g_0, g_1, ..., g_s\}$. Нехай композицію $f_2(f_1(x))$ визначено. Тоді $M_1(1^x)=1^{f_1(x)}$ і $q_0 1^x \longrightarrow q_m 1^{f_1(x)}$. Машина M пройде ту саму послідовність конфігурацій з тією різницею, що замість $q_m 1^{f_1(x)}$ вона перейде в $q_{m+1} 1^{f_1(x)} \left(g_0 1^{f_1(x)}\right)$. Ця конфігурація є стандартною початковою для машини M_2, тому $s_0 1^{f_1(x)} \longrightarrow g_1 1^{f_2(f_1(x))}$. Однак оскільки всі команди для M_2 містяться в M, маємо $q_0 1^x \longrightarrow q_{m+1} 1^{f_1(x)} \longrightarrow q_{m+s+1} 1^{f_2(f_1(x))}$, а отже, $M(1^x)=1^{f_2(f_1(x))}$. Якщо ж композицію $f_2(f_1(x))$ не визначено, то машина M_1 або M_2 не зупиниться; отже, не зупиниться також машина M. Тому вона обчислює $f_2(f_1(x))$.

Побудовану таким чином машину M (композицію машин M_1 й M_2) будемо позначати $M_2(M_1)$ і зображати так, як показано на рис. 43.

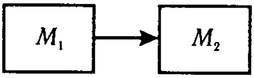


Рис. 43

Означення композиції та теорема 5.3 залишаються без змін, якщо машини M_1 та M_2 обчислюють функції кількох змінних. Важливо лише, щоб дані для M_2 були у зумовленому вигляді підготовлені машиною M_1.

Універсальна машина Тьюрінга. Суттєвий факт, який було доведено за допомогою МТ, — це те, що існує така МТ, яка за певними початковими значеннями обчислює будь-яку задану функцію, що є обчислюваною за Тьюрінгом. Як зазначено вище, систему команд для МТ можна інтерпретувати також як опис роботи конкретного механізму і як програму, тобто сукупність розпоряджень, які однозначно приводять до результату. Розбираючи приклади, читач мимоволі вибирає другу інтерпретацію, виступаючи в ролі механізму, який здатний відтворити роботу будь-якої МТ.

Упевненість у тому, що користувачі все це будуть робити однаково (якщо не наroблять помилок, що, до речі, передбачається при роботі МТ), — це, по суті, упевненість в існуванні алгоритму відтворення роботи МТ за заданою програмою, тобто системою команд. Справді, словесний опис такого алгоритму дати неважко.

Як уже зазначалося у п. 5.1, словесний опис алгоритму може бути неточним і потребувати формалізації. Оскільки як такою формалізацією поняття алгоритму розглядається МТ, природно поставити задачу побудови такої МТ, яка реалізує описаний алгоритм відтворення. Для МТ, що обчислює функцію однієї змінної, формулювання цієї задачі таке: побудувати МТ U, яка обчислює функцію двох змінних, причому $U(\Sigma_T, \alpha) = T(\alpha)$ для будь-якої машини T із системою команд Σ_T, якщо $T(\alpha)$ визначено (тобто машина $U(\Sigma_T, \alpha)$ зупиняється, якщо зупиняється машина T при початкових даних α, і машина $U(\Sigma_T, \alpha)$ не зупиняється, якщо не зупиняється машина $T(\alpha)$).

Будь-яку машину U, що має цю властивість, будемо називати універсальною МТ. Неважко узагальнити це формулювання на будь-яке число змінних. Докладніше про побудову універсальної МТ йтиметься у п. 6.5.

Проблема зупину. Серед загальних вимог до алгоритмів (див. п. 5.1) згадувалася вимога результативності. Найрадикальнішим формулюванням тут була б вимога, щоб за будь-яким алгоритмом A і даними α можна було б визначити, чи приведе робота A за початкових даних α до результату, чи ні. Іншими словами, треба побудувати алгоритм B такий, що $B(A, \alpha) = I$, якщо $A(\alpha)$ дає результат, та $B(A, \alpha) = X$, якщо $A(\alpha)$ не дає результату. З урахуванням тези Тьюрінга цю задачу можна сформулювати як задачу про побудову МТ: побудувати машину T_0 таку, що для будь-якої МТ T і будь-яких початкових даних для машини $T_0(\Sigma_T, \alpha) = I$, якщо машина $T(\alpha)$ зупиняється, та $T_0(\Sigma_T, \alpha) = X$, якщо вона не зупиняється.

Ця задача називається проблемою зупину. Її формулювання трохи нагадує задачу про побудову універсальної машини і, зокрема, також передбачає вибір відповідного кодування Σ_T та α в алфавіті машини T_0. Проте у цьому випадку ніяке кодування не приводить до успіху. У п. 6.5 доводиться, що справджується така теорема.

Теорема 5.4. *Не існує МТ T_0, яка вирішує проблему зупину для довільної машини Тьюрінга T.*

5.3.4. Теза Тьюрінга. Зв'язок рекурсивних функцій з машинами Тьюрінга

А. Тьюрінг запропонував гіпотезу, що формулюється так: «Класи обчислюваних й обчислюваних за Тьюрінгом функцій збігаються». Довести тезу Тьюрінга, як і тезу Черча, неможливо, оскільки саме поняття алгоритму (або ефективної процедури) є неточним. Це не теорема і не постулат математичної теорії, а твердження, яке пов'язує теорію з тими об'єктами, для опису яких її створено. За своїм значенням теза Тьюрінга нагадує гіпотези фізики про адекватність математичних моделей фізичним явищам та процесам. Підтвердженням тези Тьюрінга є, по-перше, математична практика, а по-друге, той факт, що опис алгоритму у термінах будь-якої іншої відомої алгоритмічної моделі може бути зведений до його опису у вигляді МТ.

Теза Тьюрінга дає змогу, з одного боку, замінити неточні твердження про існування ефективних процедур (алгоритмів) точними твердженнями про існування МТ, а з іншого — твердження про неіснування МТ тлумачити як твердження про неіснування алгоритмів узагалі. Проте не слід розуміти тезу Тьюрінга у тому сенсі, що вся теорія алгоритмів може бути зведена до теорії МТ. Наприклад, у теорії складності алгоритмів (що швидко розвивається зараз і розглядає порівняльну складність алгоритмів за пам'яттю, кількістю дій тощо) результати, правильні в межах однієї алгоритмічної моделі (скажімо, про кількість дій, потрібних для обчислення заданої функції), можуть виявитися неправильними в іншій моделі.

Розглянута у п. 5.2.3 теза Черча (всяка функція, обчислювана за деяким алгоритмом, є частково-рекурсивною) історично була сформульована раніше ніж теза Тьюрінга. Із зіставлення цих двох тез випливає таке твердження: функція обчислюється МТ тоді й тільки тоді, коли вона є частково-рекурсивною. Це твердження про еквівалентність двох алгоритмічних моделей (на відміну від тез) є точним математичним твердженням і потребує доведення.

Теорема 5.5. *Усяка частково-рекурсивна функція є обчислюваною на МТ.*

План доведення зрозумілий: спочатку доводиться, що найпростіші функції, які складають повну систему, є обчислюваними (це доведено у прикладах 5.15 — 5.17), а потім те, що оператори суперпозиції, примітивної рекурсії та μ-оператор зберігають обчислюваність (доведено у [2]).

Можна довести й супротивне [2].

Теорема 5.6. *Усяка функція, обчислювана на МТ, є частково-рекурсивною.*

5.3.5. Алгоритмічна нерозв'язуваність і машини Тьюрінга

Алгоритмічна проблема — це проблема, що зводиться до відшукання загального методу (алгоритму) для розв'язання всіх задач деякого класу. З уточненням поняття алгоритму за допомогою МТ з'явилась ще одна можливість установлювати наявність алгоритмічно нерозв'язуваних проблем, тобто проблем, для яких шуканий алгоритм відсутній.

Кожна МТ за означенням може бути задана зовнішнім алфавітом A, алфавітом внутрішніх станів Q і програмою P:

$$q_{i_a} \alpha_{j_a} \to q_{r_a} \alpha_{s_a} d_{k_a}, \text{ де } \alpha = 1, \ldots, t. \tag{5.9}$$

При цьому можна вважати, що існують деякі узагальнені алфавіти A_0 та Q_0, в яких записано символи зовнішніх і внутрішніх алфавітів для всіх МТ (див. поняття універсальної МТ). Тоді символи $q_{i_a} \alpha_{j_a}, q_{r_a}, \alpha_{s_a}$ із (5. 9) є символами алфавітів A_0 та Q_0.

Зазначимо один із способів нумерації всіх МТ (геделева нумерація). Нехай $p_1, p_2, p_3 \ldots$ — послідовність усіх простих чисел, упорядкованих за зростанням (тобто послідовність 2, 3, 5, 7, 11, ...). Номером МТ M із програмою (5.9) назвемо число

$$n(M) = p_1^{i_1} p_2^{j_1} p_3^{r_1} p_4^{s_1} p_5^{k_1} \ldots p_{5t-4}^{i_t} p_{5t-3}^{j_t} p_{5t-2}^{r_t} p_{5t-1}^{s_t} p_{5t}^{k_t}. \tag{5.10}$$

Приклад. Користуючись виразом (5.10), визначимо номер машини M_1, розглянутої у прикладі 5.15. Це число $n(M_1) = 2^1 3^1 5^1 7^1 11^1 13^1 23^1 29^3$.

Зрозуміло, що не всі натуральні числа є номерами МТ. Якщо n — номер деякої машини в алфавітах A та Q, то її програму можна однозначно встановити за цим номером. Як і раніше, кодуємо натуральні числа символом 1. Розглядатимемо МТ, алфавіт яких складається тільки із символу 1. Нагадаємо, що машина називається застосовною до початкового слова, якщо вона, починаючи роботу з ним на стрічці, перейде до заключного стану.

Означення 5.15*. Машина М, що є застосовною до слова n(М) (тобто до кода свого власного номера), називається самозастосовною. Машина, яка є незастосовною до слова n(М), називається несамозастосовною.*

Машина M_1 (див. приклад 1 на с. 130) є самозастосовною. Прикладом несамозастосовної машини може бути машина, у правих частинах команд якої не зустрічаються заключні стани. Така машина не є застосовною до жодного слова.

Розглянемо алгоритмічну проблему самозастосування. Вона є такою: задати алгоритм, який за деякою МТ встановлює, чи самозастосовна вона. Відповідно до тези Тьюрінга такий алгоритм можна шукати у вигляді МТ, тобто потрібно побудувати МТ, яка була б застосовною до кодів номерів усіх МТ, і залежно від того, чи самозастосовна вона, чи ні, мала б різні заключні конфігурації.

Нехай, наприклад, у випадку самозастосовної машини заключна конфігурація має вигляд

$$\alpha_i q_z 1 \alpha_j, \tag{5.11}$$

де α_i і α_j — які завгодно слова із символів алфавіту, а у випадку несамозастосовної машини заключна конфігурація має вигляд

$$\alpha_i q_z 0 \alpha_j. \tag{5.12}$$

Теорема 5.7. *Проблема самозастосування є алгоритмічно нерозв'язуваною.*

Доведення проведемо від супротивного. Припустимо, що шукана машина *M* існує. Тоді можна побудувати машину *N*, яка:

- є застосовною до всіх кодів номерів несамозастосовних машин;
- не є застосовною до всіх кодів номерів самозастосовних машин.

Справді, машина *N* утворюється з машини *M* у такий спосіб: алфавіт зберігається незмінним, заключний стан q_z машини *M* вважається незаключним станом машини *N*, заключним станом якої є новий стан q'_z. Програма для машини *N* складається з усіх команд для машини *M* і ще двох команд: $q_z 1 \to q'_z 1E$, $q_z 0 \to q'_z 0E$. Очевидно, машина *N* задовольняє вимоги (5.11) і (5.12).

Машина *N* є або самозастосовною, або несамозастосовною. Якщо вона самозастосовна, тобто застосовна до коду свого номера, то вона застосовна також до коду самозастосовної машини, але тоді не виконується вимога (5.12). Якщо машина *N* несамозастосовна, тобто не застосовна до коду свого номера, то вона не застосовна також до коду несамозастосовної машини, але тоді не виконується вимога (5.11).

Отже, прийшли до суперечності, ґрунтуючись на допущенні про існування машини *M*, що розв'язує проблему самозастосування. Тому такої машини не існує, а це означає, що проблема самозастосування є алгоритмічно нерозв'язуваною.

Зазначимо, що не розв'язано загальну проблему: немає єдиного алгоритму, який розв'язував би проблему самозастосування. Про це свідчать розглянуті вище приклади конкретних МТ, для яких такий алгоритм існує.

Використовуючи результат теореми 5.7, можна довести нерозв'язуваність інших алгоритмічних проблем. Розглянемо, наприклад, алгоритмічну проблему застосування до початкового слова. Вона полягає ось у чому: потрібно знайти алгоритм, який за машиною *M* та словом *X* установлював би, чи застосовна машина *M* до цього слова. У термінах МТ цю проблему можна сформулювати так: чи можна побудувати машину, яка була б застосовною до всіх слів вигляду $n(M)0X$, де *M* — довільна машина, *X* — довільне слово, і у випадку, якщо машина *M* є застосовною до слова *X*, приводила б до заключної конфігурації $\alpha_i q_z 1\alpha_j$, а у протилежному випадку — до заключної конфігурації $\alpha_i q_z 0\alpha_j$, де α_i та α_j — які завгодно слова із символів алфавіту.

Теорема 5.8. *Проблема застосування МТ до початкового слова є алгоритмічно нерозв'язуваною.*

У термінах МТ це означає, що не існує такої МТ, яка розв'язала б цю проблему в зазначеному вище сенсі. Доведення проведемо від супротивного. Припустимо, що така машина *T* існує. Нехай є *H*-машина, яка подвоює слова (див. приклад 5 на с. 131). Розглянемо машину *G*, що є композицією машин *T* й *H*. Якщо в початковий момент роботи машини *G* на стрічці буде слово $n(M)$, то після роботи машини *H* на стрічці з'явиться слово $n(M)\square n(M)$. Машина *T* як застосовна до всіх таких слів зупиниться в конфігурації $\alpha_i q_z 1\alpha_j$, якщо машина M є застосовною до слова $n(M)$, і в конфігурації $\alpha_i q_z 0\alpha_j$, у протилежному випадку. Проте тоді машина T розв'язує проблему самозастосування, що неможливо за теоремою 5.7 .

Отже, прийшли до суперечності, виходячи з допущення, що існує машина, яка розв'язує проблему застосування. Тому такої машини не існує.

5.4. Нормальний алгоритм Маркова

Подальше узагальнення поняття алгоритму пов'язане з асоціативним численням, яке будується на множині всіх слів у заданому алфавіті. Цей підхід був запропонований на початку 50-х років минулого століття математиком А. А. Марковим.

Нагадаємо означення алфавіту, наведене в розд. 1. Алфавіт є будь-якою скінченною множиною деяких символів (літер). Будь-яка скінченна послідовність n літер деякого алфавіту — це слово завдовжки n у цьому алфавіті. Наприклад, в алфавіті з трьох літер $\{a, b, c\}$ словами є послідовності a, b, c, ab, bac, $abbca$, $aacbcca$ тощо. Порожнє слово, що не містить жодної літери, позначається як λ. Якщо слово α є частиною слова β, то кажуть, що слово α входить у слово β або слово α є підсловом слова β. Наприклад, у слові $\alpha = abcbcbabaa$ є чотири підслова a, два підслова bcb, одне підслово cba тощо. Визначимо операції асоціативного числення як систему підстановок, за допомогою яких одні слова перетворяться на інші.

Означення 5.16. *Нехай* γ *— слово в деякому алфавіті* Σ, *а* α *і* β *— підслова слова* γ. *Підстановка вигляду* $\alpha \leftrightarrow \beta$ *означає заміну в слові* γ *входження підслова* α *підсловом* β *так само, як і заміну підслова* β *підсловом* α.

Якщо в деякому слові γ є одне або кілька підслів α, то кожне з цих підслів може замінюватися словом β, і навпаки, якщо є підслово β, то його можна замінити словом α. Наприклад, підстановка підслів $kl \leftrightarrow lml$ застосовна чотирма способами до слова $klmlmlklkk$. Заміна кожного із двох входжень підслова lml дасть слова $kklmlkl$ та $klmklkl$, а заміна кожного із двох входжень підслова kl — слова $lmlmlmlkl$, $klmlmllml$. Водночас до слова $lkml$ ця підстановка не застосовна. Підстановка $\alpha \leftrightarrow \lambda$ означає, що з перетвореного слова викидається входження слова, а також, що між двома якими-небудь літерами перетвореного слова або слова перед чи за ним уставляється слово α.

Означення 5.17. *Асоціативне числення — це множина всіх слів у деякому алфавіті разом з якою-небудь скінченною системою підстановок.*

Отже, щоб задати асоціативне числення, досить визначити алфавіт і систему підстановок.

Приклад. Алфавіт $[a, b, c, a, e\}$ та система підстановок $ac \leftrightarrow ca$, $ad \leftrightarrow da$, $bc \leftrightarrow cb$, $bd \leftrightarrow db$, $abac \leftrightarrow abacc$, $eca \leftrightarrow ai$, $edb \leftrightarrow bi$ задають асоціативне числення.

Означення 5.18. *Два слова називаються еквівалентними, якщо одне з них можна утворити з іншого послідовним застосуванням деяких підстановок.*

Приклад. У попередньому прикладі еквівалентними є, наприклад, слова *abcae* і *caaeab*, що випливає з таких послідовних перетворень: *abcae*, *acbae*, *cabde*, *cadbe*, *cadedb*.

Очевидно, що наведене бінарне відношення дійсно має всі властивості відношення еквівалентності: рефлексивність, симетричність, транзитивність.

Еквівалентність слів α і β позначається як $\alpha \sim \beta$. Якщо $\alpha \sim \beta$, то за наявності в якому-небудь слові γ підслова α після підстановки $\alpha \leftrightarrow \beta$ утворюється слово, еквівалентне γ. Наприклад, скориставшись еквівалентністю *abcae* ~ *cadbe*, зі слова *bbabcaec* утворюємо еквівалентне йому слово *bbcadbec*.

У кожному асоціативному численні існує нескінченна множина різних слів. Відношення еквівалентності задає розбиття цієї множини на неперерізні підмножини слів, які є еквівалентними між собою.

Означення 5.19. *Послідовність слів* $\alpha_1, \alpha_2, ..., \alpha_n$, *коли кожне наступне слово є результатом одноразового застосування допустимої підстановки до попереднього слова, утворює дедуктивний ланцюжок, сусідні слова в якому називаються суміжними.*

У кожному асоціативному численні виникає своя проблема слів, а саме: для будь-яких двох слів у заданому численні потрібно визначити, еквівалентні вони чи ні. Очевидно, еквівалентність двох слів означає, що існує дедуктивний ланцюжок від одного слова до іншого або що обидва слова належать до одного класу еквівалентності. Оскільки в будь-якому асоціативному численні міститься нескінченна множина різних слів, відповідне вирішення питання про еквівалентність будь-яких двох слів потребує перебору нескінченної множини варіантів.

За допомогою алгоритму перебору вирішується обмежена проблема слів: потрібно встановити, чи можна одне із заданих слів перетворити на інше застосуванням допустимих підстановок не більш як k разів, де k — довільне, але фіксоване число. Для цього досить побудувати всі слова, суміжні з одним із заданих слів, після чого для кожного з утворених слів побудувати всі слова, суміжні з ним тощо, всього k разів. У результаті матимемо список усіх слів, які можна утворити із заданого слова за допомогою допустимих підстановок, застосовуваних не більш як k разів. Якщо друге задане слово буде в цьому списку, то відповідь на поставлене питання буде позитивною, а якщо його в списку немає, то — негативною.

Однак такий шлях принципово непридатний для вирішення необмеженої проблеми слів. Оскільки довжина дедуктивного ланцюжка, що простягається між еквівалентними словами (якщо такий ланцюжок існує), може бути як завгодно довгою, не існує ніякої можливості зазначити таке скінченне число k, що гарантує вирішення проблеми за допомогою простого перебору. Тому для здобуття бажаних результатів треба застосовувати інші ідеї, які ґрунтуються на аналізі механізму перетворення слів з використанням допустимих підстановок.

У деяких випадках можуть бути виявлені також властивості, незмінні для всіх слів дедуктивного ланцюжка.

Означення 5.20. *Властивості, що залишаються незмінними для всіх слів дедуктивного ланцюжка, називаються дедуктивно інваріантними.*

Приклад.У кожній із допустимих підстановок числення із прикладу на с. 137 ліва і права частини містять ту саму кількість підслів *a*. Отже, у будь-якому дедуктивному ланцюжку всі слова також мають містити те саме число підслів *a*. На основі цього дедуктивного інваріанта можна встановити, які слова не можуть бути еквівалентними. Так, слова *abacuac* та *abaaaac* — не еквівалентні в численні із зазначеного прикладу.

Проблема слів в асоціативному численні має величезне значення у зв'язку з тим, що до неї зводяться багато геометричних, алгебричних і логічних задач. Так, будь-яку формулу логіки висловлень та предикатів можна трактувати як слово в деякому алфавіті, яке містить логічні символи, висловлення, предикати і предметні змінні. Процес еквівалентного перетворення або виведення логічного висновку може бути поданий як перетворення слів, причому роль допустимих підстановок відіграють логічні закони або аксіоми.

Таким чином, питання про вивідність якої-небудь формули стає питанням існування дедуктивних ланцюжків, що ведуть від слів, які є посилками, до слів, що є наслідком. У низці інтерпретацій асоціативного числення, зокрема в теорії виведення, використовуються орієнтовані підстановки виду $\alpha \to \beta$, що допускають лише підстановку зліва направо (підслова α замість підслова β).

Система допустимих підстановок у деякому алфавіті із точним розпорядженням про порядок і спосіб їх використання дає змогу здійснити детермінований процес, який послідовно перетворює деяке слово на нові слова, еквівалентні початковому.

Означення 5.21. *Говорять, що задано алгоритм в алфавіті* Σ, *який є застосовним до слова* α *і перетворює його на слово* β, *якщо, виходячи з* α *і діючи відповідно до розпоряджень, зрештою він дає* β. *Множина слів, до яких застосований цей алгоритм, називається областю його застосування. Два алгоритми в деякому алфавіті є еквівалентними, якщо області їх застосування та результати перетворення ними будь-якого слова з загальної області збігаються.*

Важливий крок на шляху уточнення поняття алгоритму зроблено А. А. Марковим. Означення нормального алгоритму Маркова зводиться ось до чого.

Означення 5.22. *Нехай задано алфавіт* Σ *і зафіксовано впоряковану (задану в певному порядку) систему орієнтованих підстановок P. Виходячи з довільного слова* α *в алфавіті* Σ *розглядаються підстановки в тому порядку, в якому їх задано. Перша підстановка, що зустрілася, з лівою частиною, яка є підсловом* α, *використовується для перетворення* α, *в яке замість першого входження лівої частини підстановки підставляється її права частина, внаслідок чого утворюється нове слово* α_1. *Далі, виходячи зі слова* α_1, *процес повторюється, поки він не зупиниться, ознаками чого є два випадки:*

- *коли утворюється таке слово* α_n, *що жодна з лівих частин допустимих підстановок не є його підсловом;*
- *коли при утворенні* α_n *використано останню підстановку.*

Пара $(\Sigma,\ P)$ *визначає нормальний алгоритм Маркова.*

Приклади: 1. Нехай задано алфавіт $\Sigma = \{1, +\}$ і систему підстановок $+ \to 1$, $1 \to 1$ (λ — порожнє слово). Слово 1 + 11 + + 1111 + 1 алгоритм перетворює так:

1 + 11 + 1111 + 1
111+1111+1
111111 + 1
11111111
11111111

Процес закінчується застосуванням заключної підстановки, що перетворює слово на самого себе, тобто алгоритм підсумовує кількість одиниць, здійснюючи операцію додавання. Еквівалентний йому алгоритм можна задати за допомогою системи підстановок $1+ \to +1$, $+1 \to 1$, $1 \to 1$.

2. Нехай задано алфавіт $\Sigma = \{1, 0\}$ і систему підстановок: $1 \to \lambda$, $0 \to 0$ (λ — порожнє слово). Алгоритм перетворює будь-яке слово в алфавіті $\{1, 0\}$, в якому є хоча б одне входження символу 1, на слово, що утворюється викреслюванням із нього лівого входження символу 1. Порожнє слово він перетворює на самого себе, а до непорожніх слів, які не мають символу 1, алгоритм не застосовний. Так, слово 0101 перетворюється на слово 001.

3. Нехай задано алфавіт $\Sigma = \{1\}$, а система підстановок складається лише з однієї підстановки $\lambda \to 1$ (λ — порожнє слово). Алгоритм до будь-якого слова в алфавіті $\{1\}$ дописує на початку слова одну одиницю. Отже, він обчислює функцію $S(x) = x + 1$.

Означення 5.23. *Функція $f(x_1, ..., x_n)$ називається нормально обчислюваною, якщо існує деякий нормальний алгоритм (Σ, P), який розраховує значення цієї функції для всіх наборів значень аргументів $x_1, ..., x_n$ з області її визначення і який не є застосовним до наборів значень аргументів $x_1, ..., x_n$, що не належать до області визначення заданої функції.*

А. А. Марков висунув гіпотезу, що дістала назву принципу нормалізації Маркова і формулюється так: «*Кожна обчислювана функція є нормально обчислюваною*».

Довести принцип нормалізації Маркова, як і гипотези Черча та Тьюрінга, не можна за аналогічними причинами, але нормальний алгоритм Маркова приймається як ще одна стандартна форма будь-якого алгоритму. Виникає питання про взаємозв'язок цих теорій. Відповідь на це дає така теорема.

Теорема 5.9. *Збігаються класи всіх частково-рекурсивних, обчислюваних за Тьюрінгом і нормально обчислюваних функцій.*

Це вже не гіпотеза, а математична теорема, яка потребує строгого доведення, причому вже доведено еквівалентність класу частково-рекурсивних та обчислюваних за Тьюрінгом функцій. Аналогічно доводиться решта тверджень цієї теореми.

Крім розглянутих класів точних алгоритмів, існують інші класи, причому теж можна довести їх еквівалентність розглянутим класам.

5.5. Загальна теорія алгоритмів

Із теореми 5.9 випливає, що будь-які твердження про існування або неіснування алгоритмів, сформульовані в одній з цих теорій, справджуються також в іншій. Це означає, що такі твердження можна формулювати для алгоритмів узагалі, не фіксуючи конкретну модель, як це робилось раніше, і використовуючи результати всіляких теорій. Після того як поняття алгоритму було точно визначено, питання про алгоритмічну можливість розв'язання того чи іншого класу задач ставиться теж точно: чи існує яка-небудь стандартна форма алгоритму для певного класу задач? Іноді на це питання теорія алгоритмів дає негативну відповідь.

У спеціальних розділах математики строго доведено нерозв'язуваність низки задач, наприклад неможливість розв'язання в радикалах алгебричних рівнянь вище четвертого степеня, неможливість трисекції кута за допомогою циркуля і лінійки тощо. Відмінна риса теорії алгоритмів полягає в тому, що вона намагається розв'язувати найзагальніші проблеми.

Наріжним каменем теорії алгоритмів стала проблема розпізнавання вивідності: для будь-яких двох формул *R* та *S* у логічному численні з'ясувати, чи існує дедуктивний ланцюжок, що веде від *R* до *S*. Виявилося, що ця проблема алгоритмічно нерозв'язувана. Негативну відповідь одержано також для проблеми розпізнавання еквівалентності слів у будь-якому асоціативному численні. Були побудовані конкретні приклади асоціативних числень, в яких не існує алгоритму, що дає змогу для будь-якої пари слів установити, еквівалентні вони чи ні.

Таким чином, можливим є викладання теорії алгоритмів, інваріантне відносно способу формалізації поняття «алгоритм», коли при будь-якій формалізації основні властивості алгоритмів залишаються тими самими. Основне поняття такої інваріантної теорії (будемо називати її загальною теорією алгоритмів) — це алгоритм (рекурсивний опис функції, система команд МТ, нормальний алгоритм Маркова або опис у якій-небудь іншій моделі; вважається, що вибрано якусь модель, але яку саме — не є важливим) та обчислювана функція. Нагадаємо, що функція називається обчислюваною, якщо існує алгоритм, який її обчислює. При цьому несуттєво, числова це функція чи ні. Термін «загальнорекурсивна функція», в інваріантному значенні є синонімом терміна «всюди визначена обчислювана функція».

Еквівалентність тверджень «функція *f* обчислювана» й «існує алгоритм, що обчислює функцію *f*» іноді приводить до змішування понять алгоритму й обчислюваної функції; зокрема, кажучи про рекурсивну функцію, часто мають на увазі її конкретний рекурсивний опис. Насправді ж відмінність між обчислюваною функцією та алгоритмом — це відмінність між функцією і способом її задання. Без цих відмінностей неможливою є коректна інтерпретація деяких важливих результатів теорії алгоритмів.

Це так, коли йдеться про існування чи не існування алгоритмів. Характеристики якості алгоритмів (їх складності в тому чи іншому розумінні), взагалі кажучи, неінваріантні відносно вибраної формалізації. Водночас традиція викладу теорії алгоритмів дає змогу не розрізнювати поняття алгоритму та функції в тих випадках, коли це не призводить до плутанини.

Тут резюмовано одержані результати теорії алгоритмів в їх інваріантному вигляді. Вони стосуються будь-якої формалізації поняття «алгоритм», що не заважає використати в кожному конкретному міркуванні конкретну алгоритмічну модель, яка найкраще відповідає тому чи іншому випадку.

5.6. Прикладна теорія алгоритмів

Стандартні форми подання алгоритмів, такі як нормальний алгоритм Маркова, через їхній надзвичайно високий ступінь деталізації непридатні для інженерної практики. МТ є зручною абстрактною моделлю реалізації будь-якого алгоритму людиною або обчислювальною машиною, але в реальних умовах будь-який вид пам'яті та час функціонування не є нескінченними, а, навпаки, жорстко обмежені. Водночас при розробці й реалізації конкретних алгоритмів в інженерній практиці досить виходити з їхніх загальних властивостей.

Прикладну теорію алгоритмів мало турбує власне існування алгоритмів (це просто мається на увазі), свої зусилля вона спрямовує в основному на розробку найефективніших методів їх опису, перетворення і реалізації. Алгоритм розглядається як сукупність пов'язаних між собою операторів, які відображають елементарні операції перетворення множини об'єктів. Способи реалізації операторів є строго визначеними (як правило, оператори самі є деякими стандартними алгоритмами), а при конкретній реалізації алгоритму задаються також значення початкових даних та параметрів, що входять в опис операторів.

Для опису алгоритмів використовуються різні методи, які різняться ступенем деталізації і формалізації. Теоретичний опис дається у формалізованому вигляді, мета якого — обґрунтувати без зайвих подробиць процедуру, пропоновану як алгоритм. Для наочного подання структури алгоритмів широко застосовуються графічні засоби: графи, схеми, мережі. Формальний та повний опис алгоритмів здійснюється на спеціально розроблених для цієї мети алгоритмічних мовах; він містить усю потрібну для реалізації алгоритму інформацію, не пов'язану безпосередньо зі специфічними особливостями обчислювальних машин. Машинна реалізація алгоритму потребує перекладу його на мову, властиву певній машині, у вигляді програми. Роль автоматичних перекладачів з алгоритмічних мов відіграють спеціальні програми-транслятори. Часто загальний опис алгоритму безпосередньо переводиться на машинну мову розшифруванням операторів алгоритму в обчислювальній машині.

На відміну від загальної теорії алгоритмів прикладна теорія розглядає не тільки детерміновані, а й імовірнісні (статистичні) й евристичні алгоритми. В останньому випадку, крім детермінованих або статистично заданих правил, алгоритм включає також змістовні вказівки про доцільний напрям процесу.

5.7. Теорема Райса

Розглядаючи накопичений запас алгоритмічно нерозв'язуваних проблем, неважко помітити, що майже всі вони так чи інакше пов'язані з самозастосовністю — досить екзотичною ситуацією, коли алгоритм працює з власним описом. Виходячи із прикладної теорії алгоритмів можна вирішити, що оскільки поняття самозастосовності далеке від алгоритмічної практики, нерозв'язуваність у ній також ніколи не зустрінеться. Проте це не так.

Теорема 5.10 (теорема Райса). *Нехай C — будь-який клас обчислюваних функцій однієї змінної, нетривіальний у тому значенні, що існують як функції, які належать до C, так і функції, які не належать до C. Тоді не існує алгоритму, який би за номером x функції* f_x *визначав, належить* f_x *класу C чи ні (в цьому випадку кажуть, що множина* $\{x | f_x \in C\}$ *є нерозв'язуваною).*

Доведення. Припустимо, що множина $M = \{x | f_x \in C\}$ є розв'язуваною; тоді вона має характеристичну функцію

$$\chi_M(x) = \begin{cases} 1, \text{ якщо } f_x \in C \text{ (тобто } x \in M); \\ 0, \text{ якщо } f_x \notin C. \end{cases}$$

Нехай ніде не визначено функцію $f_0 \in \bar{M}$. Виберемо будь-яку конкретну обчислювану функцію $f_a \in M$ і визначимо функцію $F(x, y)$:

$$F(x, y) = \begin{cases} f_a(y), \text{ якщо } f_x(x) \text{ — визначена;} \\ f_0(y), \text{ якщо } f_x(x) \text{ — невизначена.} \end{cases}$$

Функція $F(x, y)$ є обчислюваною: для її обчислення треба обчислювати $f_x(x)$; якщо $f_x(x)$ визначено, то цей процес коли-небудь зупиниться і тоді треба перейти до обчислення $f_a(y)$; якщо ж $f_x(x)$ є невизначеною, то процес не зупиниться, що рівносильно обчисленню ніде не визначеної функції $f_0(x)$. Якщо в $F(x, y)$ зафіксувати x, то вона стане обчислюваною функцією однієї змінної y. Номер цієї функції залежить від значення x, тобто всюди визначеною є функція $g(x)$; неважко показати, що $g(x)$ — обчислювана функція. Отже,

$$f_{g(x)}(y) = \begin{cases} f_a(y), \text{ якщо } f_x(x) \text{ — визначена;} \\ f_0(y), \text{ якщо } f_x(x) \text{ — невизначена.} \end{cases}$$

Таким чином, $f_{g(x)} \in M$, якщо і тільки якщо $f_x(x)$ є визначеною функцією (оскільки $f_a \in M$, $f_0 \in \overline{M}$). Звідси

$$\chi_M(g(x)) = \begin{cases} 1, \text{ якщо } f_x(x) \text{ — визначена;} \\ 0, \text{ якщо } f_x(x) \text{ — невизначена,} \end{cases}$$

тобто побудовано функцію для проблеми самозастосування, що неможливо.

Для випадку, коли $f_0 \in M$, вибираємо $f_a \in \bar{M}$; подальші міркування аналогічні, а розв'язувальна функція для проблеми самозастосування матиме вигляд $1-\chi_M(g(x))$.

Із теореми Райса випливає, що за номером обчислюваної функції не можна дізнатися, чи є ця функція сталою, періодичною, обмеженою, чи містить вона серед своїх значень задане число і т. д. Створюється враження, що взагалі нічого не можна дізнатися й усе на світі — нерозв'язуване. З іншого боку, чи не суперечить теоремі Райса той очевидний факт, що за номером машини T завжди можна дізнатися, наприклад, чи містить вона більше 10 команд, чи ні (див. п. 5.2).

Для того щоб розібратися в значенні теореми Райса, треба передусім пригадати, що номер x функції f — це номер алгоритму A_x, який обчислює f; у свою чергу, за номером алгоритму однозначно відновлюється його опис, а різним номерам відповідають різні алгоритми. Для функцій це не так: при $x \neq y$ f_x й f_y можуть бути однією і тією самою функцією (в її класичному значенні див. розд. 1). У теоремі Райса беруть участь як алгоритми, так і функції, які потрібно чітко розрізнювати. Клас C — це клас (або властивість) функцій; водночас f_x означає «функцію, що обчислюється алгоритмом A_x».

Таким чином, значення теореми Райса в тому, що за описом алгоритму нічого не можна дізнатися про властивості функції, яку він обчислює. Зокрема, виявляється нерозв'язуваною проблема еквівалентності алгоритмів: за двома заданими алгоритмами не можна встановити, обчислюють вони одну і ту саму функцію чи ні. Кількість команд - це властивість не функції, а опису алгоритму; теореми Райса вона не стосується.

Досвідчений програміст знає, що за текстом складної програми, не запускаючи її в роботу, важко зрозуміти, що вона робить (тобто яку функцію обчислює), не маючи гіпотез про те, що вона має робити; якщо це розуміння і настає, то кожний раз по-різному; систематичного методу тут не існує. З іншого боку, в тексті можна виявити алгоритмічним способом синтаксичні помилки (що і роблять компілятори з алгоритмічних мов), тобто встановити ті або інші властивості опису алгоритму. Тут варто згадати два поняття, про які детальніше йтиметься в наступному розділі: синтаксис і семантику. Синтаксичні властивості алгоритму — це властивості текстів, які його описують, тобто властивості скінченних слів у фіксованому алфавіті. Семантичні (або смислові) властивості алгоритмів пов'язані з тим, що він робить; їх природно описувати в термінах функцій та класів еквівалентності функцій, які обчислюються алгоритмами. Добре відомо, що у процесі налагодження програм синтаксичні помилки відшукуються досить швидко; всі прикрощі пов'язані з аналізом семантики неналагодженої програми, тобто зі спробами встановити, що ж вона робить замість того, щоб робити задумане. У трохи вільній і неформальній інтерпретації теорема Райса могла б формулюватися так: «за синтаксисом алгоритму нічого не можна дізнатися про його семантику».

Кілька разів повторюваний вираз «нічого не можна дізнатися» — це, строго кажучи, перебільшення. Його точний еквівалент «не існує единого алгоритму, що дає змогу дізнатися». До того ж ідеться про неможливість розпізнавання властивостей обчислюваних функцій, записаних в універсальній алгоритмічній мові (мові МТ та ін.). Властивості підкласів обчислюваних функцій, описаних у спеціальних мовах, цілком можуть бути розв'язуваними. Наприклад, у розд. 3 розглядалися алгоритми розпізнавання властивостей логічних функцій, описаних на мові формул у різних базисах.

Досі йшлося про нерозв'язувані проблеми загальної теорії алгоритмів. Деякі з них, такі як проблема зупину або теорема Райса, мають реальну інтерпретацію у практиці програмування. Нерозв'язуваність виникає також в інших теоріях: автоматів, мов і граматик тощо. Поступово вони стають побутом дискретної математики, і з їх існуванням доводиться рахуватися. З теоретичної точки зору нерозв'язуваність — не невдача, а науковий факт. Знання основної нерозв'язуваності теорії алгоритмів має бути для фахівця з дискретної математики таким самим елементом наукової культури, як для фізика знання про неможливість створення вічного двигуна. Якщо ж важливо мати справу з розв'язуваною задачею (а для прикладних наук це прагнення природне), то потрібно чітко уявляти собі дві обставини.

По-перше, відсутність загального алгоритму, який вирішує цю проблему, не означає, що в кожному окремому випадку не можна досягти успіху. Тому, якщо проблема є нерозв'язуваною, треба шукати її розв'язувані окремі випадки. По-друге, поява нерозв'язуваності — це, як правило, результат надмірної загальності задачі (або мови, на якій описано об'єкти задачі). Задача в більш загальній постановці має більше шансів виявитися нерозв'язуваною.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Які основні вимоги до алгоритмів?
2. Що таке рекурсивна функція? Що таке примітивна рекурсія?
3. Які основні властивості алгоритмів?
4. Що таке примітивно-рекурсивна функція?
5. Що таке частково-рекурсивна функція?
6. Як формулюється теорема Черча?
7. Як формально описується МТ?
8. Що таке внутрішній і зовнішній алфавіти МТ?
9. Що таке конфігурація на стрічці МТ?
10. Що таке застосовність МТ?
11. Які операції виконуються над МТ?
12. Що таке функція, обчислювана за Тьюрінгом?
13. Що таке універсальна МТ?
14. Що таке функціональна схема МТ?
15. Як формулюється теза Тьюрінга? У чому суть проблеми її зупину?
16. Що таке алгоритмічна розв'язуваність?
17. Який зв'язок рекурсивних функцій з МТ?
18. Що таке алгоритмічна проблема?
19. Які є приклади алгоритмічно нерозв'язуваних проблем?

20. Що таке асоціативне числення?
21. Що таке нормальний алгоритм Маркова?
22. У чому суть еквівалентності трьох алгоритмічних моделей?
23. Як формулюється теорема Райса?

ПЕРЕЛІК ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

1. Побудова МТ.
2. Побудова розв'язувальних процедур.

СПИСОК ЛІТЕРАТУРИ

1. *Мальцев А. И.* Алгоритмы и рекурсивные функции. — М.: Наука, 1986. — 368 с.
2. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера. — М.: Энергоатомиздат, 1988. — 480 с.
3. *Сигорский В. П.* Математический алгоритм инженера. — К.: Техніка, 1977. — 768 с.
4. *Минский М.* Вычисления и автоматы. — М.: Мир, 1971.
5. *Трахтенброт Б. А.* Алгоритмы и вычислительные автоматы. — М.: Сов. радио, 1974.
6. *Катленд Н.* Вычислимость. Введение в теорию рекурсивных функций. — М.: Мир, 1983.
7. *Клини С. К.* Введение в математику. — М.: Изд-во иностр. лит., 1957.
8. *Игошин В. И.* Математическая логика и теория алгоритмов. — Саратов.: Изд-во Сарат. ун-та, 1991. - 256 с.
9. *Касаткин В. Н., Владыкина Л. И.* Алгоритмы и игры. — К.: Рад. шк., 1984. — 80 с.
10. *Лавров И. А., Максимова Л. Л.* Задачи по теории множеств, математической логике и теории алгоритмов. — М.: Наука, 1975. — 240 с.
11. *Эббинхауз Г. Д., Янобс К., Ман Ф. К., Херми Г.* Машины Тьюринга и рекурсивные функции. — М.: Мир, 1973. — 264 с.

РОЗДІЛ 6

ФОРМАЛЬНІ СИСТЕМИ

6.1. Основні поняття

6.1.1. Поняття формальної системи

Формальні системи — це системи операцій над об'єктами, які розуміють як послідовності символів (тобто як слова у фіксованих алфавітах); самі операції також є операціями над символами. Термін «формальний» підкреслює, що об'єкти й операції над ними розглядаються суто формально, без будь-яких змістовних інтерпретацій символів. Передбачається, що між символами не існує ніяких зв'язків і відношень, крім тих, які явно описуються засобами самої формальної системи.

Означення 6.1. *Під формальною системою будемо розуміти деяку абстрактну (математичну) систему, що складається з множини аксіом та множини правил виведення, яку можна реалізувати машинним способом, наприклад за допомогою програми ЕОМ.*

Означення 6.2. *Аксіоми — це твердження про задані об'єкти і взаємозв'язки між ними, сформульовані у вигляді рядків деякої мови.*

У формальних системах прагнуть використати тільки синтаксичні аспекти мови, абстрагуючись від питань семантики та прагматики. Множиною аксіом є множина заздалегідь визначених рядків, яка може бути як скінченною, так і нескінченною. В останньому випадку має існувати алгоритм, який дає змогу генерувати всі можливі аксіоми.

Означення 6.3. *Правилами виведення називають чітко сформульовані процедури, які дають змогу утворювати нові рядки з тих, що існують.*

Нові рядки є теоремами формальної системи. Питання скінченності або нескінченності множини правил виведення вирішується так, як і у випадку з аксіомами.

Розглянемо більш детально й абстрактно поняття формальної системи та її функціонування при доведенні теорем.

Позначимо формальну систему через F, множину аксіом, як і множину рядків, — через $(\beta_1, \beta_2, ..., \beta_n)$. Правило виведення в найзагальнішому вигляді можна подати обчислюваною (тобто такою, значення якої можна знайти за скінченне число кроків) функцією $R\,(\beta, \beta_1, \beta_2, ..., \beta_k)$ з числом аргументів (k + 1). Тут як аргументи використано рядки. Функція набуває значення істинності або хибності (I, X). Якщо $R\,(\beta, \beta_1, \beta_2, ..., \beta_k) = I$, то кажуть, що рядок β безпосередньо виводиться з рядків $\beta, \beta_1, \beta_2, ..., \beta_k$ за правилом виведення. Надаючи далі правилу виведення деякої конкретної форми, дістаємо формальні системи окремого вигляду, які дають змогу ефективно розв'язувати деякий клас прикладних задач.

Означення 6.4. *Доведенням в F теореми (рядка) називається скінченна послідовність рядків* $w_1, w_2, ..., w_e$ *така, що кожний рядок з цієї послідовності є або аксіомою в F, або безпосередньо виводиться з деякої множини рядків, кожний з яких передує* w_i *у послідовності; теоремою є останній рядок у послідовності.*

Розглянемо приклад, що ілюструє означення і процес функціонування формальної системи.

Нехай задано алфавіт $A = \{a, b, c, d, e, f\}$, з якого складено рядки формальної системи. Множина аксіом формальної системи: 1) *abffe*; 2) *eedca*; 3) *ffb*.

Правила виведення:

$$R_1\,(ff, eedca, abffe) = I;$$

$$R_2\,(abca, ff, ffb, eedca) = I.$$

Знайдемо можливе виведення в цій формальній системі, нумеруючи для зручності кожний рядок виведення:

1) *eedca* — аксіома 2;
2) *abffe* — аксіома 1;
3) *ff* — безпосередньо виводиться з (1), (2) та R_1;
4) *ffb* —аксіома 3;
5) *abca* — безпосередньо виводиться з (1), (3), (4) й R_2.

Таким чином, формальна система виводить як теорему рядок *abca*. Зазначимо, що рядкам, які утворюються за допомогою цієї системи, не надається ніякого значення, а ними тільки маніпулюють на суто формальному рівні.

Розглянемо приклад формальної системи, рядки якої мають певне значення, а саме: є правильними формулами логіки висловлень, а будь-який рядок, що є теорією формальної системи, — це загальнозначуща формула. Зазначимо, що в цьому випадку маніпуляція рядками проводиться на суто формальному рівні за правилами функціонування формальної системи.

Нехай задано алфавіт $A = \{p, q, r, (,), \supset, -\}$, де $\supset$ — операція імплікації. Множина аксіом:

1) $(p \supset q) \supset ((q \supset r) \supset p \supset r))$;
2) $(\overline{p} \supset p) \supset p$;
3) $p \supset (\overline{p} \supset q)$.

Ці аксіоми в математичній логіці відомо як аксіоми Лукасевича.

Правила виведення:

$$R_1(\beta, \alpha, \alpha \supset \beta) = I\ ;$$

$$R_2(\alpha_\beta^a, \alpha) = I\ .$$

У правилах виведення α, β є будь-якими правильними формулами логіки висловлень (тобто правильними рядками і підрядками формальної системи), а рядок α_β^a у правилі R_2 утворено з рядка α підстановкою рядка β замість усіх входжень у нього символу a.

Покажемо, що $p \supset p$ є теоремою цієї формальної системи, для чого наведемо виведення рядка $p \supset p$:

1) $(p \supset q) \supset ((q \supset r) \supset p \supset r))$ — аксіома 1;

2) $(p \supset (\bar{p} \supset q)) \supset (((\bar{p} \supset q) \supset r) \supset (p \supset r))$ — знайдено з (1) та $R_2(\alpha_{\bar{p} \supset q}^q, \alpha)$, де α — рядок;

3) $p \supset (\bar{p} \supset q)$ — аксіома 3;

4) $((\bar{p} \supset q) \supset r) \supset (p \supset r)$ — знайдено з (3), (2), R_1. При цьому $\alpha = p \supset (\bar{p} \supset q)$, $\beta = ((\bar{p} \supset q) \supset r) \supset (p \supset r)$;

5) $((\bar{p} \supset p) \supset r) \supset (p \supset r)$ — знайдено з (4) й $R_2(\alpha_p^q, \alpha)$;

6) $((\bar{p} \supset p) \supset p) \supset (p \supset p)$ — знайдено з (5) та $R_2(\alpha_p^r, \alpha)$;

7) $(\bar{p} \supset p) \supset p$ — аксіома 2;

8) $p \supset p$ — знайдено з (7), (6), R_1.

Отже, щоб довести теорему у формальній системі, треба спочатку виписати аксіоми і застосувати до них правила виведення. Усі рядки, утворені таким чином, складають теореми першого рівня. Якщо правила виведення застосовують до теорем першого рівня, то одержують теореми наступного рівня і т. д. Іноді аксіоми розглядають як теореми нульового рівня. При такому підході будь-який рядок доведення (виведення) є теоремою формальної системи.

Для будь-якого заданого рядка нелегко перевірити, чи є він теоремою формальної системи, що розглядається. Загалом не існує алгоритму, який би визначив, чи є цей рядок теоремою. Можна було б спробувати виконати всі можливі виведення цього рядка, і, якщо цей рядок є теоремою, виведення буде знайдено (розроблено алгоритми відшукання такого виведення). Якщо ж рядок не є теоремою, то процес пошуку потрібного виведення ніколи не закінчиться. У цьому разі не можна ні довести, ні спростувати заданий рядок. Якщо ж задано послідовність рядків, яка претендує на те, щоб бути виведенням, то її легко перевірити. Якщо кожний рядок послідовності є аксіомою або може бути утворений з попередніх рядків за правилами виведення, то послідовність стає виведенням.

Формальна система має бути такою, яка алгоритмічно реалізується. Мається на увазі, що для пошуку будь-якого виведення у формальній системі повинен не тільки існувати алгоритм виведення, але цей алгоритм має ще допускати реалізацію на існуючих ЕОМ. Інакше формальна система залишиться абстрактною побудовою.

Історично теорія формальних систем, як і теорія алгоритмів, виникла в межах основ математики при дослідженні побудови аксіоматичних теорій та методів доведення в них. З їх вивчення і продовжимо знайомство з формальними системами.

6.1.2. Аксіоматичний спосіб опису висловлень

Таблиці істинності дають змогу відповісти на багато питань, що стосуються формул логіки висловлень (наприклад, на питання про рівнозначність двох формул). Однак складніші питання логіки висловлень уже не можуть бути вирішені за допомогою таблиць істинності. Для розв'язання цих проблем використовується інший спосіб опису — аксіоматичний, при якому застосовуються аксіоматичні теорії.

Аксіоматичні теорії — це формальні системи, які відповідають численню висловлень. Як і кожні формальні системи, вони містять алфавіт, систему аксіом, правила виведення. Аксіоматична теорія будується таким чином:

- визначається набір символів теорії *T*, тобто скінченних послідовностей, які називаються виразами теорії *T*;
- визначається множина формул, тобто підмножин виразів теорії Т;
- визначається підмножина формул, які називаються аксіомами теорії *T*;
- задається множина $R_1, ..., R_m$ відношень між формулами, які називаються правилами виведення.

Означення 6.5. *Якщо формула А і формули* $A_1, ..., A_j$ *знаходяться в деякому відношенні* R_i*, то А є безпосереднім висновком із формул* $A_1, ..., A_j$*, сформульованим за правилом* R_i.

Означення 6.6. *Виведенням є будь-яка послідовність формул* $A_1, ..., A_n$*, в якій для будь-якого і формула* A_i *є або аксіомою, або висновком з яких-небудь попередніх формул.*

Означення 6.7. *Формула А називається теоремою теорії Т, якщо в ній існує виведення, в якому останньою формулою є А. Це виведення називається виведенням формули А. Отже, теореми аксіоматичної теорії — це формули, які можуть бути виведені за певними правилами.*

Означення 6.8. *Формула А називається висновком із множини формул Г, якщо існує така послідовність формул* $A_1, ..., A_n$*, що* $A_n = A$*, і для будь-якого* i*,* $1 < i < n$*,* A_i *є або аксіомою, або формулою із Г, або висновком із деяких попередніх формул. Ця послідовність називається виведенням А з Г. Елементи Г називаються посилками виведення, або гіпотезами. Цей факт будемо позначати як* $\Gamma \vdash A$ *(тобто А є висновком Г).* Якщо множина *Г* складається із формул $B_1, ..., B_k$, то це можна записати як $B_1, ..., B_k \vdash A$. Якщо *Г* — порожня множина, то в цьому випадку писатимемо $\vdash A$ (це означає, що *А* — аксіома).

Розглянемо деякі властивості поняття вивідності з системи гіпотез. Нехай *Г* — довільна множина формул $\Gamma = \{A_1, \ldots, A_k; B_1, \ldots, B_k; C_1, \ldots, C_k; D_1, \ldots, D_k\}$, а *A*, *B*, *C* — довільні формули. Тоді:

I. $\Gamma, A \vdash A$. Справді, виведення *А* з системи гіпотез *Г*, *А* складається з однієї формули *А*.

II. Якщо $\Gamma, A, B \vdash C$, то $\Gamma, B, A \vdash C$.

III. Якщо $\Gamma \vdash A$ і *B* — формула з *Г*, то $\Gamma, B \vdash A$. Справді, як виведення *А* з системи гіпотез *Г*, *В* можна взяти виведення *А* з системи гіпотез *Г*.

IV. Якщо $\Gamma \vdash A$, $\Gamma \vdash B$ й $A, B \vdash C$, то $\Gamma \vdash C$. Нехай $A_1, ..., A_n$ — виведення *А* з *Г*; $B_1, ..., B_m$ — виведення *В* з *Г*; $C_1, ..., C_k$ — виведення *С* з *А*, *В*. Тому $C_1, ..., C_k$ є виведенням *С* з *Г*.

V. Якщо $\Gamma, A \vdash B$ та $\Gamma \vdash A$, то $\Gamma \vdash B$ (правило вилучення гіпотези, що виводиться). Нехай $B_1, ..., B_n$ — виведення *В* з *Г*, *А*. Якщо в ньому не зустрічається формула A, то маємо виведення B з Γ. Якщо в цьому виведенні зустрічається формула A, то нехай $B_{i_1}, ..., B_{i_k}$ — усі входження формули A у виведення. Нехай також $A_1, ..., A_m$ — виведення A з Γ. Тоді $B_1, \ldots, B_{i_1-1}; A_1, \ldots, A_m; B_{i_1+1}, \ldots, B_{i_2-1}; A_1, \ldots, A_m; B_{i_2+1}, \ldots, B_{i_k-1}; A_1, \ldots, A_m;$ $B_{i_k+1}, \ldots, B_n$ — виведення B із Γ.

6.1.3. Властивості числення висловлень

У численні висловлень знову зустрічаємося з об'єктами, з якими вже мали справу, — з формулами алгебри логіки. Однак тут вони розглядаються не як список подання функцій, а як складові висловлення, утворені з елементарних висловлень (змінних) за допомогою логічних операцій або, як кажуть у логіці, зв'язок $\vee, \&(\cdot), {}^{-}, \rightarrow$. При цьому особлива увага приділяється тотожно-істинним висловленням, оскільки вони мають входити в будь-яку теорію як загальнологічні закони. Їх породження і є основною задачею числення висловлень.

Розглянемо один із способів формалізації логіки висловлень — числення висловлень, що входить до складу формальної аксіоматичної теорії. Числення висловлень визначається таким чином.

1. Задаються символи числення висловлень: ${}^{-}, \rightarrow, (,), \&(\cdot), (\sim), \cup$ і змінні (літери) з цілими додатними числами як індексами: X_1, X_2, X_3 ... Символи ${}^{-}$ та $\rightarrow$ (заперечення й імплікація відповідно) — логічні символи.

2. Задаються формули:

а) усі змінні X_i — формули;

б) якщо A і B — формули, то $\bar{A}$ та $(A \rightarrow B)$ — також формули.

Якщо A, B, C — формули, то $(C \rightarrow (A \rightarrow B)), (((\bar{A}) \rightarrow B) \rightarrow \bar{C})$ — також формули.

Для скорочення запису можна опускати у формулі зовнішні дужки і ті пари дужок, без яких можна відновити формулу.

3. Використовуються аксіоми числення висловлень. Які б не були формули A, B та C, такі формули є аксіомами:

А1. $A \rightarrow (B \rightarrow A)$;

А2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$;

А3. $(\bar{B} \rightarrow \bar{A}) \rightarrow ((\bar{B} \rightarrow A) \rightarrow B)$.

Вирази А1—А3 називаються схемами аксіом, оскільки кожен з них породжує нескінченну множину формул, що є аксіомами числення висловлень. Наприклад, формула $X_1 \rightarrow (X_2 \rightarrow X_1)$ є аксіомою, утвореною за схемою А1, формула $(\bar{A} \rightarrow \bar{A}) \rightarrow ((\bar{A} \rightarrow A) \rightarrow A)$ (де A — будь-яка формула) — аксіома, утворена за схемою А3.

4. Застосовуються правила виведення. Використовується тільки одне правило виведення modus ponens (скорочено m. p.). Нехай існують три формули: A, $A \rightarrow B$ і B. Про формулу B будемо говорити, що вона утворюється за правилом виведення m.p. з формул A та $A \rightarrow B$.

Хоча для числення висловлень вибрано тільки два логічних символи ${}^{-}$ і $\rightarrow$, можна ввести й інші операції: $\&(\cdot), (\sim)$.

Поряд із властивостями вивідності I—V (див. п. 6.1.2) в численні висловлень виконується також така властивість:

VI. Якщо $\Gamma \vdash A \to B$ та $\Gamma \vdash A$, то $\Gamma \vdash B$. Нехай $A_1, ..., A_n$ — виведення A з Γ, де A_n збігається з A. Нехай $B_1, ..., B_m$ — виведення $A \to B$ з Γ, де B_m збігається з $A \to B$. Тоді $A_1, ..., A_n, B_1, ..., B_m, B$ — виведення B з Γ. Останню формулу в цьому виведенні утворено застосуванням правила m. p. до формул A_n і B_m.

Твердження 6.1. *$\vdash A \to A$ для будь-якої формули A.*

Побудуємо виведення:

1) $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$ (підстановка в схему аксіом A2 формули $A \to A$ замість B і формули A замість C);

2) $A \to ((A \to A) \to A)$ (підстановка в схему аксіом A1 формули $A \to \to A$ замість B);

3) $(A \to (A \to A)) \to (A \to A)$ — з (1) і (2) за m. p.;

4) $A \to (A \to A)$ — (підстановка в схему аксіом A1 формули A замість B);

5) $A \to A$ — з (3) та (4) за m. p.

Твердження 6.2. *Якщо $\Gamma \vdash A$ і B — будь-яка формула, то $\Gamma \vdash B \to A$.*

Нехай $A_1, ..., A_n$ — виведення A з Γ, де A_n збігається з A. Тоді послідовність $A_1, ..., A_n, A \to (B \to A), B \to A$ є виведенням $B \to A$ з Γ.

6.1.4. Основні теореми числення висловлень

У математичних міркуваннях часто якесь твердження B доводять у припущенні правильності якогось іншого твердження A, після чого встановлюють, що правильним є твердження «якщо A, то B». У численні висловлень цей метод обґрунтовується такою теоремою.

Теорема 6.1 (теорема дедукції). *Нехай Γ — множина формул, A і B — формули й Γ, $A \vdash B$. Тоді $\Gamma \vdash A \to B$.*

Доведення. Нехай

$$B_1, ..., B_n \qquad (6.1)$$

є виведенням B з Γ та A. Доведення проведемо методом індукції за n-довжиною виведення (6.1). При $n = 1$ формула B збігається з B_1. Згідно з означенням виведення можливими є три випадки:

- B_1 — аксіома;
- B_1 — формула з множини Γ;
- B_1 збігається з A.

У перших двох випадках маємо $\Gamma \vdash B_1$. Тоді згідно з твердженням 6.2 $\Gamma \vdash A \to B_1$, тобто $\Gamma \vdash A \to B$. У третьому випадку формула $A \to B$ має вигляд $A \to A$. Згідно з твердженням 6.1 $\vdash A \to A$, звідки $\Gamma \vdash A \to A$.

Припустимо тепер, що коли довжина виведення B з Γ, A менша від n, твердження теореми є правильним. Доведемо його для випадку, коли довжина виведення (6.1) дорівнює n. При цьому можливо, що:

- B_n — аксіома;
- B_n — формула з Γ;

- B_n — збігається з A;
- B_n — утворено за m. p. з B_i, B_j, де $i < n, j < n$.

У перших трьох випадках доведення проводиться так само, як при $n = 1$. У четвертому випадку або формула B_j має вигляд $B_i \to B_n$, або формула B_i — вигляд $B_j \to B_n$. Обмежимося розглядом випадку, коли B_j має вигляд $B_i \to B_n$. Відкидаючи останні $n—i$ формули з (6.1), дістаємо виведення B_i з Γ і A, а відкидаючи останні $n—j$ формули з (6.1) — виведення B_j з Γ та A; довжини цих ланцюжків виведення менша від n. За індуктивним припущенням:

1) $\Gamma \vdash A \to B_i$;
2) $\Gamma \vdash A \to (B_i \to B_n)$.

За схемою аксіом А2 маємо

3) $\Gamma \vdash (A \to (B_i \to B_n)) \to ((A \to B_i) \to (A \to B_n))$.

Застосовуючи властивість VI до (2) і (3), знаходимо:

4) $\Gamma \vdash (A \to B_i) \to (A \to B_n)$.

Застосовуючи властивість VI до (1) і (4), маємо

$\Gamma \vdash A \to B_n$.

Теорему доведено.

Наслідок 6.1 (правило силогізму): $A \to B, B \to C \vdash A \to C$.

Побудуємо виведення

1) $A \to B$ — гіпотеза;
2) $B \to C$ — гіпотеза;
3) A — гіпотеза;
4) B — застосовуємо правило m. p. до (1) і (3);
5) C — застосовуємо правило m. p. до (2) та (4).

Тоді $A \to B, B \to C, A \vdash C$. За теоремою про дедукцію $A \to B, B \to C \vdash \mathrm{A} \to \mathrm{C}$.

Наслідок 6.2: $A \to (B \to C), B \vdash A \to C$.

Після двократного застосування правила m. p. дістаємо $A, A \to (B \to C), B \vdash C$. Звідси за теоремою про дедукцію $A \to (B \to C), B \vdash A \to C$.

Для будь-яких формул A і B у численні висловлень правильними є такі твердження (наведемо їх без доведень).

Твердження 6.3. $\vdash \overline{\overline{A}} \to A$.

Твердження 6.4. $\vdash A \to \overline{\overline{A}}$.

Твердження 6.5. $\vdash \overline{A} \to (A \to B)$.

Твердження 6.6. $\vdash \left(\overline{B} \to \overline{A}\right) \to (A \to B)$.

Твердження 6.7. $\vdash (A \to B) \to (\overline{B} \to \overline{A})$.

Твердження 6.8. $\vdash A \to (\overline{B} \to \overline{(A \to B)})$.

Твердження 6.9. $\vdash (A \to B) \to \left(\left(\overline{A} \to B\right) \to B\right)$.

Наслідок 6.3. *Якщо* $\Gamma, A \vdash B$ *та* $\Gamma \vdash \overline{A} \to B$, *то* $\Gamma \vdash B$.

За теоремою про дедукцію $\Gamma \vdash A \to B$ і $\Gamma \vdash \overline{A} \to B$.Звідси внаслідок твердження 6.9 $\Gamma \vdash B$.

6.1.5. Повнота і несуперечність числення висловлень. Незалежність аксіом

Ми формалізували логіку висловлень і побудували числення висловлень як аксіоматичну теорію. Покажемо, що множина теорем числення висловлень збігається з множиною тотожно-істинних формул логіки висловлень. Доведемо спочатку таке твердження.

Теорема 6.2. *Усяка формула числення висловлень, що виводиться з порожньої системи гіпотез, є тотожно-істинною.*

Безпосередньою перевіркою пересвідчуємося в тому, що аксіоми числення висловлень — тотожно-істинні формули. Внаслідок властивостей імплікації формула, яка випливає з тотожно-істинних формул за правилом m. p., є тотожно-істинною. Отже, будь-яка формула, що виводиться, — тотожно-істинна. Називатимемо формулу, яка виводиться, вивідною.

Доведемо тепер обернену теорему 6.2 про те, що будь-яка тотожно-істинна формула виводиться в численні висловлень, тобто є теоремою.

Введемо символи X_i^{ε} й A^{ε}, які мають значення булевих змінних:

$$X_i^{\varepsilon} = \begin{cases} X_i, \text{ якщо } \varepsilon = 1; \\ \bar{X}_i, \text{ якщо } \varepsilon = 0; \end{cases} \quad A^{\varepsilon} = \begin{cases} A, \text{ якщо } \varepsilon = 1; \\ \bar{A}, \text{ якщо } \varepsilon = 0. \end{cases}$$

Для позначення поняття «істинність» застосуємо символ 1, а поняття «хибність» — символ 0.

Нехай $X_1, ..., X_n$ — усі змінні формули A, і задано деяку оцінку списку змінних $< \varepsilon_1, ..., \varepsilon_n >$, що складається з нулів й одиниць. Через $A(\varepsilon_1, ..., \varepsilon_n)$ будемо позначати значення формули A на цій оцінці.

Приклад. Нехай формула A має вигляд $(X_1 \to X_2) \to \bar{X}_1$, де X_1 та X_2 — змінні. Тоді X_1^0 — це $\bar{X}_1$, X_2^1 — це X_2, формула A^1 — це A, формула A^0 — це $\overline{((X_1 \to X_2) \to \bar{X}_1)}$.

Розглянуті чотири можливі оцінки відповідають різним розподілам істинносних значень змінних:

$$<0,0>, <0, 1>, <1, 0>, <1, 1>.$$

Тоді A (0, 0) = 1, A (0, 1) = 1, A (1, 0) = 1, A (1, 1) = 0.

Лема 6.1. *Нехай A — довільна формула, X_l, ..., X_n — усі змінні, що входять в A (для зручності ми позначили їх так, ніби вони були першими змінними), і нехай також задано довільну оцінку списку змінних $<\varepsilon_1, ..., \varepsilon_n>$. Тоді $X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash A^{\varepsilon}$, де A^{ε} — значення формули A на оцінці $<\varepsilon_1, ..., \varepsilon_n>$.*

Приклад. Нехай A — формула, розглянута у попередньому прикладі. Тоді з леми 6.1 випливають такі вивідності: $X_1, \bar{X}_2 \vdash (X_1 \to X_2) \to \bar{X}_1$; $X_1, X_2 \vdash \left(\overline{(X_1 \to X_2) \to \bar{X}_1}\right)$.

Доведемо лему 6.1. Під довжиною формули A будемо розуміти число входжень логічних символів в A. Доведення проведемо методом індукції за k — довжиною формули A. При $k = 0$ формула A стає змінною X_i і твердження леми зводиться до $X_i^{\varepsilon} \vdash X_i^{\varepsilon}$.

Нехай для формул, довжина яких менша від k, твердження леми справджується. Доведемо його для формул завдовжки k. Можливими є два випадки.

Випадок 1. Формула A має вигляд $\overline{B}$. Довжина формули B дорівнює k–1, множина змінних формул B збігається із множиною змінних формули A. Нехай $B(\varepsilon_1, ..., \varepsilon_n) = \varepsilon'$. Тоді $\varepsilon = \overline{\varepsilon'}$. Якщо $\varepsilon' = 0$, то $\varepsilon = 1$. За індуктивним припущенням:

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash B^0 .$$

Однак B^0 — це $\overline{B}$ або в цьому випадку A^1. Отже,

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash A^{\varepsilon} .$$

Якщо $\varepsilon' = 1$, то $\varepsilon = 0$. За індуктивним припущенням

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash B^1 .$$

Згідно з твердженням 6.4 і правилом m. p.

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash \overline{\overline{B}} .$$

Однак $\overline{\overline{B}}$ — це A^0. Отже,

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash A^{\varepsilon} .$$

Випадок 2. Формула A має вигляд $B \to C$. Довжина формул B і C менша від k. Нехай значення формули B на оцінці $<\varepsilon_1, ..., \varepsilon_n>$ дорівнює ε', а значення формули C на оцінці $<\varepsilon_1, ..., \varepsilon_2>$ становить ε''. Тоді $\varepsilon = \varepsilon' \to \varepsilon''$.

Якщо $\varepsilon' = 0$, то $\varepsilon = 1$. За індуктивним припущенням

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash B^{\varepsilon'} ,$$

тобто

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash \overline{B} .$$

Згідно з твердженням 6.5 $\vdash \overline{B} \to (B \to C)$. Застосовуючи правило m. p., знаходимо

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash B \to C .$$

Отже,

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash A^{\varepsilon} .$$

Нехай тепер $\varepsilon' = 1$, $\varepsilon'' = 0$. Тоді $\varepsilon = 0$ й A^{ε} — це $\overline{(B \to C)}$. За індуктивним припущенням

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash B , \; X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash \overline{C} .$$

Згідно з твердженням 6.8 $\vdash B \to \left(\overline{C} \to \overline{(B \to C)}\right)$. Після двократного застосування правила m. p. дістаємо

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash \left(\overline{B \to C}\right) .$$

Якщо $\varepsilon' = 1$, $\varepsilon'' = 1$, то $\varepsilon = 1$, а A^{ε} — це $B \to C$. За індуктивним припущенням

$$X_1^{\varepsilon_1}, ..., X_n^{\varepsilon_n} \vdash C .$$

За схемою аксіом А1 маємо $\vdash C \to (B \to C)$. Застосовуючи правило m.p., знаходимо

$$X_1^{\varepsilon_1}, \dots, X_n^{\varepsilon_n} \vdash B \to C .$$

Теорема 6.3. *Якщо формула A числення висловлень є тотожно-істинною, то вона — вивідна.*

Нехай A — тотожно-істинна формула, а $X_1, \dots, X_n$ — усі її змінні. Тоді на підставі леми 6.1 на будь-якій оцінці $<\varepsilon_1, \dots, \varepsilon_n>$, що складається з нулів й одиниць, матимемо

$$X_1^{\varepsilon_1}, \dots, X_n^{\varepsilon_n} \vdash A .$$

Тому у випадку, коли $\varepsilon_n = 1$, дістаємо

$$X_1^{\varepsilon_1}, \dots, X_{n-1}^{\varepsilon_{n-1}}, X_n \vdash A ,$$

а у випадку, коли $\varepsilon = 0$, знаходимо

$$X_1^{\varepsilon_1}, \dots, X_{n-1}^{\varepsilon_{n-1}}, \overline{X}_n \vdash A .$$

Застосовуючи наслідок твердження 6.9, одержуємо

$$X_1^{\varepsilon_1}, \dots, X_{n-1}^{\varepsilon_{n-1}} \vdash A .$$

Так само, розглядаючи випадки, коли ε_{n-1} набуває значень 1 та 0, знаходимо

$$X_1^{\varepsilon_1}, \dots, X_{n-2}^{\varepsilon_{n-2}} \vdash A$$

і т. д. Нарешті, приходимо до $\vdash$ A.

Із теорем 6.2 та 6.3 безпосередньо випливає таке твердження.

Теорема 6.4. *Формула A числення висловлень є вивідною тоді й тільки тоді, коли вона — тотожно-істинна.*

Властивість аксіоматичної теорії, яка полягає в тому, що коли формула A виражає логічний закон (як, наприклад, тотожно-істинна формула), то вона є вивідною в цій теорії і називається повнотою аксіоматичної теорії (або повнотою в широкому сенсі). З теореми 6.3 випливає, що числення висловлень є повною аксіоматичною теорією.

Формальну аксіоматичну теорію називають несуперечливою, коли не існує формули A такої, що одночасно виводяться формули A й $\overline{A}$.

Теорема 6.5. *Числення висловлень є несуперечливим.*

Справді, згідно з теоремою 6.2 кожна формула, що виводиться, — тотожно-істинна. Заперечення цієї формули не є тотожно-істинною формулою. Отже, для жодної формули A неможливо, щоб одночасно $\vdash \overline{A}$ й $\vdash A$.

Нарівні з повнотою аксіоматичної теорії в широкому сенсі розглядають її повноту у вузькому сенсі. Формальну аксіоматичну теорію називають повною у вузькому сенсі, якщо додання будь-якої формули, яка не виводиться, як схеми аксіом приводить до суперечливої теорії.

Теорема 6.6. *Числення висловлень є повним у вузькому сенсі.*

Нехай F — довільна формула , що не виводиться (згідно з теоремою 6.5 як F можна взяти будь-яку не тотожно-істинну формулу); $X_1, \ldots, X_n$ — список її змінних, а $<\varepsilon_1, \ldots, \varepsilon_n>$ — така оцінка списку змінних, на якій формула F набуває хибного значення (тобто 0). Нехай також

$$B_i = \begin{cases} A \to A, & \text{якщо} \quad \varepsilon_i = 1; \\ \overline{(A \to A)}, & \text{якщо} \quad \varepsilon_i = 0, \end{cases}$$

де A — довільна формула. Тоді формула $F(B_1, \ldots, B_n)$ буде тотожно-хибною.

Розглянемо числення, в якому до схем аксіом А1 — А3 як ще одну схему аксіом додано формулу А4, тобто $F(B_1, \ldots, B_n)$.

Вивідність у цьому численні позначимо символом $\underset{F}{\vdash}$. Оскільки $F(B_1, \ldots, B_n)$ утворено за схемою А4, маємо $\underset{F}{\vdash} F(B_1, \ldots, B_n)$. Нехай C — довільна формула. Тоді $F(B_1, \ldots, B_n) \to C$ — тотожно-істинна формула на підставі властивостей імплікацій. Отже, $\vdash F(B_1, \ldots, B_n) \to C$, а значить, $\underset{F}{\vdash} (B_1, \ldots, B_n) \to C$. Застосувавши правило m. p., матимемо $\underset{F}{\vdash} C$. Якщо замість формули C розглянути формулу $\overline{C}$, то також дістанемо, що $\underset{F}{\vdash} \overline{C}$, тобто розширена теорія виявилася суперечливою.

У кожній формальній теорії виникає питання про незалежність її аксіом, тобто питання про те, чи можна яку-небудь аксіому вивести з інших, застосовуючи правила виведення цієї теорії.

Виявляється, що система аксіом А1—А3 числення висловлень є незалежною. Встановимо незалежність аксіоми А3 від інших. Будемо вважати, що змінні набувають значень із множини $\{a, b\}$. Операції і $\rightarrow$ задамо в табл. 6.1 і 6.2 відповідно.

Легко перевірити, що аксіоми А1 й А2 при такій інтерпретації матимуть значення a. З означення $\rightarrow$ випливає, що застосування правила виведення m. p. до формул, тотожних a, дає формулу, також тотожну a. Отже, формули, які виводяться з аксіом А1 та А2 за правилом m. p., є тотожними a.

З іншого боку, аксіома А3

$$(\overline{B} \to \overline{A}) \to ((\overline{B} \to A) \to B)$$

не є тотожною a, оскільки, наприклад, при $A = a$, $B = b$ вона набуває значення b. Можна встановити незалежність кожної з аксіом А1 й А2 від інших (при цьому зручно використати тризначні логіки).

Слід зазначити, що числення висловлень можна описати системами аксіом, відмінними від А1—А3. Крім того, замість символів $^-$ і $\rightarrow$ можна використати інші набори символів, аби з їх допомогою можна було виразити всі основні логічні операції.

Таблиця 6.1

X	$\overline{X}$
A	$\overline{a}$
B	$\overline{b}$

Таблиця 6.2

X	Y	$X \to Y$
a	A	A
a	B	B
b	A	A
b	B	A

6.2. Логіка і числення предикатів

Якщо числення висловлень дає змогу доводити теореми для внутрішніх потреб логіки, то числення предикатів забезпечує можливість описувати й доводити теореми для конкретних розділів математики. Логіка предикатів дає змогу формулювати співвідношення між елементами реального світу і виводити подібні відношення або теореми в математиці. Числення висловлень — досить вузька логічна система. Існують, наприклад, такі типи логічних міркувань, які не можуть бути здійснені в межах логіки висловлень:

- Кожний друг Івана є другом Петра. Сидір не є другом Івана. Отже, Сидір не є другом Петра.
- Просте число два — парне. Отже, існують прості парні числа.

Коректність цих висновків ґрунтується на внутрішній структурі самих речень і значенні слів «кожний» та «існують».

6.2.1. Предикати, квантори. Формули логіки предикатів

Розглянемо речення, що залежать від параметрів, наприклад «x — парне число», «x менше y», «$x + y = z$», «x — батько y», «x та y — брати» тощо. Якщо x, y, z у перших трьох реченнях замінити деякими числами, то матимемо певні висловлення, які можуть бути істинними або хибними. Наприклад, «3 — парне число», «2 менше 5», «3 + 2 = 7». Останні два речення виражають родинні відносини між членами сім'ї і перетворюються на певні висловлення, істинні або хибні, при заміні x й y іменами членів сім'ї: «Іван —батько Петра», «Іван і Олег брати».

Речення такого типу називаються предикатами. Точніше предикатом $P(x_1, ..., x_n)$ називається функція, змінні якої набувають значень із деякої множини M, а сама вона набуває двох значень: 1 (істинне) і 0 (хибне), тобто $P(x_1, ..., x_n)$: $M^n \to \{1, 0\}$.

Предикат n аргументів називають n-місним. Множина M значень змінних визначається математичним контекстом. Наприклад, основне співвідношення елементарної геометрії «будь-які дві точки x, y лежать на одній прямій» можна виразити предикатом змінних x, y; а «будь-які три точки лежать в одній площині» — предикатом трьох змінних x, y, z.

Предикати позначають великими літерами латинського алфавіту. Іноді буває зручно вказувати число змінних предикатів. У таких випадках символи предикатів доповнюють верхнім індексом, який вказує число аргументів, наприклад $P^{(n)}(x_1, ..., x_n)$ — n-місний предикат. Висловлення вважається нуль-місним предикатом.

Над предикатами можна виконувати звичайні логічні операції. У підсумку утворюються нові предикати.

Приклади: 1. Нехай $P^{(1)}(x)$ позначає предикат «x ділиться на два», $Q^{(1)}(x)$ — предикат «x ділиться на три». Тоді вираз $P^{(1)}(x)$ & $Q^{(1)}(x)$ позначає предикат «x ділиться на два та x ділиться на три», тобто позначає предикат ділення на 6.

2. Нехай $S^{(2)}(x, y)$ позначає предикат «$x = y$». Він набуває значення 1 тоді й тільки тоді, коли $x = y$. У цьому випадку вираз $\overline{S}^{(2)}(x, x) \to S^{(2)}(x, y)$ позначає предикат, що набуває значення 1 при будь-яких x та y.

Крім операцій логіки висловлень будемо застосовувати ще операції зв'язування квантором.

Квантор загальності. Нехай $P(x)$ — деякий предикат, який набуває значення 1 або 0 для кожного елемента x множини M. Тоді під виразом $(\forall x)P(x)$ матимемо на увазі істинне висловлення, коли $P(x)$ — істинне для кожного елемента x із множини M, і хибне — в іншому випадку. Читається цей вираз так: «для всіх x $P(x)$» або «для будь-якого x $P(x)$». Це висловлення вже не залежить від х. Символ $\forall$ називається квантором загальності.

Квантор існування. Нехай $P(x)$ — деякий предикат. Під виразом $(\exists x)P(x)$ будемо розуміти істинне висловлення, коли існує елемент множини M, для якого $P(x)$ — істинне, та хибне — в іншому випадку. Читається цей вираз так: «існує x таке, що $P(x)$» або «існує x, для якого $P(x)$».Символ $\exists$ називається квантором існування. Операцію зв'язування квантором можна застосовувати також до предикатів більшого числа змінних (детальніше про це йтиметься далі).

Приклад. Для предикатів, наведених у попередньому прикладі, маємо $(\exists x)\left(P^{(1)}(x) \& Q^{(1)}(x)\right)$ — істинне висловлення, а $(\forall x)\left(P^{(1)}(x) \& Q^{(1)}(x)\right)$ — хибне.

На мові предикатів можна скласти набагато складніші речення, ніж на мові логіки висловлень. Уведемо поняття формули логіки предикатів. Алфавіт цієї логіки містить такі символи:

- предметних змінних $x_1, x_2, \ldots, x_n \ldots$;
- предикатів $A_1^{(t)}$, $A_2^{(t)}$, ..., $A_k^{(t)}$, ..., де $t = 0, 1{,}2 \ldots$;
- логічні $\rceil, \&, \vee, \to, \sim$;
- кванторів $\exists, \forall$;
- дужки і кому) , (.

Щоб зменшити кількість індексів, символи предметних змінних будемо позначати через x, y, z, а символи предикатів — через P, S, Q, R тощо.

Означення 6.9. *Слово в алфавіті логіки предикатів називається формулою, якщо воно задовольняє таке індуктивне означення (одночасно визначаються поняття вільної та зв'язаної змінних у формулі):*

1. *Якщо* $A_i^{(t)}$ *— символ предиката,* $x_{i_1}, x_{i_2}, \ldots, x_{i_t}$ *— символи предметних змінних, не обов'язково різних, то* $A_i^{(t)}(x_{i_1}, x_{i_2}, \ldots, x_{i_t})$ *— формула, яка називається атомарною. Всі предметні змінні атомарних формул є вільними, зв'язаних змінних немає.*

2. *Нехай* A *— формула. Тоді* $\overline{A}$ *— також формула. Вільні й зв'язані змінні формули* $\overline{A}$ *— це відповідно вільні та зв'язані змінні формули* A.

3. *Нехай A і B — формули, причому немає таких предметних змінних, які були б зв'язаними в одній формулі, але вільними в іншій. Тоді*

$$(A \vee B), (A \,\&\, B), (A \to B), (A \sim B)$$

є формулами, в яких вільні змінні формул A та B залишаються вільними, а зв'язані змінні формул A і B — зв'язаними.

4. *Нехай A — формула, яка містить вільну змінну x. Тоді*

$$(\forall x) A, (\exists x) A \tag{6.2}$$

також є формулами. Змінна x у них — зв'язана. Інші ж змінні, які у формулі A є вільними, залишаються вільними й у формулах (6.2). Змінні, які у формулі A є зв'язаними, залишаються зв'язаними й у формулах (6.2). У першій із формул (6.2) формула A називається областю дії квантора $\forall$*, а в другій — областю дії квантора* $\exists$.

5. *Слово в алфавіті логіки предикатів є формулою тільки в тому випадку, якщо це випливає з правил* 1—4. *Зазначимо, що за означенням формули жодна змінна не може бути одночасно вільною та зв'язаною.*

Приклади: 1. Такі вирази є формулами логіки предикатів: $A_5^{(3)}(x_1, x_5, x_7)$ — атомарна формула, в якій x_1, x_5, x_7 — вільні змінні; $(\forall x_1)(\exists x_2) A_1^{(3)}(x_1, x_2, x_3) \to (\forall x_1) A_1^{(2)}(x_1, x_4)$ — формула, в якій x_1, x_2 — зв'язані, а x_3, x_4 — вільні змінні.

2. Вираз $(\exists x_2)(\forall x_2) A_1^{(2)}(x_1, x_3) \,\&\, A_2^{(2)}(x_1, x_3)$ не є формулою.

Значення формули визначено лише тоді, коли задано яку-небудь інтерпретацію символів, що входять до неї.

Означення 6.10. *Під інтерпретацією розуміють систему* $M_f = \langle M, f \rangle$, *яка складається з непорожньої множини M і відповідності f, що зіставляє з кожним предикатним символом* $A_j^{(t)}$ *певний t-місний предикат (будемо позначати предикати, поставлені у відповідність предикатним символам, тими самими символами).*

При заданій інтерпретації вважають, що предметні змінні пробігають множину M, а символи $^-$, $\vee$, &, $\to$, $\sim$, і символи кванторів мають своє звичайне значення. Для заданої інтерпретації кожна формула без вільних змінних є висловленням, яке істинне чи хибне, а кожна формула з вільними змінними виражає деякий предикат на множині M, який істинний при одних значеннях змінних з цієї множини та хибний при інших.

Визначимо значення формули в цій інтерпретації за індуктивними кроками означення формули. Значення формули F на наборі $\langle a_1, ..., a_n \rangle, a_i \in M$, своїх вільних змінних $x_{i_1}, ..., x_{i_n}$ позначимо символом $F|_{\langle a_1, ..., a_n \rangle}$.

1. Формула F — це атомарна формула $A_j^{(t)}(x_{i_1}, ..., x_{i_t})$. Нехай $x_{j_1}, ..., x_{j_s}$ — різні вільні змінні цієї формули, виписані в певному порядку. Значенням формули F у наборі $\langle a_1, ..., a_s \rangle, a_j \in M$, називається значення t-місного предиката, зіставленого символу $A_j^{(t)}$ при відповідному заміщенні його змінних елементами $a_1, \ldots, a_s$.

2. Формула F має вигляд $\overline{A}$. Нехай значення формули A на наборі $\langle a_1,\dots,a_n\rangle$, $a_i\in M$, дорівнює ε . Тоді $F|_{\langle a_1,\dots,a_n\rangle}=\overline{\varepsilon}$.

3. Формула F має вигляд $A\vee B$, $A\,\&\,B$, $A\to B$ або $A\sim B$. Значення формули F на наборі $\langle a_1,\dots,a_n\rangle$ значень своїх вільних змінних дорівнює відповідно $\varepsilon_1\vee\varepsilon_2$, $\varepsilon_1\,\&\,\varepsilon_2$, $\varepsilon_1\to\varepsilon_2$, $\varepsilon_1\sim\varepsilon_2$, де ε_1 — значення формули A, а ε_2 — значення формули B у цьому наборі.

4. Формула F має вигляд $(\forall x)A$. Якщо $x_{i_1},\dots,x_{i_n}$ — сукупність усіх вільних змінних формули F, то $x_{i_1},\dots,x_{i_n}$ — вільні змінні формули A. Значення $(\forall x)\mathrm{A}|_{\langle a_1,\dots,a_n\rangle}=1$ тоді й тільки тоді, коли для будь-якого $a\in M$ маємо $A|_{\langle a,a_1,\dots,a_n\rangle}=1$.

5. Формула F має вигляд $(\exists x)A$. Якщо $x_{i_1},\dots,x_{i_n}$ — сукупність усіх вільних змінних формули F, то $x_{i_1},\dots,x_{i_n}$ — вільні змінні формули A. Значення $(\exists x)A\,|_{\langle a,a_1,\dots,a_n\rangle}=1$ тоді й тільки тоді, коли для деякого $a\in M$ матимемо $A|_{\langle a,a_1,\dots,a_n\rangle}=1$.

Приклади: **1.** Розглянемо три формули:

$$A_1^{(2)}(x_1,x_2)\,;$$

$$(\forall x_2)A_1^{(2)}(x_1,x_2)\,;$$

$$(\exists x_1)(\forall x_2)A_1^{(2)}(x_1,x_2)\,.$$

Візьмемо як область інтерпретації множину цілих додатних чисел й інтерпретуємо $A_1^{(2)}(x,y)$ як $x\le y$. Тоді перша формула — це предикат $x_1\le x_2$, який набуває істинного значення для всіх пар (a, b) цілих додатних чисел таких, що $a\le b$. Друга формула виражає властивість «для кожного цілого додатного числа y матимемо $x\le y$ », яка виконується тільки при $x = 1$. Нарешті, третя формула — це істинне висловлення про існування найменшого цілого додатного числа. Якби як область інтерпретації ми розглядали множину цілих чисел, то третя формула була б хибним висловленням.

2. Нехай $M_f=\langle N,f\rangle$, де N — множина натуральних чисел; f — відповідність, що зіставляє із предикатними символами $S^{(3)}(x, y, z)$, $P^{(3)}(x, y, z)$ предикати $S^{(3)}(x, y, z)$: $: x + y = z$; $P^{(3)}(x, y, z)$: $xy = z$.

Запишемо формули, істинні в M тоді й тільки тоді, коли виконано такі умови:

- $x = 0$;
- $x = 1$;
- x — парне число;
- x — просте число;
- $x = y$;
- $x\le y$;
- x ділить y;
- комутативність додавання.

Це відповідно формули:

- $F_1(x) = (\forall y)\, S^{(3)}(x, y, y)$, оскільки $x + y = y$ для будь-якого y тоді й тільки тоді, коли $x = 0$;
- $F_2(x) = (\forall y)\, P^{(3)}(x, y, y)$;
- $F_3(x) = (\exists y)\, S^{(3)}(y, y, x)$;
- $F_4(x) = \overline{F_1}(x)\ \&\ (\forall y)(\forall z)(P^{(3)}(y, z, x)\to(F_2(y)\vee F_2(z)))$, де F_1, F_2 —формули, визначені вище;

- $F_5(x) = (\forall z)(\forall u)(S^{(3)}(x, z, u) \to S^{(3)}(y, z, u))$;
- $F_6(x, y) = (\exists z)\, S^{(3)}(x, z, y)$;
- $F_7(x, y) = (\exists z)\, P^{(3)}(x, z, y)$;
- $(\forall x)(\forall y)(\forall z)(S^{(3)}(x, y, z) \to S^{(3)}(y, x, z))$.

3. Нехай $f(x)$ — довільна фіксована функція, задана на відрізку $[a, b]$.

Розглянемо інтерпретацію $M_f = \langle M, f_1 \rangle$, де M — множина дійсних чисел; f_1 — відповідність, що зіставляє із предикатними символами $P(x, \delta), Q(x, \varepsilon)$ та $R(\varepsilon)$ предикати $P(x, \delta): |x - x_0| < \delta, Q(x, \varepsilon): |f(x) - A| < \varepsilon; R(\varepsilon): \varepsilon > 0$. Тут x_0 — фіксований елемент відрізка $[a, b]$; A — деяке фіксоване дійсне число. Тоді твердження про те, що число A — межа функції $f(x)$ при $x \to x_0$, записується формулою

$$(\forall\varepsilon)(\exists\delta)(\forall x)((R(\varepsilon) \& P(x, \delta)) \to Q(x, \varepsilon)).$$

Розглянемо інтерпретацію $M_f = \langle M, f_2 \rangle$, де M — множина дійсних чисел; f_2 — відповідність, що зіставляє із предикатними символами $P(x, \delta), R(\varepsilon), S(x, \varepsilon)$ предикати $P(x, \delta): |x - x_0| < \delta, R(\varepsilon): \varepsilon > 0, S(x, \varepsilon): |f(x) - f(x_0)|$. Тут x_0 — довільний фіксований елемент відрізка $[a, b]$. Тоді твердження про те, що функція $f(x)$ є неперервною в точці x_0, записується формулою

$$(\forall\varepsilon)(\exists\delta)(\forall x)((R(\varepsilon) \& P(x, \delta)) \to S(x, \varepsilon)).$$

Розглянемо інтерпретацію $M_f = \langle M, f_3 \rangle$, де M — множина дійсних чисел; f_3 — відповідність, що зіставляє із предикатними символами $P_1(x, x_1, \delta), R(\varepsilon), S_1(x, x_1, \delta)$, $D(x)$ предикати $P_1(x, x_1, \delta): |x - x_1| < \delta, R(\varepsilon): \varepsilon > 0, S_1(x, x_1, \varepsilon): |f(x) - f(x_1)| < \varepsilon; D(x): x \in$ $\in [a; b]$. Тоді твердження про те, що функція $f(x)$ є небезперервною на відрізку $[a, b]$, записується формулою:

$$(\forall x_1)(\forall\varepsilon)(\exists\delta)(\forall x)((D(x_1) \& R(\varepsilon) \& P_1(x, x_1, \delta)) \to S_1(x, x_1, \varepsilon)).$$

6.2.2. Рівносильність формул

Нехай формули F і G мають одну й ту саму множину вільних змінних (зокрема, порожню).

Означення 6.11. *Формули F та G є рівносильними в заданій інтерпретації, якщо на будь-якому наборі значень вільних змінних вони набувають однакових значень (тобто якщо формули виражають у цій інтерпретації один і той самий предикат). Формули F і G є рівносильними на множині M, якщо вони рівносильні у всіх інтерпретаціях, заданих на множині M. Формули F та G є рівносильними (в логіці предикатів), якщо вони рівносильні на всіх множинах (тоді писатимемо $F \equiv G$).*

Приклад. На множині $M = \{a, b\}$ задамо предикати $P_1(x, y)$ та $P_2(x, y)$ (табл. 6.3 і 6.4). Розглянемо дві формули:

$$A_1^{(2)}(x_1, x_2) \& A_1^{(2)}(x_1, x_3); \tag{6.3}$$

$$A_1^{(2)}(x_1, x_2) \& A_1^{(2)}(x_2, x_3). \tag{6.4}$$

Якщо областю інтерпретації слугує множина M і формула $A_1^{(2)}$ інтерпретується як предикат P_1, то формули (6.3) і (6.4) є рівносильними в цій інтерпретації, оскільки набувають значення 1 тільки на двох наборах вільних змінних: $\langle a, a, a \rangle$ та $\langle b, b, b \rangle$.

Таблиця 6.3

x	y	$P_1(x, y)$
a	A	1
a	B	0
b	A	0
b	B	1

Таблиця 6.4

x	y	$P_2(x, y)$
a	A	1
a	B	1
b	A	0
b	B	0

Якщо областю інтерпретації є множина M, але формула $A_1^{(2)}$ інтерпретується як предикат P_2, то формули (6.3) і (6.4) — нерівносильні, оскільки в наборі $\langle a, b, b\rangle$ формула (6.3) набуває значення 1, а формула (6.4) — значення 0.

Формули $(\forall x_1)A_1^{(1)}(x_1)$ й $(\exists x_1)A_1^{(1)}(x_1)$ є рівносильними на одноелементній множині. Справді, якщо область інтерпретації — одноелементна множина, то який би предикат не взяти як інтерпретацію $A_1^{(1)}$ на цій множині, він набуває тільки одного значення: 1 або 0. У першому випадку обидві формули набувають значення 1, у другому — 0, і, отже, вони є рівносильними на цій множині. З іншого боку, на двоелементній множині $\{a, b\}$ ці формули — нерівносильні. Досить як інтерпретацію $A_1^{(1)}$ розглянути предикат P такий, що $P(a) = 1$, $P(b) = 0$. Наведемо кілька правил переходу від одних формул до інших, рівносильних їм (у всіх інтерпретаціях).

Очевидно, для формул логіки предикатів зберігаються всі рівносильності та правила рівносильних перетворень логіки висловлень. Крім того, можна довести такі правила.

1. Перенесення квантора через заперечення. Нехай A — формула, що містить вільну змінну x. Тоді справджуються рівносильності:

$$\left(\overline{\forall x}\right)A(x) \equiv (\exists x)\overline{A}(x); \tag{6.5}$$

$$\left(\overline{\exists x}\right)A(x) \equiv (\forall x)\overline{A}(x). \tag{6.6}$$

Доведемо спочатку рівносильність (6.5). Нехай $x_1, ..., x_n$ — множина (може бути порожньою) всіх вільних змінних формули A, відмінних від x. Нехай $M_f = \langle M, f\rangle$ — довільна інтерпретація. Доведемо, що на будь-якому наборі значень своїх вільних змінних $\langle a_1, ..., a_n\rangle, a_i \in M$, формули $\left(\overline{\forall x}\right)A(x)$ й $(\exists x)\overline{A}(x)$ набувають однакових значень істинності.

Можливими є два випадки:

- для будь-якого елемента $a \in M$ $A(x)|_{\langle a, a_1, ..., a_n\rangle} = 1$;
- для деякого елемента $a_0 \in M$ $A(x)|_{\langle a_0, a_1, ..., a_n\rangle} = 0$.

У першому випадку для будь-якого елемента $a \in M$ маємо $\overline{A}(x)|_{\langle a, a_1, ..., a_n\rangle} = 0$. Звідси за означенням $(\exists x)\overline{A}(x)|_{\langle a_1, ..., a_n\rangle} = 0$. З іншого боку, в цьому випадку $(\forall x)A(x)|_{\langle a_1, ..., a_n\rangle} = 1$. Звідси $\left(\overline{\forall x}\right)A(x)|_{\langle a_1, ..., a_n\rangle} = 0$.

У другому випадку для елемента $a_0 \in M$ маємо $\overline{A}(x)|_{\langle a_0, a_1, ..., a_n\rangle} = 1$. Звідси $(\exists x)\overline{A}(x)|_{\langle a_1, ..., a_n\rangle} = 1$. З іншого боку, в цьому випадку

$(\forall x)\ A(x)|_{\langle a_1, ..., a_n\rangle} = 0$. Звідси $(\overline{\forall x})A(x)|_{\langle a_1, ..., a_n\rangle} = 1$. Рівносильність (6.5) доведено.

Доведемо тепер рівносильність (6.6). Застосуємо рівносильність (4.5) до формули $\overline{A}^x$. Тоді $(\overline{\forall x})\overline{A}(x) \equiv (\exists x)\overline{\overline{A}}(x) \equiv (\exists x)A(x)$. Застосувавши рівносильність 11 основних рівносильностей логіки висловлення, дістанемо: $(\overline{\exists x})A(x) \equiv (\overline{\overline{\forall x}})\overline{A}(x) \equiv (\forall x)\overline{A}(x)$.

2. Винесення квантора за дужки. Нехай формула A (x) містить вільну змінну x, формула B не містить змінної x й обидві вони задовольняють п. 3 означення формул. Тоді

$$(\exists x)(A(x) \& B) \equiv (\exists x)A(x) \& B;$$

$$(\forall x)(A(x) \& B) \equiv (\forall x)A(x) \& B;$$

$$(\exists x)(A(x) \vee B) \equiv (\exists x)A(x) \vee B;$$

$$(\forall x)(A(x) \vee B) \equiv (\forall x)A(x) \vee B.$$

Доведемо першу з цих рівносильностей (інші доводяться аналогічно).

Нехай $x_{i_1}, ..., x_{i_n}$ — усі вільні змінні формули $(\exists x)(A(x) \& B)$. Тоді вони ж будуть усіма вільними змінними формули $(\exists x)A(x) \& B$.

Розглянемо довільну інтерпретацію з множиною M. Нехай $\langle a_1, ..., a_n\rangle$, $a_i \in M$, — довільний набір значень вільних змінних $x_{i_1}, ..., x_{i_n}$. Оскільки формула B не містить змінної x, можна визначити значення цієї формули в наборі $\langle a_1, ..., a_n\rangle$ (точніше, в його частині, що стосується вільних змінних формули B). Якщо $B|_{\langle a_1, ..., a_n\rangle} = 0$, то $((\exists x)A(x) \& B)_{\langle a_1, ..., a_n\rangle} = 0$ і для будь-якого елемента a з множини M у наборі значень $\langle a, a_1, ..., a_n\rangle$ своїх вільних змінних $x, x_{i_1}, ..., x_{i_n}$ формула $A(x) \& B$ набуває значення 0. Звідси $(\exists x)A(x) \& B|_{\langle a_1, ..., a_n\rangle} = 0$. Якщо $B_{<a_1, ..., a_n>} = 1$, то для будь-якого елемента a з множини M у наборі $\langle a, a_1, ..., a_n\rangle$ формули $A(x) \& B$ й $A(x)$ набувають однакових істинностей. Звідси $((\exists x)A(x) \& B)|_{\langle a_1, ..., a_n\rangle} = (\exists x)A(x)|_{\langle a_1, ..., a_n\rangle} = (\exists x)A(x) \& B|_{\langle a_1, ..., a_n\rangle}$.

Зазначимо, що коли не вимагати, щоб формула B не містила змінної x, то будуть справджуватися тільки дві рівносильності:

$$(\forall x)(A(x)) \& B(x) \equiv (\forall x)A(x) \& (\forall x)B(x);$$

$$(\exists x)(A(x) \vee B(x)) \equiv (\exists x)A(x) \vee (\exists x)B(x).$$

3. Переставлення однойменних кванторів. Для кванторів $\forall$ та $\exists$ маємо

$$(\forall y)(\forall x)A(x, y) \equiv (\forall x)(\forall y)A(x, y);$$

$$(\exists y)(\exists x)A(x, y) \equiv (\exists x)(\exists y)A(x, y).$$

4. Перейменування зв'язаних змінних. Замінюючи зв'язану змінну формули A іншою змінною, що не входить у цю формулу, у кванторі й усюди в області його дії дістаємо формулу, рівносильну A.

Це твердження легко доводиться за довжиною формули.

Розглянемо спосіб спрощення формул, що спирається на зведені рівносильності. Під довжиною формули тут і далі розумітимемо загальне число символів предикатів, логічних символів та символів кванторів, які входять до неї. Так, формула $(\forall x_1)A_1^{(2)}(x_1, x_2)\&(\exists x_3)A_2^{(1)}(x_3)$ має довжину 5.

Формули, в яких із логічних символів застосовуються тільки символи $\&, \vee, ^{-}$, причому символ $^{-}$ зустрічається лише перед символами предикатів, будемо називати зведеними.

Приклад. Розглянемо такі формули:

1. $A_1^{(1)}(x_1) \vee A_2^{(2)}(x_1, x_2)$.
2. $(\forall x_1)A_1^{(1)}(x_1)\&(\exists x_2)\overline{A}_2^{(2)}(x_2, x_3)$.
3. $\left(\overline{A}_2^{(1)}(x_1) \vee A_1^{(1)}(x_2)\right)$.
4. $(\forall x)A_1^{(1)}(x_1) \to (\exists x_2)\overline{A}_1^{(1)}(x_2)$
5. $\left(\overline{\exists x_2}\right)A_1^{(1)}(x_2) \to A_2^{(1)}(x_1)$.

Перші дві формули — зведені, а інші не є зведеними.

Теорема 6.7. *Для будь-якої формули існує рівносильна їй зведена формула, причому множини вільних і зв'язаних змінних цих формул збігаються.*

Така зведена формула називається зведеною формою заданої формули.

Користуючись рівносильностями логіки висловлення, легко вказати формулу, рівносильну заданій, що містить із логічних символів тільки символи $\&, \vee, ^{-}$. Тому будемо вважати, що формули, які розглядаються, містять тільки ці логічні символи.

Доведемо це твердження методом індукції за довжиною формули. При $n = 1$ формула є атомарною і, отже, зведеною.

Припустимо, що для формул, довжина яких менша від n, твердження теореми справджується. Доведемо його для формули F завдовжки n. Позначимо довжину формули F через $\partial(F)$.

Формула F має вигляд: $G \vee H, G \& H, \overline{G}, (\forall x)G(x)$ або $(\exists x)G(x)$.

Випадок 1. Оскільки $\partial(G) \le n-2$, $\partial(H) \le n-2$, за індуктивним припущенням існують зведені форми G_1 та H_1 формул G й H відповідно, і множини вільних та зв'язаних змінних формул G_1 і G, H_1 й H збігаються. Тоді $G_1 \vee H_1$ — формула, $G_1 \vee H_1 \equiv G \vee$ Н. Звідси $G_1 \vee H_1$ — зведена форма формули $G \vee H$ із тією самою множиною вільних і зв'язаних змінних, що й $G \vee H$.

Випадок 2. Він аналогічний випадку 1.

Випадок 3. Тут формула G може мати такий вигляд:

3.1) $G_1 \vee H_1$;
3.2) $G_1 \& H_1$;
3.3) $\overline{G}_1$;

3.4) $(\forall x)G_1(x)$;
3.5) $(\exists x)G_1(x)$.

Випадок 3.1. Тут $\overline{G} \equiv \overline{G_1} \,\&\, \overline{H_1}$, де $\partial\left(\overline{G_1}\right) \leq n-2$, $\partial\left(\overline{H_1}\right) \leq n-2$. За індуктивним припущенням існують зведені формули G_2 й H_2, рівносильні $\overline{G}_1$ та $\overline{H}_1$ відповідно, і множини вільних та зв'язаних змінних формул G_1 і G_2, H_1 й H_2 збігаються. Тоді $G_2 \,\&\, H_2$ — зведена форма формули з тією самою множиною вільних змінних.

Випадок 3.2. Він аналогічний випадку 3.1.

Випадок 3.3. Тут $\overline{G} \equiv \overline{\overline{G_1}} = G_1$, де $\partial(G_1) = n-2$. Застосовуючи індуктивне припущення до формули G_1, дістаємо зведену форму формули $\overline{G}$ з тією самою множиною вільних та зв'язаних змінних.

Випадок 3.4. Тут $\overline{G} \equiv (\exists x)\overline{G_1}(x)$, де $\partial\left(\overline{G_1}(x)\right) = n-1$. За індуктивним припущенням існує зведена формула $G_2(x)$, рівносильна формулі $\overline{G}_1(x)$, з тією самою множиною зв'язаних і вільних змінних. Тоді $(\exists x)G_2(x)$ — зведена форма формули $\overline{G}$.

Випадок 3.5. Він аналогічний випадку 3.3.

Випадок 4. Формула $G(x)$ має довжину $n - 1$. За індуктивним припущенням існує зведена форма $G_1(x)$ цієї формули з тією самою множиною вільних та зв'язаних змінних. Тоді $(\forall x)G_1(x)$ — зведена форма формули $(\forall x)G(x)$.

Випадок 5. Він аналогічний випадку 4. Теорему доведено.

Зведена формула називається нормальною, якщо вона не містить символів кванторів або всі символи кванторів записано попереду (тобто логічні символи і символи предикатів знаходяться в області дії кожного квантора).

Приклад. Розглянемо такі формули:

1. $(\forall x_1)(\exists x_2)\left(\overline{A}_1^{(1)}(x_1) \vee A_1^{(2)}(x_1, x_2)\right)$ — нормальна формула.
2. $(\forall x_1)\overline{A}_1^{(1)}(x_1) \,\&\, (\exists x_2)A_2^{(1)}(x_2)$ — зведена формула, що не є нормальною.

Теорема 6.8. *Для будь-якої зведеної формули існує рівносильна їй нормальна формула тієї самої довжини.*

Така формула називається нормальною формою заданої зведеної формули.

Доведення проведемо методом індукції за довжиною n формули. При $n = 1$ формула є атомарною і, отже, нормальною.

Припустимо, що твердження теореми справджується для всіх формул, довжина яких менша від n. Нехай F — формула завдовжки n. Тоді формула F має вигляд: $G \vee H$, $G \,\&\, H$, $\overline{G}$, $(\forall x)G(x)$ або $(\exists x)(Gx)$.

Випадок 1. За умовою $G \vee H$ — зведена формула. Тоді G та H також є зведеними формулами, довжина яких менше від n. За індуктивним припущенням існують рівносильні їм нормальні формули G_1 та H_1 відповідно, де $\partial(G_1) = \partial(G)$, $\partial(H_1) = \partial(H)$. Якщо формули G_1 й H_1 не містять символів кванторів, то $G_1 \vee H_1$ — нормальна форма завдовжки n формули $G \vee H$.

Нехай, наприклад, формула G_1 містить символ квантора. Тоді G_1 має вигляд: $(\forall x)G_2(x)$, де $\delta(G_2(x)) = \delta(G_1) - 1$ (випадок, коли G_1 має вигляд

$(\exists x)G_2(x)$, є аналогічним). Якщо змінна x входить у формулу H_1, то тільки як зв'язана (інакше порушується означення формули). Застосувавши правило перейменування зв'язаних змінних, перейдемо до формули H_2, рівносильної H_1 і такої, що не має змінної x. Очевидно, H_2 — зведена формула тієї самої довжини, що й H_1. За правилом винесення квантора за дужки $(\forall x)G_2(x) \vee H_2 \equiv (\forall x)(G_2(x) \vee H_2)$. Оскільки $\delta(G_2(x) \vee H_2) = \delta(G_1) - 1 + {}$ ${}+ 1 + \delta(H_1) = n - 1$, за індуктивним припущенням існує рівносильна їй нормальна формула $F_1(x)$ тієї самої довжини. Тоді $(\forall x)F_1(x)$ — нормальна форма формули $G \vee H$ тієї самої довжини.

Випадок 2. Він аналогічний випадку 1.

Випадок 3. Оскільки формула $\overline{G}_1$ — зведена, G є атомарною формулою, і тоді $\overline{G}_1$ — нормальна формула.

Випадок 4. Тут $G(x)$ — зведена формула, $\delta(G(x)) = n - 1$. За індуктивним припущенням існує рівносильна їй нормальна формула $G_1(x)$ тієї самої довжини. Тоді $(\forall x)G_1(x)$ — нормальна форма формули $(\forall x)G(x)$ завдовжки n.

Випадок 5. Він аналогічний випадку 4. Теорему доведено.

Із теорем 6.7 і 6.8 випливає таке твердження.

Теорема 6.9. *Для будь-якої формули існує рівносильна їй нормальна формула.*

6.2.3. Здійснюваність. Загальнозначущість

Розглянемо деяку інтерпретацію з множиною M.

Означення 6.12. *Кажуть, що формула A є здійснюваною в заданій інтерпретації, якщо існує набір $\langle a_1, \dots, a_n \rangle$, $a_i \in M$, значень вільних змінних $x_1, \dots, x_n$ формули A такий, що $A|_{<a_1,\dots,a_n>} = 1$. Кажуть, що формула A є істинною в заданій інтерпретації, якщо вона набуває значення 1 у будь-якому наборі $\langle a_1, \dots, a_n \rangle$, $a_i \in M$, значень своїх вільних змінних $x_1, \dots, x_n$. Кажуть, що формула A є загальнозначущою, або тотожно-істинною (в логіці предикатів), якщо вона — істинна в кожній інтерпретації. Кажуть, що формула A є здійснюваною (в логіці предикатів), якщо існує інтерпретація, в якій A — здійснювана формула.*

Очевидно, формула A є загальнозначущою тоді й тільки тоді, коли формула $\overline{A}$ не є здійснюваною, і здійснюваною тоді й тільки тоді, коли формула $\overline{A}$ не є загальнозначущою.

Очевидно, якщо F і G — рівносильні (в логіці предикатів) формули, то $F \sim G$ є загальнозначущою формулою.

Доведемо загальнозначущість деяких формул.

Твердження 6.10. *Формула*

$$(\forall x)A(x) \to A(y), \tag{6.7}$$

де змінна y не входить у формулу $A(x)$, — загальнозначуща.

Нехай $x, x_{i_1}, \dots, x_{i_n}$ — усі вільні змінні формули $A(x)$. Тоді $y, x_{i_1}, \dots, x_{i_n}$ — перелік вільних змінних формули (6.7). Розглянемо довільну інтерпретацію з множиною M.

Нехай $\langle b, a_1, ..., a_n \rangle$, де $b \in M$, $a_i \in M\,(1 \le \text{i} \le n)$ — довільний набір значень вільних змінних формули (6.7). Доведемо, що

$$(\forall x)A(x) \to A(y)\,|_{\langle b,\, a_1, ..., a_n \rangle} = 1\,.$$

Справді, для формули A (x) або існує елемент $a_0 \in M$ такий, що в наборі $\langle a_0, a_1, ..., a_n \rangle$ значень вільних змінних $x, x_{i_1}, ..., x_{i_n}$

$$A(x)\,|_{\langle a,\, a_1, ..., a_n \rangle} = 0,$$

або для будь-якого елемента $a \in M$ у наборі $\langle a, a_1, ..., a_n \rangle$ значень вільних змінних $x, x_{i_1}, ..., x_{i_n}$

$$A(x)\,|_{\langle a,\, a_1, ..., a_n \rangle} = 1.$$

У першому випадку

$$(\forall x)A(x)\,|_{\langle a_1, ..., a_n \rangle} = 0,$$

і тоді

$$(\forall x)A(x) \to A(y)\,|_{\langle b,\, a_1, ..., a_n \rangle} = 1\,.$$

У другому випадку

$$(\forall x)A(x)\,|_{\langle a_1, ..., a_n \rangle} = 1;\ A(y)\,|_{\langle b,\, a_1, ..., a_n \rangle} = 1,$$

і тоді

$$(\forall x)A(x) \to A(y)\,|_{\langle b,\, a_1, ..., a_n \rangle} = 1\,.$$

Твердження 6.11. *Формула $A(y) \to (\exists x)A(x)$, де змінна у не входить у формулу А (x), є загальнозначущою.*

Унаслідок твердження 6.10 формула $(\forall x)\overline{A}(x) \to \overline{A}(y)$ — загальнозначуща. Маємо:

$$(\forall x)\overline{A}(x) \to \overline{A}(y) \equiv \overline{(\forall x)}\overline{A}(x) \vee \overline{A}(y) \equiv (\exists x)\overline{\overline{A}}(x) \vee \overline{A}(y) \equiv$$
$$(\exists x)A(x) \vee \overline{A}(y) \equiv A(y) \supset (\exists x)A(x)\,.$$

Отже, формула $A(y) \to (\exists x)A(x)$ є загальнозначущою. Як зазначалося вище, однойменні квантори можна переставляти. Тому формули

$$(\exists x)(\exists y)A(x, y) \sim (\exists y)(\exists x)A(x, y);$$

$$(\forall x)(\forall y)A(x, y) \sim (\forall y)(\forall x)A(x, y)$$

будуть загальнозначущими.

Загальнозначущою є також формула $(\exists x)(\forall y)A(x, y) \to (\forall y)(\exists x)A(x, y)$. Однак формула $(\forall x)\ (\exists y)A(x, y) \to (\exists y)(\forall x)A(x, y)$ не є загальнозначущою. Справді, нехай формула A (x, y) — це атомарна формула $A_1^{(2)}(x, y)$. Розглянемо інтерпретацію, областю якої є множина цілих чисел; символу $A_1^{(2)}$ поставимо у відповідність предикат $x < y$.

Тоді формула $(\forall x)(\exists y)A_1^{(2)}(x, y)$ в цій інтерпретації є істинною, а формула $(\exists y)(\forall x)A_1^{(2)}(x, y)$ — хибною.

Твердження 6.12. *Нехай A — тотожно-істинна формула логіки висловлення, а $x_{i_1}, ..., x_{i_n}$ — список її змінних. Підставивши замість кожної змінної x_{i_k}, $k = 1,2, ..., n$, формули логіки предикатів B_k (так, щоб при цьому не порушувалися п.* 1—4 *означення формули* 6.2.1), *дістанемо загальнозначущу формулу логіки предикатів.*

Задача розпізнавання загальнозначущості формул логіки предикатів істотно складніша, ніж формул логіки висловлення. Так само, як і в логіці висловлення, вона називається *проблемою розв'язуваності* і формулюється так: указати ефективний спосіб (алгоритм) розпізнавання загальнозначущості формул (тобто чи є задана формула загальнозначущою).

Загалом ця проблема в логіці предикатів — нерозв'язувана. Наведемо це твердження без доведення.

Теорема 6.10 (теорема Черча). *Не існує алгоритму, який для будь-якої формули логіки предикатів установлює, загальнозначуща вона чи ні.*

Однак у деяких окремих випадках проблема розв'язуваності вирішується. Наприклад, якщо розглядати формули логіки предикатів, які містять тільки одномісні предикатні символи, то такий алгоритм існує. Логіка, в якій використовуються тільки одномісні предикати, відповідає логіці, що була описана ще Арістотелем.

Алгоритм перевірки загальнозначущості формул, які містять тільки одномісні предикатні символи, ґрунтується на такому твердженні.

Твердження 6.13. *Нехай F — формула, що містить n одномісних предикатних символів. Для того щоб формула F була здійснюваною, необхідно й достатньо, щоб вона була здійснюваною в усіх інтерпретаціях $\langle M, f\rangle$ із множиною M, які містять не більш як 2^n елементів.*

Наведемо схему доведення цього твердження.

Нехай в інтерпретаціях $M_{f_1} = \langle M_1, f_1\rangle$ та $M_{f_2} = \langle M_2, f_2\rangle$ одномісним предикатним символам $A_j^{(1)}$ формули поставлені у відповідність предикати P_j і Q_j, $j = 1,2, ...$. Кажуть, що інтерпретації M_{f_1} та M_{f_2} — гомоморфні , якщо існує сюр'єкція φ: $M_1 \to M_2$ така, що для будь-якого $a \in M_1$ і для будь-якого $j = 1,2, ...$ $P_j(a) = Q_j(\varphi(a))$. Тоді, як випливає з індуктивного означення, формула, що містить тільки одномісні предикатні символи, в гомоморфних інтерпретаціях одночасно або здійснювана, або нездійснювана.

Покажемо, що коли формула F, яка містить n однотипних предикатних символів $A_1^{(1)}, ..., A_n^{(1)}$, є здійснюваною, вона здійснювана також у деякій інтерпретації зі скінченною множиною M, що містить не більш як 2^n елементів. Нехай $M_{f_1} = \langle M_1, f_1\rangle$ — інтерпретація, в якій є здійснюваною формула F, і нехай у цій інтерпретації предикатним символам $A_j^{(1)}$ відповідають предикати P_j, $j = 1,2, ..., n$. Для будь-якого $a \in M$ розглянемо підмножину $M_a \subseteq M_1$, яка складається з таких елементів b, що

$$\langle P_1(a), P_2(a), \ldots, P_n(a)\rangle = \langle P_1(b), P_2(b), \ldots, P_n(b)\rangle .$$

Число таких підмножин M_a не перевищує числа наборів з 1 і 0 завдовжки n, тобто не перевищує 2^n. Виберемо в кожній з цих підмножин по одному представнику і складемо з них множину M_2. Тоді інтерпретації $M_{f_1} = \langle M_1, f_1\rangle$ та $M_{f_2} = \langle M_2, f\rangle$, де f_2 — обмеження функції f_1 на $M_2 \subseteq M_1$, — гомоморфні, й, отже, формула F є здійснюваною в інтерпретації M_{f_2} з множиною M_2, що містить не більш як 2^n елементів. Звідси випливає твердження 6.13.

6.2.4. Числення предикатів

У логіці предикатів, на відміну від логіки висловлення, немає ефективного способу для розпізнавання загальнозначущості формул. Тому аксіоматичний метод стає істотним при вивченні формул, які містять квантори. Визначення загальнозначущих формул, так само, як і в численні висловлень, здійснюється заданням деякої сукупності формул, що називаються аксіомами, і заданням правил виведення, що дають змогу з одних загальнозначущих формул одержувати інші.

На відміну від загальноприйнятого викладу, розглянемо деяку вузьку аксіоматичну теорію, яку також називатимемо численням предикатів.

Означення 6.13. *Числення предикатів — це аксіоматична теорія, символами якої є взагалі ті самі символи, що й у логіці предикатів* (6.2.1):

- *символи предметних змінних* $x_1, x_2, \ldots, x_n, \ldots$;
- *символи предикатів* $A_1^{(t)}, A_2^{(t)}, \ldots, A_n^{(t)}, \ldots (t = 0,1,2, \ldots)$;
- *логічні символи* $^-$, $\rightarrow$;
- *символи кванторів* $\forall, \exists$;
- дужки й кома (,).

Сформульоване в п. 6.2.1 означення формули поширюється також для числення предикатів з тією лише різницею, що тут застосовується тільки два логічних символи $^-$ і $\rightarrow$; інші зв'язки можна ввести, наприклад, так, як це зроблено в численні висловлень, у п. 6.1.2.

Аксіоми числення предикатів. Які б не були формули A та B, такі формули є аксіомами (при цьому не має порушуватися означення формули):

А1. $A \rightarrow (B \rightarrow A)$;

А2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$;

А3. $(\overline{B} \rightarrow \overline{A}) \rightarrow ((\overline{B} \rightarrow A) \rightarrow B)$;

А4. $(\forall x_i) A(x_i) \rightarrow A(x_i)$, де формула $A(x_i)$ не містить змінної x_i;

А5. $A(x_i) \rightarrow (\exists x_i) A(x_i)$, де формула $A(x_i)$ не містить змінної x_i.

Правила виведення предикатів (при цьому не повинно порушуватися означення формули):

1. Правило m. p. $\dfrac{A, A \rightarrow B}{B}$.

2. Правило зв'язування квантором загальності $\dfrac{B \to A(x_i)}{B \to (\forall x_i)A(x_i)}$, де формула B не містить змінної x_i.

3. Правило зв'язування квантором існування $\dfrac{A(x_i) \to B}{(\exists x_i)A(x_i) \to B}$, де формула B не містить змінної x_i.

4. Правило перейменування зв'язаної змінної. Зв'язану змінну формули A можна замінити (в кванторі й у всіх входженнях в області дії квантора) іншою змінною, яка не є вільною в A.

Поняття виведення, теореми, виведення з системи гіпотез визначаються в численні предикатів так само, як і в будь-якій аксіоматичній теорії (див. п. 6.1.1).

Теорема 6.11 (ослаблена теорема про дедукцію). *Якщо Γ, $A \vdash B$ й існує виведення в численні предикатів, побудоване із застосуванням тільки правила* 1, *то* $\Gamma \vdash A \to B$.

Ця теорема доводиться аналогічно теоремі про дедукцію в численні висловлень.

Можна показати, що клас усіх теорем числення предикатів збігається з класом загальнозначущих формул.

Твердження 6.14. *Аксіоми числення предикатів є загальнозначущими формулами.*

Для аксіом А1—А3 це випливає із твердження 6.12, для аксіом А4 й А5 — із тверджень 6.10 і 6.11.

Твердження 6.15. *Формула, що випливає із загальнозначущої формули за будь-яким із правил виведення* 1—4, *є загальнозначущою.*

Для правила виведення 1 це твердження випливає з властивостей імплікації.

Розглянемо правило виведення 2. Нехай $B \to A(x_i)$ — загальнозначуща формула. Доведемо, що формула $B \to (\forall x_i)A(x_i)$ також є загальнозначущою. Нехай $x_{i_1}, \ldots, x_{i_n}$ — усі вільні змінні цієї формули. Тоді $x_i, x_{i_1}, \ldots, x_{i_n}$ — перелік вільних змінних формули $B \to A(x_i)$ (формула B не містить змінної x_i).

Нехай задано довільну інтерпретацію з множиною M і нехай $\langle a_i, \ldots, a_n \rangle$, $a_i \in M, 1 \le i \le n$, — довільний набір значень вільних змінних формули $B \to (\forall x_i)A(x_i)$.

Можливими є два випадки:

- $B|_{\langle a_1, \ldots, a_n \rangle} = 1$;
- $B|_{\langle a_1, \ldots, a_n \rangle} = 0$.

У першому випадку із загальнозначущості формули $B \to A(x_i)$ випливає, що для будь-якого елемента a з множини M маємо $A(x_i)|_{\langle a, a_1, \ldots, a_n \rangle} = 1$. Тоді $(\forall x_i)A(x_i)|_{\langle a_1, \ldots, a_n \rangle} = 1$, звідки

$$B \to (\forall x_i)A(x_i)|_{\langle a_1, \ldots, a_n \rangle} = 1.$$

У другому випадку внаслідок властивості імплікації

$$B \to (\forall x_i) A(x_i)|_{\langle a_1, \ldots, a_n \rangle} = 1 .$$

Розглянемо правило виведення 3. Нехай $A(x_i) \to B$ — загальнозначуща формула. Доведемо, що формула $(\exists x_i) A x_i \to B$ також є загальнозначущою. Нехай $x_{k_1}, \ldots, x_{k_n}$ — усі вільні змінні цієї формули. Тоді $x_i, x_{k_1}, \ldots, x_{k_n}$ — перелік вільних змінних формули $A(x_i) \to B$ (формула B не містить змінної x_i).

Нехай задано довільну інтерпретацію з множиною M і нехай $\langle a_i, \ldots, a_n \rangle$, $a_i \in M, 1 \leq \mathrm{i} \leq n$ — довільний набір значень вільних змінних формули $(\exists x_i) A x_i \to B$.

Можливими є два випадки:

- $B|_{\langle a_1, \ldots, a_n \rangle} = 1$;
- $B|_{\langle a_1, \ldots, a_n \rangle} = 0$.

У першому випадку внаслідок властивостей імплікації

$$(\exists x_i) A(x_i) \to B|_{\langle a_1, \ldots, a_n \rangle} = 1 .$$

У другому випадку із загальнозначущості формули $A(x_i) \to B$ випливає, що для будь-якого елемента a з множини M матимемо $A(x_i)|_{\langle a, a_1, \ldots, a_n \rangle} =$ $= 0$. Тоді $(\exists x_i) A(x_i)|_{\langle a_1, \ldots, a_n \rangle} = 0$, звідки $(\exists x_i) A(x_i) \to B|_{\langle a_1, \ldots, a_n \rangle} = 1$.

Справедливість твердження, що доводиться для правила виведення 4, є очевидною.

Теорема 6.12. *Будь-яка вивідна в численні предикатів формула є загальнозначущою.*

Ця теорема випливає із тверджень 6.14 і 6.15.

Приклад. Доведемо загальнозначущість формули

$$(\exists x_i)(\forall x_j) A(x_i, x_j) \to (\forall x_j)(\exists x_j) A(x_i, x_j) .$$

Спочатку покажемо, що ця формула є теоремою числення предикатів, тобто

$$\vdash (\exists x_i)(\forall x_j) A(x_i, x_j) \to (\forall x_j)(\exists x_i) A(x_i, x_j) .$$

Побудуємо це виведення:

(1) $\vdash (\forall x_j) A(x_i, x_j) \to A(x_i, x_t)$ (аксіома 4);

(2) $\vdash A(x_i, x_i) \to (\exists x_k) A(x_k, x_t)$ (аксіома 5);

(3) $A \to B, B \to C \vdash A \to C$ (за теоремою 6.11);

(4) $\vdash (\forall x_j) A(x_i, x_j) \to (\exists x_k) A(x_k, x_t)$ ((3) застосовано до (1) і (2));

(5) $\vdash (\forall x_j) A(x_i, x_j) \to (\exists x_k) A(x_k, x_t)$ (правило виведення 3 застосовано до (4));

(6) $\vdash (\exists x_i)(\forall x_j) A(x_i, x_j) \to (\forall x_t)(\exists x_k) A(x_k, x_t)$ (правило виведення 2 застосовано до (5));

(7) $\vdash (\exists x_i)(\forall x_j) A(x_i, x_j) \to (\forall x_j)(\exists x_k) A(x_k, x_j)$ (правило виведення 4 застосовано до (6));

(8) $\vdash (\exists x_i)(\forall x_j) A(x_i, x_j) \to (\forall x_j)(\exists x_i) A(x_i, x_j)$ (правило виведення 4 застосовано до (7)).

Теорема 6.13. Числення предикатів є несуперечливим.

Справді, внаслідок теореми 6.12 неможливо, щоб $\vdash A$ і $\vdash \bar{A}$ справджувалися одночасно.

Наведемо без доведення формулювання теореми, оберненої до теореми 6.12.

Теорема 6.14 (теорема Геделя про повноту числення предикатів). *Уська загальнозначуща формула виводиться в численні предикатів.*

6.3. Канонічна система Поста

Вище підкреслювалося, що виведення у формальній системі можна проводити, орієнтуючись тільки на взаємне розташування символів (синтаксис) у рядках. Можна навіть запропонувати алгоритм, який продовжуватиме послідовність рядків різної довжини. Оскільки легко перевірити, чи є задана послідовність виведенням, для будь-якої згенерованої послідовності рядків це питання вирішується. В [1] показано, що коли деякий скінченний рядок є теоремою формальної системи, то потрібне виведення за допомогою цього алгоритму буде знайдено за скінченне число кроків.

При такому підході до доведення обробляються рядки, які в будь-який момент часу містять скінченне число символів зі скінченного алфавіту з використанням скінченної (або обчислюваної) кількості аксіом і правил виведення. У попередньому розділі було сформульовано правило виведення у вигляді досить загальної обчислюваної функції. Тепер, враховуючи суто синтаксичний підхід, спробуємо сформулювати конкретну форму правил виведення, щоб вони задовольняли також описану вже функцію, і синтаксичний підхід у реалізації виведення. Для цього проаналізуємо операції, які потрібно здійснювати при виведенні рядка з множини рядків, що вже існують.

Основний принцип синтаксичного підходу — утворення подальшого рядка з попередніх розчленуванням, перестановкою символів, заміною одних підрядків іншими, об'єднанням підрядків у рядках, які вже існують.

Для виконання зазначених операцій треба:

- переглянути рядок символ за символом і визначити в ньому деякий фіксований підрядок;
- вилучити ці підрядки й рядки, зберігаючи при цьому частини, що залишилися;
- перебудувати частини рядка, вставляючи деякий фіксований підрядок у різні місця та вилучаючи які-небудь частини рядка.

Цих операцій цілком досить, щоб здійснити виведення будь-якого рядка, який є теоремою формальної системи. Для реалізації згаданих вище операцій використовують продукцію.

Правило, що точно встановлює, як розчленовувати рядок і перебудовувати його частини, називається продукцією. Звичайна форма запису продукції така: <ліва частина>→<права частина>, де рядки в трикутних дужках означають деякі поняття, які є символами метамови — нетермінальними символами, а самі поняття є рядками символів з вилученими підрядками.

Проілюструємо поняття продукції на конкретних прикладах.

Приклади: 1. Побудувати формальну систему з правилами виведення у формі продукції, теоремами якої були б парні додатні числа, виключаючи нуль, в унарній системі числення.

Нагадаємо, що в цій системі для подання чисел використовується єдиний символ — 1. Рядки, які є теоремами формальної системи, — це 11, 1111, 111111, … . Отже, алфавіт цієї системи поданий множиною $\{1\}$. Далі зазначається, що коли деякий рядок є парним числом, то, приписавши до цього рядка дві одиниці (праворуч або ліворуч), знову дістанемо парне число. У цьому разі утворення рядка з попереднього полягає у приєднанні до нього фіксованих символів. Якщо попередній рядок позначити a, то можна скласти продукцію: $a \to a11$.

Тоді формальна система матиме вигляд

- алфавіт $A = \{1\}$;
- аксіоми 1) 11;
- продукції 1) $a \to a11$.

Покажемо, що 111111 є теоремою цієї формальної системи. Виведення

1) 11 аксіома 1;
2) 1111 з (1) і продукції 1;
3) 111111 із (2) та продукції 1

переконує у цьому. Зазначимо, що для утворення нового рядка зі старого не потрібні операції розчленування і перестановлення підрядків у старому рядку. Тут достатньо приписування фіксованого числа символів.

2. Побудувати формальну систему, теореми якої є правильними твердженнями щодо результату операції додавання двох цілих додатних чисел (виключаючи 0) в унарній системі числення. Уточнимо, що теоремами системи будуть рядки 1 + 1 = 11, 1 + 11 = 111, 11 + 111 = 11111, 111 + 11 = 11111, …, які правильно задають операцію додавання.

Очевидно, для породження рядків, які є теоремами, потрібно використати три символи: 1, +, =. Зрозуміло також, що символи + і = можна застосувати як фіксовані, які розбивають рядок на підрядки, даючи змогу визначити доданки та результат як підрядки рядка. Позначимо підрядок, що передує символу +, через a_1, підрядок, наступний за символом +, через a_2, а підрядок, наступний за символом =, — через a_3. Тоді, якщо рядок $a_1 + a_2 = a_3$ є теоремою, то рядки $a_1 1 + a_2 = a_3 1$, $a_1 + a_2 1 = a_3 1$ також будуть теоремами. Як аксіоми можна взяти один рядок 1 + 11 = 11, що є твердженням про суму двох найменших чисел. Усі інші теореми можна дістати виведенням. Ураховуючи викладене, маємо таку формальну систему:

- алфавіт $A = \{1, +, =\}$;
- аксіоми 1) 1+1=11;
- продукції 1) $a_1 + a_2 = a_3 \to a_1 1 + a_2 = a_3 1$, 2) $a_1 + a_2 = a_3 \to a_1 + a_2 1 = a_3 1$.

Покажемо, що 3 + 2 = 5 — теорема формальної системи, для чого виконаємо відповідне виведення:

1) 1 + 1 = 11 аксіома 1;
2) 11 + 1 = 111 з (1) і продукції 1;
3) 111 + 1 = 1111 із (2) та продукції 1;
4) 111 + 11 = 11111 із (3) і продукції 2.

У продукціях цього прикладу треба було розчленувати рядок на підрядки за допомогою сталих символів (+, =), а також уводити всередину рядка в точно фіксованому місці сталий підрядок 1.

Розглянемо тепер приклад, в якому б використовувалися продукції з розчленуванням рядка на підрядки і перебудова підрядків.

Приклад. Побудувати формальну систему, теоремами якої були б правильні твердження про множення цілих чисел (виключаючи 0) в унарній системі числення. Теореми системи: 1*1 = 1, 11*1 = 11, 11*111 = 111111, 111*11 = 111111.

Аналогічно попередньому прикладу символи * і = є роздільниками між співмножниками та результатом. Якщо $a_1*a_2 = a_3$ — теорема системи, то $a_1 1*a_2 = a_3 a_2$, $a_1* a_2 1 = a_3 a_1$ — також теореми. Останні два рядки утворено з попереднього приписуванням символів a_2, a_1 у вказаному місці й перебудовою підрядка a_2, a_1 услід за підрядком a_3. Нагадаємо, що приписування підрядка є операцією конкатенації. Таким чином, шуканою формальною системою є така:

- алфавіт A = {1, *, =};
- аксіома 11*1 = 1;
- продукції $a_1*a_2 = a_3 \to a_1 1*a_2 = a_3 a_2$, $a_1*a_2 = a_3 \to a_1*a_2 1 = a_3 a_1$.

Розглянемо виведення 11*11=1111:

1) 1 * 1 = 1 — аксіома 1;
2) 11 * 1 = 11 з 1) і продукції 1;
3) 11 * 11 = 1111 із 2) та продукції 2.

У наведених прикладах розглянуто операції, які виконуються над рядками і підрядками продукції. Зазначимо, що сама продукція є рядком в деякому алфавіті. Дамо точне означення продукції.

Означення 6.16. *Продукцією називається рядок вигляду*

$$\gamma_0 \sum_1 \gamma_1 \sum_2 \gamma_2 \cdots \sum_n \gamma_n \to \beta_0 \sum_1^{/} \beta_1 \sum_2^{/} \beta_2 \cdots \sum_m^{/} \beta_m ,$$

де кожне $\gamma_i (0 \le i \le n)$ *та* $\beta_i (0 \le j \le m)$ *є деяким фіксованим підрядком, що може бути порожнім; кожне* $\sum_i (1 \le i \le n)$ *є рядком (підрядком), що може бути порожнім, який інтерпретується як змінна (тобто будь-який рядок); кожне* $\sum'_j (0 \le \mathrm{j} \le m)$ *є певним рядком із множини* $\{\sum_1, \ldots, \sum_n\}$.

Візьмемо, наприклад, продукцію з прикладу 2 на с. 174: $a_1 + a_2 = a_3 \to \to a_1 1 + a_2 = a_3 1$. Тут $\gamma_0 = e$, $\sum_1 = a_1$, $\gamma_1 = +$, $\sum_2 = a_2$, $\gamma_2 = =$, $\sum_3 = a_3$, $\gamma_3 = e$, $b_0 = e$, $\sum_1 = \sum_1 = \alpha_1$, $b_1 = 1+$, $\sum_2 = \sum_2 = \alpha_2$, … .

Часто ліву частину продукції називають посилкою, а праву — висновком.

Твердження 6.16. *Формальна система є канонічною системою Поста, якщо вона означається алфавітом A, набором аксіом (рядків в A), набором продукцій (згідно з означенням 6.16), підрядки яких є рядками в A.*

Усі наведені в цьому підрозділі приклади — канонічні системи Поста.

Існує теза, що будь-яка формальна система може бути виражена канонічною системою Поста. У цьому значенні канонічна система є універсальною і часто використовується як ще одне з означень алгоритму. В основному область застосування канонічних систем — теорія алгоритмів. Надалі будемо використовувати поняття продукції канонічної системи, накладаючи на неї певні обмеження для введення нових типів формальних систем.

Таким чином, формальну систему можна розглядати як окремий випадок канонічної системи Поста. Тому, з одного боку, алгоритмічна потужність такої системи не повинна бути більшою від алгоритмічних можливостей канонічної системи, а з іншого — будь-яка формальна система, визначена як окремий випадок канонічної системи, алгоритмічно розв'язувана. Іншими словами, завжди гарантовано існування алгоритму, що реалізує формальну систему.

Системою, еквівалентною канонічній системі Поста, є МТ. Хоча вона була введена незалежно від системи Поста, розглянемо її як окремий випадок цієї системи.

6.4. Машина Тьюрінга як окремий випадок канонічної системи Поста

Розглядаючи МТ таким чином, потрібно зберегти, хоча б із погляду зовнішньої аналогії, а не внутрішньої суті, вже введені елементи формальної системи:

- алфавіт;
- правила виведення у формі продукцій;
- механізм доведення за допомогою продукцій, що визначається як перетворення вже утвореного рядка за продукцією, де цей рядок є лівою частиною.

У принципі зберігається множина аксіом, що здебільшого стає нескінченною. Означення ж теореми в МТ буде трохи відмінним від наведеного вище. Така аналогія є корисною, оскільки немає потреби в доданні нових понять до тих, які вже існують при визначенні формального об'єкта. Надалі можна встановити еквівалентність наведених нижче понять з поняттями, про які йшлося вище.

Твердження 6.17. *МТ є формальною системою, що складається з алфавіту, множини аксіом (початкові рядки) та множини продукцій (команд), які називаються програмою. Теоремами МТ будуть слова спеціального вигляду, про які йтиметься нижче.*

Докладно розшифруємо всі елементи МТ. По суті, алфавіт складається з двох алфавітів: вхідних символів і станів $A_1 = A \bigcup Q$, де $A = \{a_0, a_1, ..., a_n\}$, $Q = \{q_0, q_1, ..., q_m\}$. В алфавіті вхідних символів фіксується спеціальний символ $a_0 (\lambda)$, що вказує на порожнє місце (порівняйте з порожнім символом), а в алфавіті станів — символ q_0, який вказує на заключний стан. Множиною аксіом, або вхідних рядків МТ, будуть рядки, складені з символів алфавіту A та єдиного символу $\{q_0\}$ із Q. Положення символу q_i таке, що за ним має бути підрядок із символів a_i. При цьому не будь-який вхідний рядок є аксіомою. Для кожної конкретної МТ визначаються свої правильні вхідні рядки, які будуть аксіомами. Вони задаються правилами, що визначають вхідні рядки.

Команди МТ можуть бути утворені з постівських продукцій. Продукція канонічної системи Поста потребує, щоб розбиття рядків у лівій і правій частинах продукції задовольняло такі вимоги:

- підрядки цього розбиття мають бути або константами (що складаються з фіксованих символів), або змінними (що утворюються у процесі роботи формальної системи);

- змінні підрядки правої частини обов'язково мають братися з множини змінних підрядків лівої частини. У команді МТ відмовимося від змінних підрядків, залишивши тільки фіксовані, які складаються з одного символу (або рядка, що складається з кількох фіксованих символів). Перш ніж задати конкретну форму команди МТ, зробимо невелике зауваження. У рядку $a_{i_1}a_{i_2}\ldots q_i a_{i_k}\ldots a_{i_n}$, де існує символ q_i, що визначає стан, в якому розглядається символ, який знаходиться праворуч від нього, тобто a_{i_k}, команди МТ набувають такого вигляду:

$$\begin{array}{ll}
a_{i_1}q_i a_{i_2} \to a_{i_1}q_j a_{i_2}; & a_{i_1}q_i a_{i_2} \to a_{i_1}a_{j_2}q_i; \\
a_{i_1}q_i a_{i_2} \to a_{i_1}q_i a_{j_2}; & a_{i_1}q_i a_{i_2} \to q_i a_{i_1}a_{i_2}; \\
a_{i_1}q_i a_{i_2} \to a_{i_1}q_j a_{j_2}; & a_{i_1}q_i a_{i_2} \to q_j a_{i_1}a_{i_2}; \\
a_{i_1}q_i a_{i_2} \to a_{i_1}a_{i_2}q_i; & a_{i_1}q_i a_{i_2} \to q_i a_{i_1}a_{j_2}; \\
a_{i_1}q_i a_{i_2} \to a_{i_1}a_{i_2}q_j; & a_{i_1}q_i a_{i_2} \to q_j a_{i_1}a_{j_2}. \\
a_{i_1}q_i a_{i_2} \to a_{i_1}a_{j_2}q_j; &
\end{array}$$

Ліва і права частини команди складаються із трьох фіксованих символів, два з яких належать алфавіту A, а один — алфавіту Q. Символ стану розглядає вхідний символ, який знаходиться праворуч від нього. Перші три команди показують, що символ q_1, розглядаючи символ a_{i_2}, не змінює свого положення відносно символу, який розглядається, але може в різних комбінаціях змінюватися сам або змінювати символ, що розглядається. Наступні чотири команди переміщують символ q_i відносно того, який розглядається, праворуч і можуть змінювати як q_i, так і a_{i_2} (на q_j та a_{j_2} відповідно). Аналогічно останні чотири команди зсувають символ q_i ліворуч відносно a_{i_2} з тією самою комбінацією змін.

Функціонування МТ аналогічне постівській системі. Починаємо з аксіом. Застосовуючи продукції, утворюємо подальші рядки. Проте якщо в постівській системі продукція застосовувалася повністю до попереднього рядка для утворення наступного, то в МТ команда використовується тільки до частини попереднього рядка (підрядка), що збігається з лівою частиною команди. При функціонуванні МТ може виникнути ситуація, коли символ q_i, зсуваючись ліворуч або праворуч по рядку, не має символу для розгляду, тобто символ q_i дійшов до лівої або правої межі рядка. У цьому випадку до межі рядка автоматично приписується символ a_0 (символ порожнього місця).

Теоремою МТ буде не кожний знов утворений рядок, як у постівській системі, а тільки рядок, що містить символ q_0.

Розглянемо приклади побудови МТ як канонічної системи Поста.

Приклад. Побудувати МТ, яка реалізує продукцію $a_1 + a_2 = a_3 \to a_1 1 + a_2 = a_3 1$ із прикладу 2 на с. 174.

Запропонована продукція канонічної системи перетворює правильно записаний рядок додавання двох чисел в унарній системі числення на правильний рядок, відмінний від попереднього тим, що до першого доданка та результату приписується одиниця. Шукана МТ має виконати таке приписування і перейти в заключний стан, щоб знов утворений рядок складав теорему МТ.

Дії, що виконуються МТ, мають бути такими: знаходячись поблизу лівого кінця початкового рядка (ліва частина постівської продукції) в деякому стані, зсуваємося праворуч доти, доки не дійдемо до правого кінця рядка (символ a_0). Під час руху символу q праворуч не будемо змінювати символів, які розглядаються, а стан буде змінюватися при переході через символи +, =, a_0. Дійшовши до правого кінця рядка, зсуваємо символ q ліворуч, змінивши заздалегідь символ a_0 на 1. Рух ліворуч здійснюємо до досягнення лівого кінця рядка, після чого символ a_0 замінюємо на 1, і стан стає q_0. Після таких перетворень результат (права частина заданої продукції) розташовується праворуч від символу q_0.

Як система аксіом цієї МТ використовується множина правильно побудованих рядків для операції додавання. Будь-яка теорема, утворена на попередніх кроках функціонування МТ, може бути використана як аксіома на подальших кроках.

Виходячи з викладеного вище, маємо:

- алфавіт $A = \{a_0, 1, +, =\}, Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$;
- програму

1. $a_0q_1 1 \rightarrow a_0 1 q_2$;
2. $1q_2 1 \rightarrow a_0 1 q_2$;
3. $1q_2 + \rightarrow 1 + q_3$;
4. $a_3 q_3 1 \rightarrow a_3 1 q_3$, де $a_3 \in \{+, 1\}$;
5. $1q_3 = \rightarrow 1 = q_4$;
6. $1q_4 1 \rightarrow 11 q_4$;
7. $1q_4 a_0 \rightarrow q_5 11$;
8. $a_6 q_5 a_7 \rightarrow q_5 a_6 a_7$,
де $a_6 \in \{0, 1, +, =\}$, $a_7 \in \{1, +, =\}$;
9. $a_0 q_5 a_0 \rightarrow a_0 q_1 1$.

Розглянемо приклад функціонування побудованої МТ. Оскільки перехід до наступного рядка при доведенні в МТ відбувається строго від попереднього рядка, не будемо нумерувати кожний утворений рядок, а введемо крок функціонування від попереднього рядка до наступного. Позначимо його через $\overset{i}{\Rightarrow}$, де i вказує номер команди на цьому кроці. Нехай початковий рядок 11 + 1 = 111. Аксіомою МТ, від якої починається її функціонування, буде рядок $a_0q_1 11+1=111$. Виконуючи кроки функціонування за допомогою команд, дістаємо:

$$a_0q_1 11+1=111a_0 \overset{1}{\Rightarrow} a_0 1q_2 1+1=111a_0 \overset{2}{\Rightarrow} a_0 11q_2+1=111a_0 \overset{3}{\Rightarrow}$$

$$a_0 11+q_3 1=111a_0 \overset{4}{\Rightarrow} a_0 11+1q_3=111a_0 \overset{5}{\Rightarrow} a_0 11+1=q_4 111a_0 \overset{6}{\Rightarrow}$$

$$a_0 11+1=1q_4 11a_0 \overset{6}{\Rightarrow} a_0 11+1=11q_4 1a_0 \overset{6}{\Rightarrow} a_0 11+1=111q_4 a_0 \overset{7}{\Rightarrow}$$

$$a_0 11+1=11q_5 11 \overset{8}{\Rightarrow} a_0 11+1=1q_5 111 \overset{8}{\Rightarrow} a_0 11+1=q_5 1111 \overset{8}{\Rightarrow}$$

$$a_0 11+1q_5=1111 \overset{8}{\Rightarrow} a_0 11+q_5 11+q_5 1=1111 \overset{8}{\Rightarrow} a_0 11q_5+1=1111 \overset{8}{\Rightarrow}$$

$$a_0 1q_5 1+1=1111 \overset{8}{\Rightarrow} a_0 q_5 11+1=1111 \overset{8}{\Rightarrow} a_0 q_5 a_0 11+1=1111 \overset{9}{\Rightarrow}$$

$$a_0 q_0 111+1=1111.$$

Останній рядок у цих кроках і є теоремою в сенсі наведеного вище означення. Однак цей рядок не є теоремою в канонічній системі; він може стати нею, якщо викреслити символ q_0 і всі символи a_0 з неї. А щоб цей заключний рядок МТ був аксіомою, його треба подати у вигляді $q_1 111 + 1 = 1111$ або $a_0 q_1 111 + 1 = 1111 a_0$. Зазначимо, що a_0 на межах рядка можна не вказувати. Цей символ з'являється там автоматично, якщо в ньому виникає необхідність.

Наведемо деякі додаткові міркування щодо побудови МТ. Алфавіт Q може бути сформований після того, як сформовано програму, що дасть потрібне число станів. Стани, які вводяться, є деякою внутрішньою пам'яттю. Наприклад, стан q_1 можна розглядати як початковий і зажадати, щоб він розглядав тільки символ 1 і в команді зсувався праворуч; стан q_2 означає, що читання рядка зліва направо відбувається правильно, але як тільки в q_2 розглядається +, крім зсуву праворуч, відбувається зміна стану на q_3. Стан q_3 свідчить про те, що рядок, який читається, належить (поки, принаймні) до множини аксіом. Для цієї самої мети введено стан q_4. Стан q_5 указує, що ми благополучно дійшли до правого кінця рядка, приписали до результату 1 і рухаємося до його лівого кінця.

Тепер переконуємося, що початковий рядок належить множині аксіом, тому змінювати стани під час руху ліворуч немає необхідності. Змінювати q_5 на q_0 можна тільки тоді, коли символ, який розглядається, дорівнює a_0. Тут разом зі зміною стану (що не зсувається) a_0 змінюється на 1. Тим самим першому доданку приписується 1.

Зауважимо, що запропонований спосіб побудови МТ, яка реалізує продукцію $a_1 + a_2 = a_3 \to a_1 1 + a_2 = a_3 1$, не є єдиним. Можна було б, наприклад, при досягненні символу + змінити його на 1, тим самим приписавши 1 до першого доданка. Потім першу одиницю після + перетворити в + і, зсуваючись праворуч, відновити рівність та приписати 1 до результату. Далі зсунути останній стан до лівої межі рядка і зупинитися. Ця остання дія потрібна для того, щоб результат знаходився праворуч від символу q_0.

Якщо при функціонуванні МТ зустрічається рядок, до якого не може бути застосована жодна команда з програми, то скінченного результату (теореми) не існує. У цьому випадку кажуть, що МТ непридатна до початкового рядка.

Проте часто потрібно розглядати МТ, яка працює нескінченний час (без появи символу q_0) і видає нескінченну множину теорем. У цьому разі означення МТ потребує деякого уточнення. Якщо МТ зі скінченною програмою (а програма має бути скінченною) не видає символу q_0, то вона, будучи застосовною до всіх рядків, які з'являються, мусить циклічно використати певні набори команд.

Деякі рядки, що з'являються внаслідок функціонування МТ, можна інтерпретувати як теореми, які одночасно були б аксіомами для наступного циклу функціонування. У формальних системах аксіоми означено як теореми нульового рівня. Тому можна використати для зазначення теорем той стан, що застосовується для зазначення аксіом. У прикладі 2 на с. 174 це був стан q_1.

Розглянемо приклад МТ, яка працює нескінченно і видає нескінченну множину теорем.

Приклад. Побудувати МТ, що моделює канонічну систему Поста з попереднього прикладу.

Там ми змоделювали на МТ першу продукцію постівської системи. Аналогічно змоделюємо другу та примусимо працювати команди МТ циклічно без виходу в заключний стан. Теоремами МТ будуть рядки, які містять символ q_1. Перш ніж розпочати побудову заданої МТ, зробимо кілька важливих зауважень:

- в основу побудови МТ покладемо спосіб, викладений у прикладі на с. 178;
- оскільки моделюється недетермінований процес (порядок застосування постівських продукцій є довільним), сама МТ має бути недетермінованою. Це означає, що ліві частини команд можуть збігатися (у прикладі на с. 178 МТ була строго детермінованою);
- оскільки сконструйована МТ не зупиняється, передбачається наявність зовнішньої системи, що фіксує появу теорем. Теорема після її появи завжди є правильним рядком. Тому відпадає необхідність контролю її правильності, а водночас у введенні додаткових станів. Не будемо контролювати також правильність запису аксіоми $1 + 1 = 11$, залишивши це зовнішній системі. Враховуючи викладене, можна обійтися єдиним станом під час руху праворуч, змінюючи його тільки для того, щоб зазначити, яка продукція моделюється (перша або друга);
- моделюючи другу продукцію, будемо приписувати одиницю до другого доданка заміною = на 1. Символ = записуємо замість першої одиниці результату, а потім приписуємо дві одиниці в його кінці;
- повертаючись до лівого кінця рядка, переходимо не в q_0, а в q_1.

Таким чином, шукана МТ має:

- алфавіт $A = \{a_0, 1, +, =\}$, $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$;
- аксіоми $a_0 q_1 1 + 1 = 11 a_0$;
- програму:

1) $a_0 q_1 1 \rightarrow a_0 1 q_1$;
2) $a_0 q_1 1 \rightarrow a_0 1 q_3$;
3) $1 q_1 1 \rightarrow 11 q_1$;
4) $1 q_1 + \rightarrow 1 + q_1$;
5) $1 q_1 = \rightarrow 1 = q_1$;
6) $1 q_1 a_0 \rightarrow q_2 11$;
7) $+ q_1 1 \rightarrow +1 q_1$;
8) $= q_1 1 \rightarrow =1 q_1$;
9) $1 q_2 1 \rightarrow q_2 11$;
10) $1 q_2 = \rightarrow q_2 1 =$;
11) $1 q_2 + \rightarrow q_2 1 +$;
12) $a_0 q_1 a_0 \rightarrow a_0 q_1 1$;
13) $a_0 q_2 1 \rightarrow q_2 a_0 1$;
14) $1 q_3 1 \rightarrow 11 q_3$;
15) $1 q_3 + \rightarrow 1 + q_3$;
16) $1 q_3 = \rightarrow 11 q_4$;
17) $= q_2 1 \rightarrow q_2 =1$;
18) $+ q_2 1 \rightarrow q_2 + 1$;
19) $+ q_3 1 \rightarrow +1 q_3$;
20) $1 q_4 1 \rightarrow 1 = q_5$;
21) $= q_5 1 \rightarrow = 1 q_5$;
22) $1 q_5 1 \rightarrow 11 q_5$;
23) $1 q_5 a_0 \rightarrow 11 q_6$;
24) $1 q_6 a0 \rightarrow q_7 11$;
25) $1 q_7 1 \rightarrow q_7 11$;
26) $= q_7 1 \rightarrow q_7 = 1$;
27) $1 q_7 = \rightarrow q_7 1 =$;
28) $+ q_7 1 \rightarrow q_7 + 1$;
29) $1 q_7 + \rightarrow q_7 1 +$;
30) $a_0 q_7 1 \rightarrow q_7 a_0 1$;
31) $a_0 q_7 a_0 \rightarrow a_0 a_0 q_1$.

Покажемо, що $11 + 1 = 111$ та $11 + 11 = 1111$ є теоремами побудованої МТ, для чого примусимо її функціонувати, причому

$$a_0 q_1 1 + 1 = 11 a_0 \overset{1}{\Rightarrow} a_0 1 q_1 + 1 = 11 a_0 \overset{4}{\Rightarrow} a_0 1 + q_1 1 = 11 a_0 \overset{7}{\Rightarrow} a_0 1 + 1 q_1 = 11 a_0 \overset{5}{\Rightarrow}$$

$$a_0 1 + 1 = q_1 11 a_0 \overset{8}{\Rightarrow} a_0 1 + 1 = 1 q_1 1 a_0 \overset{3}{\Rightarrow} a_0 1 + 1 = 11 q_1 a_0 \overset{6}{\Rightarrow} a_0 1 + 1 = 1 q_2 11 \overset{9}{\Rightarrow}$$

$$a_0 1+1=q_2 111 \overset{13}{\Rightarrow} a_0 1+1q_2=111 \overset{10}{\Rightarrow} a_0 1+q_2 1=111 \overset{14}{\Rightarrow} a_0 1q_2+1=111 \overset{11}{\Rightarrow}$$

$$a_0 q_2 1+1=111 \overset{15}{\Rightarrow} a_0 q_2 a_0 1+1=111 \overset{12}{\Rightarrow} a_0 q_1 11+1=111 \overset{2}{\Rightarrow} a_0 1q_3 1+1=111 \overset{*}{\Rightarrow}$$

$$a_0 11+1q_3=111 \overset{18}{\Rightarrow} a_0 11+11q_4=111 \overset{20}{\Rightarrow} a_0 11+11=q_5 11 \overset{21}{\Rightarrow} a_0 11+11=1q_5 1 \overset{22}{\Rightarrow}$$

$$\overset{22}{\Rightarrow} a_0 11+11=11q_5 a_0 \overset{23}{\Rightarrow} a_0 11+11=111q_5 a_0 \overset{24}{\Rightarrow} a_0 11+11=11q_7 11 \overset{*}{\Rightarrow}$$

$$a_0 q_7 a_0 11+11=1111 \overset{31}{\Rightarrow} a_0 a_0 q_1 11+11=1111 \Rightarrow \ldots$$

Позначення $\Rightarrow$ показує, що застосовано кілька команд. Тут такі позначення використано двічі: у першому випадку — застосовувалися команди 16, 17, 19; у другому — послідовність команд 25, 25, 26, 27, 25, 28, 29, 25, 30. При функціонуванні МТ теореми з'являлися двічі. Якщо примусити МТ продовжувати функціонування, то кожний цикл функціонування закінчується появою стану q_1 (чергової теореми). Команди 1 і 2 є недетермінованими. Якщо МТ видає команду 1, то вона виконує першу продукцію постівської системи, а якщо 2 — другу. Кожного разу, з'являючись у стані q_1, зовнішня система вибирає, яку з цих двох команд треба виконати.

Побудована в останньому прикладі МТ не є єдиною, яка дає ті самі теореми, що й постівська система. Можна побудувати МТ, що видає такі самі теореми, але вже детерміновані, та яка переходить у стан q_0. Розглянемо її побудову.

Приклад. Побудувати МТ, яка виводить ті самі теореми, що й канонічна система Поста з прикладу 2 на с. 174. Відмінність цієї машини від побудованої у попередньому прикладі повинна полягати в тому, що вона не моделює роботу постівської системи, а виводить ті самі теореми.

Як початкові рядки (аксіоми) фігурують рядки типу $\alpha_1+\alpha_2=$, де α й α_2 — рядки, які додаються з 1. При складанні програми МТ врахуємо, що результат додавання двох чисел в унарній системі числення є рядком з одиниць, кількість яких дорівнює сумі одиниць складових. Переглядаючи початковий рядок зліва направо, замінимо першу одиницю нулем та заповнимо цю заміну відповідним станом. Будемо зсувати цей стан праворуч доти, доки не досягнемо правого кінця рядка. Замінимо крайній порожній символ на одиницю і, заповнивши цю заміну станом, будемо зсувати стан ліворуч доти, доки не прийдемо до нуля. Замінимо цей нуль на одиницю, стан повернемо в початковий та зсунемо його на один символ праворуч. Повторюємо описану процедуру доти, доки всі одиниці з лівої частини рядка від знака рівності не перейдуть у праву. Після цього перейдемо в заключний стан, який має знаходитися в лівому кінці рядка. Таким чином, теоремами побудованої МТ є рядки, що містять q_0. Теореми, еквівалентні теоремам постівської системи, дістанемо, якщо викреслимо всі символи a_0 і q_0. Зазначимо, що в побудованій МТ множина аксіом є нескінченною (їх задає зовнішня система), але МТ зупиняється при утворенні чергової теореми.

Таким чином, шукана МТ має:

- алфавіт $A = \{0, 1, +, =, a_0\}$, $Q = \{q_0, q_1, q_2, q_3, q_4\}$;
- аксіоми $a_1 + a_2 =$;
- програму:

1) $a_0\, q_1\, 1 \to a_0\, 0\, q_2$;
2) $1\, q_1 = \to q_3\, 1 =$;
3) $1\, q_1 + \to 1 + q_1$;
4) $0\, q_2\, 1 \to 0\, 1\, q_2$;
5) $1\, q_2\, 1 \to 1\, 1\, q_2$;
6) $1\, q_2 + \to 1 + q_2$;
7) $+\, q_2\, 1 \to +\, 1\, q_2$;
8) $1\, q_2 = \to 1 = q_2$;
9) $= q_2\, a_3 \to q_3 = 1$;
10) $= q_2\, 1 \to = 1\, q_2$;

11) $1\,q_2\,a_0 \rightarrow q_3\,1\,1$;
12) $1\,q_3\,1 \rightarrow q_3\,1\,1$;
13) $=q_3\,1 \rightarrow q_3 = 1$;
14) $1\,q_3 = \rightarrow q_3\,1 =$;
15) $+\,q_3\,1 \rightarrow q_3 + 1$;
16) $1\,q_3 + \rightarrow q_3\,1 +$;
17) $0\,q_3\,1 \rightarrow q_3\,0\,1$;
18) $a_0\,q_3\,0 \rightarrow a_0\,1\,q_1$;
19) $1\,q_3\,0 \rightarrow 1\,1\,q_1$;
20) $+\,q_3\,0 \rightarrow +\,1\,q_1$;
21) $a_0\,q_3\,1 \rightarrow q_3\,a_0\,1$;
22) $+\,q_3\,0 \rightarrow +\,1\,q_1$;
23) $a_0\,q_3\,a_0 \rightarrow a_0\,a_0\,q_0$.

Продемонструємо виведення 11 + 11 = 1111:

$$a_0q_1 11+11=a_0 \overset{1}{\Rightarrow} a_0 0q_2 1+11=a_0 \overset{4}{\Rightarrow} a_0 01q_2+11=a_0 \overset{6}{\Rightarrow}$$

$$a_0 01+q_2 11=a_0 \overset{7}{\Rightarrow} a_0 01+1q_2 1=a_0 \overset{5}{\Rightarrow} a_0 01+11q_2=a_0 \overset{8}{\Rightarrow}$$

$$a_0 01+11=q_2a_0 \overset{9}{\Rightarrow} a_0 01+11q_3=1a_0 \overset{*}{\Rightarrow} a_0q_3 01+11=1a_0 \overset{18}{\Rightarrow}$$

$$a_0 1q_1 1+11=1a_0 \overset{*}{\Rightarrow} a_0 11q_1+11=11 \overset{*}{\Rightarrow} a_0 11+1q_1 1=111 \overset{*}{\Rightarrow}$$

$$a_0 11+10=11q_3 11 \overset{*}{\Rightarrow} a_0 11+11q_1=1111 \overset{2}{\Rightarrow} a_0 11+1q_3 1=1111 \overset{*}{\Rightarrow}$$

$$\overset{*}{\Rightarrow} a_0q_3a_0 11+11=1111 \overset{23}{\Rightarrow} a_0a_0q_0 11+11=1111.$$

6.5. Використання канонічних систем Поста і машин Тьюрінга

Під час розгляду канонічної системи Поста і машини Тьюрінга не порушувалися питання ефективності функціонування їх при виведенні. Це пояснюється тим, що описані формальні системи використовуються як означення алгоритмів і мають певне значення в теорії алгоритмів, яка є основною сферою застосування цих систем. Крім того, МТ застосовується для визначення складності алгоритму. У цих застосуваннях важливим є не безпосереднє виведення, а принципова його можливість та деякі якісні характеристики.

Зазначимо, що МТ і постівська система — еквівалентні в сенсі їхніх алгоритмічних можливостей. Відома теза Тьюрінга (Поста) стверджує, що кожна функція, яка має алгоритм її обчислення, може бути реалізована відповідною МТ (постівською системою). Між МТ та постівською системою, крім зовнішньої аналогії, існує глибокий внутрішній зв'язок. Він полягає в тому, що формальну систему можна означити як у термінах МТ, так і в термінах канонічної системи. Якщо означено МТ, то канонічна система може бути виражена відповідною МТ. Загалом формальна система — це така МТ, що видає всі теореми одну за одною і ніколи не видає щось інше.

Поняття «формальний» застосовують до математичних систем з наявністю процедури виведення із зазначенням цієї процедури, причому її ефективність у них не така важлива. У формальній системі безпосередньо не зазначають ті операції, які її реалізують. МТ задає деякий набір операцій, за допомогою якого можна реалізувати формальну систему. У наведених вище прикладах канонічних систем і МТ ці поняття явно проглядаються. У канонічній системі не порушуються проблеми: як виконати підстановлення правої частини продукції замість лівої; коли зупинитися. У МТ, навпаки, цим проблемам приділяється першорядна увага.

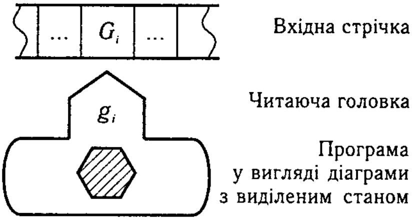


Рис. 44

У теорії формальних систем постає питання, чи завжди можна побудувати формальну систему, яка відображає певні об'єкти реального світу. На жаль, відповідь на це питання негативна. Доведено, наприклад, що не існує формальної системи, яка б вирішувала, які рядки та в яких формальних системах є теоремами. Подібні проблеми називають алгоритмічно нерозв'язуваними.

Розглянемо на прикладі МТ доведення однієї з алгоритмічно нерозв'язуваних проблем — проблему зупину МТ як застосування.

Усі доведення алгоритмічної нерозв'язуваності досить громіздкі. Щоб уникнути цього, пригадаємо означення МТ, введене в розд. 5. Зверніть увагу на подібність існуючих означень.

Отже, раніше МТ ми називали пристрій (рис. 44), що складається із вхідної стрічки, поділеної на окремі комірки і нескінченної в обидва боки; читаючої головки, яка може пересуватися вздовж стрічки в обидва боки та читати, а також змінювати вміст комірок; пристрою керування, який знаходиться в одному із станів і керує роботою машини. У кожній комірці в певний момент часу може знаходитися тільки один символ. Якщо комірка порожня, то вважаємо, що там знаходиться символ порожньої комірки (порожній символ). Ці символи складають вхідний алфавіт МТ. Символи, які вказують на стани, утворюють алфавіт внутрішніх станів, що включає заключний стан.

Машина Тьюрінга функціонує кроками. Кожний крок функціонування зводиться до виконання однієї команди. Команда МТ полягає в тому, що машина, розглядаючи поточну комірку, в якій знаходиться поточний символ, і знаходячись у поточному стані, може виконати такі дії: перейти в новий стан, змінити символ, який розглядається, зсунути читаючу головку ліворуч або праворуч від комірки, що розглядається, або залишити її на місці. Команди виконуються доти, доки МТ не вийде в заключний стан. Усі домовленості про вхідні, проміжні рядки та результати залишаються без змін. Таким чином, МТ — абстрактна система, яка складається з вхідного алфавіту, алфавіту внутрішніх станів, програми, що включає всі команди МТ, порожнього символу, символу заключного стану.

У розд. 5 було введено поняття універсальної МТ. Згідно з тезою Тьюрінга, для кожної обчислюваної функції $f(x)$ можна побудувати конкретну МТ T, яка обчислює цю функцію.

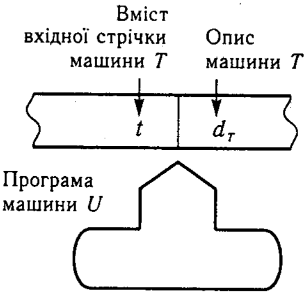


Рис. 45

При цьому на вхідній стрічці T має бути записаний ланцюжок a_x, що вказує на аргумент x. Перетворюючи a_x за допомогою команд програми МТ T, в кінці дістаємо результат $f(x)$ у вигляді ланцюжка $a_f(x)$. Якщо машину T визначено, то для кожного ланцюжка a_x можна знайти результат $a_f(x)$, примусивши функціонувати T за допомогою олівця та паперу. Для цього треба лише правильно інтерпретувати команди машини T, задані, наприклад, у вигляді згаданих п'ятірок.

Таким чином, щоб примусити функціонувати будь-яку конкретну машину T, треба мати її опис d_T у вигляді п'ятірок і вміст $a_x = t$ вхідної стрічки машини T. Процедура інтерпретації команд машини T для всіх T буде одна й та сама, тобто ця процедура може бути реалізована у вигляді сталого алгоритму. Для реалізації останнього можна запропонувати МТ U, яка називається універсальною.

На вхідній стрічці універсальної МТ U вміщують опис тієї конкретної машини T, що інтерпретується за допомогою U, а також уміст t вхідної стрічки машини T. Схематично універсальну МТ показано на рис. 45.

Для нормального функціонування машини U на її стрічці виділяється кілька зон: для зберігання поточного стану і поточного символу машини T; для зберігання опису d_T і вмісту вхідної стрічки t; для можливого розширення t у процесі роботи. Необхідно також увести маркер M, що показує, яка комірка машини T розглядається в заданий момент. Розподіл зон на стрічці машини U може бути виконаний так, як зображено на рис. 46.

Поточні параметри машини T можуть бути вміщені також у зонах для розширення, що змінить програму машини U та ефективність її функціонування, але дасть змогу зберегти для машини U таке подання, як на рис. 46.

Зазначимо, що універсальна МТ є прообразом універсальної ЕОМ, в якій алгоритм (машина T) у вигляді програми і початкові дані (t) уміщуються в оперативний запам'ятовуючий пристрій ЕОМ, який відіграє роль вхідної стрічки машини.

Сформулюємо тепер проблему зупину для МТ.

Під час роботи МТ із вхідною стрічкою може минути багато часу, доки вона закінчить процес обчислення і зупиниться. Для багатьох пар машина — стрічка зупину може взагалі не статися — «обчислення» буде продовжуватися нескінченно. З цього погляду корисно було б мати процеду-

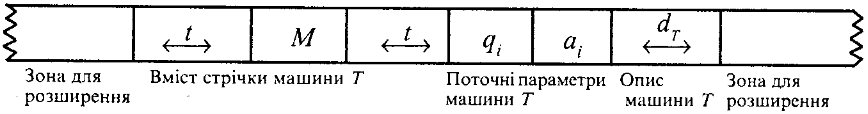


Рис. 46

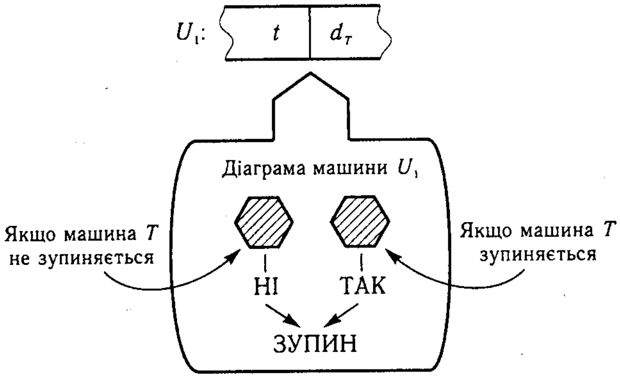


Рис. 47

ру (алгоритм), яка б за описом d_T машини T та за її вхідною стрічкою t визначала, чи зупиниться машина для всіх пар T і t. Задання такого алгоритму є проблемою зупину МТ.

Покажемо, що ця проблема є алгоритмічно нерозв'язуваною, тобто не існує єдиного алгоритму, здатного вирішити цю проблему для всіх пар T і t. Доведення нерозв'язуваності проведемо методом від супротивного, тобто спочатку припустимо, що такий алгоритм є, а потім зведемо це припущення до суперечності, що й покаже неіснування алгоритму розв'язуваності проблеми зупину МТ.

Отже, припустимо, що алгоритм, який вирішує проблему зупину МТ, існує. Цей алгоритм згідно з тезою Тьюрінга можна подати у вигляді універсальної МТ, на стрічці якої знаходиться опис d_T МТ, що досліджується, і вміст її стрічки t. Схематично універсальну МТ зображено на рис. 47.

У програмі машини U_1 є два виділених стани, показані на рис. 47. Один із них видає вказівку про те, що машина T зі стрічкою t зупиняється, а інший — що не зупиняється. Після цього машина U_1 зупиняється.

Визначимо деякі перетворення машини U_1. Оскільки вона дає результат для будь-якої пари (t, d_T), запишемо на її стрічці (d_T, d_T), що означає: вміст стрічки машини T є описом цієї машини. Цим самим вважаємо, що машина T аналізує сама себе. Тепер на стрічці машини U_1 двічі повторюється одна й та сама інформація (два рядки d_T), виражена скінченним числом символів. Уведемо U_2 — нову універсальну МТ, на стрічці якої d_T записано тільки один раз, а програма U_2 різниться від програми U_1 наявністю команд копіювання d_T у зону, де має бути записано t. Результат роботи U_2 такий самий, як і U_1. Схематичне зображення U_2 показано на рис. 48. Машина U_2 спочатку копіює опис машини T на своїй вхідній стрічці, а потім продовжує працювати, як U_1. Яка б не була внутрішня структура U_2, результатом її роботи має бути зупин із видачею результату ТАК, якщо машина T зупиняється, і НІ — в іншому випадку.

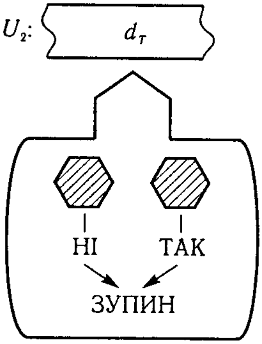


Рис. 48

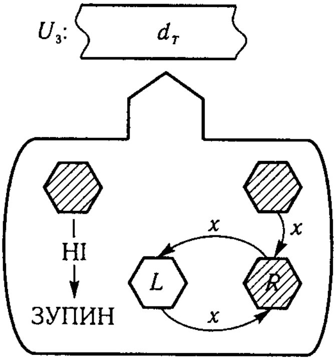


Рис. 49

Зазначимо, що введення команд копіювання в машині U_2 не вплине на кінцевий результат (зупин із видачею повідомлення), оскільки число команд для копіювання скінченного числа символів є скінченним.

Внесемо тепер невеликі зміни у структуру машини U_2. Введемо в неї два стани, які б примушували універсальну машину працювати без зупину, якщо машина, записана на її стрічці, зупиняється. Схему цієї нової машини U_3 зображено на рис. 49.

Можливість нескінченної роботи машини U_3 досягається завдяки тому, що після друку ТАК машина виходить не в заключний стан, а в новий, де «зациклюється», переходячи з одного стану в інший при читанні будь-якого символу.

Для машини U_3 є справедливим висловлення: «якщо машина U_3 зупиняється, то машина T не зупиняється». Справді, машина U_3 аналізує записану на її стрічці машину T і зупиняється тільки в разі відповіді НІ. Оскільки на стрічці машини U_3 може бути зафіксовано опис будь-якої машини, запишемо на ній опис U_3 (dU_3), тобто її опис. У цьому випадку згадане висловлення перетворюється на «якщо машина U_3 зупиняється, то машина U_3 не зупиняється», що є абсурдом. Оскільки всі проміжні перетворення універсальної МТ були конструктивними та не абсурдними, залишається тільки початкове припущення про існування алгоритму. Цим самим доведено алгоритмічну нерозв'язуваність проблеми зупину МТ.

Звичайно, в більшості математичних посібників з теорії доведень алгоритмічно нерозв'язувані проблеми доводяться за допомогою теорії нумерацій, зокрема проблема зупину МТ розглядається із залученням канторівського діагонального методу. Всі ці доведення потребують викладу теорії нумерацій і є досить громіздкими. Тому викладений метод — більш ефективний.

Існування алгоритму розв'язування деякого класу задач не означає можливість його реалізації на ЕОМ в повному обсязі. Є група задач, для яких написано та налагоджено програми, що потребують для певного обсягу вхідних даних таких машинних ресурсів, яких ще немає ні на сучасних ЕОМ, ні на ЕОМ подальших поколінь найближчого майбутнього. З погляду реалізованості алгоритмів на ЕОМ цікавою є класифікація алгоритмів за їх складністю, оскільки складність алгоритму — одна зі сфер практичного застосування МТ. Крім того, нею можна характеризувати практично корисні параметри формальних систем. Розглянемо докладніше поняття складності алгоритму.

Існують різні категорії складності алгоритму, але найуживанішими є тимчасова та ємнісна складності, що характеризують функцію збільшення часу залежно від зростання довжини вхідних даних. Таким чином, функція, яка задає зміну часу (пам'яті) від обсягу вхідних даних, називається часовою (ємнісною) складністю алгоритму. У сучасній обчислювальній практиці найчастіше застосовуються лінійна, логарифмічна, поліномна, експоненціальна складності. Нехай X позначає довжину вхідних даних. Через $O(X)$ або cX позначають лінійну, $log(X)$ — логарифмічну, $P(X)$ — поліномну, 2^{cX} — експоненціальну складності, $2^{2^{cX}}$ — суперекспоненту, $2^{2^{\cdot^{\cdot^{2}}}}$ — ітеровану експоненту. Відповідно до наведених тут функцій класифікують алгоритми за їхньою складністю. Якщо деяку задачу не можна розв'язати швидше, ніж за експоненціальний час, то вона належить до класу важких задач, які називаються класом NP повних задач.

Якщо як нижня оцінка складності виступає ітерована експонента, то задача, що характеризується такою складністю, вважається абсолютно нереальною для обчислення на ЕОМ. Нескладні обчислення [5] показують, що верхньою межею складності, яка дає надію на реалізацію задачі на ЕОМ, є складність, що характеризується формулою $2^{2^{2^{x}}}$, де $X = 16$.

Поділ алгоритмів на ті, які реалізуються та на ті, які не реалізуються за допомогою ЕОМ, є дуже важливим при створенні програм на основі математичних описів, що спричинило різні підходи до цього питання. За одним із них програмування на конкретній машині пропонується замінити програмуванням на деякій моделі, яка дає змогу різноманіття машин звести до моделі. Проте обчислення на моделі при доведенні нижніх оцінок часу і пам'яті все ще складні. Тут потрібна модель із мінімумом команд, а цю умову задовольняє модель МТ. Тому вона застосовується при встановленні нижніх оцінок складності алгоритмів.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке формальна аксіоматична теорія?
2. Які властивості поняття вивідності?
3. Як доводиться теорема про дедукцію?
4. Що таке несуперечність числення висловлень?
5. Що таке незалежність аксіом?
6. Що таке предикати?
7. Що таке квантор загальності?
8. Що таке квантор існування?
9. Що таке система Поста?

10. Що таке МТ?
11. Що означає поняття «рівносильність формул»?
12. Що означають поняття «загальнозначущість», «здійснюваність» у логіці предикатів?
13. Як формулюється теорема Черча про проблему розв'язуваності?
14. Що означає поняття «числення предикатів»?
15. Як формулюється теорема Геделя про повноту числення предикатів?
16. Які є приклади алгоритмів?
17. Що таке обчислювана функція?
18. Що таке рекурсивна функція?
19. Що таке примітивно-рекурсивна функція?
20. Що таке частково-рекурсивна функція?
21. Як формулюється теза Черча про частково-рекурсивну функцію?

ПЕРЕЛІК ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

1. Числення висловлень.
2. Формули логіки предикатів. Рівносильність формул.
3. Числення предикатів.
4. Рекурсивні функції.
5. Машина Тьюрінга.
6. Канонічна система Поста.

СПИСОК ЛІТЕРАТУРИ

1. *Нефедов В. Н., Осипова В. А.* Курс дискретной математики: — М.: Изд-во МАИ, 1992. — 264 с.
2. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженеров. — М.: Энергоатомиздат, 1988. — 480 с.
3. *Ковальски Р.* Логика в решении проблем: Пер. с англ. — М.: Наука, 1990. — 280 с.
4. *Линдон Р.* Заметки по логике: Пер. с англ. — М.: Мир, 1968. — 128 с.
5. *Джордж Ф.* Основы кибернетики: Пер. с англ. — М.: Радио и связь, 1984. — 212 с.
6. *Мельников В. Н.* Логические задачи. — Киев; Одесса: Вища шк., 1989. — 344 с.
7. *Дюбко Г. Ф.* Введение в формальные системы. — К.: УМК ВО, 1992. — 86 с.
8. *Керрол Л.* Логическая игра. — М.: Наука, 1991. — 192 с.
9. *Столл Р. Р.* Множества. Логика. Аксиоматические теории. — М.: Просвещение, 1968. — 231 с.
10. *Эббинхауз Г. Д., Янобс К., Ман Ф. К., Херми Г.* Машины Тьюринга и рекурсивные функции. — М: Мир, 1973. —264 с.

РОЗДІЛ 7

АЛГЕБРИ

7.1. Основні поняття, означення і властивості

Композиція об'єктів. У математиці та її застосуваннях велике значення мають відношення, що ставлять у відповідність парі яких-небудь об'єктів (a, b) третій об'єкт c, тобто тернарні відношення, наприклад, дії над числами. Загалом тернарні відношення можуть бути не тільки між числами, а й між об'єктами будь-якої природи. При цьому запис $a \top b = c$ (або $a \perp b = c$) означає, що a в композиції з b дає c, де $\top$ (або $\perp$) — операція; a, b — операнди; c — результат операції або композиція об'єктів a і b.

Позначимо множини операндів A та B $(a \in A, b \in B)$ і множину результатів операції C $(c \in C)$; тоді операцію (композицію) можна означити як відображення $A \times B \to C$. Її часто називають законом композиції.

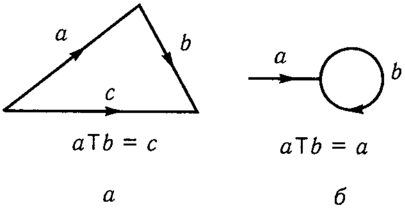
Таблиця Келі. Будь-який закон композиції $A \times B \to C$ над скінченними множинами можна задавати прямокутною матрицею (таблицею Келі):

	b_1	b_2	b_3	…
a_1	c_{11}	c_{12}	c_{13}	…
a_2	c_{21}	c_{22}	c_{23}	…
a_3	c_{31}	c_{32}	c_{33}	…
…	…	…	…	…

Тут рядки — це елементи множини A, стовпці — елементи множини B. На перетині рядка та стовпця, що відповідають (a_i, b_j), розташовується елемент $c_{ij} = a_i \top b_j$.

Закони композиції на множині. Множини A, B, C, які беруть участь в операції $A \times B \to C$, не обов'язково мають бути різними. Якщо $A = B = C = S$, то кажуть, що закон композиції означений на множині S. Розрізняють внутрішній закон композиції $S \times S \to S$ і зовнішній $Q \times S \to S$, де Q й S — різні множини $(Q \neq S)$.

У разі внутрішнього закону композиції кажуть, що множина утворює групоїд відносно операції $\top$. У разі зовнішнього закону композиції операнди $a \in Q$ називаються операторами, а Q — множиною операторів на множині S. Наприклад, множина дійсних чисел утворює групоїд відносно операцій « + » та « × », множина всіх векторів на площині — групоїд відносно операцій геометричного підсумовування.



а б

Рис. 50

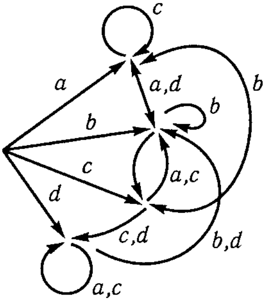


Рис. 51

Прикладами зовнішнього закону композиції можуть бути множення вектора на скаляр на множині векторів, причому операторами є скаляри — елементи з R.

Матриця і граф групоїда. Скінченний групоїд S відносно закону $\top$ визначається таблицею Келі, тобто квадратною матрицею n-го порядку, де n — число елементів групоїда.

Приклад. Таблиця Келі для групоїда $S = \{a, b, c, d\}$ відносно деякої операції $\top$ може мати такий вигляд:

	a	*b*	*c*	*d*
a	*a*	*b*	*a*	*b*
b	*a*	*b*	*c*	*a*
c	*b*	*a*	*d*	*d*
d	*d*	*b*	*d*	*b*

.

Побудова графа групоїда ґрунтується на поданні бінарного співвідношення $a \top b = c$, де дуги графа зображують елементи $a, b, c \in S$, причому операнди утворюють деякий шлях, а дуги результату операції замикають цей шлях (рис. 50, *а*). Якщо $a \top b = a$, то b зображується петлею в кінцевій вершині дуги a (рис. 50, *б*).

При побудові графа спочатку наносять дуги для всіх елементів групоїда, які виходять з однієї вершини, а потім послідовно зображають усі бінарні співвідношення.

Приклад. Для матриці з попереднього прикладу граф матиме вигляд, показаний на рис. 51.

Дуги а, b, c, d, що виходять з однієї вершини, відповідають елементам групоїда. Оскільки $a \perp a = b$, $a \perp b = c$, $a \perp c = a$ й $a \perp d = b$, з кінця дуги a проводять дуги a, b, c, d відповідно до кінцевих вершин дуг b, с, a, b. Дві паралельні дуги a і d, спрямовані до кінцевої вершини дуги b, умовно зображують однією дугою a, d. Дуга c починається та закінчується в кінцевій вершині дуги a, тобто утворює петлю. Аналогічно зображають на графі інші співвідношення, визначені матрицею групоїда.

Властивості внутрішнього закону композиції. Операції на множині S можуть мати деякі загальні властивості, що виражаються співвідношеннями між елементами S:

- комутативність $a \top b = b \top a$;
- асоціативність $a \top (b \top c) = (a \top b) \top c$;
- дистрибутивність праворуч від операції $\perp$ (відносно операції $\top$) $(a \top b) \perp c = (a \perp c) \top (b \perp c)$;
- дистрибутивність ліворуч від операції $\perp$ (відносно операції $\top$) $c \perp (a \top b) = (c \perp a) \top (c \perp b)$.

Приклади: 1. На R операції « + » і « $\times$ » є асоціативними й комунікативними, операція « $\times$ » дистрибутивною (і ліворуч, і праворуч) відносно операції « + », але « + » остання не є дистрибутивною відносно операції « $\times$ » (ні ліворуч, ні праворуч), оскільки $a+(b\times c) \neq (a+b)\times(a+c)$ та $a\times b+c \neq (a+c)\times(b+c)$. Піднесення до степеня — не асоціативне: $\left(a^b\right)^c \neq a^{(bc)}$, не комунікативне: $a^b \neq b^a$, дистрибутивне праворуч відносно операції « $\times$ »: $(ab)^c = a^c b^c$, але не дистрибутивне ліворуч відносно неї: $c^{ab} \neq c^a c^b$.

2. Операції перерізу й об'єднання множин є взаємно дистрибутивними одна відносно одної (як показано в розд. 2).

Якщо справджується внутрішній закон композиції $S^n \to S$, то функцію типу $\varphi : S^n \to S$ будемо називати *n*-арною операцією на множині *S*; *n* — арністю операції φ. Множина *S* разом із заданою на ній сукупністю операцій $\Omega = \{\varphi_1, \ldots, \varphi_m, \ldots\}$, тобто система $A = \{S; \varphi_1, \ldots, \varphi_m, \ldots\}$, називається алгеброю; *S* — основною, або множиною — носієм (чи просто носієм) алгебри A. Вектор арностей операцій алгебри є її типом, сукупність операцій Ω — сигнатурою.

Множина $F \subset S$ називається замкненою відносно *n*-арної операції φ на *F*, якщо $\varphi\left(F^n\right) \subseteq F$, тобто якщо значення φ на аргументах з *F* належать *F*. Якщо *F* —замкнена множина відносно всіх операцій $\varphi_1, \ldots, \varphi_m, \ldots$ алгебри A, то система $A' = (F, \varphi_1, \ldots, \varphi_m \ldots)$ є підалгеброю A (при цьому $\varphi_1, \ldots, \varphi_m$ розглядаються як операції на *F*).

Приклади: 1. Множина дійсних чисел з операціями додавання і множення $\{R; +, \times\}$-алгебра. Обидві операції є бінарними, тому тип цієї алгебри (2, 2). Усі скінченні підмножини *R*, крім $\{0\}$, — не замкнені відносно обох операцій. Підалгеброю цієї алгебри є, наприклад, множина раціональних чисел із тими самими операціями $\{Q; +; \times\}$.

2. Нехай $N_p = \{0, 1, 2, \ldots, p—1\}$. Означимо на *N* операції $\oplus$ («додавання за модулем *p*») та $\otimes$ («множення за модулем *p*») таким чином: $a \oplus b = c, a \otimes b = d$, де *c* і *d* — залишки від поділу на *p* чисел $a + b$ й $a \times b$ відповідно. Наприклад, якщо $p = 7$, то $N_p =$ $= \{0, 1, \ldots, 6)$, $3 \oplus 4 = 0, 3 \otimes 4 = 5, 18 \oplus 6 = 3$. Часто операції $\otimes$ та $\oplus$ означають як $a + b = c \ (\mathrm{mod}\ p)$, $a \cdot b = d \ (\mathrm{mod}\ p)$. Алгебра $\left\{N_p, \oplus, \otimes\right\}$ є підалгеброю $\{N, \oplus, \otimes\}$.

3. Алгебра $B = \left(B(U); \cup, \cap, \overline{}\right)$ називається булевою алгеброю множин над *U*, її тип (2, 2, 1). Елементами основної множини цієї алгебри є множини (підмножини *U*). Для будь-якого $U' \subset U$ $B' = \left(B(U'); \cup, \cap, \overline{}\right)$ є підалгеброю *B*. Наприклад, якщо $U = \{a\, b, c, d\}$, то основна множина алгебри *B* містить 16 елементів; алгебра $B' = \left\{B(\{a, c\}); \cup, \cap, \overline{}\right\}$ — підалгебра *B*, її основна множина містить чотири елементи.

4. Множина *F* одномісних функцій на *R*, тобто функцій $f : R \to R$, разом з операцією диференціювання є алгеброю. Елементи основної множини — функції типу $R \to R$, єдиною операцією цієї алгебри слугує диференціювання — унарна операція типу $F \to F$ (похідною функції на *R* є знову функція на *R*). Множина елементарних функцій є замкненою відносно диференціювання, оскільки похідні елементарних функцій — елементарні й, отже, утворює підалгебру цієї алгебри.

5. Розглянемо квадрат із вершинами в точках a_1, a_2, a_3, a_4, занумерованих проти руху стрілки годинника, і повороти квадрата навколо центра в тому самому напрямку, що переводять вершини у вершини. Таких поворотів є нескінченна множина: на кути $0, \pi/2, \pi, 3\pi/2, 2\pi, 5\pi/2, ...,$ однак вони задають усього чотири різних відображення множини вершин у себе, які відповідають першим чотирьом поворотам. Таким чином, маємо алгебру з основною множиною $\{a_1, a_2, a_3, a_4\}$ та чотирма унарними операціями $\alpha, \beta, \gamma, \delta$. Їх можна задати у вигляді табл. 7.1, в якій на перетині, наприклад рядка a_3 і стовпця γ, записано значення функції γ (a_3).

Таблиця 7.1

	α	β	γ	δ
a_1	a_1	a_2	a_3	a_4
a_2	a_2	a_3	a_4	a_1
a_3	a_3	a_4	a_1	a_2
a_4	a_4	a_1	a_2	a_3

Операція α, що відображає будь-який елемент у себе, називається тотожною. Вона відповідає нульовому повороту. Підалгебр у цій алгебрі немає.

7.2. Гомоморфізм та ізоморфізм алгебр

Алгебри з різними типами, очевидно, мають істотно різну будову. Якщо ж алгебри мають однаковий тип, то наявність у них подібності характеризується за допомогою понять гомоморфізму й ізоморфізму.

Нехай є дві алгебри $A = (K; \varphi_1, ..., \varphi_p)$ та $B = (M; \psi_1, ..., \psi_p)$ однакового типу. Гомоморфізмом алгебри A в алгебру B називається відображення $\Gamma : K \to M$, що задовольняє умову

$$\Gamma\left(\varphi_i\left(k_{j1}, ..., k_{jl(i)}\right)\right) = \psi_i\left(\Gamma\left(k_{j1}\right), ..., \Gamma\left(k_{jl(i)}\right)\right) \qquad (7.1)$$

для всіх $i = 1, ..., p$ (l (i) — арність операцій φ_i і ψ_i, яка у них за умовою однакова) та всіх $k_{jr} \in K$. Смисл умови (7.1) в тому, що, незалежно від того, чи здійснено спочатку операцію φ_i в A, а потім виконано відображення Γ, або спочатку зроблено відображення Γ, а потім в y здійснено відповідну операцію ψ_i, результат буде однаковим.

Ізоморфізмом алгебри A на алгебру B називається взаємно однозначний гомоморфізм. У цьому випадку існує обернене відображення $\Gamma^{-1} : M \to K$, також взаємно однозначне.

Твердження 7.1. *Якщо існує ізоморфізм A на B, то є ізоморфізм B на A.*

Нехай $\Gamma\left(k_j\right) = m_j, m_j \in M$. Тоді $k_j = \Gamma^{-1}\left(m_j\right)$. Замінимо в рівності (7.1) ліві частини на праві та застосуємо відображення Γ^{-1} до обох частин утвореної рівності. Оскільки $\Gamma^{-1}\Gamma$ є тотожним відображенням Γ^{-1} $\Gamma(a) = a$, дістанемо

$$\varphi_i\left(\Gamma^{-1}\left(m_{i1}\right), ..., \Gamma^{-1}\left(m_{il(i)}\right)\right) = \Gamma^{-1}\psi_i\left(m_{i1}, ..., m_{il(i)}\right). \qquad (7.2)$$

Рівність (7.2) — це та сама рівність (7.1) із заміною Γ на Γ^{-1}, елементів K — на елементи M і зміною місцями φ_i та ψ_i; іншими словами, Γ^{-1} — це ізоморфізм B на A. Отже, якщо існує ізоморфізм A на B, то є ізоморфізм B на A; при цьому алгебри A і B називаються ізоморфними. Потужності основних множин ізоморфних алгебр є рівними (при гомоморфізмі ця рівність може не виконуватися). Якщо $A = B$, то ізоморфізм називається ізоморфізмом на себе, або автоморфізмом; якщо $B \subset A$, то ізоморфізм є ізоморфізмом у себе.

Приклади: 1. Нехай Z_N — множина всіх цілих чисел, Z_{2N} — множина всіх парних чисел. Алгебри (Z_N; +) і (Z_{2N}; +) — ізоморфні; ізоморфізмом є відображення $\Gamma_{2n}: n \to 2n$, причому умова (7.1) тут має вигляд $2(a + b) = 2a + 2b$. Оскільки $Z_{2N} \subset Z_N$, Γ_{2n} — ізоморфізм (Z_N; +) у себе. Відображення $\Gamma_{-n}: n \to (-n)$ є для алгебри (Z_N; +) автоморфізмом; умова (7.1) має вигляд $(-a) + (-b) = -(a + b)$. Для алгебри (Z_N; ×) Γ_{-n} не є автоморфізмом, оскільки $(-a)(-b) \neq -(ab)$.

2. Розглянемо алгебри (N; +, ×) та $(N_7, \oplus, \otimes)$ (див. приклад 2 на с. 191) й означимо відображення $\Gamma_7 : N \to N$ таким чином: $\Gamma_7(n)$ дорівнює залишку від поділу n на 7; іншими словами, якщо $n = 7a + b$ $(b < 7)$, то $\Gamma_7(n) = b$. Нехай $n_1 = 7a_1 + b_1$, $n_2 = 7a_2 + b_2$. Перевіримо умову (7.1). Для додавання маємо $\Gamma_7(n_1 + n_2) = \Gamma_7(b_1 + b_2) = b_1 \oplus b_2 = \Gamma_7(n_1) \oplus \Gamma_7(n_2)$. Для множення дістаємо $\Gamma_7(n_1 n_2) = \Gamma_7(b_1 b_2) = b_1 \oplus b_2 = \Gamma_7(n_1) \oplus \Gamma_7(n_2)$. Таким чином, умова (7.1) здійснена і Γ_7 — гомоморфізм. Очевидно, Γ_7 не є ізоморфізмом, оскільки немає взаємної однозначності алгебр.

Цей приклад показує, що можливий гомоморфізм нескінченної алгебри (тобто алгебри з нескінченною основною множиною) в скінченну алгебру. При цьому N розбивається на сім класів еквівалентності за відношенням E_7: aE_7b тоді й тільки тоді, коли $\Gamma_7(a) = \Gamma_7(b)$.

Приклади: 1. Ізоморфізмом між алгебрами (R_+, ×) і (R, +), де R_+ — додатна підмножина R, є відображення $a \to \log a$. Умова (7.1) має вигляд рівності $\log ab = \log a + \log b$.

2. Розглянемо алгебри (K, φ) та (M, ψ), де $K = \{a_1, a_2, a_3, a_4\}$; $M = \{\alpha, \beta, \gamma, \delta\}$, а бінарні операції φ і ψ задано відповідно табл. 7.2, *а* та *б*.

Таблиця 7.2

	a_1	a_2	a_3	a_4
a_1	a_3	a_2	a_2	a_1
a_2	a_1	a_4	a_4	a_2
a_3	a_4	a_2	a_2	a_1
a_4	a_1	a_1	a_3	a_3

а

	α	β	γ	δ
α	δ	δ	γ	α
β	α	α	δ	γ
γ	α	α	β	γ
δ	γ	β	γ	β

б

Відображення $\Gamma : a_1 \to \gamma, a_2 \to \alpha, a_3 \to \beta, a_4 \to \delta$ є ізоморфізмом.

Перевірка умови (7.1) полягає ось у чому. В комірках (у внутрішній частині) табл. 7.2, *а* замінюємо елементи множини K на елементи множини M відповідно до Γ і дістаємо ліву частину (7.1), тобто таблицю функції $\Gamma_\varphi(a_i, a_j)$; у зовнішній частині табл. 7.2, *б* виконуємо обернену заміну й одержуємо праву частину (7.1); порівнянням утворених двох таблиць переконуємося, що вони задають одну й ту саму функцію. Справді, досить у табл. 7.2, *б* перейменувати всі елементи множини K на елементи множини M і порівняти утворену таблицю з табл. 7.2, *б*.

Зазначимо, що можна було б розглядати алгебри (K, φ) та (K, ψ), де в табл. 7.2, *б* замінено α на a_1, β на a_2, γ на a_3, δ на a_4. Тоді відображення $\Gamma : a_1 \to a_3, a_2 \to a_1, a_3 \to a_2, a_4 \to a_4$ також буде ізоморфізмом.

3. Булеві алгебри (див. приклад 3 на с. 191), утворені двома різними множинами U, U' однакової потужності, є ізоморфними: операції у них — однакові, а відображенням $Г$ може бути будь-яка взаємно однозначна відповідність між U й U'.

Легко показати, що відношення ізоморфізму є відношенням еквівалентності на множині алгебри. Справді, рефлексивність його очевидна, симетричність випливає з існування оберненого ізоморфізму, а транзитивність установлюється таким чином: якщо $Г_1$ — ізоморфізм A на B, $Г_2$ — ізоморфізм B на C, то ізоморфізмом A на C буде композиція $Г_1$ і $Г_2$. Класами еквівалентності в розбитті за відношенням ізоморфізму є класи ізоморфних між собою алгебр.

Поняття ізоморфізму є одним із найважливіших понять у математиці. Його сутність, як випливає з останніх двох прикладів, можна виразити таким чином: якщо алгебри A та B — ізоморфні, то елементи й операції B можна перейменувати так, що B збігатиметься з A. З умови (7.1) ізоморфізму виходить, що будь-яке еквівалентне співвідношення в алгебрі A зберігається в будь-якій ізоморфній їй алгебрі A'. Це дає змогу, одержавши такі співвідношення в алгебрі A, автоматично поширити їх на всю алгебру, ізоморфну A. Відомий у математиці вираз «розглядати об'єкти з точністю до ізоморфізму» означає, що розглядаються тільки ті властивості об'єктів, які зберігаються при ізоморфізмі, тобто є загальними для всіх ізоморфних об'єктів. Зокрема, ізоморфізм зберігає такі властивості: асоціативність, комутативність і дистрибутивність операцій.

7.3. Типи алгебр

Означення 7.1. *Якщо в множині $F \subset S$ композиція будь-яких двох елементів з F також належить F, то F називається замкненою множиною відносно заданого закону композиції (або відносно заданої операції). Наприклад, множина Z — замкнена відносно « + » та « × ». Елемент a називається регулярним, якщо з того, що $a \top x = a \top y$ й $x \top a = y \top a$ випливає $x = y$. Таким чином, будь-яке число — регулярне відносно « + », а для множення регулярним є всяке число, крім нуля.*

Означення 7.2. *Нейтральним елементом $e \in S$ називається такий елемент, коли для всіх елементів $x \in S$ справджується рівність $e \top x = x \top e = x$. Нейтральний елемент називають ще одиницею.*

Твердження 7.2. *Якщо нейтральний елемент існує, то він — єдиний і регулярний.*

Доведемо єдиність від супротивного: припустимо, що нейтральний елемент — не єдиний. Нехай існують два нейтральних елементи $e_1 \in S$ та $e_2 \in S$, $e_1 \neq e_2$. Оскільки e_1 — нейтральний елемент, $e_1 \top e_2 = e_2 \top e_1 = e_2$, а оскільки e_2 — теж нейтральний елемент, $e_2 \top e_1 = e_1 \top e_2 = e_1$. Одержана суперечність доводить твердження.

Доведемо регулярність від супротивного: припустимо, що нейтральний елемент — не регулярний, тобто нехай $e \top x = e \top y$ й $x \top e = y \top e$, але

$x \neq y$. Оскільки e — нейтральний елемент, $e \top x = x \top e = x$, але $e \top x = e \top y$ за умовою, а $e \top y = y \top e = y$. Одержана суперечність доводить твердження.

Приклади: 1. На множині R: 0 — нейтральний елемент відносно «+», а 1—нейтральний елемент відносно «×». У $B(U)\varnothing$ —нейтральний елемент відносно об'єднання, а U — нейтральний елемент відносно перерізу множин.

2. Розглянемо множину всіх квадратних матриць n-го порядку з числовими елементами. Очевидно, нульова матриця є нейтральним елементом відносно додавання, а одинична матриця — нейтральним елементом відносно множення.

Якщо множина містить нейтральний елемент e відносно операції $\top$, то елемент b називається симетричним (оберненим, протилежним) елементу a, коли $a \top b = b \top a = e$. При цьому a називається симетрованим елементом і b позначається через $\overline{a}$, тобто $b = \overline{a}$.

Твердження 7.3. *Якщо елемент $\overline{a}$, симетричний елементу a, існує, то він — єдиний та регулярний.*

Пропонуємо читачеві довести це твердження самостійно.

Приклади: 1. На множині R для операції «+» будь-якому елементу x симетричним є елемент —x, а для операції «×» — елемент x^{-1}.

2. Симетричними елементами на множині квадратних матриць n-го порядку відносно операції «множення матриць» є взаємно обернені матриці.

Означення 7.3. *Множина, в якій всякий елемент має симетричний, називається симетрованою.*

Якщо множина наділена двома законами композиції, то найчастіше перший з них $\top$ вважається адитивним і записується « + », а другий $\perp$ — мультиплікативним і записується «×» або «.», де «+» та «.» не обов'язково позначають додавання і множення.

Другий закон композиції (якщо він означений) є дистрибутивним ліворуч та праворуч відносно першого закону.

Означаючи на деякій множині S один або два закони композиції і наділяючи їх певними властивостями, а також визначаючи структуру множини відносно законів композиції (наявність нейтрального елемента та симетрованість множини), дістаємо різні алгебри.

Наведемо в табл. 7.3 ті, які найчастіше вживаються, і вкажемо (*) властивості та відповідні елементи першого (адитивного) і другого (мультиплікативного) законів алгебри.

Означення 7.4. *Півгрупою називається алгебра з однією асоціативною бінарною операцією.*

Ця операція є додаванням, тому результат її застосування до елементів a та b записується як $a + b$. Загалом $a + b \neq b + a$. Якщо ж додавання — комутативне, то півгрупа називається комутативною, або абелевою. Півгрупа з одиницею є моноїдом. Одиниця у півгрупі — завжди єдина. Справді, якщо існують дві одиниці e_1 і e_2, то $e_1 + e_2 = e_1$ й $e_1 + e_2 = e_2$; отже, $e_1 = e_2$.

Композиція відображень є асоціативною операцією. Тому кожна множина перетворень (відображень деякої множини в себе), замкнена відносно композиції, є півгрупою.

Таблиця 7.3

Тип алгебри	Перший закон (адитивний)				Другий закон (мультиплікативний)			
	Властивості		Елементи		Властивості		Елементи	
	Асоціативність	Комутативність	Нейтральний	Симетричний	Асоціативність	Комутативність	Нейтральний	Симетричний
Півгрупа (моноїд)	*							
Абелева (комутативна) півгрупа	*	*						
Півгрупа з нулем (одиницею)	*		*					
Абелева півгрупа з нулем (одиницею)	*	*	*					
Група	*		*	*				
Абелева (комутативна) група	*	*	*	*				
Асоціативне кільце	*	*	*	*	*			
Абелеве (комутативне) кільце	*	*	*	*	*	*		
Кільце з одиницею (унітарне кільце)	*	*	*	*	*		*	
Абелеве кільце з одиницею	*	*	*	*	*	*	*	
Тіло	*	*	*	*	*		*	*
Поле (комутативне тіло)	*	*	*	*	*	*	*	*

Приклад. Нехай на множині $\{1, 2, 3\}$ задано перетворення $\alpha = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 2 \end{pmatrix}$ і $\beta = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 2 \end{pmatrix}$. Їх композиції мають вигляд $\alpha\beta = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix}$ та $\beta\alpha = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 2 \end{pmatrix}$, тобто не збігаються з α і β. Тому множина $\{\alpha, \beta\}$ — не замкнена відносно композиції та не утворює півгрупи. Проте якщо до неї додати перетворення $\gamma = \beta^2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \end{pmatrix}$, $\delta = \alpha\beta$ і $\zeta = \beta\alpha$, то можна переконатися, що множина $\Gamma = \{\alpha, \beta, \gamma, \delta, \zeta\}$ разом з операцією композиції утворює півгрупу. Таблиця Келі цієї півгрупи має такий вигляд:

	α	β	γ	δ	ζ
α	α	δ	ζ	δ	ζ
β	ζ	γ	β	δ	ζ
γ	ζ	β	γ	δ	ζ
δ	ζ	ζ	δ	δ	ζ
ζ	ζ	ζ	ζ	δ	ζ

Якщо ж до Γ додати тотожне відображення $\varepsilon = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$, то дістанемо півгрупу з одиницею.

Теорема 7.1. *Будь-яка півгрупа з одиницею є ізоморфною деякій півгрупі перетворень.*

Доведення. Нехай задано півгрупу з множиною $M = \{e, a_1, a_2, ...\}$. Кожному елементу a_i півгрупи поставимо у відповідність перетворення f_i множини M таким чином: $f_i(x) = x + a_i$ для всіх $x \in M$. Тоді сумі $a_i + a_j$ відповідатиме перетворення $f_{ij}(x) = x + a_i + a_j = f_i(x)\, a_j = f_j(f_i(x))$, тобто композиція перетворень f_i та f_j; отже, умову (7.1) гомоморфізму виконано. Крім того, різним елементам відповідають різні відображення, оскільки $f_i(e) = a_i$, $f_j(e) = a_j$, й, отже, $f_i \neq f_j$. Таким чином, відповідність $a_i \to f_i(x)$ є ізоморфізмом.

Якщо будь-який елемент півгрупи $P = (M; \text{o})$ можна відобразити як результат застосування операції о до деякого числа елементів множини $M_0 \subset M$, то множина M_0 називається породною множиною або системою твірних півгрупи P, а її елементи — твірними. У прикладі 7.19 твірними є α та β, оскільки $\gamma = \beta^2, \delta = \alpha\beta, \zeta = \beta\alpha$. У півгрупі $(N, \times)$ породною множиною слугує нескінченна множина простих чисел. Якщо півгрупа має тільки одну твірну, то всі елементи є мірами цієї твірної. Така півгрупа називається циклічною. Нею є, наприклад, півгрупа $(N; +)$, оскільки всі натуральні числа — це суми деякої кількості одиниць.

Нехай півгрупа P має скінченну систему твірних $\{a_1, ..., a_n\}$. Якщо позначення операції опустити (як це робиться для множення), то всі елементи P можна розглядати як слова в алфавіті $\{a_1, ..., a_n\}$. Деякі слова можуть виявитися однаковими як елементи. У прикладі півгрупи перетворень виконуються, наприклад, рівності $\beta^3 = \beta$, $\beta\alpha = \alpha\beta^2$. У комутативній півгрупі для будь-яких елементів a, b справджується рівність $ab = ba$. Такі рівності називаються визначальними співвідношеннями. Якщо ж у півгрупі їх немає, тобто будь-які два різних слова є різними елементами півгрупи, то остання називається вільною.

Усяку півгрупу можна утворити з вільної півгрупи введенням деяких визначальних співвідношень. Елементи такої півгрупи — це слова в алфавіті твірних, причому деякі слова однакові (тобто задають один і той самий елемент) завдяки визначальним співвідношенням. Відношення рівності слів є відношенням еквівалентності. З будь-якого слова, використовуючи визначальні співвідношення, легко утворити різні еквівалентні йому слова. Набагато складнішою є проблема: для двох заданих слів з'ясувати, чи можна здобути одне з іншого, застосовуючи визначальні співвідношення. Її дослідження вплинуло значним чином на теорію алгоритмів.

Означення 7.5. *Групою називається півгрупа з одиницею, в якій для кожного елемента а існує елемент* $\overline{a}$ *, який називається оберненим (протилежним, симетричним) до а і який задовольняє умову* $a + \overline{a} = \overline{a} + a = e$ *. Число елементів групи називається її порядком. Група, в якій операція —комутативна, є комутативною, або абелевою. Група, всі елементи якої є степенями одного елемента а, називається циклічною.*

Циклічна група — завжди абелева. Для таких груп часто застосовується адитивний запис: операція позначається як додавання, а одиниця — як 0.

Приклади: 1. Множина раціональних чисел, що не містить нуля, з операцією множення є абелевою групою. Оберненим до елемента a буде елемент $1/a$.

2. Множина цілих чисел Z з операцією додавання є абелевою циклічною групою. Роль одиниці тут відіграє 0, оберненим до елемента a буде елемент $-a$.

3. Множина невироджених квадратних матриць порядку n з операцією множення є некомутативною групою.

4. Множина $\{0, 1, 2, 3, 4\}$ з операцією «додавання за mod 5» — скінченна абелева циклічна група. Її одиницею є 0. У цій групі $\bar{3} = 2, \bar{1} = 4$.

5. Взаємно однозначне відображення множини $N_n = \{1, 2, ..., n\}$ на себе називається підстановкою n чисел (n-го степеня). Звичайно прийнято записувати підстановку двома рядками, взятими в дужки. Перший рядок — аргументи (перші координати), другий рядок — образи (другі координати). Наприклад, взаємно однозначна відповідність чотирьох чисел, задана множиною $\{(1, 2), (2, 4), (3, 3), (4, 1)\}$, запишеться як підстановка a четвертого степеня:

$$a = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}.$$

Оскільки байдуже, в якому порядку записано впорядковані пари відображення, одна й та сама підстановка допускає різні подання, наприклад,

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 1 & 3 & 4 & 2 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 3 & 1 \\ 1 & 4 & 3 & 2 \end{pmatrix} \cdots$$

Це означає, що асоціативність виконується.

Кожний рядок у запису підстановки n-го степеня містить n різних чисел, розташованих у певному порядку, тобто є деякою перестановкою n чисел 1, 2, 3, ..., n.

Якщо позначити i-ті елементи перестановки через α_i і β_i ($i = 1, 2, ..., n$); $\alpha_i, \beta_i \in N$; то

$$\alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \beta_1 & \beta_2 & \dots & \beta_n \end{pmatrix}.$$

Оскільки число всіх перестановок з n чисел дорівнює $n!$, число всіх різних підстановок n-го степеня, як і число всіляких способів запису будь-якої з них, дорівнює $n!$

Тотожна підстановка $e_n = \begin{pmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{pmatrix}$ відіграє роль нейтрального елемента.

Якщо в підстановці a поміняти місцями її рядки, то дістанемо підстановку, симетричну a, наприклад

$$a = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}; \ \bar{a} = \begin{pmatrix} 2 & 4 & 3 & 1 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

Композицією підстановок n-го степеня a та b називається підстановка $c = ab$ n-го степеня, що є результатом послідовного виконання спочатку операції a, потім b:

$$c = ab = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 3 & 1 \end{pmatrix}.$$

Очевидно, якщо a — підстановка n-го степеня, то $ae_n = e_na = a$; $\bar{a} = \bar{a}a = e_n$.

Інверсію утворюють два числа в перестановці, коли менше з них розташовується правіше від більшого. Підстановка називається парною, якщо загальне число інверсій в її рядках (перестановках) — парне, і непарною в іншому випадку.

Для кожного з чисел визначається кількість менших чисел правіше від нього, а одержані результати додаються.

Наприклад $\begin{pmatrix} 4 & 2 & 5 & 1 & 3 & 6 \\ 5 & 3 & 1 & 4 & 2 & 6 \end{pmatrix}$ — непарна підстановка, оскільки у верхньому рядку $3 + 1 + 2 + 0 + 0 = 6$, у нижньому $4 + 2 + 0 + 1 + 0 + 0 = 7$, а загальне число інверсій $6 + 7 = 13$.

Розкладання підстановки на цикли. Усяку підстановку можна розкласти на добуток циклів, множина елементів яких попарно не перерізається. Цикл — це така підстановка

$$\begin{pmatrix} \alpha_1 & \alpha_2 & ..., & \alpha_{k-1} & \alpha_k & \alpha_{k+1} & ... & \alpha_n \\ \alpha_2 & \alpha_3 & ..., & \alpha_k & \alpha_1 & \alpha_{k+1} & ... & \alpha_n \end{pmatrix} = (\alpha_1, \alpha_2, ..., \alpha_k),$$

що переводить α_1 в α_2, α_2 в $\alpha_3, ..., \alpha_{k-1}$ в α_k, α_k в α_1, а решту елементів — у самих себе. Скорочений запис циклу $(\alpha_1, \alpha_2, ..., \alpha_k)$ зводиться до переліку множини елементів, які циклічно переходять один в одного, а кількість цих елементів k позначає довжину (порядок) циклу.

Так,

$$\begin{pmatrix} 2 & 4 & 5 & 1 & 3 & 6 \\ 5 & 3 & 1 & 4 & 2 & 6 \end{pmatrix} = (1,4,5)(1,3)(6).$$

Цикл завдовжки 1 є тотожною підстановкою і часто не записується. Підстановка, всі n елементів якої утворюють цикл, називається циклічною. Цикл завдовжки 2 є транспозицією (перестановкою тільки двох елементів). Усяка підстановка зображується добутком транспозицій, наприклад

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 3 & 4 & 5 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 & 4 & 5 \\ 2 & 5 & 3 & 4 & 1 \end{pmatrix} \begin{pmatrix} 2 & 5 & 3 & 4 & 1 \\ 2 & 5 & 4 & 3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 5 & 4 & 3 & 1 \\ 2 & 5 & 4 & 1 & 3 \end{pmatrix} =$$

$$= (1,2)\ (1,5)\ (3,4)\ (1,3).$$

Подібне розкладання множини містить цикли із загальними елементами, і при цьому воно не є єдиним. Водночас розкладання підстановки на незалежні цикли (без загальних елементів) завжди можна здійснити тільки єдиним способом. Різниця між числом усіх елементів підстановки n та кількістю її циклів m (з урахуванням циклів завдовжки 1) називається декрементом підстановки $d = n — m$. Парність підстановки збігається з парністю її декремента.

Наочне уявлення про підстановки дають їхні графи, побудовані на множині n вершин, що відповідають числам 1, 2, 3, ..., n. На рис. 52 ліворуч показано графи підстановок a (суцільними лініями) і b (штриховими), а праворуч — граф композиції.

Покажемо, що підстановки утворюють групу з операцією — композицією підстановок.

Композиція (добуток) підстановок a та b — це композиція двох взаємно однозначних відображень множини об'єктів N на себе, тобто $N^a \to N^b \to N$, завдяки чому дістаємо деяку підстановку ab.

Нейтральним елементом у групі підстановок є тотожна

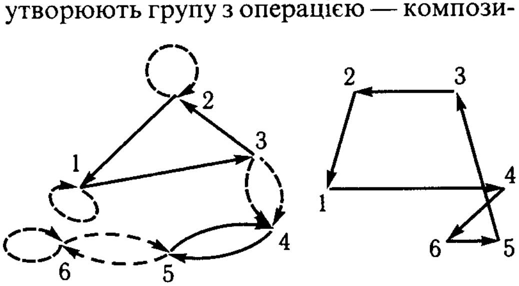


Рис. 52

підстановка e, а симетричним елементом для $\forall$ підстановки a — симетрична підстановка $\overline{a}$. Оскільки композиція підстановок не підпорядковується комутативному закону $(ab \neq ba)$, група підстановок n-го степеня при $n \geq 3$ не є комутативною.

Якщо множина N скінченна і містить n чисел, то множина S усіх підстановок n-го степеня — також скінченна й містить $n!$ елементів. Така група називається симетричною групою порядку n. Підгрупи симетричних груп є групами підстановок.

У будь-якій скінченній групі її операція може бути задана таблицею Келі. Для груп ця таблиця має важливу особливість: будь-який її стовпець містить усі елементи групи. Справді, якщо стовпець a_i не містить якогонебудь елемента, то деякий інший елемент a_j в ньому має зустрітися двічі, скажімо, в k-му й l-му рядках. Однак тоді $a_k a_i = a_j,$ $a_l a_i = a_j$ і, отже, $a_k a_i = a_l a_i$. Якщо помножимо обидві частини цієї рівності на $\overline{a}_i$, дістанемо $a_k = a_l$, що неправильно. Таким чином, i-й стовпець таблиці Келі містить усі елементи групи, тобто множення на a_i є підстановкою на множині елементів групи. Перевіривши, що ця відповідність є ізоморфізмом, маємо таку теорему Келі:

Теорема 7.2. *Будь-яка скінченна група є ізоморфною групі підстановок на множині її елементів.*

Із порівняння теорем про зв'язок півгруп з перетвореннями (теорема 7.1) і груп з підстановками випливає, що група — це півгрупа взаємно однозначних перетворень, причому саме взаємна однозначність гарантує наявність оберненого перетворення. Можна сказати, що в групі при будь-якому числі множень не втрачається інформація про початковий елемент: якщо відомо, на що множили, завжди можна дізнатися, що множили.

Для півгрупи це є правильним не завжди. Використовуючи термінологію дискретних систем, те саме можна сказати інакше. Нехай існує дискретна система із скінченним числом станів $S = \{s_1, ..., s_n\}$, на вхід якої може бути подано вхідний вплив із множини $\{x_1, ..., x_m\}$. Усякий вхідний вплив однозначно переводить стан системи в деякий інший стан і, отже, є перетворенням множини S. Послідовності впливів — це композиції перетворень, тому множина всіх послідовностей є півгрупою з твірними $\{x_1,..., x_m\}$. Якщо така півгрупа виявляється групою, то за будь-якою вхідною послідовністю і заключним станом системи можна однозначно визначити її початковий стан.

Означення 7.6. *Усяка частина алгебричної системи, що знову є системою відносно тих самих законів, називається підсистемою.*

Зокрема, кожна підгрупа R містить нейтральний елемент групи. Підкільце утворює підгрупу адитивної групи кільця і є замкненим відносно мультиплікативного закону.

До них належать одинична група, що містить тільки нейтральний елемент (тотожну підстановку), і сама симетрична група.

Крім цих тривіальних груп, є підгрупи симетричної групи, які є групами підстановок.

Група — це множина всіх парних підстановок (знакозмінна група). Множина всіх підстановок, що переводять який-небудь елемент у себе, також є групою.

Приклади: 1. Розглянемо множину многочленів (поліномів) змінної x над числовим полем $P f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$, де n — додатне ціле; $a_0, a_1, a_2, \ldots a_n \in P$. Очевидно, ця множина утворює кільце з операціями додавання та множення многочленів.

2. Непорожня система S множин утворює кільце множин, якщо для будь-яких множин A і B цієї системи $A \oplus B \in S$ й $A \cap B \in S$. Тут означено два внутрішніх закони композиції: диз'юнктивну суму та переріз множин. Перший закон — асоціативний. Нейтральним елементом відносно $\oplus$ слугує $\varnothing$, тому що $\forall A (A + \varnothing = A)$ є симетричним для $\forall A$ є $\overline{A}$, оскільки $A \oplus \overline{A} = \varnothing$. Другий закон — також асоціативний, тому що $A \cap (B \cap C) = (A \cap B) \cap C$, і дистрибутивний відносно першого, тобто $A \cap (B \oplus C) = (A \cap B) \oplus (A \cap C)$. Нейтральний елемент (одиниця) U відносно другого закону $(\cap)$ означається співвідношенням $A \cap U = A$, звідки випливає, що U є не чим іншим, як максимальною множиною цієї системи, яка містить решту множин, що входять у систему (універсум U). Якщо такий елемент існує, то маємо унітарне кільце (тобто кільце з одиницею). Так, множина всіх підмножин довільної множини T утворює унітарне кільце ($U = T$).

3. Множина всіх обмежених відрізків числової прямої, утворює кільце без одиниці, оскільки не існує обмеженого відрізка, який був би одиницею кільця, тобто містив усі обмежені відрізки прямої.

4. Комплексне число $z = a + bi$, де $a = \mathrm{Re}(z)$ — дійсна частина, а $b = \mathrm{Im}(z)$ — уявна частина, можна розглядати як пару (a, b) двох дійсних чисел $(a, b \in R)$. Тут означено два внутрішніх закони композиції: додавання $z_1 + z_2 = (a_1 + a_2, b_1 + b_2)$ і множення $z_1 z_2 = (a_1 a_2 — b_1 b_2, a_1 b_2 + a_2 b_1)$. Два комплексних числа — рівні: $z_1 = z_2$, якщо $a_1 = a_2$, $b_1 = b_2$.

Дії над комплексними числами у формі $z = a + bi$ можна виконати, як із дійсними числами, замінюючи i^2 на (-1). Комплексно-спряженим числом до числа $z = a + bi$ є число $z^* = a - bi$. Множина комплексних чисел складає комутативну групу відносно додавання. Справді, додавання — комутативне й асоціативне. Нейтральним елементом слугує нуль (0, 0), а симетричним числу $z = (a, b)$ є число $-z = (-a, -b)$.

Відносно множення нейтральним елементом є одиниця (1, 0), і кожне відмінне від нуля комплексне число $z = a + bi$ має симетричне (обернене) $\frac{1}{z} = \frac{1}{|z|^2}(a - bi) = z^{-1}$, де $|z| = \sqrt{a^2 + b^2}$ — модуль комплексного числа.

Оскільки множення — дистрибутивне відносно додавання, множина комплексних чисел є полем.

Поле Галуа. Добре відомі поля цілого і дійсних чисел — це нескінченні множини. Скінченне поле називають полем Галуа.

7.4. Алгебричні системи

Досі розглядалися алгебри, тобто множини, на яких задано операції. Множини, де крім операцій задано відношення, називаються алгебричними системами [2]. Таким чином, алгебри можна вважати окремим випадком алгебричних систем (в яких множина відношень — порожня). Іншим окремим випадком алгебричних систем є моделі множин, на яких задано тільки відношення. Поняття ізоморфізму для алгебричних систем уводиться аналогічно тому, як це було зроблено вище для алгебри, з тією різницею, що до умови (7.1) збереження операцій додається умова збереження відношень при ізоморфізмі.

Розглянемо ґратку як приклад алгебричної системи, що найчастіше зустрічається в теоретичній алгебрі та її застосуваннях.

Означення 7.7. *Ґраткою називається частково впорядкована множина, в якій для будь-яких двох елементів a і b існують:* $a \perp b = c$ *— така нижня грань a та b, що будь-яка інша нижня грань a і b менша за c;* $a \top b = d$ *— така верхня грань a та b, що будь-яка інша верхня грань a і b більша за d.*

Таким чином, ґратка — це алгебрична система $\{M; \leq; \perp, \top\}$ з одним бінарним відношенням та двома бінарними операціями. Зазначимо, що операції $\perp$ і $\top$ тут є абстрактними операціями алгебричної системи.

Уведені таким чином операції $\perp$ та $\top$ — асоціативні (що легко довести), тому можна застосовувати їх до будь-яких підмножин ґраток.

Приклади: 1. Будь-яку абсолютно впорядковану множину (наприклад, множину цілих чисел) можна перетворити на ґратку, означивши для будь-яких $a, b \in M$ $a \perp b = \max(a, b), a \top b = \min(a, b)$.

2. Означимо на N відношення часткового порядку $<$ таким чином: $a < b$, якщо a ділить b. Тоді $a \perp b$ — найменше спільне кратне a і b, $a \top b$ — найбільший спільний дільник a, b. Наприклад, $9 \perp 12 = 36, 9 \top 12 = 3, 5 \top 7 = 1, 5 \perp 7 = 35$.

3. Система всіх підмножин будь-якої множини A — частково впорядкована за включенням $M_i \subseteq M_j$ (доведено вище). Ця система є ґраткою, елементами якої є множини, а операціями — звичайні теоретико-множинні операції об'єднання та перерізу.

4. Розглянемо множину B_n двійкових векторів завдовжки n, частково впорядковану так: vAw, якщо у векторі w одиниці знаходяться на всіх тих місцях, на яких вони є у v (і, можливо, ще на деяких). Наприклад, $(010)A(011)$, а (010) і (100) — незрівнянні. Множина B_n, упорядкована таким чином, є ґраткою; в ній $v \perp w$ — вектор, в якому одиниці знаходяться на тих (і тільки тих) місцях, де є одиниці або у v, або у w, а $v \top w$ — вектор, в якому одиниці знаходяться на тих і тільки тих місцях, де є одиниці й у v, й у w. Наприклад, $(010) \perp (100) = (110), (010) \top (100) = (000)$.

При доведенні теореми про потужність множини всіх підмножин скінченної множини була встановлена взаємно однозначна відповідність між множиною B_n і системою всіх підмножин будь-якої множини A потужністю n. Легко перевірити, що ця відповідність є ізоморфізмом відповідних ґраток. Таким чином, ґратки, описані у двох останніх прикладах, — ізоморфні.

На множині $P = \{\pi_1, ..., \pi_m\}$ усіх можливих розбиттів скінченної множини M ґратки будуються так. Частковий порядок $\pi_i < \pi_j$, якщо будь-які два елементи M, які знаходяться в одному блоці розбиття π_i, є в одному блоці розбиття π_j, тобто якщо будь-який блок π_i є підмножиною деякого блока π_j. Наприклад, для $M = \{a, b, c, d, e, f)$ розбиття π_1 $\{ab, c, de, f\}$ менше за розбиття $\pi_2 = \{abf, cde\}$.

Мінімальним є розбиття $\pi_{\min} = \{a, b, c, d, e, f\}$, в якому кожний блок складається з одного елемента; максимальним — розбиття $\pi_{\max} = \{abcdef\}$,

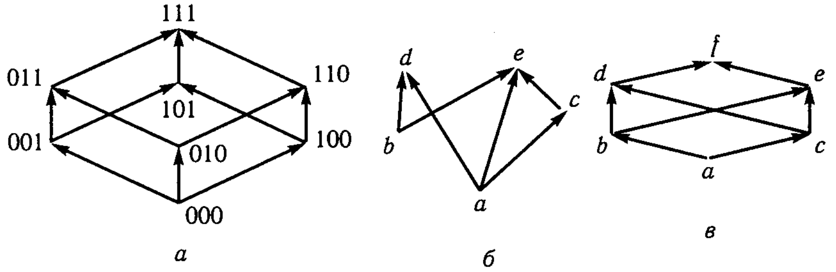


Рис. 53

що складається з одного блока. Перерізи π_i і π_j — це розбиття π_k, в якому два елементи містяться в одному блоці тоді й тільки тоді, коли вони є в одному блоці і в π_i, і в π_j. Наприклад, для $\pi_3 = \{a, bc, de, f\}$ маємо $\pi_1 \cap \pi_3 = \{a, b, c, d, e, f\}$. Об'єднання π_i та π_j — це розбиття π_1, в якому два елементи містяться в одному блоці тоді й тільки тоді, коли вони є в одному блоці в π_i або в π_j. Наприклад, $\pi_1 \cup \pi_3 = \{abc, de, f\}$.

Скінченну впорядковану множину можна зобразити діаграмою, в якій елементам відповідають точки; з точки a йде стрілка в точку b, якщо $a >$ $> b$, і немає такої точки c, що $a < c < b$. Наприклад, ґратка B_3 зображається діаграмою, показаною на рис. 53, *а*.

На мові діаграм добре ілюструються всі основні поняття, пов'язані з ґраткою: $a \leq b$ тоді й тільки тоді, коли існує шлях із стрілок, що йде з точок a і b; верхня грань a та b — це елемент, до якого є шлях і з a, і з b; нижня грань a та b — це елемент, з якого є шлях і в a, і в b.

Упорядкована множина не є ґраткою у двох випадках:

- коли які-небудь два елементи не мають верхньої або нижньої грані (на рис. 53, *б* елементи d та e, c і d не мають верхньої грані, елементи b та a — нижньої);
- коли для деякої пари елементів найменша верхня (або найбільша нижня) грань не є єдиною (на рис. 53, *в* всі елементи мають верхні й нижні грані, однак у b і c є дві найменші та незрівнянні верхні грані, а d і e мають дві найбільші нижні грані, через що зображена на цьому рисунку множина не є ґраткою).

Конкретний приклад першого випадку можна одержати з ґратки вилученням деяких її елементів. На рис. 53, *а* після вилучення елемента 101 ґратка B_3 залишається ґраткою, а після вилучення елемента 111 — ні. Вилученням елементів з ґратки можна дістати також приклад другого випадку: якщо у ґратці B_4 (див. приклад 4 на с. 202) вилучити всі елементи, крім 0000, 0100, 0010, 0111, 1110, 1111, то діаграма для елементів, які залишилися, в точності збігається з рис. 53, *в*.

Ґратка, в якій операції $\perp$ і $\top$ є можливими для будь-якої підмножини її елементів, називається повною. У зв'язку з зазначеною вище асоціативністю перерізу й об'єднання скінченні ґратки — завжди повні. Об'єднання всіх елементів повної структури — це максимальний елемент ґратки, який називається одиницею структури. Переріз усіх елементів повних ґраток — це мінімальний елемент ґратки, тобто її нуль. Ґратка з прикладу 3 на с. 202 — завжди повна (в тому числі для нескінченного *A*). Одиницею цієї ґратки слугує сама множина (містить будь-яку свою підмножину), нулем — порожня множина.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Як формулюється закон композиції об'єктів?
2. Чим різняться внутрішній і зовнішній закони композиції?
3. Що таке групоїд? Наведіть приклади групоїдів.
4. Як будується граф групоїда?
5. Що таке нейтральний елемент?
6. Як доводиться твердження: якщо нейтральний елемент існує, то він — єдиний і регулярний?
7. Що таке симетричний елемент?
8. Яка множина називається симетрованою?
9. Як доводиться твердження: якщо елемент, симетричний елементу *a* відносно асоціативного закону, існує, то він — єдиний та регулярний?
10. Що таке півгрупа (моноїду)? Наведіть приклади.
11. Що таке комутативна півгрупа? Наведіть приклади.
12. Що таке група? Наведіть приклади.
13. Які є приклади абелевої групи?
14. Які є приклади асоціативного й абелева кілець?
15. Які є приклади абелева кільця з одиницею?
16. Які є приклади тіла?
17. Які є приклади поля?

ПЕРЕЛІК ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

1. Побудова матриці та графа групоїда.
2. Визначення властивостей закону композиції.
3. Визначення типу алгебри.

СПИСОК ЛІТЕРАТУРИ

1. *Сигорский В. П.* Математический аппарат инженера. — К.: Техніка, 1977. — 768 с.
2. *Биркгоф Г., Барти Т.* Современная прикладная алгебра. — М.: Мир, 1976.
3. *Мальцев А. И.* Алгебраические системы. — М.: Наука, 1970.
4. *Гастев Ю. А.* Гомоморфизмы и модели. — М.: Наука, 1970.
5. *Шиханович Ю. А.* Введение в современную математику. — М.: Наука, 1965.
6. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера. — М.: Энергоатомиздат, 1988.

РОЗДІЛ

8

КОМБІНАТОРИКА

Комбінаторика — це наука, основним завданням якої є перерахунок і переліченя елементів у скінченних множинах. Якщо в задачі визначається, скільки елементів, які належать заданій скінченній множині, мають деяку властивість або заданий набір властивостей, то ця задача називається задачею *перерахунку*. В інших випадках для яких-небудь цілей треба вилучити всі елементи множини, які задовольняють задані властивості. Такі задачі називаються задачами *перелічення*.

Зустрічаються також задачі, коли на початковій скінченній множині елементів означено деяку цільову функцію, причому інтерес становлять елементи множини, що забезпечують мінімальне (або максимальне) значення цієї функції. У цьому разі маємо окремий випадок задачі оптимізації. Тут під її розв'язком у *сильному значенні* розуміють знаходження сукупності всіх елементів, які забезпечують мінімальне (або максимальне) значення цільової функції, а під розв'язком у *слабкому значенні* — відшукання довільного елемента, що забезпечує мінімальне (або максимальне) значення цільової функції. Іноді цікавляться лише мінімальним (або максимальним) значенням функції.

Усі зазначені задачі пов'язані одна з одною. Так, при розв'язуванні задач оптимізації припускається, що в розпорядженні є метод перелічення елементів початкової множини (яка, як правило, є сукупністю елементів деякої великої множини, що задовольняє задану властивість), а для того щоб оцінити ефективність методів перелічення або оптимізації, часто доцільно розв'язати задачу перерахунку елементів (у початковій множині або в деяких її підмножинах).

8.1. Комбінаторні схеми

8.1.1. Правила суми і добутку. Вибірки, перестановки, сполучення

У розд. 2 доведено таку теорему: *Якщо* $X_1, X_2, \ldots, X_k$ *— скінченні множини, які попарно не перерізаються, тобто* $X_i \cap X_j = \varnothing$ *при* $i \neq j$, *то* $\left|\bigcup_{i=1}^{k} X_i\right| = \sum_{i=1}^{k}|X_i|$.

У комбінаториці цей факт називається *правилом суми*. Для $k = 2$ воно формулюється так: *Якщо об'єкт а може бути вибраний р способами, а* *об'єкт b — іншими q способами, то вибір «або a, або b» може бути здійснений p + q способами.*

Вибори a та b є взаємовиключними, тому що $X_i \cap X_j = \varnothing$, тобто жоден із способів вибору об'єкта a не збігається з жодним із способів вибору об'єкта b. За наявності таких збігів, тобто коли припустити, що $X_i \cap X_j = C \neq \varnothing$, де $|C| = c$, результат дорівнює $p + q - c$ (c — число збігів).

Іншим правилом, яке часто застосовується в комбінаториці, є правило добутку: *Якщо об'єкт a може бути вибраний p способами і після кожного з таких виборів об'єкт b, у свою чергу, може бути вибраний q способами, то вибір «a та b» у вказаному порядку можна здійснити pq способами.*

Це правило використовується тоді, коли способи вибору a і b — незалежні.

Доведемо правило добутку. Нехай $a \in A, b \in B$, де $|A| = p, |B| = q$. Розглянемо множини $X_i \left(i = \overline{1, p}\right)$ пар (x, b), де $x = a_i$, $b \in B$. Тоді множини X_i попарно не перерізаються, $|X_i| = q$, $i = \overline{1, p}$. Множина всіх пар $(x, y) \in X = \bigcup\limits_{i=1}^{p} X_i$, і за правилом суми маємо

$$|X| = \left|\bigcup_{i=1}^{p} X_i\right| = \sum_{i=1}^{p} |X_i| = pq\,. \tag{8.1}$$

Сформульоване та доведене правило добутку для послідовного вибору двох елементів може бути узагальнене на n елементів. Загалом правило добутку формулюється так:

Якщо об'єкт x_1 може бути вибраний n_1 способами, після чого об'єкт x_2 може бути вибраний n_2 способами, і для будь-якого i, де $2 \le i \le m - 1$, після вибору об'єктів $x_1, \ldots, x_i$ об'єкт x_{i+1} може бути вибраний n_{i+1} способами, то вибір упорядкованої послідовності m об'єктів $(x_1, x_2, \ldots, x_m)$ може бути здійснений $n_1, n_2, \ldots, n_m$ способами.

Набір елементів $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ із множини $X = \{x_1, x_2, \ldots, x_n\}$ називається *вибіркою обсягом r*, або *r-вибіркою*.

Вибірка r-елементів називається r-перестановкою, якщо враховується порядок слідування елементів, і r-сполученням, якщо беруться до уваги тільки елементи без урахування їхнього порядку; n-перестановка з n різних елементів є просто перестановкою.

Приклад. Для множини $M = \{a, b, c\}$ розрізняють шість 3-перестановок, утворених з одних і тих самих елементів: *abc*, *acb*, *bac*, *bca*, *cab*, *cba*; водночас ці вибірки є різними записами одного й того самого 3-сполучення елементів a, b, c.

Вибірки можуть допускати і не допускати повторення елементів. При r-вибірках з повторенням розрізняють два випадки:

- запас елементів, що повторяються, обмежений і визначається специфікацією $\{n_1, n_2, \ldots, n_k\}$, де n_i — кількість елементів i-го вигляду. Загальне число елементів початкової множини $n = n_1 + n_2 + \ldots + n_k$, причому $r \le n$. Кожний вигляд можна розглядати як клас еквівалентності, елементи якого важаються нерозрізнюваними і позначаються однаковими номерами або символами. Сукупність позначень різних класів утворює сім'ю представників;

- запас елементів не обмежується й у вибірці з r-елементів допускається будь-яке число повторень, що не перевищує r. Початкову множину в цьому випадку можна розглядати як таку, що складається з різних елементів, але після вибірки деякого елемента він відновлюється в цій множині. Така вибірка називається *вибіркою з поверненням.*

Приклад. Нехай початкову множину можна задати трьома класами еквівалентності зі специфікацією $\{2, 5, 4\}$; тоді

$$n = 2 + 5 + 4 = 11.$$

Визначимо представників класів *a*, *b*, *c*. Так, *aabbbc*, *abbabc*, *bbbaac* — приклади різних перестановок із шести сполучень *aabbbc*; *aabbbbcccc*, *abbbbacccc* — приклади різних перестановок із 10 сполучень *aabbbbcccc*; *aabbbc*, *aabbcc*, *bbbbbc*, *aacccc* — різні 6-сполучення, а 11-сполучення тільки одне: *aabbbbbcccc*.

Визначимо число r-перестановок з n елементів, позначивши його через $P(n, r)$.

Твердження 8.1. *Справджується рівність* $P(n,r) = n(n-1)(n-2)\ldots$ $\ldots(n-r+1) = \dfrac{n!}{(n-r)!}$.

Доведемо це твердження. Кожна r-перестановка є впорядкованою послідовністю завдовжки r, члени якої — попарно різні й вибираються з n-елементної множини. Тоді перший член цієї послідовності може бути вибраний n способами, після кожного вибору першого члена послідовності другий — $(n-1)$ способами і т. д. Відповідно після кожного вибору першого, другого і т. д. аж до $(r - 1)$-го членів послідовності r-й член може бути вибраний $n - (r - 1) = n - r + 1$ способами, звідки за узагальненим правилом добутку дістаємо наведену вище.

Наслідок 8.1. *Справджується рівність* $P(n, n) = n!$

Наслідок 8.2. *Справджується рівність* $P(n, r) = \dfrac{P(n, n)}{P(n-1,\ n-r)}$.

Позначимо число r-сполучень з n елементів через $C(n, r)$. Щоб знайти $C(n, r)$, задамося питанням, скільки r-перестановок можна утворити з кожного r-сполучення. Очевидно, що $r!$. Тому шукане число буде в $r!$ разів меншим, ніж число r-перестановок з n елементів.

Твердження 8.2. *Справджується рівність* $\mathrm{C}(n,\ r) = \dfrac{n(n,r)}{r!} =$ $= \dfrac{n(n-1)\ldots(n-r+1)}{r!} = \dfrac{n!}{r!(n-r)!}$.

Наприклад, для множини з чотирьох елементів $A = \{a, b, c, d\}$ кількість 2-сполучень

$$C(4, 2) = \frac{4!}{2!\cdot 2!} = 6\ .$$

Перелічимо їх: *ab*, *ac*, *ad*, *bc*, *bd*, *cd*.

$C(n, r)$ означають ще C_n^r. Очевидно, що $C_n^r = C_n^{n-r}$.

Розглянемо перестановку з повторенням з n елементів, специфікація яких $\{n_1, n_2, \ldots, n_k\}$, де $n = n_1 + n_2 + \ldots + n_k$. Позначимо їх число P_n $(n_1, n_2, \ldots, n_k)$.

Через збіг деяких елементів число перестановок менше, ніж $n!$, тому що переставлення однакових елементів нічого не міняє.

Елементи j-го класу містять n_i елементів, які можна переставити $n_j!$ способами, і в кожному класі такі переставлення здійснюються незалежно. Тому відповідно до правила добутку можна здійснити $n_1!\, n_2! \ldots n_k!$ перестановок, які не змінюють задану перестановку. Отже, число різних перестановок із повтореннями, що утворюються з n елементів, менше за $P(n, n)$ у $n_1!\, n_2! \ldots n_k!$ разів. Таким чином, доведено таке твердження.

Твердження 8.3. *Справджується рівність* $P_n(n_1, n_2, \ldots, n_k) = \dfrac{n!}{n_1!\, n_2! \ldots n_k!}$.

Приклад. Нехай є проекти трьох типів. Треба визначити, скільки існує різних планів забудови вулиці 10 будинками, коли відомо, що мають бути три будинки I типу, п'ять будинків II типу і два будинки III типу.

Використовуючи формулу для визначення числа перестановок із повтореннями, знаходимо

$$P_{10}(3, 5, 2) = \frac{10!}{3!\,5!\,2!} = 2520.$$

Розглянемо r-перестановку з n різних елементів з поверненням, тобто будемо вважати, що запас об'єктів необмежений. Позначимо число r-перестановок з n різних елементів з поверненням через $U(n, r)$.

Твердження 8.4. *Справджується рівність* $U(n, r) = n^r$.

Доведемо це твердження. Кожна з шуканих перестановок є впорядкованою послідовністю завдовжки r, причому кожний член цієї послідовності може бути вибраний будь-яким з n способів, звідки за узагальненим правилом добутку одержуємо шукану формулу.

Приклад. Як приклад визначимо, скільки існує різних двозначних чисел у десятковій системі. Очевидно, це задача визначення числа 2-перестановок з 10 елементів з поверненням. Застосувавши наведену вище формулу, дістанемо

$$U(10, 2) = 10^2.$$

Розглянемо r-сполучення із n різних елементів з необмеженими повтореннями.

Позначимо їх число через $F(n, r)$.

Твердження 8.5. *Справджується рівність*

$$F(n, r) = \frac{(r+n-1)!}{r!(n-1)!}. \qquad (8.2)$$

Доведення проведемо так. Кожному сполученню поставимо у відповідність перестановку, в якій всі елементи заданого сполучення закодовано одиницями, причому різні класи елементів розділяються нулем (навіть тоді, коли елементи яких-небудь класів не ввійшли в сполучення).

Наприклад, для множини $A = \{a, b, c, d, e\}$ 4-сполученню *abbce* відповідає перестановка 101101001; 4-сполученню *bbbec* — перестановка 011100011, 7-сполученню *aabbdde* — перестановка 11011001101.

Очевидно, кожна перестановка для r-сполучення з n елементів із повтореннями містить r одиниць і $n - 1$ нулів, причому це перестановка з повтореннями з $r + n - 1$ елементів зі специфікацією $\{r, n - 1\}$. Відповідність між множиною таких перестановок і множиною сполучень, що розглядаються, є ін'єкцією (читачеві пропонується це довести самостійно). Отже, їхні потужності однакові, тобто шукане число r-сполучень збігається з числом перестановок з обмеженими повтореннями з $r + n - 1$ елементів зі специфікацією $\{r, n - 1\}$. Таким чином

$$F(n, r) = P_{n+r-1} = \frac{(r+n-1)!}{r!(n-1)!} = C^r_{n+r-1}.$$

Приклад. Розглянемо множину $A = \{a, b, c, d\}$. Знайдемо число 2-сполучень із чотирьох елементів із необмеженими повтореннями, користуючись останньою формулою:

$$F(4, 2) = C_5^2 = \frac{5!}{2!3!} = 10 .$$

Перелічимо їх: *aa*, *ab*, *ac*, *ad*, *bb*, *bc*, *bd*, *cc*, *cd*, *dd*.

8.1.2. Рекурентні співвідношення

Підрахунок числа перестановок і сполучень можна виконати за допомогою рекурентних співвідношень, які відіграють важливу роль у комбінаториці. Множину r-перестановок з n елементів можна розбити на два класи так, щоб перестановки одного з них не містили деякого фіксованого елемента початкової множини, а всі перестановки іншого класу обов'язково містили його. Очевидно, перший клас складається з $P(n - 1, r)$ перестановок, а другий — з $P(n - 1, r - 1)$, оскільки фіксований елемент може займати одне з r положень у кожній з $P(n - 1, r - 1)$ перестановок.

Отже, рекурентна формула має вигляд:

$$P(n, r) = P(n - 1, r) + r\,P(n - 1, r - 1),$$

де $P(k, 0) = 1$ (не має комбінаторного значення); $P(k, 1) = k \forall k$; $P(k, S) = 0$ при $k < S$.

Це граничні умови для здобутого рекурентного співвідношення.

Поклавши $n = r$, дістанемо:

$$\begin{aligned} P(n, n) &= P(n - 1, n) + nP(n - 1, n - 1) = \\ &= 0 + nP(n - 2, n - 1) + n(n - 1)\ P(n - 2, n - 2) = \\ &= n(n - 1)\ P(n - 3, n - 2) + n(n - 1)(n - 2)\ P(n - 3, n - 3) = \ldots = \\ &= n(n - 1)(n - 2) \ldots 2 \cdot 1 = n! \end{aligned}$$

Для числа r-сполучень з n елементів маємо

$$C(n,r)=\frac{P(n,r)}{r!}=\frac{P(n-1,r)}{r!}+\frac{rP(n-1,r-1)}{r!}=$$

$$=C(n-1,r)+\frac{P(n-1,r-1)}{(r-1)\ !}=C(n-1,r)+C(n-1,r-1).$$

Граничні умови: $C(n,0)=C(1,1)=1, C(k,S)=0\ \forall k<S$.

8.1.3. Біном Ньютона

Поставимо у відповідність кожному об'єкту з множини $\{\alpha_1,\alpha_2,...,\alpha_n\}$ двочлени вигляду $1+\alpha_i x(i=1,2,...,n)$, перемножимо їх і зведемо подібні члени: $(1+\alpha_1 x)(1+\alpha_2 x)(1+\alpha_3 x)...(1+\alpha_n x)=1+a_1x+a_2x^2+a_3x^3+...+a_nx^n$.

Очевидно, коефіцієнт a_r множини є сумою добутків, кожен з яких утвориться r-елементами з n (r-сполучення), причому в a_r таких добутків C (n, r). Якщо покласти $\alpha_1=\alpha_2=...=\alpha_n=1$, то будь-який добуток r-сполучень елементів дорівнює 1 і, отже, $a_r=C(n,r)=\left(C_n^r\right)$. Таким чином,

$$(1+x)^n=C_n^0+C_n^1x+C_n^2x^2+...+C_n^{n-1}x^{n-1}+C_n^nx^n. \tag{8.3}$$

Цей вираз називається *біномом Ньютона*, а r-сполучення з n різних елементів C_n^r є біномними коефіцієнтами.

За допомогою бінома Ньютона можна вивести різні формули для сполучень. Поклавши x = 1, матимемо $\sum_{i=1}^{n}C_n^i=2^n$. Ця формула визначає кількість усіх підмножин деякої множини.

При x = –1 знаходимо $\sum_{i=0}^{n}(-1)^i C_n^i=0$.

8.1.4. Поліномні твірні функції

Добуток $(1+\alpha_1x)(1+\alpha_2x)$ ... $(1+\alpha_nx)$ породжує r-сполучення з n різних елементів, в яке кожний об'єкт із множин може входити не більш як один раз. Щоб одержати інші вигляди сполучень, треба брати й інший вигляд співмножників.

Якщо об'єкт α_i може входити в сполучення 0, 1, ..., k разів, то замість $1+\alpha_i x$ потрібно взяти співмножник $1+\alpha_i x+\alpha_i^2x^2+...+\alpha_i^kx^k$.

Тоді при $\alpha_1=\alpha_2=...=\alpha_n=1$ коефіцієнти a_r многочлена $A(x)=1+a_1x+$ $+a_2x^2+...+a_nx^n$ будуть r-сполученнями з n різних елементів із повтореннями.

Приклад. Знайдемо за наведеною схемою для сполучення з трьох елементів $\{a, b, c\}$ зі специфікацією {3, 1, 2} добуток

$$(1+x+x^2+x^3)(1+x)(1+x+x^2)=1+3x+5x^2+6x^3+5x^4+3x^5+x^6.$$

Тут коефіцієнт при x^n дає шукане число n-сполучень, тобто:

3 — число 1-сполучень(це a, b, c);

5 — число 2-сполучень(це aa, ad, ac, bc, cc);
6 — число 3-сполучень(це aaa, aab, aac, abc, acc, bcc).

Многочлен $A(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n + \ldots$ називається поліномною твірною функцією (енумератором) для послідовності $a_0, a_1, \ldots, a_n, \ldots$. У випадку, що розглядається, ця послідовність є r-сполученням з n елементів із повтореннями. Біном Ньютона — це твірна функція для сполучень без повторень. Слід мати на увазі, що змінна x енумератора $A(x)$ ніяк не визначена і вважається просто абстрактним символом. Його роль зводиться лише до того, щоб розрізняти елементи послідовності $a_0, a_1, a_2, \ldots, a_n$. При цьому різні перетворення таких послідовностей замінюються відповідними операціями над твірними функціями.

Для сполучень із необмеженими повтореннями елементів n типів енумератором буде вираз $(1 + x + x^2 + \ldots + x^n + \ldots)^n$, який можна записати у вигляді

$$1+x+x^2+\ldots+x^n+\ldots=\frac{1}{1-x}=(1-x)^{-1}.$$

Розглядаючи вираз $(1-x)^{-n}$ як біном Ньютона з від'ємним показником $(-n)$, формально дістаємо

$$(1-x)^{-n} = \sum_{r=0}^{\infty} C_{-n}^{r}(-x)^r = \sum_{r=0}^{\infty} \frac{-n(-n-1)\ldots(-n-r+1)}{r!}(-x)^r =$$

$$= \sum_{r=0}^{\infty} \frac{(n+r-1)\ldots(n+1)n}{r!}x^r = \sum_{r=0}^{\infty} C_{n+r-1}^{r}x^r = \sum_{r=0}^{\infty} F(n,r)x^r,$$

що збігається з раніше одержаним результатом. Звідси також маємо

$$C_{-n}^{r} = C_{n+r-1}^{2}.$$

Якщо потрібно визначити число r-сполучень з n типів елементів із необмеженими повтореннями, яке обов'язково містить хоча б один елемент кожного типу, то енумератор

$$(x+x^2+x^3+\ldots)^n = x^n(1+x+x^2+\ldots)^n = x^n(1-x)^{-n} = x^n\sum_{r=0}^{\infty} C_{n+r-1}^{r}x^r =$$

$$= \sum_{r=0}^{\infty} C_{n+r-1}^{r}x^{n+r}.$$

8.1.5. Розміщення і функціональні відображення

Позначимо через Y^X множину всіх відображень $f: X \to Y$.

Твердження 8.6. *Нехай* $|X| = r, |Y| = n$. *Тоді*

$$\left|Y^X\right| = U(n,r) = n^r = |Y|^{|X|}.$$

Нехай $X = \{x_1, \ldots, x_r)$. Тоді будь-яке відображення $f: X \to Y$ можна подати у вигляді впорядкованої послідовності $<f(x_1), \ldots, f(x_r)>$ значень

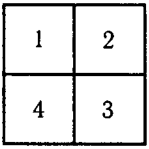


Рис. 54

функції f у точках з X, де $f(x_i) \in Y$. Очевидно, що цим самим установлено взаємно однозначну відповідність між множиною функціональних відображень та множиною впорядкованих вибірок із повтореннями обсягом r із множини Y обсягом n (тобто множиною r-перестановок з n елементів із повтореннями), звідки випливає правильність твердження 8.6.

Аналогічно доводиться таке твердження.

Твердження 8.7. *Нехай* $|X| = r, |Y| = n$. *Тоді число всіх ін'єктивних відображень вигляду* $f : X \to Y$ *дорівнює* $P(n, r)$.

Наслідок 8.3. *Для* S_n *— множини всіх бієктивних відображень* n*-елементної множини в себе маємо* $|S_n| = A_n^n = P(n, n) = n!$.

Розглянемо деякі приклади застосування наведених формул.

Приклади: 1. Задано квадрат, розділений на чотири частини (рис. 54). Скількома способами можна розфарбувати квадрат п'ятьма кольорами: а) допускаючи фарбування різних частин в один колір; б) якщо різні частини забарвлюються різними кольорами.

Для розв'язання будемо розглядати кожне розфарбування як функціональне відображення множин номерів частин квадрата $X = \{1, 2, 3, 4\}$ у множину кольорів Y, де $|Y| = 5$. Тоді, використовуючи твердження 8.6 і 8.7, маємо: а) $5^4 = 625$; б) $5!/(5 - 4)! = 5! = 120$.

2. Знайти, скількома способами можна вибрати 5 номерів із 36.

Розв'язком буде число (36, 5)-сполучень. Використовуючи твердження 8.2, знаходимо, що число способів $C_{36}^5 = 376\,992$.

3. У скількох випадках при грі в «Спортлото» (угадування 5 номерів із 36) будуть правильно вибрані: а) рівно 3 номери; б) рівно 4 номери; в) рівно 5 номерів; г) не менш як 3 номери?

Маємо: а) 3 правильних номери з 5 можна вибрати C_5^3 способами, а 2 неправильних номери, що залишилися, — C_{31}^2 способами. Далі за правилом множення визначаємо, що шукане число дорівнює $C_5^3 C_{31}^2 = \frac{5!}{3!2!}\frac{31!}{29!2!} = 4650$. В інших випадках відповідно знаходимо: б) $C_5^4 C_{31}^1 = 155$; в) 1; г) $4650 + 155 + 1 = 4806$.

4. З колоди у 52 карти вибираються 10 карт. У скількох випадках серед них виявляться всі 4 тузи?

Виключивши з розгляду тузи, дістанемо, що вибираються 6 карт із 48, а такий вибір можна здійснити C_{48}^6 способами.

5. Нехай є монети вартістю 1, 2 і 3 коп. Скільки існує різних комбінацій 30 монет (наприклад, набори: 13 монет по 1 коп., 7 — по 2 коп., 10 — по 3 коп.)?

За умовою задачі потрібно визначити кількість (3, 30)-сполучень із повтореннями. Використовуючи твердження 8.5, визначаємо, що шукане число $F(3, 30) = 496$.

6. Виведемо формулу для визначення кількості цілочислових розв'язків системи

$$x_1 + x_2 + \ldots + x_n = r,\ x_i \geq a_i,\ i = 1, 2, \ldots, n, \tag{8.4}$$

де $n \geq 1$, a_i — цілі числа.

Розглянемо рівняння $x_1 + x_2 + \ldots + x_n = r$. Зазначимо, що при будь-яких цілих $n \geq 1$, $r \geq 0$ значення $F(n, r)$ виражає кількість розв'язків серед цілих невід'ємних чисел цього рівняння.

Справді, кожному розв'язку $(x_1, x_2, \ldots, x_n)$ рівняння (8.4) можна поставити у відповідність r-сполучення з n елементів множини $A = \{a_1, a_2, \ldots, a_n\}$ таке, що в ньому містяться x_1 елементів вигляду a_1, x_2 елементів вигляду a_2 і т. д. аж до x_n елементів вигляду a_n. Неважко бачити, що зазначена відповідність є взаємно однозначною, звідки і випливає правильність твердження, яке доводиться.

Тепер для визначення кількості цілочислових розв'язків системи (8.4) виконаємо заміну змінних. Нехай $u_i = x_i - a_i$. Тоді $x_i = u_i + a_i$, а отже,

$$u_1 + u_2 + \ldots + u_n = r - \sum_{i=1}^{n} a_i, u_i \geq 0, i = 1, 2, \ldots, n\ , \tag{8.5}$$

тобто початкову задачу зведено до вже розглянутого випадку (очевидно, кількість цілочислових розв'язків систем (8.4) і (8.5) збігається). Тоді, використовуючи доведене вище твердження 8.5, знаходимо, що кількість цілочислових розв'язків системи (8.4) або (8.5) у випадку $r \geq \sum_{i=1}^{n} a_1$ становить

$$F\left(n, r - \sum_{i=1}^{n} a_i\right) = C_{n+r-\sum_{i=1}^{n} a_i - 1}^{r - \sum_{i=1}^{n} a_i}\ . \tag{8.6}$$

Якщо ж $r < \sum_{i=1}^{n} a_i$, то множина цілочислових розв'язків системи (8.4) є порожньою, і кількість їх дорівнює 0.

8.1.6. Розбиття

Пригадаємо означення розбиття з розд.2 . Підрахуємо число розбиттів скінченної множини X, де $|X| = n$, на k підмножин $X_1, X_2, \ldots, X_k$ $(k \geq 1)$, де $\sum_{i=1}^{k} n_i = n$, таких, що кожне X_i містить n_i елементів, тобто

$$\bigcup_{i=1}^{k} X_i = X,\ X_i \bigcap X_j = \varnothing \text{ при } i \neq j, |X_i| = n_i, i = 1, 2, \ldots, k\ . \tag{8.7}$$

Нехай для деяких номерів i матимемо $X_i = \varnothing$. Число зазначених розбиттів при фіксованих n_i позначимо через $C_n^{n_1, \ldots, n_k}$.

У цьому випадку набір підмножин множини X у розбитті є впорядкованим (тобто $X_1, \ldots, X_k$ — впорядкована послідовність множин). Нижче, крім того, розглядається випадок, коли набір підмножин у розбитті не є впорядкованим.

Твердження 8.8. *Справджується рівність* $C_n^{n_1, \ldots, n_k} = \dfrac{n!}{n_1! \ldots n_k!}$.

Очевидно, кожну з множин X_i можна розглядати як сполучення без повторень.

Для утворення сполучення, що відповідає множині X_1, можуть бути використані всі елементи множини X, тобто множина X_1 може бути вибрана $C_n^{n_1}$ способами. Після вибору X_1 множина X_2 може бути вибрана $C_{n-n_1}^{n_2}$ способами (оскільки X_2 є підмножиною множини $X - X_1$ та $|X - X_1| = n - n_1$), і для будь-якого i, де $2 \leq i \leq k$, після вибору множин $X_1, \ldots, X_{i-1}$ множина X_i може бути вибрана $C_{n-n_1-\ldots-n_{i-1}}^{n_i}$ способами. Проте тоді за правилом добутку вибір упорядкованої послідовності множин $X_1, \ldots, X_k$, які задовольняють (8.7), можна здійснити $C_n^{n_1} C_{n-n_1}^{n_2} \ldots C_{n-n_1-\ldots-n_{k-1}}^{n_k}$ способами.

Використовуючи цей вираз, а також твердження 8.2 і зробивши необхідні скорочення, дістанемо твердження 8.8.

Твердження 8.9. *Число* $C_n^{n_1,\dots,n_k}$, *де* $n_i \geq 0$, $\sum_{i=1}^{k} n_i = n$, *дорівнює числу* P_n $(n_1, n_2, \dots, n_k)$ *перестановок n елементів зі специфікацією* $\{n_1, n_2, \dots, n_k\}$, *серед елементів яких міститься* n_1 *елементів першого типу,* n_2 *елементів другого типу і т. д. аж до* n_k *елементів k-го типу.*

Кожній перестановці зазначеного типу поставимо у відповідність розбиття множини $X = \{1, 2, \dots, n\}$ номерів елементів у вибірці на підмножини $X_1, \dots, X_k$, де X_i — множина номерів елементів i-го типу у вибірці. Очевидно, при цьому виконується рівність (8.7).

Зазначена відповідність між перестановками заданого типу і розбиттями, які задовольняють (8.7), є взаємно однозначною, звідки завдяки твердженню 8.8 випливає правильність твердження 8.9.

Приклади: 1. У студентській групі, яка складається з 25 осіб, при виборі старости за висунену кандидатуру проголосували 12 чоловік, проти — 10, утрималися — 3. Скількома способами могло бути проведене таке голосування?

Нехай X — множина студентів у групі; X_1 — множина студентів, які проголосували за висунену кандидатуру; X_2 — множина студентів, які проголосували проти; X_3 — множина студентів, які утрималися від голосування. Тоді $|X| = 25$, $|X_1| = 12$, $|X_2| = 10$, $|X_3| = 3$, $X = X_1 \cup X_2 \cup X_3$, $X_i \cap X_j = \varnothing$ при $i \neq j$, а отже, шукане число дорівнює $C_{25}^{12,10,3}$. Використовуючи твердження 8.8, знаходимо

$$C_{25}^{12,10.3} = \frac{25!}{12!10!3!} = 1\,487\,285\,800.$$

2. Скількома способами можна розфарбувати квадрат, поділений на дев'ять частин (рис. 55), чотирма кольорами таким чином, щоб у перший колір були забарвлені 3 частини, у другий — 2, у третій — 3, у четвертий — 1?

Нехай X — множина кольорів, де $|X| = 4$. Тоді кожне розфарбування, що розглядається як послідовність кольорів, в які забарвлюються пронумеровані частини квадрата, є перестановкою 9 елементів зі специфікацією. При цьому нас цікавлять перестановки із заданою комбінацією елементів (3 елементи — перший колір, 2 — другий, 3 — третій, 1 — четвертий). Проте тоді, використовуючи твердження 8.9, дістаємо, що шукане число

$$C_9^{3,2,3,1} = \frac{9!}{3!2!3!1!} = 5040\,.$$

Підрахуємо тепер, скількома способами можна розбити множину X, де $|X| = n$, на підмножини, серед яких для кожного $i = 1, 2, \dots, n$ існують $m \geq 0$ підмножин з i елементами, де $\sum_{i=1}^{n} im_i = n$. При цьому, на відміну від розглянутого вище випадку, тут набір підмножин у розбитті не є впорядкованим (тобто порядок підмножин у розбитті не є суттєвим). Так, розбиття множини $X = \{1, 2, 3, 4, 5\}$ вигляду

$\{1, 3\}, \{4\}, \{2, 5\};$

$\{4\}, \{2, 5\}, \{1, 3\};$

$\{2, 5\}, \{4\}, \{1, 3\}$

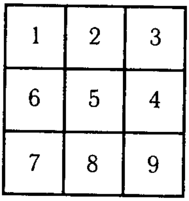


Рис. 55

вважаються однаковими. Позначимо число невпорядкованих розбиттів множини X через $N(m_1, \ldots, m_n)$.

Твердження 8.10. *Справджується рівність* $N(m_1, \ldots, m_n) = \dfrac{n!}{m_1!\ldots m_n!(1!)^{m_1}\ldots(n!)^{m_n}}$.

Зазначимо, що кожне з невпорядкованих розбиттів, розглянутих при визначенні величини $N(m_1, \ldots, m_n)$, можна, нумеруючи множини в цьому розбитті, привести $m_1!\ldots m_n!$ способами до впорядкованих розбиттів вигляду

$$X_1, \ldots, X_{m_1}, X_{m_1+1}, \ldots, X_{m_1+m_2}, \ldots, X_{m_1+\ldots+m_{n-1}+1}, \ldots, X_{m_1+\ldots+m_n}, \quad (8.8)$$

де

$$\begin{aligned}&|X_1| = \ldots = \left|X_{m_1}\right| = 1; \left|X_{m_1+1}\right| = \ldots = \left|X_{m_1+m_2}\right| = \\ &= 2; \ldots; \left|X_{m_1+\ldots+m_{n-1}+1}\right| = \ldots = \left|X_{m_1+\ldots+m_n}\right| = n.\end{aligned} \quad (8.9)$$

При цьому об'єднання попарно непересічних множин розбиття, що утворюється таким чином, вигляду (8.8), (8.9) для всіх невпорядкованих розбиттів, які розглядаються, дає сукупність усіх розбиттів вигляду (8.8), (8.9), а отже, за правилом суми, використовуючи твердження 8.8, маємо $\dfrac{n!}{(1!)^{m_1}\ldots(n!)^{m_n}} = \sum m_1!\ldots m_n! = N(m_1, \ldots, m_n)m_1!\ldots m_n!$ (де підсумовування проводиться за всіма невпорядкованими розбиттями, що розглядаються), звідки і випливає правильність твердження 8.10.

Приклади: 1. Скількома способами з групи в 25 чоловік можна сформувати 5 коаліцій по 5 чоловік?

Нехай X — множина людей у групі, m_i — число коаліцій по i чоловік, де $i = 1, 2, \ldots, 25$. Тоді за умовою задачі $|X| = 25$, $m_5 = 5$, $m_i = 0$, $i \in \{1, 2, \ldots, 25\} \setminus \{5\}$, а отже, шукане число дорівнюватиме $N(0, 0, 0, 0, 5, 0, \ldots, 0)$, де завдяки твердженню 8.10

$$N(0, 0, 0, 0, 5, 0, \ldots, 0) = \frac{25!}{5!(5!)^5} = \frac{25!}{(5!)^6}.$$

2. Скількома способами можна задати відношення еквівалентності на множині $X = \{1, 2, \ldots, 25\}$ із трьома класами еквівалентності?

Використовуючи той факт, що множина класів еквівалентності є розбиттям множини X, як доведено в розд. 2, дістаємо, що шукане число визначається за формулою

$$\sum_{\substack{m_1+2m_2+\ldots+25m_{25}=25 \\ m_1+m_2+\ldots+m_{25}=3}} N(m_1, \ldots, m_{25}) = \sum_{\substack{m_1+2m_2+\ldots+25m_{25}=25 \\ m_1+m_2+\ldots+m_{25}=3}} \frac{25!}{m_1!\ldots m_{25}!(1!)^{m_1}\ldots(25!)^{m_{25}}},$$

де $m_1 + 2m_2 + \ldots + 25m_{25} = 25$, $m_1 + m_2 + \ldots + m_{25} = 3$ — множина всіх розв'язків цієї системи рівнянь у цілих невід'ємних числах (наприклад, $m_1 = 1$, $m_{12} = 2$, $m_i = 0$, $i \neq 1$, $i \neq 12$ — один із таких розв'язків).

8.1.7. Поліномна формула

Визначимо коефіцієнти функції $(x_1 + \ldots + x_k)^n = \sum\limits_{n_1+\ldots+n_k=n} c_{n_1,\ldots,n_k} x_1^{n_1} \ldots x_k^{n_k}$, яку вище було названо поліномною, де підсумовування проводиться за всіма розв'язками рівняння $n_1 + \ldots + n_k = n$ у цілих невід'ємних числах.

Твердження 8.11. *Справджується рівність* $c_{n_1,\ldots,n_k} = C_n^{n_1,\ldots,n_k} = \frac{n!}{n_1!\ldots n_k!}$.

Уведемо позначення співмножників виразу $(x_1 + x_2 + \ldots + x_k)^n$. Позначимо $a_i = (x_1 + x_2 + \ldots + x_k)$, $i = 1, 2, \ldots, n$. Тоді $\{x_1 + x_2 + \ldots + x_k)^n = a_1 a_2 \ldots a_n$. Нехай $A = \{a_1, a_2, \ldots, a_n\}$. Перерахуємо всі одночлени після перемноження $a_1 a_2 \ldots a_n$, в яких x_1 зустрічається n_1 разів, x_2 – n_2 разів, ..., x_k – n_k разів, тобто одночлени вигляду

$$x_1^{n_1} \ldots x_k^{n_k}, \tag{8.10}$$

де $n_1 + \ldots + n_k = n$.

Розглянемо будь-який з таких одночленів. Для кожного $i \in \{1, 2, \ldots, k\}$ позначимо через A_i таку підмножину множини A, що в цей одночлен увійдуть змінні x_i з тих і тільки тих співмножників, які перераховано в A_i. Цим самим одночлену поставлено у відповідність розбиття множини A на підмножини $A_1, \ldots, A_k$ такі, що

$$|A_i| = n_i, i = 1, \ldots, k, \bigcup_{i=1}^{k} A_i = A, A_i \cap A_j = \varnothing \text{ при } i \neq j. \tag{8.11}$$

Очевидно, зазначена відповідність між одночленами (8.10) і розбиттями (8.11) є взаємно однозначною. Проте тоді кількість одночленів вигляду (8.10) становить

$$c_{n_1,\ldots,n_k} = C_n^{n_1,\ldots,n_k} = \frac{n!}{n_1!\ldots n_k!}.$$

Приклад. Визначимо коефіцієнт c в одночлені $cx_1^3 x_2^4 x_3^3$ многочлена (зі зведеними подібними членами), що утворюється з виразу $(x_1 + x_2 + x_3)^{10}$.

Завдяки твердженню 8.11, маємо $c = C_{10}^{3,4,3} = \frac{10!}{3!4!3!} = 4200$.

8.1.8. Формула включень і виключень

Нехай X_1, X_2 — дві скінченні множини. Якщо $X_1 \cap X_2 = \varnothing$, то $|X_1 \cup X_2| = |X_1| + |X_2|$. Нехай тепер $X_1 \cap X_2 \neq \varnothing$. Тоді в $|X_1| + |X_2|$ кожний елемент з $X_1 \cap X_2$ буде врахований двічі, що треба виправити.

Тому

$$|X_1 \cup X_2| = |X_1| + |X_2| - |X_1 \cap X_2|, \tag{8.12}$$

тобто кількість елементів в об'єднанні множин виражено через кількість елементів у їх перерізі. Маємо відповідну формулу для довільного числа множин, яка називається формулою включень і виключень.

Твердження 8.12. *Нехай X_i — скінченні множини, $i = 1, 2, \dots, n$, $n \geq 2$. Тоді*

$$|X_1 \cup \ldots \cup X_n| = (|X_1| + \ldots + |X_n|) - (|X_1 \cap X_2| + |X_1 \cap X_3| + \ldots + |X_{n-1} \cap X_n|) +$$
$$+ (|X_1 \cap X_2 \cap X_3| + \ldots + |X_{n-2} \cap X_{n-1} \cap X_n|) + \ldots + (-1)^{n+1} |X_1 \cap \ldots \cap X_n|. \quad (8.13)$$

Доведення проведемо методом індукції за n. При $n = 2$ формула (8.13) збігається з (8.12). Припустимо, що формула, яка доводиться, є правильною для випадку $n - 1$ підмножин, де $n \geq 3$. Доведемо її правильність для n підмножин. Розіб'ємо множини $X_1, \dots, X_n$ на дві групи: $X_1, \dots, X_{n-1}$; X_n. Тоді згідно з (8.12) маємо:

$$|X_1 \cup \ldots \cup X_n| = |(X_1 \cup \ldots \cup X_{n-1}) \cup X_n| = |X_1 \cup \ldots \cup X_{n-1}| + |X_n| -$$
$$- |(X_1 \cup \ldots \cup X_{n-1}) \cap X_n| = |X_1 \cup \ldots \cup X_{n-1}| + |X_n| - |A_1 \cup \ldots \cup A_{n-1}|, \quad (8.14)$$

де $A_i = X_i \cap X_n$, $i = 1, \dots, n-1$.

Використовуючи індуктивне припущення, знаходимо:

$$|X_1 \cup \ldots \cup X_{n-1}| = (|X_1| + \ldots + |X_{n-1}|) - (|X_1 \cap X_2| + \ldots + |X_{n-2} \cap X_{n-1}|) +$$
$$+ (|X_1 \cap X_2 \cap X_3| + \ldots + |X_{n-3} \cap X_{n-2} \cap X_{n-1}|) - \ldots + (-1)^n |X_1 \cup \ldots \cup X_{n-1}|;$$
$$|A_1 \cup \ldots \cup A_{n-1}| = (|A_1| + \ldots + |A_{n-1}|) - (|A_1 \cap A_2| + |A_1 \cap A_3| + \ldots + |A_{n-2} \cap A_{n-1}|) + \ldots$$
$$+ (-1)^n |A_1 \cap \ldots \cap A_{n-1}| = (|X_1 \cap X_n| + \ldots + |X_{n-1} \cap X_n|) -$$
$$- (|X_1 \cap X_2 \cap X_n| + |X_1 \cap X_3 \cap X_n| + \ldots + |X_{n-2} \cap X_{n-1} \cap X_n|) + \ldots$$
$$+ (-1)^n |X_1 \cap \ldots \cap X_n|.$$

Із (8.14), ураховуючи останні рівності, дістаємо (8.13).

Наслідок 8.4. *Нехай X – скінченна множина, $X_1, \dots, X_n$ — підмножини X. Тоді*

$$|X - (X_1 \cup \ldots \cup X_n)| = |X| - (|X_1| + \ldots + |X_n|) + (|X_1 \cap X_2| + \ldots + |X_{n-1} \cap X_n|) - \ldots$$
$$+ (-1)^n |X_1 \cap \ldots \cap X_n|. \quad (8.15)$$

Справді,

$$[X - (X_1 \cup \ldots \cup X_n)] \cup (X_1 \cup \ldots \cup X_n) = X;$$

$$[X - (X_1 \cup \ldots \cup X_n)] \cap (X_1 \cup \ldots \cup X_n) = \varnothing,$$

звідки

$$|X - (X_1 \cup \ldots \cup X_n)| + |X_1 \cup \ldots \cup X_n| = |X|,$$

а отже,

$$|X-(X_1 \cup ... \cup X_n)| = |X| - |X_1 \cup ... \cup X_n|. \qquad (8.16)$$

Для здобуття (8.15) залишається тільки у (8.16) застосувати формулу (8.13).

Наведемо ще одну (найпоширенішу) форму запису формули включень і виключень. Нехай X — скінченна множина, що складається з N елементів; $\alpha_1, ..., \alpha_n$ — деякі властивості (одномісні предикати, визначені на X), які можуть мати або не мати елементи з X. Позначимо через $\forall i \in \{1, 2, ..., n\} \times \times X_i = \{x \in X \mid \alpha_i(x)\}$ множину елементів в X, що мають властивість α_i. Позначимо також через $N\left(\alpha_{i_1}, ..., \alpha_{i_k}\right) = \left|X_{i_1} \cap ... \cap X_{i_k}\right| = \left|\left\{x \in X \mid \alpha_{i_1}(x) \& ... ... \& \alpha_{i_k}(x)\right\}\right|$ кількість елементів в X, які мають одночасно властивості $\alpha_{i_1}, ..., \alpha_{i_k}$; $N_0 = |X-(X_1 \cup ... \cup X_n|$ — кількість елементів в X, що не мають жодної з властивостей $\alpha_1, ..., \alpha_n$. Тоді за формулою (8.15) дістанемо

$$N_0 = N - S_1 + S_2 - ... + (-1)^n S_n, \qquad (8.17)$$

де

$$S_k = \sum_{1 \le i_1 < ... < i_k \le n} N(\alpha_{i_1}, ..., \alpha_{i_k}), k = 1, 2, ..., n.$$

Приклади: **1.** Нехай $n = 3$, тобто задано три властивості $\alpha_1, \alpha_2, \alpha_3$. Тоді за формулою (8.17):

$$N_0 = N - N(\alpha_1) - N(\alpha_2) - N(\alpha_3) + N(\alpha_1, \alpha_2) +$$
$$+ N(\alpha_1, \alpha_3) + N(\alpha_2, \alpha_3) - N(\alpha_1, \alpha_2, \alpha_3). \qquad (8.18)$$

2. Нехай $X = \{0, 1, ..., 10\}$; $\alpha_1(x)$: «x — парне»; $\alpha_2(x)$: «$x > 6$»; $\alpha_3(x)$: «$2 < x < 8$». Підрахуємо кількість N_0 елементів в X, що не мають властивостей $\alpha_1, \alpha_2, \alpha_3$. Використовуючи формулу (8.18), знаходимо $N_0 = 11 - 6 - 4 - 5 + 2 + 2 + 1 - 0 = 1$ (неважко бачити, що єдиним елементом в X, який не має властивостей $\alpha_1, \alpha_2, \alpha_3$, є число 1).

3. Застосовуючи формулу включень і виключень, задачу визначення кількості цілочислових розв'язків системи

$$x_1 + x_2 + ... + x_n = r, a_i \le x_i \le b_i, i = 1, 2, ..., n, \qquad (8.19)$$

де a_i, b_i, r — цілі числа; $a_i \le b_i, i = 1, 2, ..., n$, легко звести до сукупності задач визначення кількості цілочислових розв'язків систем вигляду (8.4). Для цього слід скористатися властивостями $\alpha_i(x)$: «$x_i \ge b_i + 1$», де x = $(x_1, ..., x_n)$, $i = 1, 2, ..., n$, а за початкову множину X узяти сукупність цілочислових розв'язків x системи (8.4). Нехай N — кількість цілочислових розв'язків системи (8.4). Тоді N_0, що визначається за формулою (8.17), і виражатиме кількість цілочислових розв'язків системи (8.19).

4. Визначимо кількість тризначних чисел, в яких сума цифр дорівнює 20.

Якщо через x_1, x_2, x_3 позначити відповідно першу, другу та третю цифри в довільному тризначному числі $a = x_1 \cdot 10^2 + x_2 \cdot 10 + x_3$, то для розв'язання задачі досить визначити кількість цілочислових розв'язків системи

$$x_1 + x_2 + x_3 = 20, 1 \le x_1 \le 9, 0 \le x_2 \le 9, 0 \le x_3 \le 9. \qquad (8.20)$$

Нехай X — множина цілочислових розв'язків $x = \langle x_1, x_2, x_3 \rangle$ системи

$$x_1 + x_2 + x_3 = 20, x_1 \geq 1, x_2 \geq 0, x_3 \geq 0 \ ;$$

N — кількість елементів в X. Уведемо такі три властивості:

$$\alpha_1(x): «x_1 \geq 10»; \alpha_2(x): «x_2 \geq 10»; \alpha_3(x): «x_3 \geq 10» .$$

Використовуючи формулу включень і виключень (8.18), а також формулу (8.6) для підрахунку кількості цілочислових розв'язків системи вигляду (8.4), визначимо кількість N_0 цілочислових розв'язків системи (8.20):

$$N_0 = F(3, 20 - 1) - F(3, 20 - 10) - F(3, 20 - 10 - 1) - F(3, 20 - 10 - 1) + F(3, 20 - 10 - -10) + F(3, 20 - 10 - 10) + 0 - 0 = 210 - 66 - 110 + 2 = 36.$$

5. (Задача про безладдя). Задано n різних предметів $a_1, a_2, \ldots, a_n$ та n різних кліток $b_1, b_2, \ldots, b_n$. Скількома способами можна розмістити предмети по клітках так, щоб жоден предмет a_i не потрапив у клітку b_i?

Як початкову множину X візьмемо сукупність всіляких розміщень предметів по клітках. Тоді $N = |X| = n!$. Уведемо властивості a_i: «a_i знаходиться в клітці b_i», $i = 1, 2, \ldots, n$. Число $N\left(\alpha_{i_1}, \ldots, \alpha_{i_k}\right)$ розміщень, при якому предмет a_{i_v} знаходиться в клітці b_{i_v}, $v = 1, \ldots, k_i$, дорівнює $(n - k)!$. Однак тоді

$$S_k = \sum_{1 \leq i_1 < \ldots < i_k \leq n} N(\alpha_{i_1}, \ldots, \alpha_{i_k}) = C_n^k (n-k)! = \frac{n!}{k!} .$$

Використовуючи формулу включень і виключень (8.17), знаходимо, що кількість N_0 розміщень, при якому жодна з властивостей не виконується (тобто жоден із предметів a_i не потрапив у клітку b_i), становить

$$N + \sum_{k=1}^{n} (-1)^k S_k = n! + \sum_{k=1}^{n} (-1)^k \frac{n!}{k!} = n! \sum_{k=0}^{n} (-1)^k \frac{1}{k!} .$$

8.2. Розв'язання комбінаторних задач методом Поя

8.2.1. Реалізація групи

Нехай X — деяка скінченна множина елементів, $S(x)$ — група всіх бієктивних відображень множини X у себе, G — деяка група. Під реалізацією G у $S(X)$ будемо розуміти будь-який гомоморфізм $\tau : G \to S(X)$. Для простоти позначень замість $\tau(g)$, де $g \in G$, пишемо τ_g, а замість $\tau_g(x)$, де $x \in X$, — gx.

Приклад. Нехай $X = R^2$ — множина точок на площині, $S(R^2)$ — група бієктивних відображень R^2 у себе, G — група обертань на площині навколо початку координат. Елементи групи G будемо позначати кутами обертань, при цьому як додатний виберемо напрямок проти руху годинникової стрілки. Нехай, наприклад, $G = \{\alpha \in R \mid 0 \leq \leq \alpha < 2\pi\}$. Бінарною операцією на G є послідовне виконання обертань. Позначимо її символом $\oplus$. Тоді, очевидно, виконується рівність

$$\alpha \oplus \beta = \begin{cases} \alpha + \beta, & \text{якщо} \quad \alpha + \beta < 2\pi; \\ \alpha + \beta - 2\pi, & \text{якщо} \quad \alpha + \beta \geq 2\pi. \end{cases}$$

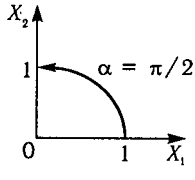


Рис. 56

Одиничним елементом у групі G є обертання на кут 0. Крім того, $\forall\alpha\in G-\{0\}\,\alpha^{-1}=2\pi-\alpha$. Під реалізацією введеної групи G будемо розуміти відображення, яке ставить у відповідність кожному обертанню α бієкцію $\tau_\alpha : R^2 \to R^2$ таку, що для будь-якої точки $M \in R^2$ точка $\tau_\alpha\underline{(M)}$ є результатом переміщення точки M після повороту вектора $\overline{OM}$ на кут α проти руху годинникової стрілки. Тоді, наприклад, $\pi/2(1,0)=\tau_{\pi/2}[(1,0)]=(0,1)$ (рис. 56).

8.2.2. Дія групи на множині

Нехай зберігаються умови, описані в п. 8.2.1, і τ — деяка реалізація G в S (X). Зауважимо, що оскільки при гомоморфізмі одиничний елемент відображається в одиничний елемент,

$$\forall x\in X\ ex=x\ , \tag{8.21}$$

де e — одиничний елемент із G. Крім того, за означенням гомоморфізму

$$\forall x\in X, \forall g,h\in G\ \tau_{gh}(x)=\tau_g(\tau_h(x))=g(hx),$$

або у введених вище позначеннях

$$\forall x\in X, \forall g,h\in G\ (gh)x=g(hx)\,. \tag{8.22}$$

Усякий раз, коли існує відображення $G\times X$ в X, яке задовольняє властивості (8.21) і (8.22), будемо говорити, що група G діє на множині X. В означенні дії групи G на множині X явним чином не використовується гомоморфізм τ, а отже, дію групи на множині можна вводити безпосередньо, тобто без попереднього зазначення гомоморфізму τ. Проте в останньому випадку можна за формулою

$$\tau_g(x)=gx, x\in X. \tag{8.23}$$

для кожного $g\in G$ однозначно визначити відображення $\tau_g : X\to X$, і при цьому відображення $\tau : g\to\tau_g$ буде гомоморфізмом групи G в S (X). Справді, використовуючи (8.21) та (8.22), знаходимо, що для будь-якого елемента $g\in G$ виконуються умови:

$$\forall x\in X\ \tau_{g-1}\left(\tau_g(x)\right)=g^{-1}(gx)=\left(g^{-1}g\right)x=ex=x\ ; \tag{8.24}$$

$$\forall x\in X\ \tau_g\left(\tau_{g-1}(x)\right)=g\left(g^{-1}x\right)=\left(gg^{-1}\right)x=ex=x\ . \tag{8.25}$$

Звідси випливає, що $\forall g\in G\ \tau_g$ — бієкція з X в X (із (8.25) одержуємо сюр'єктивність τ_g , а з (8.24) – ін'єктивність: із того, що $x_1, x_2\in X, \tau_g(x_1)=$ $=\tau_g(x_2)$, виходить, що $x_1=\tau_{g-1}\left(\tau_g(x_1)\right)=\tau_{g-1}\left(\tau_g(x_2)\right)=x_2$, і, крім того, з урахуванням (8.21) $\forall g,h\in G$, маємо

$$\forall x\in X\ \left(\tau_g\tau_h\right)=(x)\tau_g(\tau_h(x))=g(hx)=(gh)x=\tau_{gh}(x),$$

а отже,

$$\forall g, h \in G\ \tau_{gh} = \tau_g \tau_h ,$$

тобто показано, що τ — гомоморфізм.

Таким чином, у тих випадках, коли це зручно, будемо визначати дію групи на множині заданням гомоморфізму τ, а в інших випадках — безпосереднім заданням відображення $(g, x) \to gx$, яке задовольняє умови (8.21), (8.22).

Зазначимо далі, що із (8.21), (8.22) випливає

$$\forall g \in G, \forall x_1, x_2 \in X\ x_2 = gx_1 \Rightarrow x_1 = g^{-1} x_2 \qquad (8.26)$$

(оскільки, використовуючи $x_2 = gx_1$, маємо $g^{-1x_2} = g^{-1}x_2 = g^{-1}(g(x_1)) = (g^{-1}g)x_1 = ex_1 = x_1$).

8.2.3. Орбіти. Лема Бернсайда про кількість орбіт

Дві точки $x_1, x_2 \in X$ називаються еквівалентними відносно групи G, що діє на множині X (або G-еквівалентними), якщо існує $g \in G$, таке, що $x_2 = gx_1$.

Із (8.21), (8.22), (8.26) безпосередньо випливає, що G-еквівалентність є відношенням еквівалентності на X ((8.22) показує рефлексивність цього відношення, (8.26) — симетричність, а з (8.23) маємо транзитивність: оскільки $x_2 = g_1 x_1$, $x_3 = g_2 x_2$, маємо $x_3 = g_2 (g_1 x_1) = (g_2 g_1) x_1$) і, отже, зазначене відношення розбиває множину X на класи еквівалентності, які називають G-орбітами. Орбіту, що містить елемент $x_0 \in X$, будемо позначати через $G(x_0)$, тобто $G(x_0) = \{gx_0 \mid g \in G\}$.

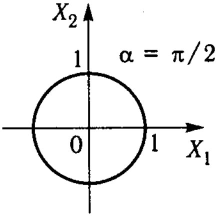


Рис. 57

Приклад. Орбітою точки (0,1) у попередньому прикладі є коло з центром у точці 0 радіусом 1, яке проходить через точку (0, 1) (рис. 57).

Нехай тепер X — скінченна множина елементів; G — скінченна група, що діє на X. Розглянемо задачу про визначення кількості G-орбіт. Позначимо через $N(g)$, де $g \in G$, кількість елементів з X, які залишаються на місці при дії g, тобто $N(g) = |\{x \in X \mid gx = x\}|$.

Лема (лема Бернсайда). $N = \dfrac{1}{|G|} \sum\limits_{g \in G} N(g)$.

Для доведення леми доведемо таке твердження.

Твердження 8.13. *Нехай* $x_0 \in X, St(x_0) = \{g \in |gx = x_0|\}$. *Тоді* $St(x_0)$ *— підгрупа групи* G *(стаціонарна підгрупа точки* x_0*).*

Оскільки $ex_0 = x_0$ (див. (8.21)), маємо $e \in St(x_0)$. Нехай g_1, $g_2 \in St(x_0)$. Тоді, використовуючи (8.22), дістаємо: $(g_1 g_2)x_0 = g_1(g_2 x_0) = g_1 x_0 = x_0$, тобто g_1, $g_2 \in St(x_0)$. Нехай $g \in St(x_0)$. Тоді $gx_0 = x_0$, звідки, застосовуючи (8.26), маємо: $x_0 = g^{-1} x_0$, а отже, $g^{-1} \in St(x_0)$.

Твердження 8.14. *Нехай* $x_0 \in X$. *Тоді* $|G(x_0)| = |G| - |St(x_0)|$.

Уведемо поняття суміжних класів у підгрупі. Якщо H — підгрупа групи G, то лівим суміжним класом G в H називається множина GH усіх елементів вигляду gh, де g — фіксований елемент із G, а $h \in H$, тобто $GH = \{gh \mid h \in H\}$, а правим класом HG називається множина $HG = \{hg \mid h \in H\}$.

Множина лівих суміжних класів GH позначається G/H.

Розглянемо ліві суміжні класи групи G у підгрупі $H = St\ (x_0)$. Поставимо у відповідність кожній точці $x \in G(x_0)$ суміжний клас $gH \in G - H$, де g — елемент із G такий, що $x = gx_0$. Покажемо, що кожному елементу x_0 поставлено у відповідність єдиний суміжний клас, і цим задано деяке відображення $\varphi : G(x_0) \rightarrow G/H$. Справді, нехай для деякого елемента $g_1 \in G$ виконується умова $x = g_1x_0$. Тоді $g_1x_0 = gx_0$, звідки $(g^{-1}g_1)\ x_0 = x_0$, а отже, $g^{-1}g_1 \in St(x_0) = H$. Однак тоді $g_1 \in gH$, і, отже, $g_1\ H = gH$.

Покажемо тепер, що відображення $\varphi : G(x_0) \rightarrow G/H$ є взаємно однозначним, звідки і матимемо, що $|G(x_0)| = |G/H| = |G|/|H|$. Нехай $g_1, g_2 \in G$, $x_1 = g_1x_0, x_2 = g_2x_0, x_1 \neq x_2$. Покажемо, що $g_1H \neq g_2H$. Справді, якщо $g_1H = g_2H$, то $g_2^{-1}g_1H = H$, звідки $g_2^{-1}g_1 \in H$. Однак тоді $(g_2{}^{-1}g_1)\ x_0 = x_0$, а отже, $g_1x_0 = g_2x_0$, що суперечить умові $x_1 \neq x_2$.

Таким чином, доведено ін'єктивність відображення φ. Для доведення його сюр'єктивності залишилося зазначити, що $\forall gH \in G/H$ є образом точки $gx_0 \in G(x_0)$.

Твердження 8.15. *Нехай* $x_0 \in X$. *Тоді* $\forall x \in G(x_0) |St(x)| = |St(x_0)|$.

Нехай $x \in G(x_0)$. Тоді знайдеться $g_0 \in G$ таке, що $x = g_0x_0$. Розглянемо довільний елемент $g \in St(x)$. Тоді $gx = x$, а отже, $(gg_0)\ x_0 = g\ (g_0x_0) = gx = x = g_0x_0$, звідки $(g_0{}^{-1}gg_0)\ x_0 = x_0$, тобто $g_0^{-1}gg_0 \in St(x_0)$. Таким чином, $g_0^{-1}St(x)g_0 \subset St(x_0)$, звідки

$$g_0St(x_0)g_0^{-1} \supset St(x). \tag{8.27}$$

Нехай тепер g — довільний елемент з $St\ (x_0)$. Тоді $gx_0 = x_0$, і, отже, з урахуванням $x_0 = g_0{}^{-1}x$ маємо $(gg_0{}^{-1})\ x = g\ (g_0{}^{-1}x) = gx_0 = x_0 = g_0{}^{-1}x$, звідки $(g_0gg_0{}^{-1})\ x = x$, тобто $g_0gg_0^{-1} \in St(x)$. Таким чином, $g_0St(x_0)g_0^{-1} \subset St(x)$ і, використовуючи (8.27), дістаємо $g_0St\ (x_0)\ g_0{}^{-1} = St\ (x)$, звідки й випливає, що $|St(x)| = |St(x_0)|$.

Тепер доведемо лему Бернсайда.

Позначимо через X_G множину G-орбіт. Нехай також $\forall g \in G, \forall x \in X$, а

$$\alpha(gx = x) = \begin{cases} 1, & \text{якщо} \quad gx = x; \\ 0, & \text{якщо} \quad gx \neq x. \end{cases}$$

Тоді, використовуючи твердження 8.14 і 8.15, маємо

$$\sum_{g \in G} N(g) = \sum_{g \in G} \sum_{x \in X} \alpha(gx = x) = \sum_{x \in X} \sum_{g \in G} \alpha(gx = x) = \sum_{G(x_0) \in X_G} \sum_{x \in G(x_0)} \sum_{g \in G} \alpha(gx = x) =$$

$$= \sum_{G(x_0)\in X_G} \sum_{x\in G(x_0)} |St(x)| = \sum_{G(x_0)\in X_G} \sum_{x\in G(x_0)} |St(x_0)| = \sum_{G(x_0)\in X_G} |G(x_0)||St(x_0)| =$$

$$= \sum_{G(x_0)\in X_G} (|G|/|St(x_0)|) = |G||X_G| = |G|N,$$

звідки і випливає правильність формули, що доводиться.

8.2.4. Застосування леми Бернсайда для розв'язання комбінаторних задач

Приклади: 1. Складаються слова завдовжки 3 з літер *a* і *b*. Слова вважаються еквівалентними, якщо утворюються одне з одного зміною місцями крайніх літер (наприклад, *abb* ~ *bba*). Визначимо кількість *N* класів еквівалентності.

Нехай $X = \{c = c_1c_2c_3 \mid c_i \in \{a, b\}, i = 1, 2, 3\} = \{a, b\}^3$. Розглянемо групу $G = \{e, \sigma\}$, де e — одиничний елемент групи G, $\sigma \neq e$, $\sigma^2 = e$. Визначимо відображення $(g, c) \to gc$ прямого добутку GX в X. Нехай $\forall c \in X\ ec = c$, $\sigma c = \sigma c_1c_2c_3 = c_3c_2c_1$ (тобто e залишає слово c на місці, а σ міняє в слові c крайні літери місцями). Покажемо, що введене відображення задовольняє (8.21), (8.22).

Умова (8.21) виконується за означенням. Покажемо виконання умови (8.22). Розглянемо нетривіальний випадок із $g = h = \sigma$ (випадок з $g = e$ або $h = e$ очевидний). Тоді

$$\forall c \in X\ (\sigma\sigma)c = \sigma^2 c = ec = c,\ \sigma(\sigma c) = \sigma c_3c_2c_1 = c_1c_2c_3 = c\ ,$$

звідки і випливає правильність (8.22).

За умовою задачі шукана кількість *N* збігається з кількістю *G*-орбіт, а це означає, що, використовуючи лему Бернсайда, маємо $N = \frac{1}{2}(N(e) + N(\sigma))$. Зауважимо, що

$$N(e) = |X| = 2^3 = 8;$$

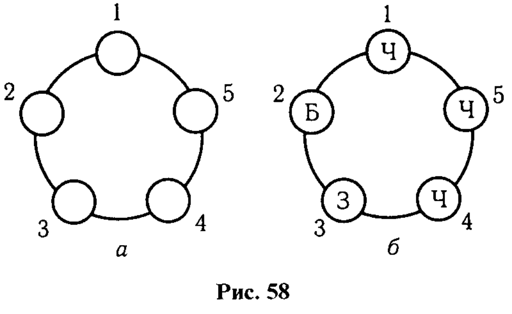
$$N(\sigma) = |\{c \in X \mid \sigma c_1c_2c_3 = c_1c_2c_3\}| = |\{c \in X \mid c_3c_2c_1 = c_1c_2c_3\}| =$$

$$= |\{c \in X \mid c_1 = c_3\}| = 2^2 = 4,$$

а отже, $N = \frac{1}{2}(8 + 4) = 6$.

2. Збирається намисто з плоских намистин трьох кольорів, при цьому забарвленим є тільки один бік намистин. Кожне намисто складається з п'яти намистин. Визначити кількість *N* різного намиста.

Пронумеруємо намистини в намисті, починаючи з деякої намистини і збільшуючи номери при обході намиста в напрямку, зворотному руху годинникової стрілки (рис. 58, *а*). Кожній намистині можна поставити у відповідність один із трьох можливих кольорів, наприклад: Ч (червоний), Б (блакитним), З (зелений). Тоді будь-яке розфарбування намиста з пронумерованими намистинами можна описати впорядкованим набором кольорів завдовжки 5, припускаючи, що *i*-й елемент цього на-



а *б*

Рис. 58

бору відповідає кольору i-ї намистини. Наприклад, упорядкована п'ятірка c = (Ч, Б, З, Ч, Ч) означає, що першу намистину забарвлено в червоний колір, другу — у блакитний і т. д. (рис. 58, *б*). Розглянемо множину X розфарбовувань намиста з пронумерованими намистинами:

$$X = \{c = (c_1, c_2, c_3, c_4, c_5) \mid c_i \in \{\text{Ч, Ч, З}\}\} = \{\text{Ч, Б, З}\}^5.$$

Очевидно, $|X| = |\{\text{Ч, Б, З}\}|^5 = 3^5 = 243$. Розглянемо також групу G обертання намиста на площині (навколо центра цього намиста), яке суміщає його з самим собою. При цьому під композицією $\alpha \circ \beta$ двох довільних обертань $\alpha, \beta \in G$ будемо розуміти результат послідовних обертань β, α, що застосовується до намиста (тобто спочатку виконується обертання β, а потім до одержаного результату застосовується обертання α). Очевидно, група G складається з п'яти елементів — обертань (проти руху годинникової стрілки) на кути $2\pi i/5$, де i = 0, 1, 2, 3, 4. Позначимо елементи групи G значеннями відповідних кутів.

Зазначимо, що група G, яка розглядається в цій задачі, є комутативною, що робить зайвим указівку, в якій послідовності треба здійснювати обертання в композиції $\alpha \circ \beta$. Однак в інших задачах група обертань може виявитися некомутативною, і тоді зазначення послідовності обертання при визначенні композиції $\alpha \circ \beta$ є необхідним (вибране означення композиції збережемо при розв'язуванні всіх подальших задач).

Поставимо у відповідність кожному обертанню $\alpha \in G$ підстановку τ_a на множині номерів намистин {1, 2, 3, 4, 5} таку, що для будь-якого $i \in \{1, 2, 3, 4, 5,\}$ $\tau_a(i)$ буде номером i-ї намистини намиста, одержаного внаслідок обертання α відносно початкового положення намиста (це можна уявити собі так: одне намисто закріплено на площині, і з ним поєднується деяке намисто, яке може пересуватися і до якого застосовується обертання α). Тоді $\tau_0 = e$, $\tau_{2\pi/5} = (1, 2, 3, 4, 5)$, $\tau_{4\pi/5} = (1, 3, 5, 2, 4)$, $\tau_{6\pi/5} =$ $= (1, 4, 2, 5, 3)$, $\tau_{8\pi/5} = (1, 5, 4, 3, 2)$. Позначимо $T = \{e, \tau_{2\pi/5}, \tau_{4\pi/5}, \tau_{6\pi/5}, \tau_{8\pi/5}\}$. Безпосередньою перевіркою переконуємося, що відображення $\alpha \to \tau_\alpha$ групи G на T є бієкцією. Зазначимо далі, що (за означенням композиції і відображення $\alpha \to \tau_\alpha$) виконується рівність

$$\forall \alpha, \beta \in G \ \tau_{\alpha \circ \beta} = \tau_\alpha \tau_\beta, \tag{8.28}$$

звідки випливає, що група G є ізоморфною групі $T \subset S_5$ (для доведення того, що T — група, можна скористатися формулою (8.28), а також тим, що $\tau_0 = e$). Проте тоді групи G і T будуть нерозрізненими і для простоти можна вважати, що $G = T$.

Означимо відображення $(g, c) \to gc$ прямого добутку GX в X. Нехай

$$\forall c \in X, \forall g \in G \ gc = g(c_1, c_2, c_3, c_4, c_5) = \left(c_{g^{-1}(1)}, c_{g^{-1}(2)}, c_{g^{-1}(3)}, c_{g^{-1}(4)}, c_{g^{-1}(5)}\right),$$

тобто $\forall c \in X, \forall \tau_a \in G \ \tau_\alpha c$ — розфарбування, яке виходить внаслідок обертання α намиста з розфарбуванням c. Наприклад, $\tau_{2\pi/5}\langle \text{Ч, Ч, Ч, Б, З} \rangle = \langle \text{З, Ч, Ч, Ч, Б} \rangle$ (рис. 59, *а* — розфарбування c = (Ч, Ч, Ч, Б, З); рис. 59, *б* — розфарбування $\tau_{2\pi/5}c$).

Покажемо, що введене відображення $(g, c) \to gc$ прямого добутку GX в X задовольняє (8.22) (виконання (8.21) є очевидним). Доведемо правильність (8.22). Нехай $\tau_a, \tau_\beta \in G$. Тоді $\tau_\alpha(\tau_\beta c) = \tau_\alpha(1)\left\langle c_{\tau_\beta^{-1}(1)}, \dots, c_{\tau_\beta^{-1}(5)} \right\rangle$. Позначимо $c_i' = c_{\tau_\beta^{-1}(1)}$, $i = 1, 2, 3, 4, 5$. Використовуючи введені позначення, дістаємо

$$\tau_\alpha(\tau_\beta c) = \tau_\alpha \langle c_1', \dots, c_5' \rangle = \left\langle c'_{\tau_\alpha^{-1}(1)}, \dots, c'_{\tau_\alpha^{-1}(5)} \right\rangle = \left\langle c'_{\tau_\beta^{-1}\left(\tau_\alpha^{-1}(1)\right)}, \dots, c'_{\tau_\beta^{-1}\left(\tau_\alpha^{-1}(5)\right)} \right\rangle =$$

$$= \left\langle c'_{(\tau_\alpha \tau_\beta)^{-1}(1)}, \dots, c'_{(\tau_\alpha \tau_\beta)^{-1}(5)} \right\rangle = (\tau_\alpha \tau_\beta)c,$$

тобто умова (8.22) виконується.

Повернемося до початкового намиста з непронумерованими намистинами. Їх кількість N дорівнює кількості орбіт при дії групи G на множині X, тобто за лемою Бернсайда маємо

$$N = \frac{1}{|G|}\sum_{g\in G} N(g) = \frac{1}{5}\left[N(e) + N(\tau_{2\pi/5}) + N(\tau_{4\pi/5}) + N(\tau_{6\pi/5}) + N(\tau_{8\pi/5})\right].$$

Зауважимо, що обертання на кут 0 залишає на місці будь-яке розфарбування, а отже, $N(e) = |X| = 3^5 = 243$. Обертання на кут $2\pi/5$ залишає на місці розфарбування, при яких намистину з номером i забарвлено так само, як і намистину з номером $\tau_{2\pi/5}^{-1}(i)$, $i = 1, 2, 3, 4, 5$, тобто якщо $c_1 = c_5 = c_4 = c_3 = c_2$ (всі намистини забарвлено в один колір). Очевидно, таких розфарбувань може бути три, а значить, $N(\tau_{2\pi/5}) = 3$. Аналогічно знаходимо $N(\tau_{2\pi/5}) = 3, i = 2, 3, 4$, і, отже, $N = \frac{1}{5}(243 + 4\cdot 3) = 255/5 = 51$.

3. Нехай $\Omega = \{1, 2, ..., n\}$ — множина номерів елементів деякої фігури Φ; R — скінченна множина кольорів, в які можуть бути забарвлені елементи фігури Φ; $X = \{(c_1, ..., c_n) \mid c_i \in R, i = 1, 2, ..., n\} = R^n$ — множина розфарбувань фігури Φ. Нехай, далі, σ — деяка підстановка з S_n (тобто $\sigma : \Omega \to \Omega$). Визначимо дію підстановки σ на довільне розфарбування $c \in X$ за формулою

$$\sigma c = \sigma(c_1, ..., c_n) = (c_{\sigma^{-1}(1)}, ..., c_{\sigma^{-1}(n)}). \tag{8.29}$$

Виведемо формулу для $N(\sigma)$ — кількості розфарбувань, які залишаються на місці при дії на них підстановки σ. Розкладемо σ на добуток незалежних елементарних циклів

$$\sigma = \sigma_1, ..., \sigma_k, \tag{8.30}$$

де k — кількість циклів (у (8.30) враховуються також усі цикли завдовжки 1).

Нагадаємо, що кожному елементарному циклу $(i_1, i_2, \ldots, i_m)$ відповідає σ-орбіта $\{i_1, i_2, \ldots, i_m\}$, і при цьому сукупність усіх σ-орбіт є розбиттям множини $\{1, 2, \ldots, \mathrm{n}\}$. Нехай c — деяке розфарбування фігури Φ, яке залишається на місці при дії на нього підстановки σ. Для будь-якого циклу $\sigma_j(i_1, i_2, ..., i_m)$, що входить у розкладання (8.30), маємо

$$i_1 = \sigma^{-1}(i_2), i_2 = \sigma^{-1}(i_3), ..., i_{m-1} = \sigma^{-1}(i_m),$$

звідки, використовуючи ту обставину, що $c = \sigma c$, з урахуванням (8.29) дістаємо

$$c_{i_m} = c_{\sigma^{-1}(i_m)} = c_{i_{m-1}} = c_{\sigma^{-1}(i_{m-1})} = c_{i_{m-2}} = ... = c_{\sigma^{-1}(i_2)} = c_{i_1},$$

а отже, елементи фігури Φ, що входять в одну й ту саму σ-орбіту, мають бути забарвлені однаково. Оскільки σ-орбіти попарно не перетинаються, вони можуть бути забарвлені незалежно одна від одної. Проте тоді за правилом добутку $N(\sigma) = |R|^k$.

4. Складається намисто з плоских намистин трьох кольорів, забарвлених однаково з обох боків. Кожне намисто складається з п'яти намистин. Визначити кількість N різного намиста.

Порівняно з прикладом 2 на с. 223 до групи обертань G фігури, зображеної на рис. 58, *а*, додаються осьові перетворення симетрії. У цієї фігури, очевидно, п'ять осей симетрії: $\ell_1, \ell_2, \ell_3, \ell_4, \ell_5$, де $\forall i \in \{1, 2, 3, 4, 5\}$; ℓ_1 — пряма, що проходить через i-ту намистину і центр кола, на якому розташовано намистини в намисті. Тоді,

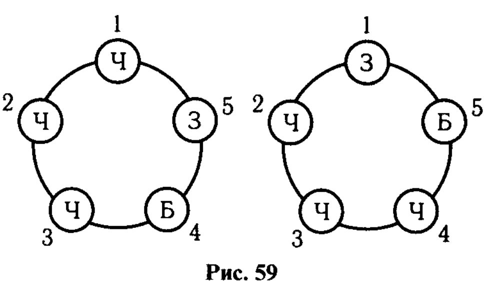


Рис. 59

наприклад, перетворенню симетрії відносно осі ℓ_1 відповідає підстановка $\sigma_1 = (1)(2\ 5)(3\ 4)$ й одержуємо $N(\sigma_1) = 3^3 = 27$. Аналогічно розглядаються перетворення симетрії відносно інших осей. З урахуванням леми Бернсайда знаходимо $N = \frac{1}{10}\left(3^5 + 4\cdot 3 + 5\cdot 3^3\right) = 39$.

8.2.5. Цикловий індекс групи, що діє на множині

Нехай G — скінченна група, а X — скінченна множина, де $|X| = n$. Нехай також τ — деяка реалізація групи G на множині в $S(X)$, що визначає дію групи G на множині X (тобто така, яка задає відображення $(g, x) \to gx$ прямого добутку GX в X, що задовольняє (8.21), (8.22)). Для будь-якого елемента $g \in G$ позначимо через $j_k(g)$ кількість циклів завдовжки k в розкладанні підстановки $\tau_g \in S(X)$ на добуток незалежних елементарних циклів, де $k = 1, 2, \ldots, n$. Кожному елементу $g \in G$ поставимо у відповідність вагу

$$\omega_G(g) = t_1^{j_1(g)}\ t_2^{j_2(g)} \ldots t_n^{j_n(g)}, \tag{8.31}$$

тобто елемент кільця $Z[t_1, \ldots, t_n]$. Тоді *цикловий індекс* $P(G, X, t_1, \ldots, t_n)$ групи G, що діє на X, є многочленом змінних $t_1, \ldots, t_n$, який визначається виразом

$$P(G, X, t_1, \ldots, t_n) = \frac{1}{|G|}\sum_{g\in G}\omega_g(g) = \frac{1}{|G|}\sum_{g\in G} t_1^{j_1(g)}\ t_2^{j_2(g)} \ldots t_n^{j_n(g)}. \tag{8.32}$$

Приклади: 1. Нехай $X = \{1, 2, \ldots, 10\}$; G — група, що діє на X, і для деякого елемента $g \in G$ виконується рівність

$$\tau_g = \begin{pmatrix} 1 & 3 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 9 & 6 & 1 & 8 & 5 & 2 & 7 & 10 & 3 & 4 \end{pmatrix}.$$

Визначимо вагу $\omega\{g\}$ за формулою (8.31). Маємо $\tau_g = (1\,9\,3)(2\,6)(4\,8\,10)(5)(7)$, звідки $w_G(g) = t_1^2 t_2^1 t_3^2 t_4^0 \ldots t_{10}^0 = t_1^2 t_2 t_3^2$.

2. Нехай G — група обертань на площині фігури Ф (рис. 60) навколо центра цієї фігури, що суміщують її з самою собою; $X = \{1, 2, 3, 4, 5\}$ — множина номерів елементів фігури. Під композицією $\alpha \circ \beta$ двох довільних обертань $\alpha, \beta \in G$ будемо розуміти обертання, яке є результатом послідовних обертань α, β, що застосовуються до фігури *Ф* (тобто спочатку виконується обертання β, а потім до утвореної внаслідок обертання β фігури застосовується обертання α).

Очевидно, група G складається з чотирьох елементів — обертань (проти руху годинникової стрілки) на кути $2\pi i/4$, $i = 0, 1, 2, 3$. Позначимо елементи групи G значеннями відповідних кутів. Поставимо у відповідність кожному обертанню $\alpha \in G$ підстановку $\tau_\alpha \in S(X) = S_5$ таку, що для будь-якого $i \in X$ $\tau_\alpha(i)$ буде номером i-го елемента фігури, утвореної внаслідок обертання α відносно початкового положення фігури. Тоді $\tau_0 = e, \tau_{\pi/2} = (1\,4\,3\,2)(5), \tau_p = = (1\,3)(2\,4)(5), \tau_{3\pi/2} = (1\,2\,3\,4)(5)$.

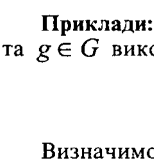


Рис. 60

Зазначимо, що за означенням композиції обертань і відображення $\alpha \to \tau_\alpha$ групи G в S_5 виконується рівність $\forall \alpha, \beta \in G\ \tau_{\alpha\circ\beta} = \tau_\alpha, \tau_\beta$, звідки випливає, що відображення $\tau : G \to S_5$, яке ставить у відповідність кожному елементу $\alpha \in G$ підстановку $\tau_\alpha \in S_5$, є гомоморфізмом, тобто τ — реалізація групи G в $S(X)$. При цьому у відповідності з (8.32) цикловий індекс групи G, що діє на множині X, становить

$$P(G,X,t_1,t_2,t_3,t_4,t_5)=\frac{1}{|G|}\sum_{g\in G}\omega_G(g)=\frac{1}{4}(\omega_G(0)+\omega_G(\pi/2)+\omega_G(\pi)+$$

$$+\omega_G(3\pi/2))=\frac{1}{4}\left(t_1^5+t_1t_4+t_1t_2^2+t_1t_4\right)=\frac{1}{4}\left(t_1^5+2t_1t_4+t_1t_2^2\right). \tag{8.33}$$

8.2.6. *G*-еквівалентні відображення

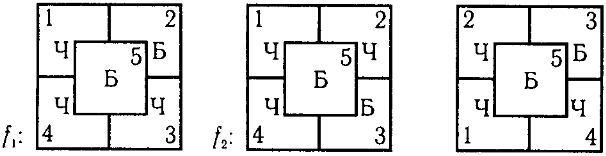
Припустимо, що зберігаються умови, описані в п. 8.2.5. Нехай, R — деяка скінченна множина. Дві функції $f_1, f_2 \in R^X$ будемо називати G-еквівалентними й позначати $f_1 \sim f_2$, якщо $\exists g \in G$ таке, що $\forall x \in X\ f_1(x) = f_2(gx)$, або, що те саме, якщо $\exists g \in G$ таке, що $\forall x \in X\ f_1(x) = f_2\left(g^{-1}x\right)$.

Очевидно, введене бінарне відношення ~ на множині R^X є еквівалентністю, а отже, воно породжує розбиття множини R^X на класи еквівалентності $F \in R^X/\sim$.

Нехай кожному елементу $r \in R$ надано деяку вагу $\omega_R(r) \in K$, де K — кільце многочленів над Z від деяких змінних. Тоді вага функції $f \in R^X$ за означенням $\omega(f) = \prod\limits_{x\in X}\omega_R(f(x))$. Якщо $f_1 \sim f_2$, то $\omega(f_1) = \omega(f_2)$; тому можна визначити вагу класу еквівалентності $\omega(F)$, де $F \in R^X/\sim$, як вагу $\omega(f)$ будь-якого елемента $f \in F$.

Приклад. Нехай R = {червоний, блакитний} — множина кольорів; $X = \{1, 2, 3, 4, 5\}$ — множина елементів фігури Φ, розглянутої у попередньому прикладі. Тоді будь-яке відображення $f: X \to R$ можна розглядати як розфарбування елементів фігури Φ у кольори з R. Надамо червоному кольору вагу a, блакитному — вагу b (зазначимо, що a, b — елементи кільця $Z[a, b]$). Тоді, наприклад, будь-яке розфарбування f таке, що два елементи фігури Φ будуть забарвлені в червоний колір, а решта три — в блакитній, має вагу $\omega(f) = a^2b^3$.

Нехай, далі, на множині X діє група G, взята з попереднього прикладу. Тоді відповідно до введеного вище означення розфарбування f_1, f_2 є еквівалентними, якщо $\exists\alpha \in G: \forall i \in X\ f_1(i) = f_2\left(\alpha^{-2}i\right)$, тобто якщо знайдеться таке обертання $\alpha \in G$, яке при поєднанні фігури Φ із розфарбуванням f_1 з фігурою, що утворюється з фігури Φ із розфарбуванням f_2 обертанням на кут α (проти руху годинникової стрілки), покаже, що суміщені один з одним їхні елементи будуть забарвлені однаково. Наприклад, роз-



а *б* *в*

Рис. 61

фарбування f_1, f_2, показані на рис. 61, *а, б* — еквівалентні, оскільки при поєднанні фігури Φ з розфарбуванням f_1 із фігурою, яка утворюється з фігури Φ із розфарбуванням f_2 обертанням на кут $\pi/2$ (проти руху годинникової стрілки), елементи, що суміщуються, будуть розфарбовані однаково (див. на рис. 61, *в* результат обертання на кут $\pi/2$ фігури Φ із розфарбуванням f_2 у порівнянні з рис. 61, *а*). При цьому $\forall i \in X\ f_1(i) = f_2\left((\pi/2)^{-1} i\right)$.

8.2.7. Твірна функція запасу класів еквівалентності

Припустимо, що зберігаються умови, описані в п. 8.2.5 і 8.2.6. Нехай запас — множина класів еквівалентності $F \in R^X / \sim$. Виникає питання: яка твірна функція запасу $\sum\limits_F \omega(F)$?

Знаючи твірну функцію запасу, можна розв'язувати деякі досить складні комбінаторні задачі. Справді, нехай, наприклад, $R = \{r_1, r_2\}$, $\omega(r_1) = a, \omega(r_2) = b$ і відомо твірну функції запасу, подану у вигляді

$$\sum_F W(F) = \sum_{m+k=n} c_{mk} a^m b^k$$

(тобто її зведено до вигляду многочлена, де підсумовування у виразі праворуч проводиться за всіма цілими невід'ємними m, k, що задовольняють рівність $m + k = n$). Однак тоді відомо також усі числа c_{mk}, які виражають кількість класів еквівалентності вагою $a_m b_m$.

Ці числа дають відповіді на багато прикладних комбінаторних задач (зокрема, якщо r_1, r_2 — кольори), даючи змогу визначити кількість різних (попарно нееквівалентних) розфарбувань при заданій комбінації кольорів. Так, в умовах прикладу на с. 227 $C_{3,2}$ виражає кількість різних розфарбувань фігури Φ таких, що 3 елементи забарвлено в червоний колір, а 2 — у блакитний; $c_{3,2} + c_{2,3} + c_{1,4} + c_{0,5}$ — кількість різних розфарбувань фігури Φ таких, що принаймні 2 елементи забарвлено в блакитний колір.

Для практичної побудови твірної функції запасу класів еквівалентності скористаємося такою теоремою.

Теорема 8.1 (теорема Поя). *Твірна функція запасу класів еквівалентності задовольняє рівність*

$$\sum_F \omega(F) = P\left(G, X, \sum_{r\in R} \omega_R(r), \sum_{r\in R} [\omega_R(r)]^2, \ldots, \sum_{r\in R} [\omega_R(r)]^n \right),$$

де $P(G, X, t_1, t_2, \ldots, t_n)$ — цикловий індекс групи G, що діє на X.

Наслідок 8.5. *Якщо ваги* $\omega_R(r)$, $r \in R$, *дорівнюють* 1, *то кількість класів еквівалентності становить*

$$P(G, X, |R|, \ldots, |R|). \tag{8.34}$$

Приклад. На підставі попередніх прикладів, визначимо твірну функцію запасу класів еквівалентності. Внаслідок теореми Поя, застосовуючи формулу (8.33), маємо

$$\sum_F \omega(F) = P\left(G, X, a+b, a^2+b^2, a^3+b^3, a^4+b^4, a^5+b^5\right) =$$
$$= \frac{1}{4}\left[(a+b)^5 + 2(a+b)\left(a^4+b^4\right) + (a+b)\left(a^2+b^2\right)^2\right]. \tag{8.35}$$

Скориставшись рівністю (8.35), визначимо, скількома попарно нееквівалентними способами можна розфарбувати фігуру Φ так, щоб три її елементи були забарвлені в червоний колір, а два — у блакитний. Для цього знайдемо коефіцієнт c при членові ca^3b^2 у твірній функції запасу класів еквівалентності, зведеній до вигляду многочлена. Очевидно, $c = \frac{1}{4}\left(C_5^3 + 2\right) = \frac{1}{4}(10+2) = 3$, тобто існують три попарно нееквівалентних способи розфарбування із заданою комбінацією кольорів. Використовуючи тепер наслідок із теореми Поя, знаходимо, що загальна кількість попарно нееквівалентних способів розфарбування, які виражаються формулою (8.34), — однакова.

8.2.8. Обґрунтування теореми Поя

Для доведення теореми Поя будуть потрібні два допоміжних твердження. Розглянемо будь-яке натуральне число ℓ і позначимо $I_\ell = \{1, 2, ..., \ell\}$.

Твердження 8.16. *Нехай K — комутативне кільце; $a_{ij} \in K$, $i = 1, \ldots k$, $j = 1, \ldots, m$. Тоді*

$$\prod_{i=1}^{k} \sum_{j=1}^{m} a_{ij} = \sum_{\varphi \in I_m^{I_k}} \prod_{i=1}^{k} a_{i,\varphi(i)}, \tag{8.36}$$

де $I_m^{I_k}$ — множина всіх відображень $\varphi : I_k \to I_m$.

Доведення проведемо методом індукції за k. Очевидно, при $k = 1$ рівність (8.36) виконується. Нехай вона є правильною також при $k - 1$ (де $k \geq 2$). Покажемо її виконання при k:

$$\prod_{i=1}^{k} \sum_{j=1}^{m} a_{ij} = \left(\prod_{i=1}^{k-1} \sum_{j=1}^{m} a_{ij}\right)\left(\sum_{j=1}^{m} a_{kj}\right) = \left(\sum_{\varphi \in I_m^{I_{k-1}}} \prod_{i=1}^{k-1} a_{i,\varphi(i)}\right)\left(\sum_{j=1}^{m} a_{kj}\right) =$$
$$= \sum_{\varphi \in I_m^{I_{k-1}}} \sum_{j=1}^{m} a_{kj} \prod_{i=1}^{k-1} a_{i,\varphi(i)} = \sum_{\varphi \in I_m^{I_k}} \prod_{i=1}^{k} a_{i,\varphi(i)}.$$

Твердження 8.17. *Нехай $X_1, \ldots, X_k$ — непорожні підмножини множини X, що створюють розбиття множини X, тобто*

$$\bigcup_{i=1}^{k} X_i = X,\ X_i \cap X_j = \varnothing \text{ при } i \neq j.$$

Нехай далі $n_i = |X_i|$, $i = 1, ..., k$. Тоді твірна функція запасу функцій з R^X, які набувають сталих значень на кожному з X_i, дорівнює

$$\prod_{i=1}^{k} \sum_{r \in R} [\omega_r(R)]^{n_i}. \tag{8.37}$$

Розглянемо клас функцій $R^{\bar{X}}$, де $\bar{X} = \{X_1, ..., X_k\}$. Визначимо ваги $\omega(\varphi)$ функцій $\varphi \in R^{\bar{X}}$ за формулою

$$\omega(\varphi) = \prod_{i=1}^{k} \left[\omega_R(\varphi(X_i))\right]^{n_i} .$$

Очевидно, між функціями запасу, що розглядаються, і функціями з $R^{\bar{X}}$ можна встановити взаємно однозначну відповідність. Для цього кожній функції $f : X \to R$ запасу, яка розглядається, поставимо у відповідність функцію $\varphi_f \in R^{\bar{X}}$ таку, що $\forall i \in \{1, ..., k\} \varphi_f(X_i) = f(x_i)$, де $x_i \in X_i$. При такій відповідності, очевидно, $\omega(\varphi_f) = \omega(f)$, а отже, твірна функція запасу, яка розглядається, збігається з твірною функцією запасу $R^{\bar{X}}$. Однак тоді для доведення твердження 8.17 залишається помітити, що внаслідок твердження 8.16 твірна функція запасу $R^{\bar{X}}$ визначається виразом (8.37) (тут $a_{ij} = \left[\omega_R(r_j)\right]^{n_i}$ де $R = \{r_1, ..., r_m\}$).

Тепер доведемо теорему Поя.

Нехай D — множина значень ваги елементів $f \in R^{\bar{X}}$ й $H(d) = \left\{f \in R^{\bar{X}} \mid \omega(f) = d\right\}$, де $d \in D$. Нехай група G діє на $H(d)$ і при цьому $\forall f \in H(d) gf$ визначається за правилом

$$(gf)(x) = f\left(g^{-1}x\right), x \in X .$$

Зауважимо, що $\forall f \in H(d), \forall g \in G\ gf \sim f$, а отже, $\forall f \in H(d), \forall g \in G$, $gf \in H(d)$. Покажемо, що введене відображення $<g, f> \to gf$ прямого добутку $G \times H(d)$ в $H(d)$ справді задовольняє умови (8.21), (8.22). Умова (8.21) очевидним чином виконується. Покажемо правильність (8.22). Дійсно, $\forall g, h \in G$, а позначивши $\tilde{f} = hf$, дістанемо

$$[(gh)f](x) = f\left((gh)^{-1}x\right) = f\left(h^{-1}\left(g^{-1}x\right)\right) = \tilde{f}\left(g^{-1}x\right) = \left(g\tilde{f}\right)(x) = [g(hf)](x), \quad x \in X.$$

Використовуючи лему Бернсайда, маємо

$$\forall d \in D \left|H(d)/\sim\right| = \frac{1}{|G|} \sum_{g \in G} \left| \{f \in H(d) | gf = f\}\right|. \tag{8.38}$$

Зазначимо, що $\forall F \in R^X/\sim F \subseteq H(d)$, де $d = \omega(F)$, звідки

$$\forall F \in R^X/\sim F \in H(\omega(F))/^\wedge . \tag{8.39}$$

Нехай $d \in D$. З урахуванням (8.39) $\left\{F \in R^X/\sim \mid \omega(F) = d\right\} \subseteq H(d)/\sim$. Покажемо також протилежне включення. Нехай $f_1 \in H(d)/\sim$. Розглянемо довільну функцію $f \in F$, а також клас еквівалентності $F \in R^X/\sim$ такий, що $f \in F$. Тоді $\omega(F) = \omega(f) = d$, а отже, з урахуванням (8.39) $F \in H(d)/\sim$. Таким чином, $F, F_1 \in H(d)/\sim, F \cap F_1 \neq \varnothing$, звідки, $F = F_1$, тобто $F_1 \in \left\{F \in R^X/\sim \mid \omega(F) = d\right\}$. Отже, ми довели, що

$$\forall d\in D\left\{F\in R^X/\sim\mid\omega(F)=d\right\}=H(d)/\sim. \qquad (8.40)$$

Використовуючи тепер той факт, що $H(d_1)\cap H(d_2)=\varnothing$ при $d_1\neq d_2$, з урахуванням (8.40) маємо $\{H(d)/\sim\mid d\in D\}$ — розбиття множини $R^X/\sim$, звідки, скориставшись (8.38), а також тим, що $\forall F\in H(d)/\sim\ \omega(F)=d$, дістаємо

$$\sum_{F\in R^X/\sim}\omega(F)=\sum_{d\in D}\sum_{F\in H(d)/\sim}\omega(F)=\sum_{d\in D}d\left|H(d)/\sim\right|=$$

$$=\sum_{d\in D}\frac{1}{|G|}\sum_{g\in G}d\left|\{f\in H(d)|gf=f\}\right|= \qquad (8.41)$$

$$=\sum_{d\in D}\frac{1}{|G|}\sum_{g\in G}\sum_{f\in\{f\in H(d)\ |gf=f\}}\omega(f)=\frac{1}{|G|}\sum_{g\in G}\sum_{d\in D}\sum_{f\in\{f\in H(d)\ |gf=f\}}\omega(f)=$$

$$=\frac{1}{|G|}\sum_{g\in G}\sum_{f\in\left\{f\in R^X\ |gf=f\right\}}\omega(f).$$

Зауважимо, далі, що $\forall g\in G$ підстановка τ_g розбиває множину X на τ_g-орбіти $X^{(g)_1},\ldots,X^{(g)_{k(g)}}$, де $k(g)$ — кількість орбіт. Позначимо $n_i(g)=n_i(g)=\left|X^{(g)_i}\right|$, де $i=1,\ldots,k(g)$. Тепер видно, що

$$gf=f\Leftrightarrow\forall i\in\{1,\ldots,k(g)\}\exists c_i\in R:\forall x\in X^{(g)_i}f(x)=c_i.$$

Однак тоді, використовуючи твердження 8.17, з урахуванням (8.41), знаходимо

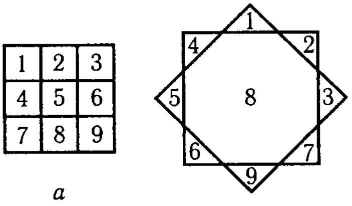
$$\sum_{F\in R^X/\sim}\omega(F)=\frac{1}{|G|}\sum_{g\in G}\sum_{f\in\left\{f\in R^X\ |gf=f\right\}}\omega(f)=\frac{1}{|G|}\sum_{g\in G}\prod_{i=1}^{k(g)}\sum_{r\in R}[\omega_R(r)]^{n_i(g)}=$$

$$=\frac{1}{|G|}\sum_{g\in G}\prod_{i=1}^{n}\left\{\sum_{r\in R}[\omega_R(r)]^i\right\}^{j_i(g)}=$$

$$=P\left(G,X,\sum_{r\in R}\omega_R(r),\sum_{r\in R}[\omega_R(r)]^2,\ldots,\sum_{r\in R}[\omega_R(r)]^n\right).$$

КОНТРОЛЬНІ ЗАПИТАННЯ ТА ЗАДАЧІ

1. Скількома способами можна дати клички чотирьом щенятам, маючи сім можливих варіантів (щенята названо по-різному)?
2. Скількома способами можна розфарбувати квадрат, поділений на дев'ять частин (див. рис. 55), п'ятьма кольорами, допускаючи фарбування різних частин в один колір?



а

б

Рис. 62

3. Скількома способами можна розмістити 10 різних куль у трьох різних урнах?
4. Обґрунтувати (без використання формул) рівність $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$.
5. Визначити кількість прямокутних матриць розміром $m \times n$ елементів із $\{0,1\}$ із попарно різними рядками $\left(m \le 2^n\right)$.
6. З колоди, що складається із 52 карт, вибрали 10 карт. Визначити, у скількох випадках серед них виявляться: а) пікова дама; б) всі чотири королі; в) всі карти однієї масті; г) жодного туза; д) рівно один туз; е) хоча б один туз; ж) рівно два тузи.
7. Є монети по 1, 2 і 3 коп.; всього 20 монет. Скільки існує різних комбінацій монет?
8. Скількома способами можна розмістити 20 однакових куль у чотирьох різних урнах?
9. Визначити кількість цілочислових розв'язків системи $x_1 + x_2 + x_3 = 40, x_1 \ge$ $\ge 3, x_2 \ge 0, x_3 \ge 2$.
10. Скількома способами можна розмістити 20 різних куль у трьох різних урнах так, щоб у першій, другій і третій урнах знаходилося відповідно 5, 3 та 12 куль?
11. Скількома способами групу з 25 чоловік можна поділити на сім коаліцій: дві — по 5 чоловік, одна — 7 чоловік, чотири — по 2 чоловіки?
12. Визначити коефіцієнт k членів многочлена (зі зведеними подібними членами), що утворюються з алгебричного виразу $(a + b + c)^2 (a^2 + b^2 + c^2)^4$: а) $ka^3b^3c^4$; б) $ka^2b^4c^4$; в) ka^5b^5; г) $ka^2b^2c^6$.
13. Визначити кількість цілочислових розв'язків системи $x_1 + x_2 + x_3 = 40,\ 4 \le \sigma_1 \le$ $\le 15, 9 \le \sigma_2 \le 18, 5 \le \sigma_3 \le 16$.
14. Визначити кількість шестизначних чисел, в яких сума перших трьох цифр збігається з сумою останніх трьох цифр.
15. Знайти кількість цілих додатних чисел, які не перевищують 200 і не діляться на жодне з простих чисел: а) 2, 3, 5; б) 7, 11, 13.
16. На квадратних аркушах паперу пишуть п'ятизначні числа від 0 до 99999. Якщо в деякому числі менше п'яти значущих цифр, то його доповнюють ліворуч нулями (наприклад, пишуть 00536 замість 536). Вважається, що після обертання аркуша на кут цифри 0, 1, 8 не міняються, а цифри 6, 9 переходять одна в одну. Скільки треба мати аркушів паперу для запису всіх чисел з урахуванням можливості обертання аркушів?
17. Складається намисто з плоских намистин трьох кольорів, забарвлених однаково з обох боків. Кожне намисто складається з шести намистин. Визначити кількість різного намиста.
18. Визначити, скількома способами можна розфарбувати трьома кольорами пронумеровані елементи фігури *Ф*. Два способи розфарбування вважаються однаковими, якщо один виходить з іншого обертанням фігури на площині (навколо центра). Варіанти фігури *Ф* показано на рис. 62.
19. Визначити, скількома способами можна розфарбувати трьома кольорами (червоним, блакитним, зеленим) пронумеровані елементи фігури *Ф*, припускаючи, що n_1 елементів мають бути забарвлені в червоний колір, n_2 – у блакитний, n_3 — в зелений. Два способи розфарбування вважаються однаковими, якщо один виходить з іншого обертанням фігури на площині (навколо центра). Варіанти фігури *Ф* показано на рис. 62. Відповідні набори значень n_1, n_2, n_3 становлять $n_1 = 3$, $n_2 = 4$, $n_3 = 2$ або $n_1 = 1$, $n_2 = 4$, $n_3 = 4$.

20. Розв'язати задачі 18 і 19 у припущенні, що два способи розфарбування вважаються однаковими, якщо один випливає з іншого обертанням фігури Φ на площині (навколо центра) або внаслідок осьового перетворення симетрії.

ПЕРЕЛІК ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

1. Застосування основних комбінаторних схем для розв'язання задач.
2. Застосування поліномних твірних функцій для розв'язання задач.
3. Розв'язання задач на розбиття.
4. Застосування формули включень і виключень для розв'язання задач.
5. Розв'язання комбінаторних задач методом Поя.

СПИСОК ЛІТЕРАТУРИ

1. *Яблонский С. В.* Введение в дискретную математику. — М.: Наука, 1989. — 384 с.
2. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера. — М.: Энергоатомиздат, 1988. — 480 с.
3. *Нефедов В. Н., Осипова В. А.* Курс дискретной математики. — М.: Изд-во МАИ, 1992. — 264 с.
4. *Сигорский В. П.* Математический аппарат инженера. — К.: Техніка, 1977. — 768 с.

РОЗДІЛ 9

ГРАФИ

Існує кілька причин зростання інтересу до теорії графів. Незаперечним є той факт, що теорія графів застосовується в таких сферах, як фізика, хімія, теорія зв'язку, проектування обчислювальних машин, електротехніка, машинобудування, архітектура, дослідження операцій, генетика, психологія, соціологія, економіка, антропологія і лінгвістика. Ця теорія також тісно пов'язана з багатьма розділами математики, серед яких теорія груп, теорія матриць, чисельний аналіз, теорія ймовірностей, топологія та комбінаторний аналіз. Вірогідним є й те, що теорія графів слугує математичною моделлю для кожної системи, що містить бінарне відношення. Графи діють привабливо і мають естетичну привабливість завдяки їх поданню у вигляді діаграм. Хоч у теорії графів є багато результатів, елементарних за своєю природою, в ній також — величезна кількість надто тонких комбінаторних проблем, гідних уваги найдосвідченіших математиків.

Ф. Харарі

9.1. Основні поняття теорії графів

Нехай V — довільна множина, E — деяка сукупність пар вигляду $\left(v_i, v_j\right)$, де $v_i, v_j \in V$. Термін сукупність означає можливість наявності однакових пар.

Упорядкована пара $G = (V, E)$, що складається з множини V та сукупності E, називається графом із множиною вершин V і множиною ребер E. Як зазначено в епіграфі, графи зручно зображувати графічно, що і спричинило появу їхньої назви (рис. 63). При цьому елементи множини V зображають крапками на площині, а ребра $\left(v_i, v_j\right)$ — відрізками (прямолінійними або криволінійними), які з'єднують крапки v_i та v_j. Граф називається скінченним, якщо множини його вершин і ребер є скінченними. Множину вершин графа G позначають $V(G)$, а множину ребер — $E(G)$. Кількість вершин графа $n(G) = |V(G)|$, а кількість ребер $m(G) = |E(G)|$.

Кількість вершин $n(G)$ графа називають його порядком.

Кількість ребер графа, інцидентних деякій вершині v, називається локальним степенем, або просто степенем, вершини v і позначається $\rho(v)$. Якщо $e = (v, w) \in E(G)$, то кажуть:

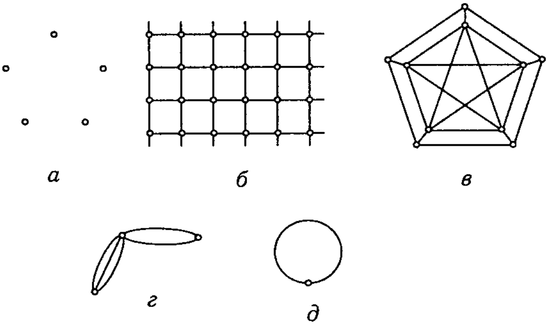


Рис. 63

- вершини v та w суміжні;
- вершини v і w є кінцями ребра e;
- вершини v та w — інцидентні ребру e;
- ребро e — інцидентне вершинам v і w.

Два ребра називаються суміжними, якщо обидва вони є інцидентними одній вершині.

Множина ребер E може бути порожньою (рис. 63, *а*). Такий граф називається нуль-графом і позначається $\varnothing$. Якщо ж множина вершин V — порожня, то порожньою є також множина E. Такий граф називається порожнім. Лінії, що зображають ребра графа, можуть перетинатися, але точки перетину не є вершинами (рис. 63, *в*); різні ребра можуть бути інцидентними одній і тій самій парі вершин (рис. 63, *г*), такі ребра називаються кратними. Цей випадок відповідає наявності кількох однакових пар $(v_i, v_j) \in E(G)$. Граф, що містить кратні ребра, називається мультиграфом. Ребро може з'єднувати деяку вершину саму з собою (рис. 63, *д*), таке ребро називається петлею. Цей випадок відповідає наявності в множині E пар вигляду (v, v). Граф із петлями та кратними ребрами називається псевдографом.

На рис. 63, *б* зображено фрагмент нескінченного графа. Його вершини — це точки площини з цілими координатами (x, y), а ребра, що з'єднують їх, — горизонтальні й вертикальні відрізки.

Графи, які найчастіше розглядаються, є скінченними, тобто скінченні множини їхніх елементів (вершин і ребер). Їх і розглядатимемо. Проте деякі поняття та результати, про які йтиметься, належать до довільних графів. Скінченний неорієнтований граф без петель і кратних ребер називається звичайним.

Якщо пари $(v, w) \in E$ вважаються впорядкованими, то граф є орієнтованим. Інакше граф називається неорієнтованим. Ребра орієнтованого графа прийнято називати дугами та зображати напрямленими відрізками, як на рис. 64.

При зображенні орієнтованих графів напрямки ребер позначаються стрілками (рис. 64, *а*). Орієнтований граф може мати кратні ребра (рис. 64, *б*), петлі (рис. 64, *в*), а також петлі, що з'єднують одні й ті самі вершини ребра, але у зворотних напрямках (рис. 64, *г*).

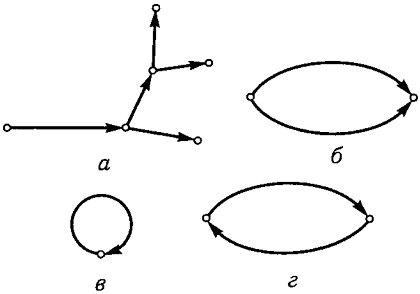


Рис. 64

Кожному неорієнтованому графу можна поставити у відповідність орієнтований граф з тією самою множиною вершин, в якій кожне ребро замінено двома орієнтованими ребрами, що є інцидентними тим самим вершинам і мають зворотні напрямки. Таку відповідність будемо називати канонічною.

Граф, що має як ребра, так і дуги, називається мішаним.

Про дугу (v, w) кажуть, що вона виходить із вершини v та входить у вершину w. Іноді вершини v і w називають відповідно початком та кінцем дуги (v, w). Домовимося позначати орграфи літерою D або D з індексами.

9.2. Задання графа за допомогою матриці інцидентності та списку ребер

Задати граф означає задати множини його вершин і ребер, а також відношення інцидентності. Коли граф G — скінченний, для опису його вершин та ребер досить їх занумерувати. Нехай $v_1, v_2, ..., v_n$ — вершини графа G; $e_1, e_2, ..., e_m$ — його ребра. Відношення інцидентності можна означити матрицею $\mathrm{E} = \left\| \varepsilon_{ij} \right\|$, що має m рядків й n стовпців. Стовпці відповідають вершинам графа, а рядки — його ребрам. Якщо ребро e_i є інцидентним вершині v_j, то $\varepsilon_{ij} = 1$, в іншому випадку $\varepsilon_{ij} = 0$. Це матриця інцидентності звичайного графа G, яка є одним із способів його визначення (для графа на рис. 65) її задано в табл. 9.1, *а*.

У матриці інцидентності $\left\| \varepsilon_{ij} \right\|$ орієнтованого графа G, якщо вершина v_j — початок ребра e_i, то $\varepsilon_{ij} = -1$; якщо v_j — кінець e_i, то $\varepsilon_{ij} = 1$; якщо e_i — петля, а v_j — інцидентна їй вершина, то $\varepsilon_{ij} = \alpha$, де α — будь-яке число, відмінне від 1, 0 та –1, в інших випадках $\varepsilon_{ij} = 0$ (приклад — табл. 9.1, *в* для графа на рис. 66).

У кожному рядку матриці інцидентності для неорієнтованого або орієнтованого графа тільки два елементи відмінні від 0 (або один, якщо ребро є петлею). Тому такий спосіб задання графа — не досить економний. Відношення інцидентності можна задати ще списком ребер графа. Кожний рядок цього списку відповідає ребру, в ньому записано номери вершин, інцидентних йому. Для неорієнтованого графа порядок цих вершин у рядку довільний, для орієнтованого першим записується номер або інше найменування початку ребра, а другим — його кінця. У табл. 9.1, *б* і *г* наведено списки ребер для графів, зображених на рис. 65 та 66.

За списком ребер графа можна легко визначити матрицю інцидентності. Справді, кожний рядок цього списку відповідає рядку матриці з

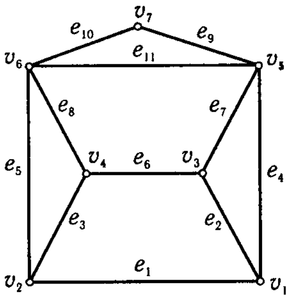


Рис. 65

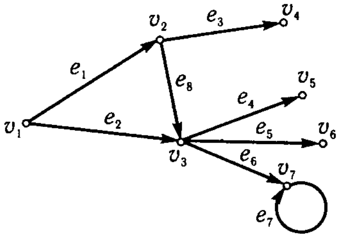


Рис. 66

тим самим номером. Для неорієнтованого графа в рядку списку записуються номери елементів рядка матриці інцидентності, що дорівнюють 1, а для орієнтованого графа в цьому рядку першим зазначається номер елемента рядка матриці, який дорівнює –1, другим — номер елемента, що дорівнює 1.

Поняття матриці інцидентності та списку ребер можна легко узагальнити на випадок мультиграфа.

Таблиця 9.1

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
e_1	1	1	0	0	0	0	0
e_2	1	0	1	0	0	0	0
e_3	0	1	0	1	0	0	0
e_4	1	0	0	0	1	0	0
e_5	0	1	0	0	0	1	0
e_6	0	0	1	1	0	0	0
e_7	0	0	1	0	1	0	0
e_8	0	0	0	1	0	1	0
e_9	0	0	0	0	1	0	1
e_{10}	0	0	0	0	0	1	1
e_{11}	0	0	0	0	1	1	0

а

Ребро	Вершина
e_1	v_1, v_2
e_2	v_1, v_3
e_3	v_2, v_4
e_4	v_1, v_5
e_5	v_2, v_6
e_6	v_3, v_4
e_7	v_3, v_5
e_8	v_4, v_6
e_9	v_5, v_7
e_{10}	v_6, v_7
e_{11}	v_5, v_6

б

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
e_1	–1	1	0	0	0	0	0
e_2	–1	0	1	0	0	0	0
e_3	0	–1	0	1	0	0	0
e_4	0	0	–1	0	1	0	0
e_5	0	0	–1	0	0	1	0
e_6	0	0	–1	0	0	0	1
e_7	0	0	0	0	0	0	2
e_8	0	–1	1	0	0	0	0

в

Ребро	Вершина
e_1	v_1, v_2
e_2	v_1, v_3
e_3	v_2, v_4
e_4	v_3, v_5
e_5	v_3, v_6
e_6	v_3, v_7
e_7	v_7, v_7
e_8	v_2, v_3

г

9.3. Задання графа за допомогою матриці суміжності

Матриця суміжності графа — це квадратна матриця $\Delta\left\|\delta_{ij}\right\|$, стовпцям і рядкам якої відповідають вершини графа. Для неорієнтованого графа δ_{ij} дорівнює кількості ребер, інцидентних *і*- та *j*-й вершинам, для орієнтованого графа цей елемент матриці суміжності відповідає кількості ребер з початком в *і*-й вершині й кінцем у *j*-й. Таким чином, матриця суміжності неорієнтованого графа є симетричною $\left(\delta_{ij}=\delta_{ji}\right)$, а орієнтованого — необов'язково. Якщо вона все ж симетрична, то для кожного ребра орієнтованого графа існує ребро, яке з'єднує ті самі вершини, але йде у зворотному напрямку. Очевидно, орієнтований граф із симетричною матрицею суміжності канонічно відповідає неорієнтованому графу, що має ту саму матрицю суміжності.

Матриці суміжності розглянутих вище графів (див. рис. 65 і 66) наведено в табл. 9.2. Матриця суміжності повністю визначає відповідний неорієнтований або орієнтований граф. Число його вершин дорівнює вимірності матриці *n*, *і*- та *j*-й вершинам графа інцидентними є δ_{ij} ребер. Для неорієнтованого графа $\delta_{ij}=\delta_{ji}$, і всі його ребра визначаються верхнім правим трикутником матриці, розташованим над діагоналлю, включаючи останню.

Кількість їх дорівнює сумі δ_{ij} у цьому трикутнику, тобто $\sum_{i=1}^{n}\sum_{j=i}^{n}\delta_{ij}$. Ребра орієнтованого графа визначаються всіма елементами δ_{ij} матриці суміжності. В обох випадках за допомогою матриці суміжності легко будується, наприклад, список ребер, що визначає граф. Елементу матриці суміжності, розташованому в *і*-му рядку та *j*-му стовпці, відповідають δ_{ij} рядків списку ребер (при $\delta_{ij}=0$ немає жодного рядка), в кожному з яких записуються номери *і*, *j*. Для неорієнтованого графа ці рядки відповідають тільки елементам названого вище верхнього правого трикутника матриці суміжності, тобто елементам δ_{ij} з $j \geq i$, а для орієнтованого графа треба розглядати всі елементи δ_{ij}.

Отже, граф може бути поданий різними способами. Він може бути зображений на кресленні (рисунку), заданий матрицею інцидентності, списком ребер або матрицею суміжності. Вигляд креслення залежить від форми ліній і взаємного розташування вершин. Іноді не так легко зрозуміти, чи однакові графи, зображені різними кресленнями, як, наприклад, на рис. 67. Вигляд матриць

Таблиця 9.2

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	1	0	1	0	0
v_2	1	0	0	1	0	1	0
v_3	1	0	0	1	1	0	0
v_4	0	1	1	0	0	1	0
v_5	1	0	1	0	0	1	1
v_6	0	1	0	1	1	0	1
v_7	0	0	0	0	1	1	0

а

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	1	0	0	0	0
v_2	0	0	1	1	0	0	0
v_3	0	0	0	0	1	1	1
v_4	0	0	0	0	0	0	0
v_5	0	0	0	0	0	0	0
v_6	0	0	0	0	0	0	0
v_7	0	0	0	0	0	0	1

б

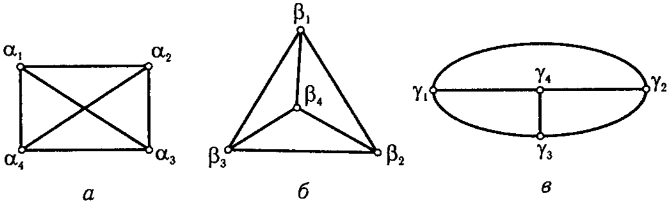


Рис. 67

та списку ребер залежить від нумерації вершин і ребер графа. Строго кажучи, граф вважається повністю заданим, якщо нумерацію його вершин зафіксовано.

Означення 9.1. *Нехай існує бієкція* φ, *яка діє з множини вершин графа G на множину вершин графа H так, що для будь-яких вершин* v_1 *та* v_2 *графа G їхні образи* $\varphi(v_1)$ *і* $\varphi(v_2)$ *є суміжними в H тоді й тільки тоді, коли* v_1 *та* v_2 *— суміжні в G. Така бієкція називається ізоморфізмом графа G на граф H, а графи G і H є ізоморфними.*

Таким чином, графи, зображені на рис. 67, — ізоморфні. Іншими словами, графи є ізоморфними, якщо вони різняться тільки нумерацією вершин.

Перенумерація вершин графа задається рядком $\alpha_1, \alpha_2, ..., \alpha_n$ нових номерів вершин, розташованих у початковому порядку. Нова матриця суміжності утворюється з початкової переміщенням кожного елемента δ_{ij} в α_i-й рядок та α_j-й стовпець, тобто внаслідок перестановки $(\alpha_1, \alpha_2, ..., \alpha_n)$ рядків і стовпців початкової матриці. Тому, щоб дізнатися, чи зображують дві матриці суміжності ізоморфні графи, можна, наприклад, здійснити всілякі перестановки рядків та стовпців першої матриці. Якщо після однієї з цих перестановок виникне матриця, що тотожно збігається з другою, графи, які зображаються цими матрицями суміжності, будуть ізоморфними. Проте, щоб пересвідчитися таким способом у тому, що графи не є ізоморфними, доведеться виконати всі $n!$ перестановок рядків і стовпців, а це — досить трудомістка операція.

Матриця інцидентності графа та список його ребер залежать від нумерації ребер і вершин. Перехід від однієї пари нумерації до іншої визначається перестановками $(\alpha_1, \alpha_2, ..., \alpha_n)$ вершин та $(\beta_1, \beta_2, ..., \beta_m)$ ребер графа, який розглядається. Матриця інцидентності утворюється з початкової внаслідок перестановки рядків (i-й на β_i-те місце) і стовпців (j-й на α_i-те). Рядки списку ребер переставляються так само, як і рядки матриці інцидентності, причому кожний номер j в рядках списку замінюється номером α_j.

9.4. Локальні степені вершин графа

Якщо задано матриці суміжності Δ або інцидентності Е графа, то можна визначити локальні степені всіх його вершин. Справді, в j-му стовпці матриці інцидентності, що відповідає вершині v_j, одиниці знаходяться на перетині з рядками, яким відповідають інцидентні цій вершині ребра, а інші елементи стовпця дорівнюють 0. Отже, $\rho(v_i) = \sum_{i=1}^{m} \varepsilon_{ij}$.

Елементи ж δ_{ij} матриці суміжності — це кількість ребер, інцидентних вершинам v_i і v_j. Звідси $\rho\left(v_j\right)=\sum_{i=1}^{n}\delta_{ij}$.

При підрахунку степенів вершин за цими формулами кожна петля вносить у степінь інцидентної їй вершини 1. Проте при зображенні петлі на рисунку до цієї вершини примикають два кінці петлі, тобто петля вносить у цей степінь 2. Щоб таким чином ураховувати внесок петель у степінь, треба трохи ускладнити формули для його обчислення. Через коефіцієнти матриці інцидентності степінь можна розрахувати, наприклад, за формулою $\rho'(v_j)=\sum_{i=1}^{m}\varepsilon_{ij}(3-\sum_{k=1}^{n}\varepsilon_{ik})$. Коли i-те ребро звичайне, $\sum_{k=1}^{n}\varepsilon_{ik}=2$, і відповідний доданок зовнішньої суми дорівнює ε_{ij}, тобто 1 для ребер, інцидентних вершині v_j, та 0 для інших. Якщо ж воно є петлею, то $\sum_{k=1}^{n}\varepsilon_{ik}=1$, а доданок зовнішньої суми дорівнює $2\varepsilon_{ij}$, тобто 2 для петель, інцидентних вершині v_j, та 0 для інших.

Це саме значення степеня визначається через коефіцієнти δ_{ij} матриці суміжності графа G за формулою $\rho'(v_j)=\sum_{i=1}^{n}\delta_{ij}+\delta_{jj}=\sum_{k=1}^{n}\delta_{jk}+\delta_{jj}$.

Залежно від задачі, що розглядається, може бути потрібен той чи інший спосіб визначення степеня вершини. Тому в кожному випадку має бути відомо, чи є петля один раз або двічі інцидентною своїй вершині.

Оскільки кожне ребро має два кінці, в сумі $\sum_{v\in G}\rho'(v)$ ребра враховуються двічі. Таким чином, ця сума дорівнює подвоєному числу ребер графа, тобто є парною. Отже, парна також кількість непарних доданків цієї суми, тобто кількість вершин непарного степеня.

Означення 9.2. *Граф називається однорідним степеня k, якщо степені всіх його вершин дорівнюють k і, отже, є рівними між собою.*

Якщо однорідний граф степеня k має n вершин та m ребер, то $m=\frac{1}{2}\sum_{v\in V}\rho'(v)=\frac{kn}{2}, n=\frac{2m}{k}$.

Означення 9.3. *Звичайний граф називається повним, якщо кожна пара його вершин сполучається ребром.*

У звичайного повного графа P_n на n вершинах степені всіх вершин однакові й дорівнюють $n-1$.

9.5. Локальні степені вершин орієнтованих графів

Для вершин орієнтованого графа визначаються два локальних степеня: $\rho_1(v)$ — кількість ребер із початком у вершині v, або, інакше, кількість ребер, які виходять із v, і $\rho_2(v)$ — кількість ребер, що входять у ребра v, тобто ребер, для яких ця вершина є кінцем. Петля дає внесок 1 в обидва ці степені. Локальні степені вершин орієнтованого графа визначаються через коефіцієнти δ_{ij} його матриці суміжності: $\rho_1(v_i) = \sum_{j=1}^{n} \delta_{ij}, \rho_2(v_i) = \sum_{h=1}^{n} \delta_{hi}$. Вираз їх через коефіцієнти матриці інцидентності — значно складніший.

Оскільки кожне ребро орієнтованого графа G має один початок й один кінець, суми $\sum_{v \in G} \rho_1(v)$ та $\sum_{v \in G} \rho_2(v)$ дорівнюють кількості ребер цього графа, а отже, є рівними між собою. Звідси випливає, що в однорідному орієнтованому графі степеня k з n вершинами й m ребрами $m = \sum_{v \in V} \rho_1(v) = \sum_{v \in V} \rho_2(v) = kn, \; n = \frac{m}{k}$.

9.6. Частини графа, суграфи та підграфи

Означення 9.4. *Граф H називається частиною графа $G (H \subset G)$, якщо множина його вершин $V(H)$ міститься в множині $V(G)$, а множина $E(H)$ ребер — в $E(G)$. Якщо $V(H) = V(G)$, то частина графа називається суграфом.*

Наприклад, існує нульовий суграф, множина ребер якого є порожньою. Суграф H покриває вершини неорієнтованого графа G (або є покривним), якщо будь-яка вершина останнього — інцидентна хоча б одному ребру з H. Таким чином, якщо в графі G існує ізольована вершина v, не інцидентна жодному ребру, то покривного суграфа цього графа не існує.

Будь-яку множину B ребер графа G можна вважати множиною ребер деякої частини H. Множина вершин цієї частини складається з вершин, інцидентних елементам множини B. Якщо B є множиною ребер іншої частини H', то $H \subset H'$, причому вершини H', що не належать H, у графі H' є ізольованими.

Означення 9.5. *Підграфом графа G називається частина графа з множиною вершин $U \subset V(G)$, якщо її ребрами є всі ребра з $E(G)$, обидва кінці яких належать U.*

Зірчастий граф для вершини $v \in G$ складається з усіх ребер із початком або кінцем у вершині v і вершин, інцидентних цим ребрам (включаючи вершину v).

9.7. Операції з частинами графа

Доповнення $\bar{H}$ частини H визначається множиною всіх ребер графа G, що не належать H.

Об'єднання $H_1 \cup H_2$ та переріз $H_1 \cap H_2$ частин H_1 і H_2 графа G визначаються природно:

$$V(H_1 \cup H_2) = V(H_1) \cup V(H_2);$$

$$E(H_1 \cup H_2) = E(H_1) \cup E(H_2);$$

$$V(H_1 \cap H_2) = V(H_1) \cap V(H_2);$$

$$E(H_1 \cap H_2) = E(H_1) \cap E(H_2).$$

Дві частини H_1 та H_2 не перерізаються по вершинах, якщо вони не мають спільних вершин, а значить, і спільних ребер. Об'єднання $H_1 \cup H_2$ частин, що не перерізаються по вершинах, називається прямою сумою. Аналогічно визначається пряма сума будь-якої кількості частин. Частини H_1 й H_2 не перерізаються по ребрах, якщо $E(H_1) \cap E(H_2) = \varnothing$. Наприклад, для будь-якої частини H та її доповнення $\bar{H}$ сума $G = H \cup \bar{H}$ є прямою по ребрах.

9.8. Графи та бінарні відношення

Між орієнтованими графами без кратних ребер із множиною вершин $V = \{v_i, ..., v_n\}$ і бінарними відношеннями на множині V існує взаємно однозначна відповідність: відношенню R відповідає орієнтований граф $G(R)$, в якому ребро (v_i, v_j) існує тоді й тільки тоді, коли виконано співвідношення $v_i R v_j$. Аналогічна взаємно однозначна відповідність є між симетричними бінарними відношеннями і неорієнтованими графами (див. розд. 4).

Розглянемо відповідність між операціями над відношеннями й операціями над графами. Кожне відношення R має заперечення $\bar{R}$, істинне тоді й тільки тоді, коли R — хибне. Наприклад, для відношення рівності $a = b$ запереченням є відношення нерівності $a \neq b$, а для відношення ортогональності $a \perp b$, визначеного для елементів векторного простору, запереченням є відношення відмінності скалярного добутку від 0: $(a, b) \neq 0$. Граф $G(\bar{R})$ є доповненням графа $G(R)$ відносно повного орієнтованого графа $P(V)$ з множиною вершин V, на якій задано бінарне відношення R, що розглядається, і множиною дуг $E(P(V)) = V \times V$. Граф $G(R^{-1})$, де R^{-1} — відношення, обернене до R, різниться від графа $G(R)$ тим, що напрямки всіх дуг замінено на зворотні.

Відношення R' містить відношення R, якщо вони визначені на одній і тій самій множині V та з $v_i R v_j$ випливає $v_i R' v_j$. У цьому випадку кажуть також, що відношення R' утворюється з відношення R, і пишуть $R' \supset R$. Відповідні графи $G(R)$ та $G(R')$ мають одну й ту саму множину вершин V, а множина $E(R')$ ребер першого є підмножиною множини $E(R')$ ребер другого. Таким чином, $G(R)$ є суграфом графа $G(R')$, тобто $G(R') \supset G(R)$.

Для будь-яких бінарних відношень R_1 й R_2, заданих на одній і тій самій множині V, можна визначити суму (об'єднання) $R_1 \cup R_2$ і переріз $R_1 \cap R_2$:

$$v_i (R_1 \cup R_2) v_j \Leftrightarrow v_i R_1 v_j \vee v_i R_2 v_j;$$

$$v_i (R_1 \cap R_2) v_j \Leftrightarrow v_i R_1 v_j \,\&\, v_i R_2 v_j.$$

Відповідні графи також є сумою та перерізом:

$$G(R_1 \cup R_2) = G(R_1) \cup G(R_2);$$

$$G(R_1 \cap R_2) = G(R_1) \cap G(R_2).$$

Деякі типи графів добре описуються на мові бінарних відношень Наприклад, нуль-граф $\varnothing(V)$, що не має ребер, відповідає нульовому відношенню $v_i O v_j$, яке не містить жодної пари $(v_i, v_j) \in V \times V$; повному орієнтованому графу $D(V)$ відповідає універсальне (повне) відношення P, завжди істинне.

Якщо R — рефлексивне, то $G(R)$ має петлі у всіх вершинах; якщо R — антирефлексивне, то $G(R)$ не має петель. Якщо R — транзитивне, то в графі $G(R)$ для кожної пари ребер (v_i, v_j) і (v_j, v_k) існує замикальне ребро (v_j, v_k).

9.9. Маршрути, ланцюги та цикли

Означення 9.6. *Нехай G — неорієнтований граф. Маршрутом M у графі G називається така скінченна або нескінченна послідовність вершин і ребер, які чергуються,*

$$(\ldots v_1, e_1, v_2, e_2, \ldots, v_{n-1}, e_{n-1}, v_n, \ldots), \tag{9.1}$$

що кожні два сусідні ребра e_{i-1} та e_i мають спільну інцидентну вершину v_i.

Очевидно, маршрут M можна задавати послідовністю $(\ldots v_1, v_2, \ldots, v_n, \ldots)$ його вершин (у звичайному графі), а також послідовністю $(\ldots e_1, e_2, \ldots, e_n, \ldots)$ ребер, що й робитимемо далі. Одне і те саме ребро може зустрічатися в маршруті кілька разів. Надалі будуть розглядатися в основному скінченні маршрути. У таких маршрутах існують перше e_1 й останнє e_n ребра. Вершина v_0, інцидентна ребру e_1, називається початком маршруту. Якщо ребра e_1 та e_2 — кратні, то потрібна спеціальна вказівка, яку з двох інцидентних ним вершин слід вважати початком маршруту. Аналогічно означається кінець маршруту.

Означення 9.7. *Вершини, інцидентні ребрам маршруту, крім початкової і кінцевої, називаються внутрішніми, або проміжними.*

Оскільки різні ребра маршруту можуть бути інцидентними одній і тій самій вершині, початок або кінець маршруту може одночасно виявитися внутрішньою вершиною (див. маршрут $(e_1, e_2, e_3, e_4, e_5)$ на рис. 68).

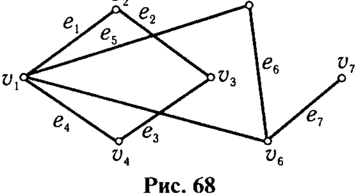


Рис. 68

Означення 9.8. *Нехай маршрут $M(e_1, e_2, \ldots, e_n)$ має початок v_0 і кінець v_n. Тоді його називають сполучним. Число ребер маршруту є його довжиною. Якщо $v_0 = v_n$, то маршрут називають замкненим, або циклічним. Відрізок $(e_i, e_{i+1}, \ldots, e_j)$ скін-*

ченного або нескінченного маршруту M є маршрутом. Він називається ділянкою маршруту M.

Означення 9.9. *Маршрут M називається ланцюгом, якщо кожне ребро зустрічається в ньому не більше ніж один раз, і простим ланцюгом, якщо будь-яка вершина, крім, можливо, початкової, зустрічається в ньому не більш як один раз. Якщо ланцюг є замкненим, то його називають циклом, а якщо простий ланцюг — замкнений, то це — простий цикл.*

Наприклад, послідовності (e_1, e_2, e_3, e_4), (e_2, e_3, e_4, e_1), (e_3, e_4, e_1, e_2) та (e_4, e_1, e_2, e_3) зображують один і той самий цикл. Часто вважається, що можна міняти порядок ребер циклу на зворотний, тобто, наприклад, послідовність (e_4, e_3, e_2, e_1) зображує той самий цикл (див. рис. 68).

Визначення маршруту можна перенести з графа на орієнтований граф. Маршрут в останньому називатимемо шляхом. Відповідно можна перенести також визначення ланцюга, простого ланцюга та циклу. Простий цикл в орієнтованому графі ще називається контуром.

9.10. Зв'язність

Означення 9.10. *Дві вершини v і w називаються зв'язаними, якщо існує маршрут вигляду* (9.1) *із кінцями v та w. Граф називається зв'язним, якщо будь-яка пара його вершин є зв'язаною.*

Означення 9.11. *Зв'язністю графа називається мінімальна кількість вершин, вилучення яких приводить до утворення незв'язного графа.*

Означення 9.12. *Кількість вершин у максимальному повному підграфі графа G називається щільністю* $\alpha(G)$ *графа G. Кількість вершин у максимальному порожньому підграфі графа G називається нещільністю* $\varepsilon(G)$ *графа G.*

Означення 9.13. *Довжина найменшого ланцюга між вершинами v і w звичайного зв'язного графа G називається відстанню d(v, w) між цими вершинами.*

Очевидно, вона задовольняє всі аксіоми метрики:

- $d(v, w) \geq 0$;
- $d(v, w) = 0 \Leftrightarrow v = w$;
- $d(v, w) = d(w, v)$;
- $d(v, w) + d(w, u) \geq d(v, u)$.

Означення 9.14. *Діаметром графа G називається величина* $d(G) = \max\limits_{v, w \in V} d(v, w)$.

Виберемо деяку фіксовану вершину c і позначимо $r(c) = \max\limits_{v \in V} d(c, v)$. *Величина r(c) називається максимальною віддалю від вершини c. Назвемо* c_0 *центром графа G, якщо* $r(G) = r(c_0) = \min\limits_{c \in V} r(c)$.

Величина r(G) називається радіусом графа G, а будь-який найкоротший ланцюг від центра c_0 *до максимально віддаленої від нього вершини — радіальним.*

Означення 9.15. *Дводольний граф* $G_{m,n}$ *— це граф, множину вершин якого можна розбити на дві підмножини* V_1 *і* V_2 $(|V_1| = m, |V_2| = n)$ *таким чином, що кожне ребро графа з'єднує вершини з різних підмножин.*

Означення 9.16. *Граф називається k-зв'язним, якщо, вилучаючи будь-яку його $k-1$ вершину, не можна порушити його зв'язність, а при вилученні деяких k вершин зв'язність може порушитися.*

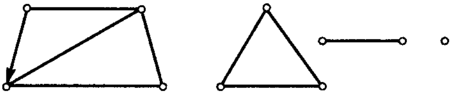


Рис. 69

Означення 9.17. *Граф називається k-ребернозв'язним, якщо k — максимальне з таких p, що при вилученні будь-яких $p-1$ ребер зв'язність графа не порушується.*

Означення 9.18. *Орієнтований граф називається сильнозв'язним, якщо для будь-яких двох його вершин v і w існує шлях в обох напрямках.*

Означення 9.19. *Орієнтований граф називається однобічно зв'язним, якщо для будь-яких двох його вершин v_i та v_j існує шлях хоча б в одному напрямку.*

Означення 9.20. *Псевдографом, асоційованим з орієнтованим псевдографом $D_1(V)$, називається псевдограф $D_2(V)$ такий, коли кожну дугу графа D_1 замінено на ребро графа D_2.*

Орієнтований граф називається слабкозв'язним, якщо зв'язним є асоційований з ним псевдограф.

Означення 9.21. *Якщо граф (орієнтований граф) не є зв'язним (слабкозв'язним), то він називається незв'язним.*

Компонентою зв'язності (сильної зв'язності) графа G (орієнтованого графа D) називається його зв'язний (сильнозв'язний) підграф, який не є підграфом жодного зв'язного (сильнозв'язного) підграфа графа G (орієнтованого графа D).

Приклади: 1. На рис. 69 зображено граф із чотирма компонентами зв'язності.

2. На рис. 70, *а* показано орієнтований граф, що має три компоненти зв'язності, зображені на рис. 70, *б — г*.

З означення зв'язності випливає, що коли $G_1 = (V_1, E_1)$ — компоненти зв'язності графа G, то G_1 — підграф графа G, породженого множиною V_1.

Неважко показати, що справджується таке твердження.

Твердження 9.1. *Нехай $G = (V, E)$ — граф із p компонентами зв'язності $G_1 = (V_1, E_1), \ldots, G_p = (V_p, E_p)$. Тоді*

$$V = V_1 \cup \ldots \cup V_p,\ E = E_1 \cup \ldots \cup E_p;$$

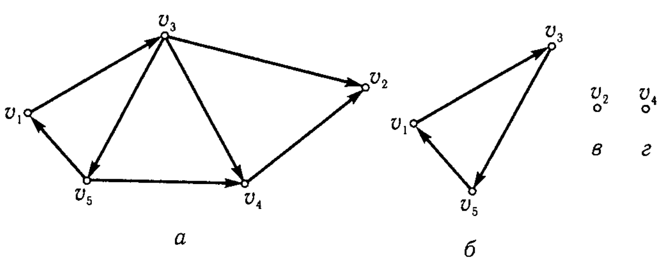


Рис. 70

$$V_i \cap V_j = \varnothing, E_i \cap E_j = \varnothing \text{ при } i \neq j;$$

$$n(G_1) + \ldots + n(G_p) = n(G);\ m(G_1) + \ldots + m(G_p) = m(G).$$

Виходячи з означення зв'язності двох вершин, можна говорити про бінарне відношення зв'язності (позначимо його σ), задане на множині $V(G)$.

Твердження 9.2. *Якщо припустити, що вершина зв'язана сама з собою, тобто існує маршрут із v у v, то:*

- σ *— відношення еквівалентності на множині V;*
- $v\sigma w$ *тоді й тільки тоді, коли вершини v і w належать одній компоненті зв'язності графа G;*
- *для будь-якого класу еквівалентності $V_1 \in \{[v]_\sigma\}$ граф G_1, породжений множиною V_1, є компонентою зв'язності графа G;*
- *для будь-якої компоненти зв'язності $G_1 = (V_1, E_1)$ графа G маємо $V_1 \in \{[v]_\sigma\}$.*

Надалі кількість компонент зв'язності графа G будемо позначати $n_\sigma(G)$. Під операцією вилучення вершини з графа будемо розуміти операцію вилучення деякої вершини разом із ребрами, інцидентними їй.

Означення 9.22. *Вершина графа, вилучення якої збільшує кількість компонент зв'язності, називається відокремлювальною (або точкою зчленування).*

Приклад. Для графа, зображеного на рис. 71, точками зчленування будуть вершини v_3, v_4, v_7, v_{10}.

Наведені вище твердження 9.1 і 9.2 можна легко поширити на випадок орієнтованих графів: твердження 9.1 перефразовується дослівно, а твердження 9.2 можна сформулювати так.

Твердження 9.3. *Нехай σ_1 — відношення досяжності на множині V вершин орієнтованого псевдографа D. Нехай σ_2 — відношення двобічної досяжності на V, тобто $\sigma_2 = \sigma_1 \cap \sigma_1^{-1}$. Тоді:*

- σ_1 *— рефлексивне й транзитивне;*
- σ_2 *— відношення еквівалентності на V;*
- $v\sigma_2 w$ *тоді й тільки тоді, коли вершини v і w належать одній компоненті сильної зв'язності орієнтованого псевдографа D;*

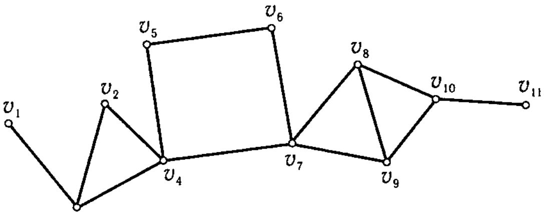


Рис. 71

- *для будь-якого класу еквівалентності $V_1 \in \{[v]_\sigma\}$ орієнтований псевдограф D_1, породжений множиною V_1, є компонентою сильної зв'язності орієнтованого псевдографа D;*
- *для будь-якої компоненти сильної зв'язності $D_1 = (V_1, E_1)$ орієнтованого псевдографа D маємо $V_1 \in \left\{[v]_{\sigma_2}\right\}$.*

9.11. Дерева

Означення 9.23. *Граф G називається деревом, якщо він є зв'язним і не має циклів, а граф G, усі компоненти зв'язності якого є деревами, — лісом.*

Приклад. Граф, зображений на рис. 72, є деревом.

Розглянемо деякі властивості дерев.

Твердження 9.4. *Такі твердження є еквівалентними:*

1) *граф G — дерево;*

2) *граф G є зв'язним і не має простих циклів;*

3) *граф G є зв'язним, і $n(G) = m(G)+1$;*

4) *для будь-яких двох різних вершин графа G існує єдиний (і притому простий) ланцюг;*

5) *граф G не містить циклів, але, додаючи до нього будь-яке нове ребро, дістаємо рівно один і притому простий цикл.*

Доведемо еквівалентність тверджень 1) та 4).

Розглянемо будь-які дві вершини v_1 і v_2 дерева.

Необхідність доведемо від супротивного: нехай M_1 й M_2 — два різних ланцюги з початком у v_1 і кінцем у v_2. Оскільки ланцюги — різні, в ланцюзі M_2 існує хоча б одне ребро e, що не належить ланцюгу M_1. Нехай v' та v'' — його кінці. Оскільки $e \notin M_1$, або $v' \notin M_1$, або $v'' \notin M_1$. Без обмеження загальності можна вважати, що $v' \in M_1$, а $v'' \notin M_1$. Тоді, очевидно, знайдеться ребро e_1, яке виходить з v''. Нехай другим кінцем ребра e_1 буде вершина v'''. Якщо $v''' \in M_1$, то покладемо $v^* = v'''$. Якщо ж $v''' \in M_1$, то цей процес можна продовжити, і за скінченне число кроків (оскільки розглядаємо скінченні графи) знайти вершину $v^* : v^* \in M_1 \wedge v^* \in$ $\in M_2$. Проте тоді вершини v' та v^* сполучаються двома ланцюгами, які не мають спільних ребер. Значить, у початковому графі існує цикл, що суперечить означенню дерева. Одержана суперечність доводить єдність ланцюга, який з'єднує дві будь-які вершини дерева. Відсутність циклів у дереві показує, що цей ланцюг може бути тільки простим.

Доведемо достатність. Оскільки будь-які дві вершини можна з'єднати ланцюгом, граф — зв'язний. Припущення про наявність циклів спростовується єдиністю простого ланцюга, який з'єднує будь-які дві вершини.

Аналогічно доводиться еквівалентність будь-якої пари зі сформульованих тверджень.

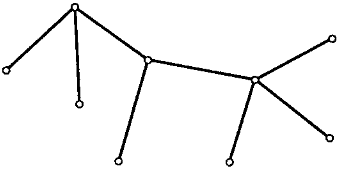


Рис. 72

9.12. Цикломатичне число графа

Означення 9.24. *Нехай $G = (V, E)$ — скінченний неорієнтований граф. Його цикломатичним числом називається число*

$$\gamma(G) = n_{\sigma}(G) + m(G) - n(G),$$

де $n_{\sigma}(G)$ — кількість зв'язних компонент графа; $m(G)$ — кількість його ребер, а $n(G)$ — кількість вершин.

Цикломатичне число дерева дорівнює нулю, цикломатичне число лісу — сумі цикломатичних чисел своїх зв'язних компонент-дерев, тобто також дорівнює нулю.

Твердження 9.5. *Цикломатичне число скінченного графа є додатним.*

Доведення. Оскільки цикломатичне число незв'язного графа дорівнює сумі цикломатичних чисел його зв'язних компонент, досить розглянути зв'язний граф G.

Якщо G — не дерево, то в ньому існує хоча б один цикл. Якщо вилучимо одне з ребер циклу, то побудуємо з графа G граф G'. Якщо в G' існує цикл, то вилучимо знов одне з ребер цього циклу, побудуємо граф G'' і т. д. k разів, доки не побудуємо граф G^k, в якому немає циклів. Отже, G^k — дерево й $n\left(G^k\right) = m\left(G^k\right) + 1$, а $\gamma\left(G^k\right) = 0$.

Оскільки $n(G) = n\left(G^k\right)$, а $m(G) = m\left(G^k\right) + k$, маємо

$$\gamma(G) = n_{\delta}(G) + m(G) - n(G) = 1 + m\left(G^k\right) + k + n\left(G^k\right) = \gamma\left(G^k\right) + k = k > 0.$$

У доведенні цього твердження закладено конструктивну ідею побудови об'єкта, який буде означений нижче, а сама ідея зазнає подальшої формалізації.

9.13. Кістякове дерево зв'язного графа

Означення 9.25. *Кістяковим деревом зв'язного графа G називається будь-яка його частина, що містить усі вершини графа G і є деревом.*

Нехай G — зв'язний граф. Тоді внаслідок твердження 9.4 кістякове дерево графа G (якщо воно існує) має містити $n(G) - 1$ ребер, а внаслідок твердження 9.5 будь-яке кістякове дерево графа є результатом вилучення з G рівно $m(G) - (n(G) - 1) = m(G) - n(G) + 1$ ребер. Покажемо існування кістякового дерева для довільного зв'язного псевдографа $G = (V, X)$, описавши алгоритм його вибору.

Алгоритм 9.1. Він включає такі кроки:

Крок 1. Вважаємо $i = 1$. Вибираємо в G довільну вершину v_1, що утворює підграф G_1 псевдографа G, який є деревом.

Крок 2. Якщо $i = n$, де $n = n(G)$, то задачу розв'язано, і G_i — шукане кістякове дерево псевдографа G. В іншому випадку переходимо до кроку 3.

Крок 3. Нехай уже побудовано дерево G_i, що є підграфом псевдографа G і містить деякі вершини $v_1, ..., v_i$, де $1 \le i \le n-1$. Будуємо граф G_{i+1}, додаючи до графа G_i нову вершину $v_{i+1} \in V$, суміжну в G з деякою вершиною v_j графа G_i, та нове ребро, інцидентне вершинам v_i і v_{i+1} (внаслідок зв'язності G та тієї обставини, що $i < n$, вершина v_{i+1} обов'язково знайдеться). Очевидно, граф G_{i+1} також є деревом. Прирівнюємо $i: = i + 1$ і переходимо до кроку 2.

9.14. Мінімальні кістякові дерева зважених графів

Означення 9.26. *Граф $G(V, E)$ називається зваженим, якщо на множині його ребер означено вагову функцію $1: E \to R$, тобто кожному ребру $e \in E$ зв'язного графа $G = (V, E)$ з непорожньою множиною ребер E поставлено у відповідність величина $l(e)$ — вага ребра e.*

Наведемо алгоритм, що дає змогу знайти кістякове дерево графа G з мінімальною сумою ваг ребер, які містяться в ньому (порівняно з рештою кістякових дерев графа G). Кістякове дерево зв'язного зваженого графа G з мінімальною сумою ваг ребер, що містяться в ньому, будемо називати мінімальним кістяковим деревом графа G.

Алгоритм 9.2. Він включає такі кроки:

Крок 1. Вибираємо в графі G ребро мінімальної ваги. Разом з інцидентними йому вершинами воно утворить підграф G_2 графа G. Покладемо $i = 2$.

Крок 2. Якщо $i = n$, де $n = n(G)$, то задачу розв'язано, і G_i — шукане мінімальне кісткове дерево графа G. В іншому випадку переходимо до кроку 3.

Крок 3. Будуємо граф G_{i+1}, додаючи до графа G_i нове ребро мінімальної ваги, вибране серед усіх ребер графа G, кожне з яких є інцидентним якій-небудь вершині графа G_i й одночасно інцидентним якій-небудь вершині графа G, що не міститься в G_i. Разом із цим ребром включаємо в G_{i+1} інцидентну йому вершину, яка не міститься в G_i. Покладаємо $i: = i + 1$ і переходимо до кроку 2.

9.15. Хроматичне число графа

Означення 9.27. *Нехай G — звичайний граф. Граф G називається k-розфарбовуваним, якщо існує таке розкладання множини його вершин V на k неперерізних класів $V_1, V_2, ..., V_k$, що*

$$V = \bigcup_{i=1}^{k} V_i,\ E(V_i) \cap E(V_j) = \varnothing\ \forall i, j. \qquad (9.2)$$

Подання (9.2) *називається k-розфарбуванням графа G. Найменша кількість $\beta(G)$ класів у можливому розфарбуванні* (9.2) *називається хроматичним числом графа, а сам граф —β-хроматичним.*

9.16. Задача про кенігсберзькі мости. Ейлерові графи

Народженням теорії графів ми зобов'язані постановці та розв'язанню цієї задачі Л. Ейлером. Розташування мостів у м. Кенігсберзі в часи його життя показано на рис. 73.

Задача, яку він розв'язував, формулюється так: якщо вийти з деякої частини міста, то чи можна пройти кожен міст один раз і повернутися в початкову частину міста. Можна побудувати граф задачі, в якому кожній частині міста відповідає вершина, а кожному мосту — ребро, інцидентне вершинам, що стосуються тих частин міста, які з'єднуються цим мостом (рис. 74).

Очевидно, обходу мостів відповідає послідовність ребер графа задачі, в якій два сусідніх ребра мають спільну вершину, тобто маршрут. Оскільки в кінці обходу треба повернутися в початкову частину міста і на кожному мості побувати по одному разу, цей маршрут є циклом, що містить усі ребра графа.

Означення 9.28. *Цикли, що містять усі ребра графа, називаються ейлеровими, а графи, які мають цей цикл, — ейлеровими.*

Можна сказати, що ейлерові — це такі графи, які можна зобразити одним розчерком пера, причому процес цього зображення починається й закінчується в одній і тій самій точці. Л. Ейлер розв'язав поставлену задачу.

Теорема 9.1 **(*теорема Ейлера*).** *Скінченний неорієнтований граф G є ейлеровим тоді й тільки тоді, коли він зв'язний і степені всіх його вершин парні.*

Доведемо необхідність. У незв'язному графі кожний цикл належить якій-небудь його зв'язній частині, тобто не проходить через усі ребра графа (крім випадку, коли всі компоненти зв'язності графа, за винятком однієї, — ізольовані вершини, але це не відповідає умові задачі). Отже, граф має бути зв'язним.

Окрім того, кожний раз, коли ейлерів цикл приходить в яку-небудь вершину, він має вийти з неї по іншому ребру, тобто інцидентні вершинам ребра можна розбити на пари сусідніх у циклі Ейлера. Винятком є ребра, інцидентні початку ейлерова циклу, що збігається з його кінцем. Спочатку цикл виходить із цієї вершини по якому-небудь ребру. Потім він, мож-

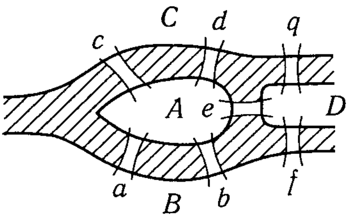


Рис. 73

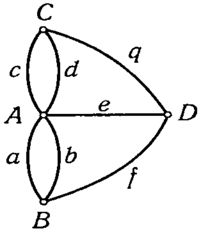


Рис. 74

ливо, кілька разів повертається в цю вершину, кожний раз входячи по новому ребру і виходячи також по новому, іншому ребру. Зрештою, він повертається у початкову вершину по незачепленому раніше ребру. Таким чином, й у цьому випадку інцидентні цій вершині ребра також розпадаються на пари сусідніх у циклі Ейлера, але одна з цих пар складається з ребра, яке проходиться на початку процесу, й іншого, що замикає цей процес.

Доведемо достатність. Нехай G — скінченний зв'язний неорієнтований граф з парними степенями всіх вершин. Почнемо побудову циклу Ейлера з довільної вершини v графа G. Кожен раз, коли додаємо до маршруту нове ребро і приходимо в нову вершину, число вільних ребер у ній змінюється на одиницю. Якщо до цього воно було парним, то тепер стає непарним і не може виявитися нулем, інакше ця вершина початково повинна була мати непарний степінь, що суперечить умові. Після виходу з цієї вершини число вільних, інцидентних їй ребер, зменшується ще на одиницю і знову стає парним. Винятком є початкова вершина, в якої після початку процесу число вільних ребер — непарне і залишається таким після кожного повернення в цю вершину й відходу з неї.

Описаний процес побудови циклу може закінчитися лише в тій вершині, звідки починався, але при цьому не обов'язково, щоб він проходив через усі ребра графа. Ребра, що належать йому, породжують зв'язну частину G_1 графа G, в якій степені всіх вершин — парні. Отже, вони є такими і для різниці $G - G_1$. Оскільки граф G — зв'язний, в доповненні $G - G_1$ існує хоча б одна вершина v', що належить також частині G_1. Починаючи з цієї вершини, можна виконати, як і вище, побудову циклу G_2 у $G - G_1$, що закінчиться також у вершині v'.

Однак тоді $G_1 \cup G_2$ —цикл, який починається та закінчується у вершині v, але який має більшу кількість ребер. Якщо і цей цикл не проходить через усі ребра, то процес його розширення можна повторити. Кожний раз число ребер у циклі збільшується не менш як на два, значить, зрештою цикл Ейлера буде побудований.

У графі задачі про кенігсберзькі мости (див. рис. 74) усі вершини мають непарний степінь. Отже, її розв'язання — неможливе.

Означення 9.29. *Ейлеровим називається ланцюг, що включає всі ребра заданого скінченного неорієнтованого графа G, але має різні початок і кінець.*

Щоб у графі існував ейлерів ланцюг, необхідні його зв'язність та парність степенів усіх вершин, окрім початкової v' і кінцевої v''. Останні дві вершини повинні мати непарні степені: з v' ми зайвий раз виходимо, а в v'' зайвий раз входимо. Ці умови є достатніми для існування ейлерового ланцюга. Доведення таке саме, як і для умов, достатніх для існування циклу Ейлера.

Твердження 9.6. *Найменша кількість неперетинних по ребрах ланцюгів, що покривають скінченний зв'язний граф G, дорівнює k/2, де k — кількість вершин непарного степеня графа G.*

Доведення. Нехай G не є ейлеровим графом, а k — кількість його вершин непарного степеня. Вище було показано, що k — парне число. Кожна вершина непарного степеня має бути кінцем хоча б одного з покривних ланцюгів. Отже, кількість останніх не менша, ніж *k*/2. Проте обмежитися *k*/2 ланцюгами не так важко. З'єднаємо вершини непарного степеня попарно *k*/2 ребрами довільним способом. Тоді степінь кожної з цих вершин збільшиться на одиницю і стане парним. Вийде ейлерів граф, в якому існує ейлерів цикл. Вилучимо в ньому приєднані ребра. При вилученні першого ребра ейлерів цикл перетвориться на ейлерів ланцюг, а при вилученні кожного наступного ребра один із ланцюгів, які виникли до цього моменту, розіб'ється на дві частини. Таким чином, загальна кількість цих ланцюгів стане дорівнювати *k*/2. (Зауважимо, що при побудові ніколи не вилучаються сусідні в ейлеровому циклі ребра.)

Поширимо поняття ейлерова циклу на випадок скінченного орієнтованого графа. Щоб у такому графі *G* існував ейлерів цикл, необхідно й достатньо, щоб виконувалась рівність степенів вершин цього графа по вхідних та вихідних дугах: $\forall v \in V(G)(\rho_1(v) = \rho_2(v))$. Доведення буквально таке саме, що й для неорієнтованого графа, тільки замість визначення парності числа вільних ребер, які залишилися, показується, що в процесі побудови ейлерова циклу для всіх вершин, крім вихідної, для побудови чергового підциклу зберігається баланс між числами вільних (ще не пройдених) вхідних і вихідних ребер.

Твердження 9.7. *У скінченному зв'язному графі завжди можна побудувати орієнтований цикл, що проходить через кожне ребро.*

Оскільки будь-якому неорієнтованому графу канонічно відповідає орієнтований, в якому кожне ребро замінено двома ребрами, які є інцидентними тим самим вершинам і йдуть у протилежному напрямку (причому у цього графа для кожної вершини степені по ребрах, що входять та які виходять, дорівнюють степеню цієї вершини в початковому неорієнтованому графі, а отже, рівні між собою), звідси випливає правильність сформульованого твердження.

Такий цикл іноді називають способом обходу всіх ребер графа. Він використовується в багатьох прикладних задачах, пов'язаних із графами.

9.17. Гамільтонові графи

Означення 9.30. *Гамільтоновим циклом називається простий цикл, що проходить через усі вершини графа. Граф, який містить гамільтонів цикл, називається гамільтоновим.*

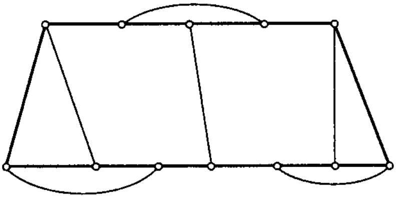


Рис. 75

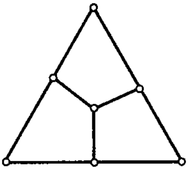


Рис. 76

Такий цикл існує далеко не в кожному графі. Більше того, через будь-які дві вершини графа, який розглядається, може пройти простий цикл (у цьому випадку граф G називається циклічно зв'язним), а гамільтонів цикл може бути відсутнім.

На рис. 75 зображено граф із гамільтоновим циклом, а на рис. 76 — циклічно зв'язний граф, в якому немає гамільтонова циклу.

Іноді можна побудувати простий ланцюг, що проходить через усі вершини графа з початком і кінцем у різних заданих вершинах v', $v'' \in G$. Такий ланцюг також називається гамільтоновим.

9.18. Планарність графів

Означення 9.31. *Граф G називається планарним (плоским), якщо існує ізоморфний йому граф, який може бути зображений на площині без перетину ребер* (див. рис. 77).

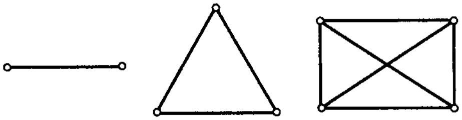
Розглянемо повні графи. Очевидно, повні графи як на двох вершинах (рис. 77, *а*), так і на трьох (рис. 77, *б*) та чотирьох (рис. 77, *в*) вершинах також є плоскими. Повний граф на п'яти вершинах P_5 (рис. 78, *а*) не є планарним, а граф P_4 — це повний плоский граф із максимальною кількістю вершин.

Розглянемо повний дводольний граф $K_{3,3}$ (рис. 78, *б*), який є математичною моделлю відомої задачі про три будинки і три колодязі, що формулюється так. Є три будинки та три колодязі. Сусіди ворогують, не хочуть зустрічатися, але хочуть користуватися всіма трьома колодязями. Чи можна прокласти стежки від кожного будинку до кожного колодязя так, щоб вони не перетиналися? Ця задача нерозв'язна, тому що граф $K_{3,3}$ — неплоский.

Графи P_5 і $K_{3,3}$ відіграють фундаментальну роль у теорії планарності. Їх називають графами Понтрягіна — Куратовського.

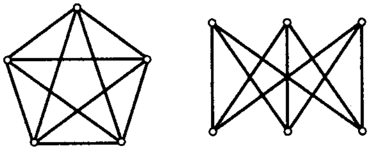
Зазначимо наступний факт: властивість планарності графа G не порушується, якщо деяке ребро розбити на два введення нової вершини (утворюється новий граф G', в якому кількість вершин та ребер збільшилася на одиницю), або замінити два ребра, інцидентних вершині другого степеня, одним ребром, вилучивши цю вершину (в цьому випадку кількість вершин і ребер зменшується на одиницю).

Означення 9.32. *Два графи називаються гомеоморфними, якщо після вилучення в них вершин другого степеня з подальшим об'єднанням інцидентних цим вершинам ребер вони стають ізоморфними.*



а *б* *в*

Рис. 77



а *б*

Рис. 78

Очевидно, граф, гомеоморфний планарному графу, також є планарним, і навпаки, граф, гомеоморфний непланарному графу, — непланарний.

Теорема 9.2. *Граф G є планарним (плоским) тоді й тільки тоді, коли він не містить підграфів, гомеоморфних одному з графів Понтрягіна — Куратовського.*

9.19. Проблема чотирьох фарб*

9.19.1. Постановка задачі

Розглянемо карти, тобто розбиття площини на зв'язні області (рис. 79). Чи досить чотирьох кольорів для розфарбування будь-якої карти так, щоб ніякі суміжні області не були однакового кольору? Ця проблема має давню історію. Досить повний її опис є в [1]. Наведемо його.

Ще в 1840 р. про неї знав відомий німецький математик А. Ф. Мебіус. Перше з помилкових «доведень» було дано А. Б. Кемпе в 1879 р., але помилка була виявлена не відразу. Її знайшов Хейвуд в 1890 р. і тоді ж довів, що області будь-якої карти можна розфарбувати потрібним чином у п'ять кольорів. Відтоді проблему чотирьох фарб, як її назвали, довго не вдавалося вирішити.

Кожній вершині можна співвіднести мітку, вказуючи, якому класу вона належить. Мітки є елементами заданої множини. Іноді вони явно вказують на властивості, що означають множину. Американські математики П. Е. Аппель та Хакен довели гіпотезу чотирьох фарб, використавши обчислення на ЕОМ. Це перший випадок, коли така відома математична задача була розв'язана за допомогою машини. Доведення задачі потребує громіздкого опису машинних розрахунків, тому тут не наводиться. Однак його ідеї повчальні, і їх розглянемо.

Спочатку кілька зауважень. Деякі області можуть межувати між собою тільки в ізольованій точці, в якій можуть сходитися скільки завгодно

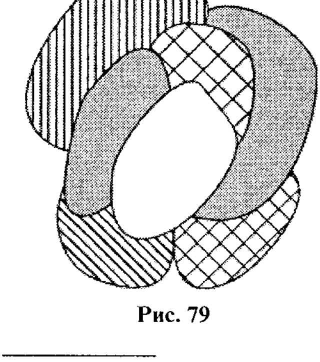
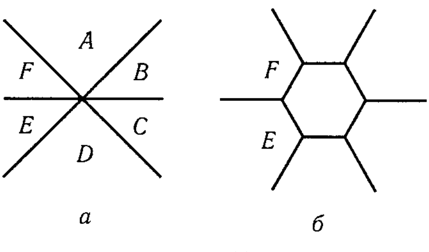


Рис. 79



а

б

Рис. 80

---

* Матеріал п. 9.19 викладено за [1].

областей (рис. 80, *а*), але ми не будемо вважати їх сусідніми. Такі точки можна «роздути», тобто оточити їх областями, що не відгороджують одну від одної області з протяжними межами (рис. 80, *б*). У кожній точці нової карти будуть сходитися не більш як по три області. Якщо її можна розфарбувати в чотири кольори, то, зберігши кольори областей старої карти, дістанемо допустиме розфарбування останньої. Надалі будемо розглядати карти, в точках яких сходяться не більш як по три області.

9.19.2. Двоїстий граф

Нехай є плоский граф. Його вершини — точки, де сходяться по три області, ребра — сполучні лінії (спільні межі двох областей). У кожній області виберемо точку (її центр), а центри сусідніх областей з'єднаємо лініями. Утвориться плоский граф, який називається двоїстим означеному вище (обидва графи зображено на рис. 81).

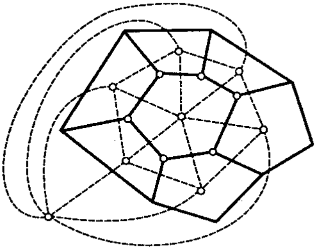


Рис. 81

Області двоїстого графа оточують вершини графа меж карти. Оскільки нас цікавлять карти, в яких у вершинах графа меж сходяться по три області, у двоїстому графі кожна область — трикутник (криволінійний, причому зовнішню область також обмежено трикутником). Таким чином, двоїстий граф означає тріангуляцію площини. Помітимо ще, що він завжди є зв'язним. Розфарбовувати області карти однаково, що й вершини двоїстого графа. Тому проблему чотирьох фарб формулюють ще й так: чи можна розфарбувати вершини будь-якого зв'язного плоского графа в чотири кольори так, щоб сусідні вершини були розфарбовані в різні кольори?

9.19.3. Конфігурації, що редукуються

Конфігурація — це зв'язний підграф плоского графа, породжений деякою підмножиною його вершин (звичайно, невеликою). Інші вершини й ребра графа становлять зовнішню частину конфігурації. Конфігурацію можна «стягнути», тобто вилучити з неї деякі вершини, що не мають інцидентних ребер із зовнішньої частини, ототожнити деякі інші та з'єднати вершини, які залишилися, ребрами, можливо, інакше. Зовнішня частина приєднується до відповідних вершин нової конфігурації, причому потрібно, щоб утворений граф виявився плоским.

Якщо початковий граф був тріангуляцією, то новий граф також має бути тріангуляцією. Якщо з правильного розфарбування вершин нового графа можна дістати правильне розфарбування початкового графа, то конфігурація називається конфігурацією, що редукується. Це означення залежить від потужності засобів редукції. Наведемо приклади конфігурацій, які редукуються.

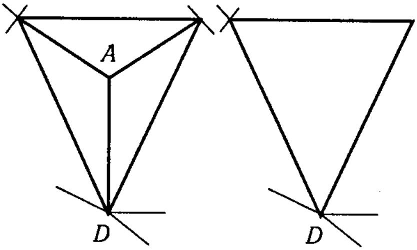


Рис. 82

Вершину степеня 3 показано на (рис. 82). Вилучивши внутрішню вершину, знову матимемо тріангуляцію (той самий рисунок). Якщо редукований граф можна розфарбувати в чотири кольори, то вершини конфігурації, що залишилися, забарвлені в три з них. Тому залишається вільний колір для забарвлення вилученої вершини.

Вершину степеня 4 зображено на рис. 83. Вилучивши внутрішню вершину і з'єднавши ребром яку-небудь пару протилежних вершин, дістанемо тріангуляцію (той самий рисунок). Нехай редукований граф розфарбовано у чотири кольори. Якщо вершини конфігурації, що залишилися, забарвлено тільки в три кольори, то є вільний колір для вилученої вершини. Нехай їх забарвлено в чотири кольори. Перефарбуємо вершину *C* в колір 1, сусідні з нею вершини кольору 1 — у колір 3, сусідні з ними вершини кольору 3 — в колір 1 тощо. Цей процес називається ланцюговим перефарбуванням кольорів 1 та 3; він починається з вершини *C*. Коли він закінчиться, редукований граф буде правильно розфарбований. Якщо при цьому колір вершини *A* не зміниться, то колір 3 виявиться вільним для забарвлення вершини *E*, вилученої в редукованому графі.

В іншому випадку в редукованому графі є ланцюг із вершин кольорів 1 і 3, що з'єднує вершини *A* та *C*. Проте тоді в ньому немає ланцюга вершин кольорів 2 і 4, який з'єднує вершини *B* та *D*, крім ребра *BD*, але останнє не належить зовнішній частині конфігурації і тому при перефарбуванні кольорів не означає сусідства. Застосуємо ланцюгове перефарбовування кольорів 2 та 4, яке починається з вершини *B*. Нове розфарбування матиме тільки один дефект: вершини *B* і *D* забарвлені однаково. Проте у початковому графі вони не є сусідніми. Разом із тим при перефарбуванні звільниться колір 4 для забарвлення вершини *E*.

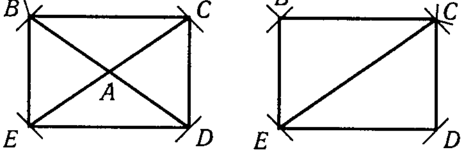


Рис. 83

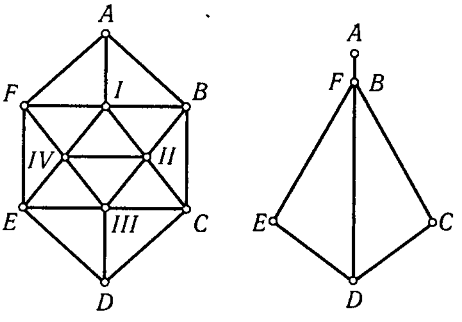


Рис. 84

Чотири вершини степеня 5 розташовуються так, як показано на рис. 84. Вилучимо ці вершини, ототожнимо вершини *B* й *F*, а вершини *B* та *O* з'єднаємо ребром. З точністю до перенумерації кольо-

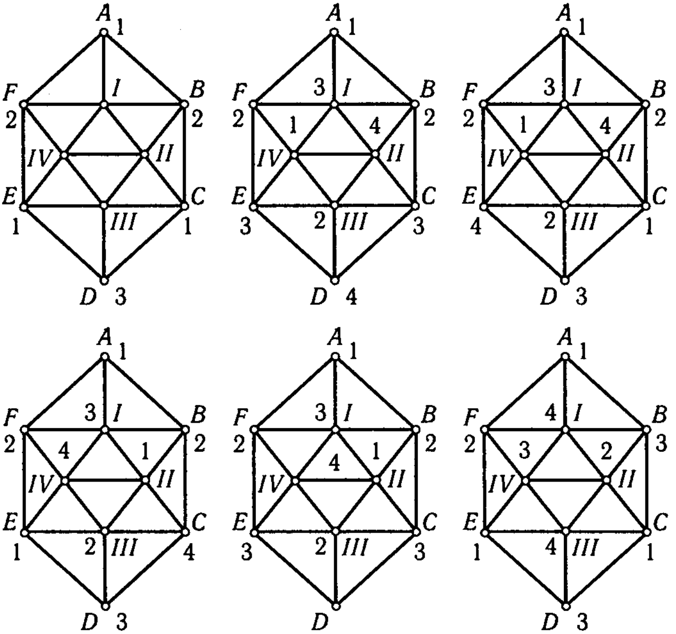


Рис. 85

рів і симетрії відносно конфігурації можливе розфарбування вершин, які залишилися, останньою наведено в рядках табл. 9.3, що відповідає шести графам рис. 85. Усі вони, крім першої, продовжуються всередину конфігурації. При першому розфарбуванні або вершини *B* і *D* не зв'язані ланцюгом вершин кольорів 2 та 3, або вершини *A* і *C* не зв'язані ланцюгом вершин кольорів 1 та 4. Застосувавши відповідні ланцюгові перефарбування, перейдемо до останніх двох розфарбувань на рис. 85, які також можна продовжити всередину конфігурації.

Якщо гіпотеза про чотири фарби є неправильною, то існує граф, вершини якого не можна розфарбувати в чотири кольори, що має мінімальну кількість вершин серед усіх таких графів. У цьому графі немає ніякої

Таблиця 9.3

A	*B*	*C*	*D*	*E*	*F*	*I*	*II*	*III*	*IV*
1	2	1	3	1	2				
1	2	1	3	4	2	3	4	2	1
1	2	3	1	3	2	3	1	2	4
1	2	3	1	4	2	3	4	2	1
1	2	3	4	3	2	3	1	2	4
1	3	1	3	1	2	4	2	4	3

конфігурації, яка редукується, інакше редукований граф також не можна було б розфарбувати в чотири кольори, адже в нього менше вершин.

Таким чином, якщо в будь-якому зв'язному плоскому графі з досить великою кількістю вершин існує конфігурація, що редукується, то гіпотеза про чотири фарби є справедливою (доведено, що всі зв'язні плоскі графи з невеликою кількістю вершин розфарбувати можна). Плоскі графи взагалі заслуговують вивчення.

9.19.4. Породження плоских графів

Будемо розглядати плоскі графи разом з їх відображеннями на площину. Отже, надалі вершини плоского графа — це точки на площині, а ребра — лінії, які з'єднують їх.

Кінцями ребер є інцидентні їм вершини, інші вершини їм не належать, і вони не мають інших спільних точок, окрім спільних інцидентних вершин. Можна вважати, що ребра є ламаними лініями, які складаються зі скінченного числа відрізків прямих. Інші точки площини розбиваються графом на деяку кількість зв'язних областей.

Мінімальний зв'язний граф складається з єдиної вершини і не має ребер. Будь-які точки площини, що не збігаються з вершиною цього графа, можна з'єднати ламаною лінією, яка не проходить через неї. Отже, у мінімального графа на площині одна область (ідеться про зв'язні області). Будь-який інший зв'язний плоский граф може бути породжений з графа з меншою кількістю вершин або ребер однією з таких операцій.

1. Додання вершини степеня 1. Поза зв'язним плоским графом G, тобто в деякій області r, утворюється вершина p і сполучається ребром q з деякою вершиною графа G, розташованою на межі області r (рис. 86). Зрозуміло, що в інших областях нічого не міняється, але те, що область r залишається зв'язною, потребує доведення (хоча це й очевидно). Ми його не наводимо, оскільки для цього треба досить глибоко знати топологію площини. В нового зв'язного плоского графа G' вершин і ребер на 1 більше, ніж у графа G, а кількість областей — така сама.

2. Додання вершини степеня 2. Всередині деякого ребра q зв'язного плоского графа G утворюється вершина p. Таким чином, це ребро розбивається на два — q' та q'' (рис. 87), тобто в новому зв'язному плоскому графі G' стає на одну вершину і на одне ребро більше. Поза графом нічого не міняється, тому кількість областей залишається тією самою.

3. Розбиття області. Нове ребро q з'єднує вершини p' та p'', розташовані на межі області r, і лежить у цій області. З відомої теореми Жордана випливає, що область розбивається на дві — r' та r''. Отже, у графа G' на одне ребро і на одну область більше, ніж у графа G, а кількість вершин не змінюється (рис. 88).

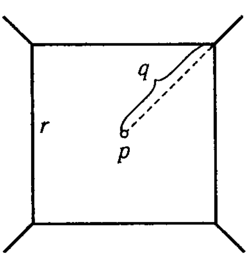


Рис. 86

Для доведення можливості породити за допомогою зазначених операцій будь-який зв'язний плоский граф використаємо обернені операції: вилучення вершин степенів 1 та 2, а також розрив циклу. Нехай задано не мінімальний зв'язний плоский граф G' із вершинами v' і ребрами e'. Якщо в ньому є вершина p' степеня 1 або 2, то можна виконати операцію її вилучення; тоді утвориться зв'язний плоский граф G. Навпаки, граф G' може бути здобутий з графа G операцією додання вершини степенів 1 або 2.

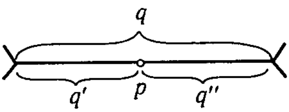


Рис. 87

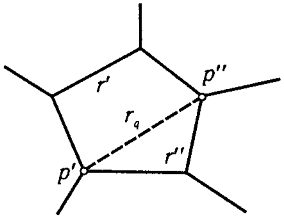


Рис. 88

Нехай тепер степені всіх вершин графа G' не менші, ніж 3. Додамо всі ці степені $n_{p'}\,(p' \in G')$. Сума дорівнює подвоєному числу ребер $2e'$, оскільки кожне ребро збільшує на 1 степені двох вершин. З іншого боку, вона не менша від потрійної кількості вершин, оскільки степінь кожної вершини не менший, ніж 3. Таким чином, $2e' = \sum_{p' \in G'} n_{p'} \geq 3v'$.

Цикломатичне число графа G' буде $e' - v' + 1 \geq 3/2v' - v' + 1 > 0$, тобто в ньому існує деякий цикл Z. Якщо вилучити з G' яке-небудь ребро q' циклу Z, то граф залишиться зв'язним. І навпаки, граф G' може бути утворений зі зв'язного плоского графа G операцією з'єднання вершин p' та p ребром q'. Оскільки при цьому виникає новий цикл Z, область r, по якій проходить ребро q', розбивається на дві.

9.19.5. Теорема Ейлера про многогранники

Нехай L — опуклий многогранник. Л. Ейлер довів, що *для кількості його вершин v, ребер e і граней g справджується співвідношення $v - e + g = 2$.* Опуклий многогранник можна спроектувати з внутрішньої точки на поверхню сфери, а з останньої — на площину. При цьому його вершини перейдуть у вершини деякої карти, ребра — в її ребра, грані — в області (одна область буде зовнішньою). Виявляється, співвідношення Ейлера справджується для кількості вершин, ребер та областей будь-якого зв'язного плоского графа.

Теорему Ейлера легко довести за методом індукції. У мінімальному зв'язному плоскому графі G_0 є одна вершина, одна область і немає ребер. Співвідношення Ейлера для нього виконується. Нехай воно справджується для всіх зв'язних плоских графів з e ребрами, а G' — довільний зв'язний плоский граф із v' вершинами, $e' = e + 1$ ребрами та r' областями. Вже було показано, що він може бути утворений з деякого зв'язного плоского графа G з e ребрами за допомогою однієї з зазначених вище операцій. Нехай v — кількість вершин графа G, r — кількість його ребер.

Якщо граф G' породжений за допомогою першої або другої операції, то $v' = v+1$, $g' = g$; якщо за допомогою третьої, то $v' = v$, $g' = g+1$. В обох випадках $v' - e' + v \ \ g = -r + g = 2$.

9.19.6. Набори конфігурацій, яких неможливо уникнути

Можна показати, що в будь-якій тріангуляції площини G є вершина, степінь якої не більший, ніж 5. Нехай v — кількість вершин тріангуляції, e — кількість ребер, g — кількість областей. Останні мають $3g$ граничних ребер, але кожне ребро є межею двох областей. Отже, $2e = 3g$, тобто $g = 2/3e$. З іншого боку, як уже зазначалося, $2e = \sum\limits_{p \in G} n_p$, де n_p — степені вершин тріангуляції. Крім того, $v = \sum\limits_{p \in G} 1$. Підставимо ці значення v, e і g у співвідношення Ейлера:

$$2 = v - e + g = v - e + 2/3e = v - 1/3e = \sum_{p \in G} 1 - 1/6 \sum_{p \in G} n_p = 1/6 \sum_{p \in G} \left(6 - n_p\right).$$

Звідси $\sum\limits_{p \in G} (6 - n_p) = 12$.

У лівій частині цієї рівності додатними є тільки члени суми зі степенями 2, 3, 4, 5 (можна показати, що вершини степеня 1 у тріангуляції немає), і, оскільки вся сума — додатна, такі члени існують, що й треба було довести.

Таким чином, набір конфігурацій, який складається з вершин степенів 2, 3, 4, 5 (рис. 89), має ту властивість, що в будь-якій тріангуляції площини є хоча б одна конфігурація з цього набору (насправді їх більше). Такі набори конфігурацій називаються наборами, яких неможливо уникнути. Наведений набір не є єдиним. Нижче будуть наведені ще два приклади. В обидва входять вершини степенів 2, 3 і 4.

Кожна вершина p тріангуляції G дає в наведену вище суму, одержану зі співвідношення Ейлера, внесок, який дорівнює $\left(6 - n_p\right)$, де n_p — її степінь. Ці внески можна по-різному перерозподілити між сусідніми вершинами. Наприклад, залишимо при кожній вершині степенів 6, 7, ... їх

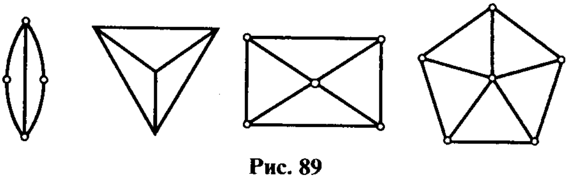


Рис. 89

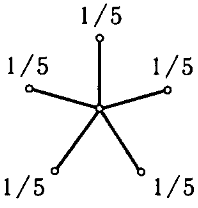


Рис. 90

внески, а внески вершин степеня 5 перенесемо в сусідні вершини, розподіливши їх порівну (рис. 90). Нехай v_p — кількість вершин степеня 5, сусідніх із вершиною p тріангуляції G. Новий внесок вершини p дорівнює $v_p/5$, якщо її степінь — 5, і $6+v_p/5-n_p$, якщо він більший. Легко показати, що

$$12=\sum_{p\in G}(6-n_p)=\sum_{p\in G,\,n_p=5} v_p/5+\sum_{p\in G,\,n_p>5}(6+v_p/5-n_p)=$$

$$=\sum_{p\in G,\,n_p\le 6} v_p/5+\sum_{p\in G,\,n_p>6}(6+v_p/5-n_p).$$

Отже, в останній сумі є додатні члени. Всі члени першої суми — додатні, але вони присутні тільки тоді, коли в тріангуляції G є дві поруч розташовані вершини степеня 5 або степенів 5 і 6 (рис. 91, *а*).

Для дослідження доданків першої суми зазначимо, що $v_p \le n_p$. Отже, $6+v_p/5-n_p \le 6-4n_p/5$. При $n_p \ge 8$ маємо $6-4n_p/5<0$. Залишається дослідити випадок $n_p=7$, $6+v_p/5-n_p>0$, тобто $v_p/5>\ell$, звідки $v_p>5$. Однак якщо серед сусідів вершини степеня 7 є шість вершин степеня 5, то серед останніх будуть сусіди між собою. Досить навіть, щоб були чотири сусідні вершини степеня 5 (узагалі, коли у вершини степеня k більше, ніж $k/2$ сусідніх вершин мають степінь 5, серед них є сусідні між собою). Отже, у тріангуляції G є або вершини степенів 2, 3 або 4, або конфігурації, зображені на рис. 91, *а*.

Можна вказати трохи інший набір конфігурацій, який замість двох сусідніх вершин степенів 5 і 6 він містить конфігурацію з трьох взаємно сусідніх вершин степенів 5, 6 та 6 (рис. 91, *б*). Якщо в тріангуляції немає вершин степенів 2, 3 і 4, двох сусідніх вершин степеня 5 та зазначеної вище конфігурації, то в кожної вершини степеня 5 не менше трьох сусідів мають степені, більші ніж 6.

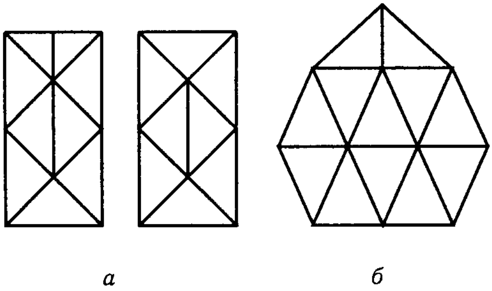

а *б*

Рис. 91

Будемо переносити додатні внески вершин степеня 5 тільки в сусідні вершини зі степенями 7, 8, ..., причому порівну. Отже, в кожну з цих вершин перенесемо не більше, ніж $v_p/3$, тобто нові внески вершин будуть не більшими, ніж $6-n_p+v_p/3$. Серед них мають бути додатні, але умова $6-n_p+v_p/3>0$ при $n_p=7$ дає $v_p>3$, при $n_p=8$ буде $v_p>6$, а при $n_p\ge 9$ це неможливо. Таким чином, у тріангуляції G є або вершина степеня 7, у якої чотири сусіди мають степінь 5, або вершина степеня 8, у якої є сім таких сусідів. Однак в обох випадках дві вершини степеня 5 мають бути сусідніми.

9.19.7. Основна ідея

Якби існував набір, якого неможливо уникнути, що складається тільки з конфігурацій, які редукуються, то мінімальним графом, що не розфарбовується, про який вже йшлося, міг би бути тільки граф G_0 з єдиною вершиною. Проте його можна розфарбувати навіть в один колір. Отже, мінімального графа, який не розфарбовується, не існувало б. Однак тоді всі зв'язні плоскі графи можна було б розфарбувати в чотири кольори. Таким чином, щоб довести гіпотезу про чотири фарби, досить знайти набір конфігурацій, які редукуються і яких неможливо уникнути.

Наведені вище приклади не підходять: вершини степенів 2, 3 та 4 є редукованими, але редукувати інші конфігурації не вдається. Треба шукати набори, яких неможливо уникнути не для всіх тріангуляцій, а тільки для тих, в яких степені всіх вершин не менші від 5.

П. Е. Аппель та Хакен будували такі набори, що складаються з великої кількості конфігурацій, за допомогою різноманітних перерозподілів внесків вершин у суму $\sum\limits_{p \in G} (6 - n_p)$. Проте як перевірити, що всі конфігурації, які входять у набір, є конфігураціями, що редукуються? Методи редукції є аналогічними тим, які використовувалися при дослідженні конфігурації, зображеної на рис. 84. Через безліч варіантів застосування цих методів розглянути їх усі вдалося тільки за допомогою машини. Було створено програму, що давала різні способи редукувати запропоновані конфігурації, але припиняла свою роботу, якщо їй не вдавалося зробити редукцію за заданий час. Тоді П. Е. Аппель і Хакен будували новий набір, який таку конфігурацію не містив (хоча, можливо, конфігурація, що вилучалася, була такою, яка редукується).

Так, надшвидкодійні машини пропробили близько 1500 год, поки не був побудований набір із 1932 конфігурацій, з яких 1931 є такими, що редукуються. Конфігурацію, яка залишилася, досліджували вручну, застосувавши більш тонкі методи редукції. Вона виявилася також конфігурацією, що редукується. Таким чином, проблему чотирьох фарб було вирішено.

9.20. Екстремальні задачі в теорії графів

Загальна постановка екстремальної задачі в теорії графів може бути сформульована таким чином: задано деякий клас графів K. У цьому класі потрібно знайти граф із максимальною $M(K)$ (мінімальною $m(K)$) кількістю ребер.

Серед перших результатів у теорії екстремальних задач треба зазначити відому теорему Турана, сформульовану в 1941 р. для класу K графів на n вершинах, що не містять повних підграфів на p вершинах.

П. Туран дістав таку оцінку максимальної кількості ребер для графів цього класу:

$$M(n, p) \le \frac{t}{2}(n - p + r + 1), \text{ де } n = t(p-1) + r; \quad 0 \le r \le p-2; \quad 3 \le p \le n.$$

Ця оцінка досягається. У класі K всіх графів на n вершинах, які не містять повних підграфів на p вершинах, є екстремальний граф $D(n, p)$. Він має n вершин, позначених $1, 2, \ldots, n$; ребро, що з'єднує i- та j-ту вершини, належить цьому графу тільки тоді, коли $i \neq j \mod(n-1)$. Графи $D(n, p)$ на n вершинах із $M(n, p)$ ребрами, які не містять повних підграфів на p вершинах, визначаються з точністю до ізоморфізму.

Розглянемо наступний клас графів K: графи на n вершинах, у яких будь-який підграф на p вершинах містить не більш як q ребер. Будемо говорити, що такі графи задовольняють (p, q) властивості. Серед цих графів треба знайти граф із максимальною кількістю ребер $M(n, p, q)$.

Доведемо кілька тверджень, які надалі будуть потрібні для розв'язання поставленої задачі.

Твердження 9.8. *Якщо у звичайному графі G маємо* $m(G) \leq n(G)-2$, *то існує вершина x із локальним степенем* $\rho_x \leq 1$.

Доведемо це твердження від супротивного. Нехай для будь-якої вершини x із $\rho_x > 2$; тоді $2m(G)=\sum\limits_{x\in V}\rho(x) \geq 2n(G)$, тобто $m(G) \geq n(G)$. Це суперечить нерівності $m(G) \leq n(G)-2$. Одержана суперечність доводить твердження 9.8.

Твердження 9.9. *Нехай G — звичайний граф й* $m(G) \leq n(G)-2$, *тоді G — незв'язний граф.*

Доведемо це твердження методом індукції за кількістю вершин у графі G.

При $n(G) = 3$ та $m(G) \leq 1$ граф є незв'язним. Нехай твердження доведено для $n(G) = n$. Доведення для $n(G) = n + 1$ проведемо від супротивного. Нехай граф має $m(G) \leq n(G)-2$ ребер, але він зв'язний; тоді жодна з вершин не може мати локальний степінь, що дорівнює нулю, інакше граф був би незв'язним. За твердженням 9.8 знайдеться вершина x із $\rho_x \leq 1$. Оскільки $\rho_x \neq 0$, маємо $\rho_x = 1$. Вилучивши цю вершину, матимемо граф G'. Очевидно, він буде зв'язним. Знайдемо кількість ребер і вершин цього графа: $m(G')=m(G')-1$, $n(G')=n(G')-1=n$. Кількість вершин графа дорівнює n, і виконується нерівність $m(G') \leq n(G')-2$. Одержана суперечність із припущенням індукції про незв'язність графа G' доводить твердження 9.9.

Твердження 9.10. *У зв'язному графі завжди існує вершина, після вилучення якої він залишається зв'язним.*

Для доведення розглянемо дві вершини x, y, відстань між якими дорівнює діаметру графа. Тоді можна вилучити вершину y, при цьому зв'язність графа не порушиться. Доведемо це від супротивного. Нехай при вилученні вершини y зв'язність графа порушилася, тобто знайшлася пара z, w незв'язних вершин. Оскільки G — зв'язний граф, усі маршрути з вершини z у вершину w проходять через вершину y, але тоді відстань від x до тієї з вершин z і w, яка не зв'язана з вершиною x, більша від діаметра графа. (Обидві вершини w та z не можуть бути зв'язаними з вершиною x, тому що в цьому випадку існував би маршрут із z у w). Прийшли до суперечності з означенням діаметра графа.

Отже, припущення про те, що при вилученні вершини y утвориться незв'язний граф, не є правильним.

Твердження 9.11. *Граф, що задовольняє властивість $(p,\ q)$ при $q \le p-2$, є незв'язним.*

Доведемо це твердження методом індукції за кількістю вершин у графі G. При $n(G) = p$ він — незв'язний, оскільки кількість ребер $\le p-2$ (за твердженням 9.9). Нехай твердження справджується для $n(G) = n$. Доведемо його для $n(G) = n + 1$ від супротивного. Нехай граф G із $n(G) = n + 1$ є зв'язним. За твердженням 9.10 існує вершина y така, що після її вилучення граф залишається зв'язним. Граф $G - y$ має задовольняти властивість (p, q), інакше граф G не задовольнятиме її. За припущенням індукції граф $G - y$ є незв'язним. Приходимо до суперечності, що й доводить твердження 9.11.

Твердження 9.12. *Якщо з графа, який задовольняє властивість $(p,\ q)$ $(q \le p-2)$, вилучити компоненту зв'язності на S вершинах із r ребрами $(S-1 \le r)$, то частина графа, що залишилася, має задовольняти властивість $(p - S,\ q - r)$.*

Доведемо це твердження від супротивного. Нехай граф G', який залишився, не задовольняє зазначену властивість. Тоді існує підграф на $p - S$ вершинах, що має ребер $\ge q-r+1$. У графі G підграф, який є об'єднанням підграфа з G' і вилученої компоненти зв'язності, має $S + p - S$ вершин та ребер $\ge r+q+1-r = q+1$, тобто не задовольняє властивість (p, q). Одержана суперечність і доводить твердження 9.12.

Наслідок. *Виконується нерівність $M_k\,(n,\ p,\ q) \le r + M\,(n-\ell,\ p-\ell,\ q-r)$.*

При розв'язуванні поставленої задачі виникають природні обмеження: $n \ge p,\ p \ge 2$.

Якщо $p = 1$, то q має дорівнювати нулю. Єдиний граф, що задовольняє властивість (1, 0), є порожнім, тобто $M(n,\ 1,\ 0) = 0$, де $M(n,\ p,\ q)$ — максимальна кількість ребер, яку може мати граф на n вершинах, що задовольняє властивість (p, q).

Очевидно, якщо $q \le [p/2]$, то $M(n, p,\ q) = q$.

Розглянемо випадок $q = [p/2]$, де $[p/2] \ne p-2$, тобто $p \ne 3,4$.

Оскільки $q = [p/2] \le p-2$, розв'язок задачі слід шукати в класі K незв'язних графів. Розіб'ємо цей клас на два класи: K_1 — клас графів, у яких існує компонента зв'язності з кількістю ребер $q_1,\ 2 \le q_1 \le [p/2]$ (кількість вершин у компоненті зв'язності має задовольняти нерівність $3 \le \ell \le ([p/2]-1)$; K_2 — клас графів, всі компоненти зв'язності якого мають ≤ 1 ребер.

Знайдемо $M_{k_2}(n, p, [p/2])$ — максимальну кількість ребер графа з класу K_2.

Твердження 9.13. *Справджується рівність M_{k_2} $(n,\ p,\ [p/2]) = [n/2]$.*

Доведемо це твердження від супротивного. Якщо $M_{k_2}(n, p, [p/2]) \ge$ $\ge [n/2]+1$, то існує компонента зв'язності з кількістю ребер ≥ 2. Потрапляємо в клас K_1. Отже, $M_{k_2}\,(n, p, [p/2]) \le [n/2]$.

Нехай граф має x_1 компонент зв'язності з нульовою кількістю ребер, а x_2 — з кількістю ребер, яка дорівнює 1; загальна кількість ребер у графі

$m(G) = x_2$, а кількість вершин $n(G) = x_1 + 2x_2 = n$. Виникає задача цілочислового програмування: знайти максимум функції $f(x_1, x_2) = x_2$ при обмеженнях $x_1 + 2x_2 = n$; $x_i \geq 0$, $i = 1, 2$; x_i — цілі числа.

Задача цілочислового програмування для двох змінних має розв'язок $x_1 = r$, $x_2 = [n/2]$, де $n = 2[n/2] + r$, $r = 0{,}1$.

Знаходимо $M_{k_2}(n, p, [p/2]) = n/2$. Екстремальний граф має $[n/2] + r$ компонент зв'язності ($n = 2[n/2] + r$, $r = 0{,}1$), з яких $[n/2]$ компонент мають дві вершини й одне ребро, а r компонент — порожній граф на r вершинах.

Покажемо, що $M_{k_1}(n, p, [p/2]) \leq M_{k_2}(n, p, [p/2])$.

Вилучимо компоненту зв'язності з кількістю ребер q_1, $2 \leq q_1 \leq [p/2]$. Згідно з наслідком із твердження 9.12

$$M_{k_1}(n, p, [p/2]) \leq q_1 + M(n - S, p - S, [p/2] - q_1).$$

Покажемо, що

$$[p/2] - q_1 \leq \left[\frac{p-S}{2}\right] - 1;$$

тоді

$$M(n - S, p - S, [p/2] - q_1) = [p/2] - q_1;$$

$$[p/2] - q_1 - \left[\frac{p-S}{2}\right] + 1 \leq 0.$$

Оскільки $\left[\frac{p-S}{2}\right] \geq [p/2] - [S/2] - 1$, маємо $-\left[\frac{p-S}{2}\right] \leq -[p/2] + [S/2] + 1$.

Отже, $[p/2] - q - [p/2] + [S/2] + 1 \leq 0$.

Оскільки $q_1 \geq S - 1$, дістаємо $-q_1 \leq -S + 1; -S + 1 + [S/2] + 1 \leq 0;$ $[S/2] + 2 \leq S$. Очевидно, при $S \geq 3$ нерівність $[S/2] + 2 \leq S$ виконується. Знаходимо $M_{k_1}(n, p, [p/2]) \leq q_1 + [p/2] - q_1 \leq [n/2] = M_{k_2}(n, p, [p/2])$.

Розглянемо випадок $q = p - 2$. Розв'язок задачі шукаємо в класі K незв'язних графів (див. твердження 9.11). Розіб'ємо клас K на два класи: K_1 — клас графів, у яких існує компонента зв'язності на S вершинах із m ребрами $(q \geq m \geq S)$; $K_2 = K - K_1$.

Знайдемо екстремальний граф у класі K_2. Нехай граф містить:

x_0 — компоненту зв'язності з кількістю ребер = 0;

x_1 — компоненту зв'язності з кількістю ребер = 1;

……………………………………………;

x_q — компоненту зв'язності з кількістю ребер = q.

Маємо задачу цілочислового програмування. Потрібно знайти максимум функції $f(x_1, x_2, ..., x_q) = x_0 + 2x_1 + ... + qx_q$ (кількість ребер) при обмеженнях $x_0 + 2x_1 + 3x_2 + ... + (q+1)x_q = n$, всі $x_i \geq 0$ — цілі числа.

Як відомо, максимум лінійної функції досягається на межі в кутових точках, що розташовуються на перетині осей координат із гіперплощиною $x_0 + 2x_1 + 3x_2 + ... + (q+1)x_q = n$. Вони мають вигляд $(n, 0, 0, ..., 0)$; $(0, n/2, 0, ..., 0)$; $(0, 0, n/3, ..., 0)$; …; $(0, 0, 0, 0, ..., n/(q + 1))$.

Очевидно, максимальне значення функції $f(x_0, x_1, ..., x_q)$ задачі з ослабленими обмеженнями досягається в точці $(0, 0, 0, 0, ..., n/(q + 1))$, де воно дорівнює $qn/(q+1)$. Нехай $n = t(q+1) + r$, де $0 \le r \le q$. Розв'язком задачі цілочислового програмування буде точка з цілими координатами, яка належить гіперплощині та близька до точки $(0, 0, 0, 0, \ldots, n/(q+1))$. Очевидно, точка $(0, 0, \ldots, 1, \ldots, [n/q+1])$ (1 записується на r-му місці) належить гіперплощині, де функція

$$f(0,\ldots,1,\ldots[n/(q+1)]) = q[n/(q+1)]+r-1 = \left[q\frac{n}{q+1}\right]-1 = \left[(p-2)\frac{n}{p-1}\right]-1.$$

Отже, $M_{k_2}(n, p, p-2) = \left[(p-2)\frac{n}{p-1}\right]-1.$

Ця оцінка досягається на наступному графі $P(n, p, p-2)$, що має n вершин ($n = t(p-1) + r$) і складається з $t + 1$ компонент зв'язності: t компонент на $p-1$ вершині містять $p-2$ ребра (є деревами), одна компонента на r вершинах містить $r-1$ ребро. Справді, $P(n, p, p-2)$ задовольняє властивість $(p, p-2)$ і має $t(p-2)+r-1 = \left[\frac{n}{p-1}\right](p-2)+r-1 = \left[(p-2)\frac{n}{p-1}\right]-1$ ребро.

Очевидно $M_{k_1}(n, p, p-2) \le M_{k_2}(n, p, p-2)$.

Розглянемо випадок $q = C_p^2$. При цьому будь-який підграф на p вершинах є повним. Очевидно, $M(n, p, C_p^2) = C_n^2$.

Розглянемо випадок $q = C_p^2 - 1$. Теорема Турана стверджує, що

$$M(n, p, C_p^2 - 1) + 1/2\frac{p-2}{p-1}\left(n^2 - r^2\right) + 1/2\, r(r-1),$$

$$\text{де } n = t(p-1) + r,\ 0 \le r \le p-2.$$

Розглянемо наступну задачу. У класі графів на n вершинах задано властивість $(p, \ge k)$: будь-який підграф на p вершинах має $\ge k$ ребер. Серед таких графів потрібно знайти граф із мінімальною кількістю ребер. Нехай $m\ (n, p, \ge k)$ — мінімальна кількість ребер у класі графів, які задовольняють властивість $(p, \ge k)$, $M(n, p, \le C_p^2 - k)$ максимальна кількість ребер у класі графів, що задовольняють властивість $(p, \le C_p^2 - k)$ (будь-який підграф на p вершинах має $\le C_p^2 - k$ ребер).

Очевидно, виконується співвідношення $M(n, p, C_p^2 - 1) + m\ (n, p, \ge k) = C_p^2$.

Знайдемо $m\ (n, p, \le 1)$. Граф, який задовольняє властивість $(p, \le 1)$, має нещільність $\varepsilon(G) \le p-1$.

Клас графів, що задовольняють властивість $(p, \ge 1)$, розіб'ємо на два класи: K_2 — клас графів із $\varepsilon(G) \le p-2$; K_1 — клас графів із $\varepsilon(G) = p-1$.

Маємо $m_{k_2}(n, p, \ge 1) \ge m_{k_1}(n, p, \ge 1)$, оскільки при вилученні ребра число $\varepsilon(G)$ може зрости тільки на одиницю, і знов утворений граф задовольнятиме властивість $(p, \ge 1)$.

Доведемо, що

$$m_{k1}(n, p, \ge 1) = 1/2t(n-p+1+r), \qquad (9.3)$$

індукцією за кількістю вершин у графі G при $n = p$. Очевидно, $m_{k_1}(p, p, \geq 1) = 1$. З іншого боку, $n = p = 1(p - 1) + 1$, тобто $t = 1$, $r = 1$; тоді $m_{k_1}(p, p, \geq 1) = 1/2\ (p - p + 1 + 1) = 1$.

Нехай рівність (9.3) справджується для $n(G) = n - 1$. Доведемо її для $n(G) = n$. Оскільки $\varepsilon(G) = p - 1$, знайдеться $\varnothing$-підграф графа G на $p - 1$ вершині. Будь-яка вершина з тих $n - p + 1$, які залишилися, сполучається принаймні з однією з вершин цього $\varnothing$-підграфа. Після вилучення $\varnothing$-підграфа на $p - 1$ вершині, граф, що залишиться, на $n - p + 1$ вершині задовольнятиме властивість $(p, \geq 1)$. Кількість його ребер не менша, ніж $m(n - \mathrm{p} + 1, p, \geq 1) \geq m_{k_1}(n - p + 1, p, \geq 1)$.

Отже, $m_{k_1}(n, p, \geq 1) \geq n - p + 1 + m_{k_1}(n - p + 1, p, \geq 1)$, оскільки $n - p + 1 = (t - 1)(p - 1) + r$. За припущенням індукції виконуються такі рівності:

$$m_{k_1}(n - p + 1, p, \geq 1) = 1/2(t - 1)(n - 2p + 2 + r);$$

$$m_{k_1}(n, p, \geq 1) \geq n - p + 1 + 1/2(t - 1)(n - 2p + 2 + r) = 1/2t(n - p + 1 + r).$$

Отже, $m_{k_1}(n, p, \geq 1) \geq 1/2t(n - p + 1 + r)$.

Ця оцінка є точною. Екстремальний граф $d(n, p)$ — це граф на n вершинах ($n = t(p - 1) + r$), який має $p - 1$ повну компоненту зв'язності: r компонент на $t + 1$ вершині; $p - 1 - r$ — на t вершинах.

Очевидно, цей граф належить класу K_1. Підрахуємо кількість ребер у графі $d(n, p)$:

$$m(d(n, p)) = rC_{t+1}^2 + (p - 1 - r)C_t^2 = \frac{r(t+1)t}{2} + (p - 1 - r)\frac{t(t-1)}{2} =$$

$$= t/2(n + r - p + 1).$$

Тоді

$$m_{k_1}(n, p, \geq 1) = 1/2t(n - p + 1 + r)$$

$$\text{й } M\left(n, p, \leq C_p^2 - 1\right) = C_n^2 - 1/2t(n - p + 1 + r) =$$

$$= \frac{n(n-1)}{2} - 1/2t(n - p + 1 + r) = \frac{(p-2)\left(n^2 - r^2\right)}{2(p-1)} + \frac{r(r-1)}{2}.$$

Для туранівського графа $D(n, p)$, що є доповненням графа $d(n, p)$, кількість ребер дорівнює

$$\frac{(p-2)(n^2 - r^2)}{2(p-1)} + \frac{r(r-1)}{2}.$$

Розглянемо випадок $q = C_p^2 - 2$. Знайдемо $m(n, p, \geq 2)$. При цьому клас графів, які задовольняють властивість $(p, \geq 2)$, розіб'ємо на три класи: K_1 — клас графів із $\varepsilon(G) = p - 1$; K_2 — клас графів із $\varepsilon(G) = p - 2$; K_3 — клас графів із $\varepsilon(G) \leq p - 3$. Тут $m_{k_3}(n, p, \geq 2) \geq m_{k_i}(n, p, \geq 2)$ ($i = 1, 2$), оскільки при вилученні ребра число $\varepsilon(G)$ може тільки зрости на одиницю, а утворений граф задовольнятиме властивість $(p, \geq 2)$.

Доведемо, що

$$m_{k_2}(n, p, \ge 2) = 1/2t(n-p+2+r), \tag{9.4}$$

де $n = t(p-2) + r$, $0 \le r \le p-3$.

Доведення проведемо індукцією за кількістю вершин n. При $n = p$, очевидно, що $m_{k_2}(n, p, \ge 2) = 2$. Нехай рівність (9.4) справджується для $n(G) \le n-1$, доведемо її для $n(G) = n$. Оскільки $\varepsilon(G) = p-2$, знайдеться $\varnothing$-підграф на $p-2$ вершинах. Будь-яка з $n-p+2$ вершин, які залишилися, є інцидентною принаймні одній з цього підграфа. Вилучаючи $\varnothing$-підграф на $p-2$ вершині, граф, що залишився, на $n-p+2$ вершині задовольнятиме властивість $(p, \ge 2)$ і потрапить у клас $K_2 \cup K_3$. Кількість його ребер не менша, ніж $m(n-p+2, p, \ge 2) \ge m_{k_2}(n-p+2, p, \ge 2)$.

Отже, $m_{k_2}(n, p, \ge 2) \ge n-p+2+m_{k_2}(n-p+2, p, \ge 2)$.

Оскільки $n-p+2 = (t-1)(p-2)+r$, за припущенням індукції випливає, що

$$m_{k_2}(n-p+2, p, \ge 2) = 1/2(t-1)(n-2p+4+r);$$

$$m_{k_2}(n, p, \ge 2) \ge n-p+2+1/2(t-1)(n-2p+4+r) = 1/2t(n-p+2+r).$$

Маємо $m_{k_2}(n, p, \ge 2) \ge 1/2t(n-p+2+r)$. Рівність досягається для графа $d(n, p-1)$.

Покажемо, що

$$m_{k_1}(n, p, \ge 2) \ge 1/2t(n-p+2+r). \tag{9.5}$$

Доведення проведемо за кількістю вершин у графі.

При $n(G) = p$ очевидно, що $m_{k_1}(p, p, \ge 2) = 2$. З іншого боку, при $p \ge 5$ матимемо $t[p/p-2] = 1$ і $m_{k_2}(p, p, \ge 2) = 1/2(p+2-p+2) = 2$.

При $p = 4$ дістаємо $m_{k_2}(p, p, \ge 2) = 1/2[4/2](4+2-2\cdot 2) = 2$, тобто при $p \ge 4$ буде $m_{k_1}(p, p, \ge 2) \ge m_{k_2}(p, p, \ge 2)$.

Нехай при $n(G) \le n-1$ нерівність (9.5) виконується, доведемо її для $n(G) = n$. Оскільки $\varepsilon(G) = p-1$, знайдеться порожній підграф на $p-1$ вершині. Будь-яка з $n-p+1$ вершин, які залишилися, є інцидентною принаймні двом вершинам із цього підграфа (граф задовольняє властивість $(p, \ge 2)$). Вилучаючи порожній підграф на $p-1$ вершині, граф, що залишився, на $n-p+1$ вершині задовольнятиме властивість $(p, \ge 2)$. Кількість його ребер — не менша, ніж

$$m(n-p+1, p, \ge 2) \ge m_{k_2}(n-p+1, p, \ge 2) = 1/2t_1(n-2p+3+r_1),$$

де $n-p+1 = t_1(p-2)+r_1$, звідки $n = (p-2)(t_1+1)+r_1+1$. Отже, $t = t_1+1$, $r = r_1+1$. Тоді

$$m_{k_1}(n, p, \ge 2) \ge 2(n-p+1)+1/2t_1(n-2p+3+r_1).$$

Легко показати, що $2(n-p+1)+1/2t_1(n-2p+3+r_1) \ge 1/2(n-p+2+r_1)$.

Отже, $m(n, p, \geq 2) = 1/2(n-p+2+r)$ й

$$M\left(n, p, \leq C_p^2 - 2\right) = \frac{(n-1)n}{2} - t/2(n-p+2+r) =$$

$$= 1/2\frac{(n^2-r^2)}{(p-2)}(p-3) + \frac{r(r-1)}{2} = \frac{(p-3)(n^2-r^2)}{2(p-2)} + \frac{r(r-1)}{2}.$$

Екстремальним є туранівський граф $D(n, p-1)$.

Розглянемо випадок $q = C_p^2 - 3$. Знайдемо $m(n, p, \geq 3)$. У цьому разі граф, який задовольняє властивість $(p, \geq 3)$, розіб'ємо на чотири класи: K_1 — клас графів із $\varepsilon(G) = p-1$; K_2 — клас графів із $\varepsilon(G) = p-2$; K_3 — клас графів із $\varepsilon(G) = p-3$; K_4 — клас графів із $\varepsilon(G) \leq p-4$.

Очевидно, $m_{k_4}(n, p, \geq 3) \geq m_{k_i}(n, p, \geq 3)$ (I = 1, 2, 3). Аналогічно доведенню рівностей (9.4) і (9.5) можна показати, що

$$m_{k_3}(n, p, \geq 3) = 1/2t(n-p+3+r), \tag{9.6}$$

де $n = t(p-3) + r$, $0 \leq r \leq p-4$.

Доведемо, що

$$m_{k_1}(n, p, \geq 3) \geq m_{k_3}(n, p, \geq 3) = 1/2t(n-p+3+r). \tag{9.7}$$

При $n = p$ маємо $m_{k_1}(p, p, \geq 3) = 3$ і, якщо $p \geq 6$, то $m_{k_2}(p, p, \geq 3) = 3$.

Нехай нерівність (9.7) справджується для $n(G) \leq n-1$, доведемо її для $n(G) = n$. Оскільки $\varepsilon(G) = p-1$, знайдеться порожній підграф на $p-1$ вершині. Будь-яка з $n-p+1$ вершин, які залишилися, сполучається принаймні з трьома вершинами цього підграфа (граф задовольняє властивість $(p, \geq 3)$). Вилучаючи $\varnothing$-підграф на $p-1$ вершині, граф, що залишиться на $n-p+1$ вершині задовольнятиме властивість $(p \geq 3)$. Кількість його ребер не менша, ніж $m(n-p+1, p, \geq 3)$. За припущенням індукції $m(n-p+1, p, \geq 3)$ маємо $n-p+1 = t_1(p-2)+r_1$; отже, $t = t_1+1, r = r_1+2$. Тоді

$$m_{k_1}(n, p, \geq 3) \geq 3(n-p+1) + 1/2t_1(n-2p+4+r_1).$$

Легко показати, що

$$t/2(n-p+3+r) - t_1/2(n-2p+4+r_1) \leq 3(n-p+1).$$

Очевидно, $m_{k_2}(n, p, \geq 3) \geq m_{1k_3}(n, p, \geq 3)$; $m(n, p, \geq 3) = 1/2 + (n-p+$ $+3+r)$, де $n = t(p-3+r)$, $0 \leq r \leq p-4$.

Покажемо, що $M_{k_1}(n, p, p-2) \leq M_{k_2}(n, p, p-2)$.

Лема. *У графах класу K_1 існує цикл D.*

Доведення є очевидним. Нехай G — граф на $n(G)$ вершинах із $m(G)$ ребрами з класу K_1, який задовольняє властивість (p, q), $q \leq p-2$. Занумеруємо вершини графа G. За лемою у G існує цикл у компоненті зв'язності на S вершинах з m ребрами $(q \geq m \geq S)$. Нехай v_i, v_{i+1} — дві суміжні вершини цього циклу. Оскільки K_1 — клас незв'язних графів, існує вершина v_k $(k \neq i, i+1)$, що належить іншій компоненті зв'язності. Зіставимо графу G граф G_1 так:

- $n(G) = n(G_1)$;
- у G_1 зберігаємо позначення вершин графа G;
- відношення суміжності зберігаємо в G_1 для всіх вершин, крім v_i, v_{i+1};
- у G_1 вершина є v_i суміжною з вершиною v_k.

Твердження 9.14. *Справджуються такі рівності:*

- $m(G_1) = m(G)$;
- *цикломатичне число* $\gamma(G_1) = \gamma(G) - 1$;
- *граф* G_1 *задовольняє властивість (p, q).*

Перші дві рівності є очевидними. Третю рівність доведемо від супротивного: нехай G_1 не задовольняє властивість (p, q) при $q \leq p-2$, тобто існує підграф G_2' графа G_1 на p вершинах із $m\left(G_2'\right) > q$. Оскільки кількість ребер у підграфах G_i може збільшитися на одиницю порівняно з підграфами на тих самих вершинах графа G, $m(G_2') = q+1$. Очевидно, G_2' містить вершини v_i, v_k і не містить вершину v_{i+1}.

Доведемо, що в G' є вершина v_x із $\rho_{vx} \leq 1$, від супротивного: нехай для будь-якої вершини $v \in G_2'$ буде $\rho_v \geq 2$. Тоді $m\left(G_2'\right) = \frac{\sum \rho_v}{2} \geq p$, а це суперечить тому, що $m\left(G_2'\right) = q+1 \leq p-1$.

Розглянемо підграф G' графа G на тих самих вершинах, що й G_2', замінивши лише вершину v_x на v_i; $\rho_{v_k} \geq 1$, оскільки вона належить у G деякому циклу.

Тоді $m(G') \geq m\left(G_2'\right) = q+1$, а це суперечить тому, що граф G задовольняє властивість (p, q). Одержана суперечність доводить твердження 9.14.

Очевидно, послідовним виконанням дій, описаних у твердженні 9.14, для будь-якого графа G із класу K_2, що задовольняє властивість (p, q), де $q = p - 2$, можна знайти граф G_1 класу K_2, який задовольняє властивість (p, q) і такий, що $n(G) = n(G_1)$, $m(G) = m(G_1)$.

9.21. Булеві матриці

Очевидно, матриці суміжності Δ й інцидентності E звичайного графа та матриця суміжності орієнтованого псевдографа D без кратних дуг складаються з нулів та одиниць. Назвемо матрицю, що складається з нулів і одиниць, булевою. Очевидно, матриця інцидентності псевдографа G також є булевою. Із булевими матрицями можна виконувати звичайні логічні операції, такі як диз'юнкція та кон'юнкція, здійснюючи їх з елементами матриць. Крім того, введемо операцію логічного множення булевих матриць, яку означимо $*$. Нехай $A = \left\| a_{ij} \right\|$ — булева матриця вимірністю $m \times n$, а $B = \left\| b_{ij} \right\|$ — булева матриця вимірністю $k \times n$. Тоді $A*B=C=\left\| c_{ij} \right\|$ — булева матриця вимірністю $n \times k$, де $c_{ij} = \bigvee\limits_{t=1}^{n} (a_{it} \,\&\, b_{tj})$, $i = \overline{1,m}$, $j = \overline{1,k}$.

Введемо ще операцію sign переходу від довільної ($m \times n$) матриці $D = \|d_{ij}\|$ із невід'ємними елементами до булевої ($m \times n$) матриці $C = \|c_{ij}\| = \text{sign}\,(D)$, в якої $c_{ij} = \text{sign}\left(d_{ij}\right)$, $i = 1, 2, ..., n$, де

$$\text{sign}\,(x) = \begin{cases} 1, \text{ якщо } x > 0; \\ 0, \text{ якщо } x = 0. \end{cases}$$

Неважко показати, що для будь-яких двох булевих матриць D_1 і D_2 справджуються рівності

$$\begin{aligned} &\text{sign}(D_1 + D_2) = \text{sign}(D_1) \vee \text{sign}(D_2); \\ &\text{sign}(D_1 D_2) = \text{sign}(D_1) * \text{sign}(D_2). \end{aligned} \tag{9.8}$$

Слід зазначити, що булеві матриці більш економічні в обчислювальному відношенні, ніж цілочислові: вони потребують меншого обсягу пам'яті; крім того, виконання на ЕОМ логічних операцій над булевими матрицями потребує меншого обсягу обчислень, ніж над цілочисловими матрицями тієї самої вимірності.

Твердження 9.15. *З усякого незамкненого маршруту (шляху) можна вилучити простий ланцюг із початковою та кінцевою вершинами початкового маршруту.*

Доведемо це твердження індукцією за k — кількістю ребер у маршруті. При $k = 1$ кожний маршрут є простим ланцюгом. Нехай твердження справджується для $k \le n-1$. Розглянемо довільний маршрут $M = v_1 e_1 v_2 ...$ $...e_n v_{n+1}$, де $v_1 \ne v_{n+1}$, завдовжки n, зупинившись на двох номерах i, та j, де $1 \le i < j \le n+1$, таких, що $v_i = v_j$. Якщо цих номерів немає, то маршрут, який розглядається, є простим ланцюгом. Якщо ж зазначені номери є, то розглядають підмаршрут

$$v_1 e_1 v_2 ... e_{i-1} v_i e_j v_{j+1} ... e_n v_{n+1}$$

(тобто передбачається, що $i \ne j$, $j \ne n+1$) завдовжки $< n$, а з нього внаслідок припущення індукції можна вилучити простий ланцюг, що сполучає вершини v_1, v_{n+1}.

Твердження 9.16. *У псевдографі G (в орієнтованому псевдографі D) з циклу можна вилучити простий цикл.*

Це твердження доводиться аналогічно попередньому.

Уведемо поняття композиції маршрутів. Нехай $M_1 = v_1 e_1 v_2 ... e_{k-1} v_k$, $M_2 = v_k e_k v_{k+1} ... e_{\ell-1} v_\ell$ — маршрути в графі D, де $k \ge 2$, $\ell \ge k+1$. Назвемо маршрут $M_1 \circ M_2 = v_1 e_1 v_2 ... e_{k-1} v_k e_k v_{k+1} ... e_{\ell-1} v_\ell$ композицією маршрутів M_1 і M_2. Аналогічно означається композиція шляхів (маршрутів) в орієнтованому графі.

Позначимо $\Delta^k = \|\delta_{ij}\|^k$ k-й степінь матриці суміжності Δ (D) орграфа D (аналогічно для графа G).

Твердження 9.17. *Нехай задано матрицю Δ^k орієнтованого псевдографа D=(V, E), де $V = \{V_1, ..., V_n\}$. Тоді елемент δ_{ij}^k дорівнює кількості всіх шляхів із $V_i y V_j$ завдовжки k.*

Доведення можна провести методом індукції за k.

Аналогічно формулюється та доводиться твердження для псевдографа $G = (V, \mathrm{E})$.

Твердження 9.18. *Для того щоб n-вершинний орграф D з матрицею суміжності Δ мав хоча б один цикл, необхідно й достатньо, щоб матриця $A = \Delta^2 + \Delta^3 + \ldots + \Delta^n$ мала ненульові діагональні елементи.*

Для доведення необхідності розглянемо деякий цикл в орграфі D. Як показано вище, із циклу можна вилучити простий цикл, довжина якого $\leq n$. Однак тоді внаслідок твердження 9.17 для будь-якої вершини v_i, що належить деякому простому циклу завдовжки ℓ, де $2 \leq \ell \leq n$, знайдеться вершина v_j, зв'язана з v_i, тобто елемент δ_{ij}^{ℓ} матриці Δ^{ℓ} відрізняється від нуля, і отже, елемент α_{ij} матриці А не дорівнює нулю.

Для доведення достатності припустимо, що в матриці $\mathrm{A} = \left\|\alpha_{ij}\right\|$ для деякого номера i виконується нерівність $k_{ij} > 0$. Це означає, що для деякого r справджується нерівність $\delta_{ij}^{r} > 0$. Це означає, що знайдуться вершина і деяке v, що δ_{ij}^{r}, тобто внаслідок твердження 9.17 знайдеться шлях у D з v_i у v_j, а внаслідок твердження 9.16 в орграфі D міститиметься простий цикл.

Аналогічно можна довести твердження 9.18 для випадку орієнтованого n-вершинного псевдографа D.

Означення 9.33. *Нехай $D = (V, E)$ — орієнтований граф. Матрицею однобічної зв'язності орієнтованого графа D називається квадратна матриця $T(D) = \left\|t_{ij}\right\|$ порядку n, у якої $t_{ij} = 1$, якщо існує шлях із вершини v_i у v_j, і $t_{ij} = 0$ — в іншому випадку. Матрицею сильної зв'язності орграфа D є квадратна матриця $S(D) = \left\|s_{ij}\right\|$ порядку n, у якої $s_{ij} = 1$, якщо існує шлях із вершини v_i у вершину v_j й одночасно існує шлях із вершини v_j у v_i, та $s_{ij} = 0$ — в іншому випадку (тобто $s_{ij} = 1$ тоді й тільки тоді, коли вершини v_i, v_j належать одній компоненті сильної зв'язності орграфа D).*

Означення 9.34. *Нехай $G = (V, E)$ — граф. Матрицею зв'язності графа G називається матриця $S(G) = \left\|s_{ij}\right\|$ порядку n, у якої $s_{ij} = 1$, якщо $i = j$ або існує маршрут, що з'єднує v_i з v_j, та $s_{ij} = 0$ — в іншому випадку (тобто $s_{ij} = 1$ тоді й тільки тоді, коли вершини v_i, v_j належать одній компоненті зв'язності графа G).*

Скориставшись твердженням 9.17, рівностями (9.8), а також тим фактом, що внаслідок твердження 9.15 із будь-якого незамкненого маршруту або шляху можна вилучити простий ланцюг із тими самими початковою та кінцевою вершинами, дістанемо правильність наступних тверджень.

Твердження 9.19. *Нехай задано граф $G = (V, E)$, $n(G) = n$, а Δ — його матриця суміжності. Тоді*

$$S(G) = \operatorname{sign}\left(E + \Delta + \Delta^2 + \ldots + \Delta^{n-1}\right) = E \vee \Delta \vee \Delta^2 \vee \ldots \vee \Delta^{n-1},$$

де E — одинична матриця порядку n.

Твердження 9.20. *Нехай задано орієнтований граф $D = (V, E)$, $n(D) = n$, а Δ — його матриця суміжності. Тоді:*

- $T(D)=\operatorname{sign}\left(E+\Delta+\Delta^2+...+\Delta^{n-1}\right)=E\vee\Delta\vee\Delta^2\vee...\vee\Delta^{n-1}$;
- $S(D)=T(D)\&[T(D)]^T$,

де T — позначення операції транспортування матриці.

Твердження 9.19 і 9.20 дають прості, легкі методи обчислення матриць $S(G)$, $T(D)$, $S(D)$, які легко реалізуються на ЕОМ. Існують також більш економічні методи обчислення цих матриць. Опишемо, наприклад, метод Уоршелла, що ґрунтується на такому твердженні, наведеному за [2].

Твердження 9.21. *Нехай* Δ *— матриця суміжності графа* $G=(V,E)$ *(орієнтованого графа* $D=(V,E)$*), де* $V=\{v_1,...,v_n\}$*. Нехай є послідовність булевих квадратних матриць* B^ℓ *порядку* n*, де* $\ell=0,1,...,n$*, а* $B^0=\Delta\vee E$*. Нехай елементи матриць* B^ℓ *обчислюються за такою ітераційною формулою:*

$$b_{ij}^{\ell}=b_{ij}^{\ell-1}\vee\left(b_{i\ell}^{\ell-1}\&b_{\ell j}^{\ell-1}\right),$$

де $\ell=1,2,...,n$*. Тоді* $S(G)=B^n$ *(і відповідно* $T(D)=B^n$*,* $S(D)=T(D)\&$ $\&[T(D)]^T$ *для орієнтованого графа* D*).*

Доведення будемо проводити для G (для D воно аналогічне). Покажемо індукцією за ℓ, що $b_{ij}^{\ell}=1$ тоді й тільки тоді, коли або $i=j$, або існує маршрут, який зв'язує v_i, v_j і внутрішні вершини якого належать множині $\{v_1...v_n\}$. Із цього твердження при $\ell=n$ випливає правильність твердження 9.21. При $\ell=0$ елементи b_{ij}^{0} матриці $B^0=A\vee E$, очевидно, задовольняють потрібну умову.

Припустимо, що при деякому ℓ, де $1\le\ell\le n$, елементи $b_{ij}^{\ell-1}$ також задовольняють цю умову. Покажемо виконання її для елементів b_{ij}^{ℓ}. Нехай $i\ne j$ та існує маршрут, що зв'язує v_i і v_j, внутрішні вершини якого належать множині $\{v_i...v_n\}$. Доведемо, що тоді $b_{ij}^{\ell}=1$. Нехай M — простий ланцюг (інакше відповідно до твердження 9.15 вилучимо з M простий ланцюг, який зв'язує v_i, v_j). Якщо v_i не є внутрішньою вершиною ланцюга M, то внаслідок індуктивного припущення $b_{ij}^{\ell-1}=1$, звідки $b_{ij}^{\ell}=1\vee$ $\vee\left(b_{i\ell}^{\ell-1}\&b_{\ell j}^{\ell-1}\right)=1$. Нехай тепер v_i є внутрішньою вершиною ланцюга M, тобто він має вигляд $M=M_1\circ M_2$, де M_1,M_2 — прості ланцюги, що зв'язують вершини v_i з v_ℓ, а v_ℓ з v_j відповідно, внутрішні вершини яких належать множині $\{v_1...v_{\ell-1}\}$. Однак тоді внаслідок індуктивного припущення $b_{i\ell}^{\ell-1}=b_{\ell j}^{\ell-1}=1$, звідки $b_{ij}^{\ell}=b_{ij}^{\ell-1}\vee(1\&1)=1$.

Нехай $b_{ij}^{\ell}=1$, $i\ne j$. Покажемо, що існує маршрут, який зв'язує v_i, v_j, а його внутрішні вершини належать множині $\{v_1...v_n\}$. Через те що $b_{ij}^{\ell}=b_{ij}^{\ell-1}\vee\left(b_{i\ell}^{\ell-1}\&b_{\ell j}^{\ell-1}\right)=1$, або $b_{ij}^{\ell-1}=0$, або $b_{i\ell}^{\ell-1}=b_{\ell j}^{\ell-1}=1$. Якщо $b_{ij}^{\ell-1}=1$, то внаслідок індуктивного припущення існує маршрут, який зв'язує v_i, v_j, а його внутрішні вершини належать множині $\{v_1...v_{\ell-1}\}$. Якщо $b_{ij}^{\ell-1}=0$, $b_{i\ell}^{\ell-1}=b_{\ell j}^{\ell-1}=1$, то внаслідок індуктивного припущення існують маршрути M_1, M_2, що зв'язують v_i, v_ℓ і v_ℓ, v_j відповідно, а їхні внутрішні вершини належать множині $\{v_1...v_{\ell-1}\}$. Отже, маршрут $M_1\circ M_2$ є шуканим.

9.22. Вилучення компонент зв'язності

Важливе значення мають знаходження числа компонент сильної зв'язності орієнтованого графа, а також вилучення їх.

Опишемо алгоритм, користуючись двома такими твердженнями.

Твердження 9.22. *Нехай D — орієнтований граф, що містить* $p\,(p \geq 2)$ *компонент сильної зв'язності* $D_1,...,D_p$. *Тоді* D_2 *— орієнтований граф, який утворюється після вилучення із D вершин, що містяться в* D_1, *має* $p-1$ *компоненту сильної зв'язності:* $D_2,...,D_p$.

Твердження 9.23. *Нехай* D_1 *— компонента сильної зв'язності орієнтованого графа D, який має не менше двох компонент сильної зв'язності;* D_2 *— орієнтований граф, що утворюється після вилучення з D вершин, які містяться в* D_1. *Тоді матриці* $\Delta(D_1)$, $S(D_2)$ *є матрицями, які утворюються внаслідок вилучення з матриць* $\Delta(D)$ *і* $S(D)$ *рядків та стовпців, що відповідають вершинам орієнтованого графа* D_1.

З означення матриці сильної зв'язності випливає справедливість такого твердження.

Твердження 9.24. *Нехай задано орієнтований граф D = (V, E), де* $V=\{v_1,...,v_n\}$, *із матрицею сильної зв'язності S(D). Тоді одиниці її i-го рядка (i-го стовпця) визначають вершини компоненти сильної зв'язності орієнтованого графа D, що містить вершину* v_i.

Із тверджень 9.22 — 9.24 випливає алгоритм визначення кількості компонент сильної зв'язності орграфа D, а також матриць суміжності цих компонент.

Алгоритм 9.3. Введемо такі позначення. Нехай D_i — i-та компонента сильної зв'язності, S_i — матриця сильної зв'язності D_i, V_i — множина вершин D_i. Тоді:

Крок 1. $i := 1$, $D_i = D, S_i := S(D)$.

Крок 2. Включаємо в множину вершин v_i вершини, що відповідають одиницям першого рядка матриці S_i. Як $\Delta(D_i)$ беремо підматрицю матриці $\Delta(D)$, утворену перетином рядків і стовпців, які відповідають вершинам із V_i.

Крок 3. Позначимо S' матрицю, що утворюється викреслюванням із S_i рядків і стовпців, які відповідають вершинам із V_i. Якщо в результаті не залишиться жодного рядка та жодного стовпця в S_i, то i — кількість компонент сильної зв'язності, а $\Delta(D_1)$, $\Delta(D_2)$, ..., $\Delta(D_i)$ — матриці компонент сильної зв'язності D_1, D_2, ..., D_i. Інакше $i := i + 1$, $S_{i+1} := S'i$ треба перейти до кроку 2.

Аналогічно розв'язуються задачі для неорієнтованого графа. Очевидно, після змін у позначеннях і термінології алгоритм можна застосувати для визначення кількості компонент зв'язності графа G, а також матриць їх суміжності. Для обґрунтування цього досить скористатися твердженнями 9.22 — 9.24, але сформульованими для неорієнтованого графа G. Більше того, алгоритм залишається справедливим також для довільних псевдографів (орієнтованих та неорієнтованих).

9.23. Задачі пошуку маршрутів у графі

При розв'язуванні широкого кола прикладних задач нерідко виникає необхідність знайти маршрут, що зв'язує задані вершини в графі G. Приклад такої задачі було розглянуто в розд. 5. Наведемо алгоритм розв'язання її в загальному вигляді.

Алгоритм 9.4 (алгоритм Террі). Задача зводиться до пошуку маршруту у зв'язному графі $G = (V, X)$, який з'єднує задані вершини v, $w \in V$, де $v \neq w$.

Нехай $G = (V, E)$ — зв'язний граф. Треба знайти маршрут, що зв'язує задані вершини v, $w \in V$, де $v \neq w$, у графі G.У зв'язному графі G завжди можна знайти такий маршрут, що зв'язує дві задані вершини v та w, якщо, виходячи з вершини v і здійснюючи послідовний перехід від кожної досягнутої вершини до суміжної з нею, керуватися такими правилами:

1) йдучи по довільному ребру, кожний раз відмічати напрямок, в якому воно було пройдене;

2) виходячи з деякої вершини v_1, завжди рухатися тільки по тому ребру, яке не було пройдене або було пройдене у зворотному напрямку;

3) для кожної вершини v_1, відмінної від v, відмічати те ребро, яке першим заходить у v_1, якщо вершина v_1 зустрічається вперше;

4) виходячи з деякої вершини v_1, відмінної від v, по першому ребру, яке заходить у v_1, рухатися лише тоді, коли немає інших можливостей.

Приклад. Використовуючи алгоритм Террі, знайти маршрут, що зв'язує вершини v_1 та v_5 у графі G, зображеному на рис. 92.

Пошук вершини v_5 у G будемо здійснювати так, неначе нічого невідомо про цей граф (нагадаємо, що граф G — це схема лабіринту, де v_5 — вихід із нього, а v_1 — розвилка, з якої починається пошук виходу).

На рис. 93 показано один із можливих варіантів руху по графу G згідно з алгоритмом Террі. Пронумерованими штриховими дугами зображено схему руху по графу G. Знаками помічено перші ребра, які заходять у вершини (помітка робиться ближче до тієї вершини, в яку ребро заходить). Ця схема руху відповідає маршруту $v_1v_2v_1v_3v_4v_3v_5$. Зазначимо, що після того, як із вершини v_1 зайшли у вершину v_3 (див. дугу *3*), внаслідок правила 4 не можна повернутися у v_1, оскільки існують інші можливості, а (v_1, v_3) є першим ребром, що заходить у v_3. Далі, після того, як із вершини v_4 зайшли у вершину v_3 (див. дугу *5*), внаслідок правила 2 не можна рухатися до вершини v_1, і, таким чином, залишається єдина можливість — рухатися до вершини v_5.

Обґрунтуємо алгоритм Террі. Припустимо, що, керуючись цим алгоритмом, зупинимося в деякій вершині u (не досягши вершини w), а всі

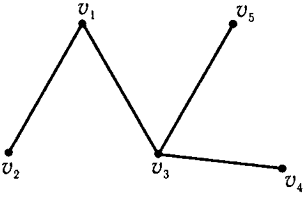


Рис. 92

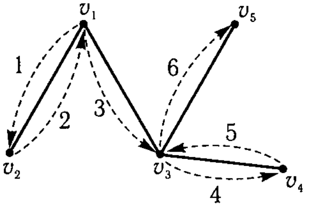


Рис. 93

ребра, інцидентні u, вже пройдено в напрямку з u (тоді внаслідок правила 2 вже не можна вийти з u). Покажемо, що в цьому випадку: а) вершина u збігається з v; б) всі вершини графа G пройдено.

Доведемо спочатку правильність твердження а). Якщо вершина u не збігається з v, то нехай у вершині u ми побували k разів (включаючи останній). Тоді ребра, інцидентні u, були пройдені k разів у напрямку до u, $k-1$ разів у напрямку з u (оскільки кількість заходів в u, за винятком останнього, відповідає кількості виходів із цієї вершини). Таким чином, використовуючи те, що за припущенням були пройдені всі ребра, інцидентні вершині u, в напрямку з u, а також те, що з урахуванням правила 2 по кожному ребру, інцидентному u, маємо $\rho(u)=k-1$, а це суперечить тому, що у напрямку до u були пройдені k різних (див. знову правило 2) ребер; отже, $\rho(u)\geq k$. Одержана суперечність підтверджує, що $u = v$.

Доведемо тепер правильність твердження б). Нехай (за твердженням а))

$$v_1, v_2, ..., v_k, \tag{9.9}$$

де $v_1 = v_k = v$ — послідовність вершин, розташованих у тому самому порядку, в якому ми рухалися, діючи згідно з алгоритмом. Очевидно, (9.9) є маршрутом у графі G. Покажемо, що цей маршрут містить усі вершини графа G. Спочатку доведемо, що кожне ребро, інцидентне будь-якій вершині v_j, де $1\leq j\leq k$, було пройдене по одному разу в обох напрямках. Доведення проведемо індукцією за кількістю вершин j.

Оскільки в замкненому маршруті для кожної вершини, яка міститься в ньому, кількість виходів із неї дорівнює кількості заходів у неї, внаслідок того, що згідно з твердженням а) і правилом 2 всі ребра, інцидентні вершині $v=v_1$, були пройдені по одному разу в напрямку з v (тобто ми $\rho(v)$ разів виходили з v), встановлюємо, що рівно $\rho(v)$ разів ми заходили у v, а оскільки внаслідок правила 2 кожний такий захід у v здійснювався по новому ребру, всі ребра, інцидентні вершині $v_1 = v$, були пройдені по разу в обох напрямках.

Припустимо, що при деякому j, де $2\leq j\leq k$, твердження, яке доводиться, справджується для всіх вершин $v_1, ..., v_{j-1}$. Доведемо його для вершини v_j. Якщо при деякому $i<j$ виконується рівність $v_i = v_j$, то правильність твердження, що доводиться для вершини v_j, випливає з того, що за індуктивним припущенням воно справджується для вершини v_i. Нехай тепер $\forall i\in\{1, 2, .., j-1\}$ $v_i\neq v_j$, тобто вершина v_i зустрілася вперше. Тоді (v_{j-1}, v_j) — перше ребро, яке заходить у вершину v_j, за індуктивним припущенням воно буде пройдене в обох напрямках, що з урахуванням правила 4 можливо лише тоді, коли всі інші ребра, інцидентні v_j, будуть пройдені в напрямку з v_j. Далі, оскільки в замкненому маршруті, як уже зазначалося, для кожної вершини, яка міститься в цьому маршруті, кількість виходів із неї дорівнює кількості заходів в неї, використовуючи правило 2, встановлюємо, що всі ребра, інцидентні v_j, будуть пройдені по разу в обох напрямках.

Отже, кожну вершину в маршруті (9.9) проходимо разом з усіма суміжними їй вершинами, звідки внаслідок зв'язності графа G випливає, що маршрут (9.9) проходить через усі вершини графа G, а це суперечить початковому припущенню, за яким вершина w не була досягнута.

Алгоритм 9.4 і його обґрунтування справджуються також для випадку, коли G — зв'язний псевдограф.

Якщо псевдограф $G = (V, X)$ не є зв'язним, то за допомогою алгоритму 9.4, виходячи з довільної вершини $v \in V$ і позначаючи пройдені вершини й ребра, можна вилучити компоненту зв'язності псевдографа G, яка містить вершину v. Алгоритм завершує свою роботу в той момент, коли в перший раз неможливо буде задовольнити правило 2 (тобто ми прийшли у вершину u, а всі ребра, інцидентні їй, пройдено в напрямку з u; при цьому, як показано при обґрунтуванні алгоритму 9.4, $u = v$).

9.24. Пошук відстані між вершинами графа

Розглянемо деякі властивості мінімальних шляхів (маршрутів). У п. 9.10 наведено означення довжини маршруту та відстані між вершинами графа G. Аналогічно визначаються поняття довжини шляху і відстані між вершинами орієнтованого графа D.

Означення 9.35. *Назвемо образом вершини x в орієнтованому графі D множину кінців дуг, початком яких є вершина x, і позначимо його $D(x)$, а множину початків дуг, кінцем яких є вершина x, назвемо прообразом вершини x і позначимо його $D^{-1}(x)$.*

Нехай $D = (V, E)$ — орієнтований граф з n вершинами $(n \geq 2)$, а v, w — задані вершини з V, де $v \neq w$. Опишемо алгоритм пошуку відстані та відповідного їй мінімального шляху з v у w в орієнтованому графі D.

Алгоритм 9.5. Він включає такі кроки:

Крок 1. Позначаємо вершину v індексом 0, а вершини, що належать образу вершини v, — індексом 1. Множину вершин з індексом k позначаємо $FW_k(v)$. Вважаємо $k = 1$.

Крок 2. Якщо $FW_k(v) = \varnothing$ або виконується $k = n - 1$ і $w \notin FW_k(v)$, то вершина w є незв'язаною з v і робота алгоритму на цьому завершується. В іншому випадку переходимо до кроку 3.

Крок 3. Якщо $w \notin FW_k(v)$, то переходимо до кроку 4. В іншому випадку існує шлях із v у w завдовжки k, причому цей шлях є мінімальним. Послідовність вершин $v, w_1, w_2, ..., w_{k-1}w,$

де
$$w_{k-1} \in FW_{k-1}(v) \cap D^{-1}(w);$$
$$w_{k-2} \in FW_{k-2}(v) \cap D^{-1}(w_{k-1}); \tag{9.10}$$
$$\dots\dots\dots\dots\dots\dots\dots\dots$$
$$w_1 \in FW_1(v) \cap D^{-1}(w_2),$$

і є шуканим мінімальним шляхом із v у w. На цьому робота алгоритму завершується.

Крок 4. Позначаємо індексом $k+1$ всі непозначені вершини, що належать образу множини вершин з індексом k. Множину вершин з індексом $k+1$ позначаємо $FW_{k+1}(v)$. Нехай $k:=k+1$ і переходимо до кроку 2.

Вершини $w_1,\ldots,w_{k-1}$ із (9.10), взагалі кажучи, можуть бути вилучені неоднозначно. Ця неоднозначність відповідає випадкам, коли існує кілька різних мінімальних шляхів з v у w в орграфі D (опишіть цей алгоритм самостійно).

Приклад. Використовуючи алгоритм 9.5, визначити мінімальний шлях із v_1 у v_6 в орієнтованому графі D, заданому матрицею суміжності у вигляді табл. 9.4.

Таблиця 9.4

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	0	0	1	1	0
v_2	1	0	0	1	1	1
v_3	1	1	0	1	1	1
v_4	0	1	1	0	1	0
v_5	1	1	1	1	0	0
v_6	1	1	1	1	1	0

Діючи згідно з алгоритмом 9.5, послідовно знаходимо $FW_1(v_1)=\{v_4,v_5\}$; $FW_2(v_1)=$ $=D(FW_1(v_1))\setminus\{v_1,v_4,v_5\}=\{v_2,v_3\}$; $FW_3(v_1)=D(FW_2(v_1))\setminus\{v_1,v_2,v_3,v_4,v_5\}=\{v_6\}$. Таким чином, $v_6\in FW_3(v_1)$, а отже, (див. крок 3) існує шлях із v_1 у v_6 завдовжки 3, і цей шлях є мінімальним.

Знайдемо тепер мінімальний шлях із v_1 у v_6. Визначимо множину

$$FW_2(v_1)\cap D^{-1}(v_6)=\{v_2,v_3\}\cap\{v_2,v_3\}=\{v_2,v_3\}.$$

Виберемо будь-яку вершину зі знайденої множини, наприклад v_5. Визначимо далі множину

$$FW_1(v_1)\cap D^{-1}(v_3)=\{v_4,v_5\}\cap\{v_4,v_5,v_6\}=\{v_4,v_5\}.$$

Виберемо будь-яку вершину зі знайденої множини, наприклад, v_5. Тоді $v_1v_5v_3v_6$ — шуканий мінімальний шлях із v_1 у v_6 в орієнтованому графі D, а відстань між v_1 і v_6 дорівнює 3.

9.25. Мінімальні шляхи (маршрути) у зважених орієнтованих (неорієнтованих) графах

Означення 9.36. *Орієнтований граф $D=(V,E)$ є зваженим, якщо на множині дуг E визначено вагову функцію $\ell:E\to R$.*

Таким чином, у зваженому орієнтованому графі D кожній дузі $e\in E$ поставлено у відповідність вагу ℓ. Значення $\ell(e)$ будемо називати вагою дуги e. Для будь-якого шляху M зваженого орієнтованого графа D позначимо через $\ell(M)$ суму ваг дуг, що входять в M; при цьому кожна дуга враховується стільки разів, скільки вона входить у шлях. Величину $\ell(M)$ будемо називати вагою шляху M у зваженому орієнтованому графі D.

Якщо ваги дуг дорівнюють 1, то $\ell(M)$ виражає введену раніше довжину шляху M у незваженому орієнтованому графі. Отже, будь-який незважений орієнтований граф можна вважати зваженим із вагами дуг, що дорівнюють 1.

Означення 9.37. *Шлях у зваженому орієнтованому графі D з вершини v у вершину w, де $v \neq w$, називається мінімальним, якщо він має найменшу вагу серед усіх шляхів з v у w.*

Якщо у зваженому графі є замкнені шляхи від'ємної ваги, то для заданих вершин v і w орієнтованого графа D, де $v \neq w$, мінімального шляху з v у w не може бути.

Наведемо деякі властивості мінімальних шляхів (маршрутів) у зваженому орієнтованому графі $D = (V, E)$ (графі $G = (V, E)$):

1) якщо $\forall e \in E\ \ell(e) > 0$, то будь-який мінімальний шлях (маршрут) є простим ланцюгом;

2) якщо $v_1, v_2, ..., v_k$ — мінімальний шлях (маршрут), то для будь-яких номерів i, j таких, що $1 \leq i \leq j \leq k$, шлях (маршрут) $v_i, v_{i+1}, ..., v_j$ також є мінімальним;

3) якщо $v, ..., u, w$ — мінімальний шлях (маршрут) серед шляхів з v у w (серед маршрутів, які з'єднують v, w), що містять не більш як $k + 1$ дуг (ребер), то $v...u$ — мінімальний шлях (маршрут) серед шляхів з v у u (серед маршрутів, які з'єднують v, u), що з'єднують не більш як k дуг (ребер).

При розв'язуванні деяких практичних задач виникає необхідність пошуку максимальних шляхів у зваженому орієнтованому графі. Така задача легко зводиться до задачі пошуку мінімальних шляхів, що досліджується нижче.

Нехай $D = (V, E)$ — зважений граф; $V = (v_1, ..., v_n)$, $n \geq 2$. Введемо величини λ_i^k, де $i = 1, ..., n$, $k = 1, 2, ...$. Для кожних фіксованих i та k величина λ_i^k дорівнює вазі мінімального шляху серед шляхів із v_1 у v_i, які містять не більш як k дуг; якщо ж таких шляхів немає, то $\lambda_i^k = \infty$. Крім того, якщо довільну вершину $v \in V$ вважати шляхом із v у v нульової ваги, то величини λ_i^k можна ввести також і для $k = 0$; при цьому

$$\lambda_1^0 = 0, \lambda_i^0 = \infty, \quad i = 2, ..., n. \tag{9.11}$$

Уведемо також у розгляд квадратну матрицю $C(D) = \|c_{ij}\|$ порядку n з елементами

$$c_{ij} = \begin{cases} \ell(v_i, v_j), \text{ якщо } (v_i, v_j) \in E; \\ \infty, \text{ якщо } (v_i, v_j) \notin E, \end{cases}$$

яку будемо називати матрицею ваг дуг зваженого орієнтованого графа D.

Твердження 9.25. *Для обчислення λ_i^k справджуються рекурентні формули:*

$$\lambda_i^{k+1} = \min_{1 \leq j \leq n} \left\{ \lambda_j^k + c_{ji} \right\} \text{ при } i = 2, ..., n;\ k \geq 0; \tag{9.12}$$

$$\lambda_i^{k+1} = \min\left\{0, \min_{1 \le j \le n}\left\{\lambda_j^k + c_{ji}\right\}\right\} \text{ при } i = 1;\ k \ge 0. \qquad (9.13)$$

Доведемо правильність цього твердження. Нехай $i \in \{2, 3, ..., n\}$, $k \ge 0$. Покажемо спочатку, що $\lambda_i^{k+1} \ge \min\limits_{1 \le j \le n}\{\lambda_j^k + c_{ij}\}$. При $\lambda_i^{k+1} = \infty$ ця нерівність, очевидно, виконується. Нехай $\lambda_i^{k+1} < \infty$ і $M = v_1, ..., v_{i-1}, v_i$ — шлях із v_1 у v_i, який містить не більш як k + 1 дуг, такий, що $\ell(M) = \lambda_i^{k+1}$. Тоді (за властивість 3) $M_1 = v_1, v_2, ..., v_{i-1}$ — мінімальний шлях серед шляхів з v_1 у v_{i-1}, який містить не більш як k дуг. При цьому $\lambda_i^{k+1} = \ell(M) = \ell(M_1) + {}$ $+ \ell(v_{i-1}, v_i) = \lambda_{i-1}^k + c_{i-1,i} \ge \min\limits_{1 \le j \le n}\left\{\lambda_j^k + c_{ji}\right\}$, тобто $\lambda_i^{k+1} \ge \min\limits_{1 \le j \le n}\left\{\lambda_j^k + c_{ji}\right\}$.

Покажемо тепер, що $\lambda_i^{k+1} \ge \min\limits_{1 \le j \le n}\left\{\lambda_j^k + c_{ji}\right\}$. При $\min\limits_{1 \le j \le n}\left\{\lambda_j^k + c_{ji}\right\} = \infty$ ця нерівність виконується. Нехай $\min\limits_{1 \le j \le n}\left\{\lambda_j^k + c_{ji}\right\} < \infty$, а значить, $\lambda_j^k < \infty$ і $c_{ji} <$ $< \infty\ \forall k,\ j$, і $i_1 \in \{1, 2, ..., n\}$ — номер такий, що $\lambda_{i_1}^k + c_{i_1 i} = \min\limits_{1 \le j \le n}\left\{\lambda_j^k + c_{ji}\right\}$. Оскільки $\lambda_{i_1}^k < \infty$ та $c_{i_1 i} < \infty$, ребро $(v_{i_1}, v_i) \in X$ і $c_{i_1 i} = \ell(v_{i_1}, v_i)$, тобто існує шлях M_1, який містить не більш як k дуг, такий, що $\ell(M_1) = \lambda_{i_1}^k$. Однак тоді для шляху $M = M_1 \circ (v_{i_1}, v_i)$, який містить не більш як k + 1 дуг, виконується рівність $\lambda_i^{k+1} \le \ell(M) = \lambda_{i_1}^k + c_{i_1 i}$, тобто маємо $\lambda_i^{k+1} \le \min\limits_{1 \le j \le n}\left\{\lambda_j^k + c_{ji}\right\}$. Таким чином, правильність формули (9.12) доведено.

Формула (9.13) доводиться аналогічно.

Приймемо без доведення такі твердження.

Твердження 9.26. *Якщо замкнений шлях має від'ємну вагу, то з нього можна вилучити простий замкнений шлях від'ємної ваги.*

Твердження 9.27. *Якщо у зваженому орієнтованому графі відсутні прості замкнені шляхи від'ємної ваги, то в ньому немає замкнених шляхів від'ємної ваги.*

Твердження 9.28. *Якщо у зваженому графі відсутні прості замкнені шляхи від'ємної ваги, то з усякого незамкненого шляху можна вилучити простий ланцюг із тими самими початковою та кінцевою вершинами.*

Будемо припускати, що в D відсутні прості цикли (контури) від'ємної ваги.

Зауваження. Величини λ_i^k можна визначити також для випадку, коли для деяких $v \in E$ виконується рівність $\ell(e) = \infty$. Усі твердження й алгоритми при цьому не змінюються.

Опишемо алгоритм знаходження таблиці значень величин λ_i^k (будемо записувати її у вигляді матриці, де i — номер рядка, k + 1 — номер стовпця). Справді, використовуючи рекурентні співвідношення, можна послідовно визначити набір величин $\lambda_i^k, ..., \lambda_i^k$ ((k + 1)-й стовпець матриці), починаючи з $k = 0$, а потім рухатися крок за кроком, збільшуючи k до будь-якого потрібного значення.

Якщо при деякому k_0, де $0 \le k_0 \le n-2$, виконується рівність $\lambda_i^{k_0} = \lambda_i^{k_0+1}$, $i = 1, 2,..., n$, то $\forall k \ge k_0$, $\lambda_i^k = \lambda_i^{k_0}$, $i = 1, 2,..., n$, і, зокрема, $\lambda_i^{n-1} = \lambda_i^{k_0}$, $i = 1, 2,..., n$, тобто в цьому разі значення λ_i^k при $k > k_0$ не несуть ніякої додаткової інформації; тоді доцільно обірвати процес послідовного визначення наборів величин $\lambda_i^k, ..., \lambda_n^k$ на значенні $k = k_0$.

Наведемо алгоритм, що дає змогу за таблицею величин λ_i^k, $i = 1, 2,..., n$, $k = 0, 1, ..., n-1$, визначити мінімальний шлях у зваженому орієнтованому графі D з v_1 у будь-яку досяжну вершину.

Алгоритм 9.6 (алгоритм Беллмана). Він включає такі кроки.

Крок 1. Якщо $\lambda_{i_1}^{n-1} = \infty$, то вершина v_i^k є недосяжною з v_i (припускаємо, що всі значення $\ell(x)$, $x \in X$, — скінченні). В цьому випадку робота алгоритму завершується.

Крок 2. Нехай $\lambda_{i_1}^{n-1} < \infty$. Тоді число $\lambda_{i_1}^{n-1}$ виражає вагу будь-якого мінімального шляху з v_1 у v_{i_1} у зваженому орієнтованому графі D. Визначимо мінімальне число $k_1 \ge 1$, при якому виконується рівність $\lambda_{i_1}^{k_1} = \lambda_{i_1}^{n-1}$. З означення чисел λ_i^k випливає, що k_1 — мінімальна кількість дуг у шляху серед усіх мінімальних шляхів із v_1 у λ_i^k у зваженому орієнтованому графі D.

Крок 3. Послідовно визначаємо номери $i_2, ..., i_{k_1+1}$ такі, що

$$\begin{gathered}
\lambda_{i_2}^{k_1-1} + c_{i_2,i_1} = \lambda_{i_1}^{k_1}; \\
\lambda_{i_3}^{k_1-2} + c_{i_3,i_2} = \lambda_{i_2}^{k_1-1}; \\
\dots\dots\dots\dots\dots\dots\dots \\
\lambda_{i_{k_1+1}}^{0} + c_{i_{k_1+1},i_{k_1}} = \lambda_{i_{k_1}}^{1}.
\end{gathered} \tag{9.14}$$

Із (9.14) з урахуванням того, що $\lambda_{i_1}^{k_1} = \lambda_{i_1}^{n-1}$, маємо $c_{i_2,i_1} < \infty, ..., c_{i_{k_1+1},i_{k_1}} < \infty$, $< \infty$, $\lambda_{i_{k_1+1}}^0 < \infty$, звідки

$$\begin{gathered}
(v_{i_2}, v_{i_1}), ..., (v_{i_{k_1+1}}, v_{i_{k_1}}) \in E,\, ' \; \ell(v_{i_2}, v_{i_1}) = c_{i_2,i_1}, ..., \\
\ell(v_{i_{k_1+1}}, v_{i_{k_1}}) = c_{i_{k_1+1},i_{k_1}};\ \lambda_{i_{k_1+1}}^0 = 0,\ \ i_{k_1+1} = 1,,\ \ v_{i_{k_1+1}} = v_1.
\end{gathered} \tag{9.15}$$

Додаючи рівності (9.14) і враховуючи (9.15), знаходимо $1(v_1, v_{k_1}, ..., v_{i_2}, v_{i_1}) = \lambda_{i_1}^{k_1}$, тобто $v_1, v_{k_1}, ..., v_{i_2}, v_{i_1}$ — шуканий мінімальний шлях із v_1 у v_{i_1} у зваженому орієнтованому графі D. Зауважимо, що в цьому шляху є k_1 дуг. Отже, визначено шлях із мінімальною кількістю дуг серед усіх мінімальних шляхів із v_1 у v_{i_1} у зваженому орієнтованому графі D.

Номери $i_2, i_3, ..., i_{k_1}$, що задовольняють (9.14), взагалі кажучи, можуть бути вилучені неоднозначно. Ця неоднозначність відповідає випадкам, коли існує кілька різних шляхів із v_1 у v_{i_1} у зваженому орієнтованому графі D.

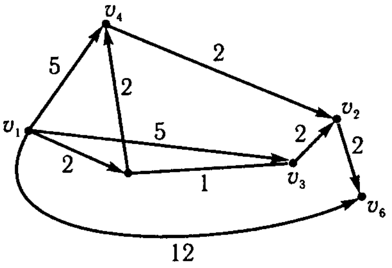


Рис. 94

Алгоритм 9.6 можна модифікувати для того, щоб визначити мінімальний шлях із v_1 у задану вершину v_{i_1} серед шляхів із v_1 у v_{i_1}, які містять не більш як k_0 дуг, де k_0 — задане число, $k_0 \geq 1$. Для цього в алгоритмі 9.6 замість $\lambda_{i_1}^{n-1}$ потрібно покласти $\lambda_{i_1}^{k_0}$.

Алгоритм 9.6 і його модифікація після відповідної зміни в термінології та позначеннях застосовні також до неорієнтованого графа G. При цьому умова відсутності простих контурів від'ємної ваги замінюється умовою відсутності ребер від'ємної ваги.

Приклади: 1. Визначити мінімальний шлях із v_1 у v_6 у зваженому орієнтованому графі D, зображеному на рис. 94, біля кожної дуги якого вказано її вагу.

Складемо матрицю $C(D)$ ваг дуг зваженого графа D у вигляді табл. 9.5, *а*. В табл. 9.5, *б* є шість стовпців $\left(\lambda_1^k, \lambda_2^k, ..., \lambda_6^k\right)^T$, де $k = 0, 1, 2, 3, 4, 5$, які будемо визначати, користуючись рекурентним співвідношенням. Величина $\lambda_6^5 = 7$ виражає вагу мінімального шляху з v_1 у v_6 у зваженому орієнтованому графі D. Знайдемо мінімальне число $k_1 \geq 1$, при якому виконується рівність $\lambda_6^{k_1} = \lambda_6^5$. Із табл. 9.5, *б* маємо $k_1 = 4$. Таким чином, мінімальна кількість дуг серед усіх мінімальних шляхів із v_1 у v_6 у зваженому орієнтованому графі D дорівнює 4.

Визначимо тепер послідовність номерів i_1, i_2, i_3, i_4, i_5, де $i_1 = 6$, які задовольняють (9.14) (для цього використовуємо формулу (9.12)). Із табл. 9.5, *б* знаходимо, що як таку послідовність треба взяти номери 6, 2, 3, 5, 1, оскільки

$$\lambda_2^3 + c_{26} = 5 + 2 = 7 = \lambda_6^4;$$

$$\lambda_3^2 + c_{32} = 2 + 3 = 5 = \lambda_2^3;$$

$$\lambda_5^1 + c_{53} = 2 + 1 = 3 = \lambda_3^2;$$

$$\lambda_1^0 + c_{15} = 0 + 2 = 2 = \lambda_5^1.$$

Тоді $v_1v_5v_3v_2v_6$ — шуканий мінімальний шлях із v_1 у v_6 у зваженому орієнтованому графі D, причому він містить мінімальну кількість дуг серед усіх можливих мінімальних шляхів із v_1 у v_6.

Таблиця 9.5

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	∞	∞	5	5	$\boxed{2}$	12
v_2	∞	∞	∞	∞	∞	$\boxed{2}$
v_3	∞	2	∞	∞	∞	∞
v_4	∞	$\boxed{2}$	∞	∞	∞	∞
v_5	∞	∞	$\boxed{1}$	2	∞	∞
v_6	∞	∞	∞	∞	∞	∞

а

λ_i^0	λ_i^1	λ_i^2	λ_i^3	λ_i^4	λ_i^5
⓪	0	0	0	0	0
∞	∞	7	⑤	5	5
∞	5	③	3	3	3
∞	5	4	4	4	4
∞	②	2	2	2	2
∞	12	12	9	⑦	7

б

2. Визначити шлях із v_1 у v_6 у зваженому орієнтованому графі D (див. попередній приклад) мінімальної ваги серед шляхів із v_1 у v_6, які містять не більш як три дуги.

Із табл. 9.5, *б* знаходимо $\lambda_6^3 = 9$, тобто шуканий шлях має вагу, що дорівнює 9. Визначаємо тепер мінімальне число k_1, при якому $\lambda_6^{k_1} = \lambda_6^3$. За табл. 9.5, *б* маємо $k_1 = 3$. Далі встановлюємо послідовність номерів i_1, i_2, i_3, i_4, де $i_1 = 6$, які задовольняють (9.15). Із табл. 9.5, *б* випливає, що як таку послідовність можна взяти номери 6, 2, 4, 1. Тоді $v_1v_4v_2v_6$ — шуканий шлях.

Наведемо твердження, яке дає необхідну й достатню умову (яка легко перевіряється) того, що в D відсутні прості цикли від'ємної ваги. При цьому будемо розглядати випадок, коли з вершини v_1 досягаються всі інші вершини орієнтованого графа D. Цього завжди можна домогтися вилученням вершин, незв'язаних із v_1.

Твердження 9.29. *Нехай в орієнтованому графі* $D = (V, E)$, *де* $V = \{v_1, ..., v_n\}$, *будь-яку вершину* v_i, $i \in \{2, ..., n\}$, *зв'язано шляхом із* v_1. *Тоді для того щоб у* D *були відсутні прості цикли від'ємної ваги, необхідно й достатньо, щоб виконувалася умова*

$$\lambda_i^{n-1} = \lambda_i^n, i = 1, ..., n.$$

9.26. Гамільтонові ланцюги та цикли у зважених графах

Для означеності будемо розглядати орієнтовані графи (наші міркування справедливі також для графів). Нерідко при розв'язуванні прикладних задач необхідно знайти не мінімальний шлях із v у w в орієнтованому графі D, де v, w — задані вершини, а шлях (або множину шляхів), який має деяку властивість $\mathcal{L}$, де $\mathcal{L}$ — одномісний предикат, означений на множині $\{M_i\}$ шляхів в орієнтованому графі D. Така властивість називається $\mathcal{L}$-властивістю, якщо з того, що шлях $M = M_1 \circ M_2$, де $M_1, M_2 \in \{M_i\}$, має $\mathcal{L}$-властивість, випливає, що шляхи M_1, M_2 також мають цю властивість.

Приклад. Наведемо деякі $\mathcal{L}$-властивості шляхів в орієнтованому графі:

- не проходити через задану вершину (або через задану множину вершин);
- не проходити через задану дугу (або через задану множину дуг);
- бути простим ланцюгом;
- бути простим ланцюгом або простим циклом;
- не проходити через кожну вершину більш як k разів (де k — задане число, $k \geq 1$).

Уведемо для будь-якого цілого $k \geq 1$ матрицю $\mathcal{L}^k(D) = \left\| \ell_{ij}^k \right\|$ вимірністю $n \times n$ таку, що ℓ_{ij}^k — множина шляхів завдовжки k з v_i у v_j, які мають $\mathcal{L}$-властивість. Нехай $M_{\mathcal{L}}$ — множина шляхів у D, що мають $\mathcal{L}$-властивість. Введемо операцію $\mathcal{L}$-композиції $\overset{\mathcal{L}}{\circ}$ на $M_{\mathcal{L}} \cup \varnothing$.

Нехай $M_1 = u_1, u_2, ..., u_k$, $M_1 \in M_{\mathcal{L}}$, $M_2 = w_1, w_2, ..., w_\ell$, $M_2 \in M_{\mathcal{L}}$.
Тоді

$$M_1 \overset{\mathcal{L}}{\circ} M_2 = \begin{cases} M_1 \circ M_2, \text{ якщо } u_k = w_1, \text{ та } M_1 \circ M_2 \text{ має } \mathcal{L}\text{ - властивість;} \\ 0 - \text{в іншому випадку.} \end{cases}$$

Приймемо без доведення таке твердження.

Твердження 9.30. *Для буль-якого $k \geq 1$справджується рівність*

$$\mathcal{L}^k(D) = \mathcal{L}^1(D) \overset{\mathcal{L}}{\circ} \mathcal{L}^1(D) \overset{\mathcal{L}}{\circ} \ldots \overset{\mathcal{L}}{\circ} \mathcal{L}^1(D),$$

тобто матриця $\mathcal{L}^k(D)$ дорівнює $\mathcal{L}$-композиції k матриць $\mathcal{L}^1(D)$.

Нехай G — псевдограф. Нагадаємо, що ланцюг (цикл) у G називається гамільтоновим, якщо він проходить через кожну вершину псевдографа G рівно один раз.

Із поняттям гамільтонових циклів тісно пов'язана задача комівояжера: у зваженому графі G визначити гамільтонів цикл мінімальної ваги (іншими словами, комерсант повинен здійснити поїздку по містах і повернутися назад, побувавши в кожному місті рівно один раз; при цьому вартість поїздки має бути мінімальною).

На перший погляд, як уже підкреслювалося, поняття гамільтонова циклу подібне до поняття ейлерова цикла. А тим часом графи, показані на рис. 95, де стовпці відповідають випадкам існування (стовпець 1) та неіснування (стовпець 2) гамільтонових циклів, а рядки — випадкам існування (рядок 1) і неіснування (рядок 2) ейлерових циклів, показують незалежність цих понять.

Гамільтонові ланцюги та цикли належать до спеціальних маршрутів у графах. Очевидно, властивість маршрутів «пройти через кожну вершину не більш як один раз» є $\mathcal{L}$-властивістю, а отже, всі гамільтонові ланцюги і цикли псевдографа G можна дістати, застосовуючи до G метод $\mathcal{L}$-композиції.

Нехай G є n-вершинним псевдографом. Використовуючи той очевидний факт, що довжина будь-якого гамільтонова ланцюга дорівнює $n - 1$, а довжина будь-якого гамільтонова циклу — n, установлюємо, що всі гамільтонові ланцюги будуть пераховані в непорожніх елементах матриці $\mathcal{L}^{n-1}(G)$, за винятком елементів на головній діагоналі, а всі гамільтонові цикли — в кожному діагональному елементі матриці $\mathcal{L}^n(G)$. Проте метод $\mathcal{L}$-композиції в цьому випадку практично застосовний лише при досить малих n, і тому цікавою є розробка більш економічних методів.

Зауважимо, що (як у випадку з ейлеровими ланцюгами та циклами) було б корисно мати порівняно прості необхідні й достатні умови існування гамільтонових ланцюгів і циклів.

Питання існування та знаходження гамільтонових ланцюгів і циклів у псевдографах, очевидно, зводяться до аналогічних питань стосовно графів; тому далі йтиметься тільки про графи.

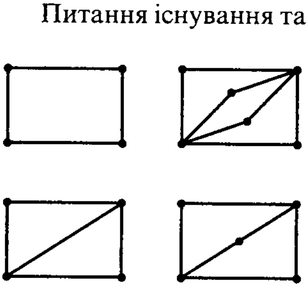


Рис. 95

Розглянемо клас графів, в яких явно існують гамільтонові ланцюги та цикли. Очевидно, в повному графі завжди існують гамільтонів цикл, а також гамільтонові ланцюги, що з'єднують дві довільні вершини цього графа. Таким чином, найпростішою достатньою умовою існування гамільтонових ланцюгів і циклів у графі є його повнота. Наведемо також найпростіші необхідні умови. Очевидною необхідною умовою існування гамільтонових ланцюгів та циклів у графі G є його зв'язність.

Твердження 9.31. *Якщо граф G має гамільтонів цикл, то в ньому відсутні точки зчленування.*

Нехай у G існує точка зчленування v_1. Доведемо, що G не може мати гамільтонова циклу. Припустимо супротивне, тобто що G має такий цикл $M = v_1 v_2, ..., v_n, v_1$, де $n = n(G)$. Оскільки v_1 — точка зчленування, після вилучення v_1 із G утвориться граф G' із компонентами зв'язності $G_1, ..., G_p$, де $p \geq 2$. Нехай компоненти зв'язності пронумеровано так, що $v_2 \in G_1 = (V_1, E_1)$. Зазначимо, що за означенням гамільтонова циклу виконуються умови $v_2 \neq v_1$, $v_3 \neq v_1$, звідки $(v_2, v_3) \in E_1$, а отже, $v_3 \in V_1$. Аналогічно $v_4, ..., v_n \in V_1$, а це суперечить тому, що $p \geq 2$.

Методи вилучення гамільтонових циклів є більш трудомісткими порівняно з методами вилучення ейлерових циклів. Нехай $G = (V, E)$ — звичайний граф, а $V = \{v_1, v_2, ..., v_n\}$. Будемо перебирати всілякі перестановки $v_{i_1}, v_{i_2}, ..., v_{i_m}$ вершин графа G з метою перевірки, чи є $v_{i_1}, v_{i_2}, ..., v_{i_m}$ маршрутом у G. Якщо так, то $v_{i_1}, v_{i_2}, ..., v_{i_m}, v_{i_1}$ — гамільтонів цикл, а якщо ні, переходимо до наступної перестановки.

У результаті будуть вилучені всі гамільтонові цикли в графі G. При цьому доведеться перебрати $n!$ перестановок.

КОНТРОЛЬНІ ЗАДАЧІ ТА ВПРАВИ

1. Показати, що в будь-якому графі кількість вершин непарних степенів — парна.
2. Показати, що з усякого замкненого маршруту можна вилучити простий ланцюг.
3. Показати, що ребро, яке входить у цикл графа, входить також у деякий його простий цикл.
4. Показати, що будь-яка вершина, яка входить у цикл, не є висячою.
5. Довести, що у зв'язному графі, який містить принаймні дві вершини, знайдеться вершина, яка не є точкою зчленування.
6. Довести, що якщо в орієнтованому графі D відсутні вершини з нульовим степенем виходу (входу), то в D існує простий цикл.
7. Довести, що вилучення з орієнтованого графа вершини v із $\delta^+(v) \leq 1$ (або $\delta^-(v) \leq 1$) приводить до орієнтованого графа, цикли якого збігаються із циклами початкового орієнтованого графа.
8. Визначити, чи мають цикли орієнтовані графи з матрицями суміжності:

$$\text{а)} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}; \text{б)} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}; \text{в)} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}; \text{г)} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix};$$

$$\text{д)} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

9. Визначити матриці зв'язності та сильної зв'язності для орієнтованих графів із матрицями суміжності з попередньої задачі.
10. Нехай орієнтований граф D задано матрицею суміжності. Визначити матрицю сильної зв'язності $S(D)$. Використовуючи алгоритм 9.1, знайти кількість

компонент сильної зв'язності орієнтованого графа D і визначити матриці їх суміжності. Побудувати зображення орієнтованого графа D і його компоненту сильної зв'язності. Розглянути випадки:

$$\text{а) }\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix};\text{ б) }\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix};\text{ в) }\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

11. Використовуючи алгоритм Террі, визначити замкнений маршрут у графі, зображеному на рис. 94, який проходить рівно два рази (по одному разу в кожному напрямку) через кожне ребро графа.

12. Довести, що в сильно зв'язному орієнтованому графі із симетричною матрицею суміжності існує цикл, який проходить по одному разу через кожну дугу орієнтованого графа.

13. Знайти мінімальний шлях із v_1 у v_7 в орієнтованих графах, заданих матрицями суміжності:

$$\text{а) }\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix};\text{ б) }\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix};$$

$$\text{в) }\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

14. Визначити мінімальний шлях із v_1 у v_7 у зважених орієнтованих графах із заданими матрицями ваг:

$$\text{а) }\begin{bmatrix} \infty & \infty & 5 & 4 & 2 & 2 & 9 \\ \infty & \infty & 1 & 1 & \infty & 1 & 1 \\ 2 & \infty & \infty & 1 & 1 & \infty & 3 \\ \infty & 2 & 1 & \infty & 1 & \infty & \infty \\ \infty & \infty & 2 & 2 & \infty & 1 & 6 \\ 1 & 5 & \infty & 1 & 1 & \infty & \infty \\ 2 & \infty & 1 & \infty & 1 & 2 & \infty \end{bmatrix};\text{ б) }\begin{bmatrix} \infty & 4 & \infty & \infty & 5 & 4 & \infty \\ 9 & \infty & 2 & 1 & \infty & \infty & \infty \\ 4 & 4 & \infty & \infty & \infty & \infty & 3 \\ \infty & 3 & 1 & \infty & 1 & \infty & \infty \\ \infty & \infty & 2 & \infty & \infty & 4 & 5 \\ \infty & 3 & \infty & 2 & 2 & \infty & \infty \\ \infty & \infty & 2 & \infty & \infty & 2 & \infty \end{bmatrix};$$

$$\text{в) }\begin{bmatrix} \infty & \infty & 9 & \infty & \infty & 2 & 12 \\ 1 & \infty & \infty & \infty & 1 & 2 & 4 \\ 2 & 1 & \infty & \infty & 1 & \infty & 2 \\ \infty & 4 & 1 & \infty & \infty & 1 & \infty \\ 1 & 2 & \infty & 2 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 4 & \infty & 5 \\ \infty & 2 & 1 & \infty & 1 & 2 & \infty \end{bmatrix}.$$

15. Визначити шлях із v_1 у v_7 мінімальної ваги в кожному зваженому орієнтованому графі (див. попередню задачу) серед шляхів із v_1 у v_7, які містять не більш як k дуг, де: а) $k = 2$; б) $k = 3$; в) $k = 4$.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке граф, ребро, дуга, порядок графа?
2. Що таке орієнтований граф, мішаний граф, мультиграф, псевдограф?
3. Що таке локальні степені вершин орієнтованого графа?
4. Що таке частина графа, суграф та підграф?
5. Що таке маршрут, ланцюг і цикл графа?
6. Як формулюються еквівалентні означення дерева графа?
7. Що таке цикломатичне число графа?
8. Що таке хроматичне число графа?
9. Що таке ейлерів граф?
10. Що таке гамільтонів граф?
11. Що таке планарний граф?
12. Як формулюється проблема чотирьох фарб?
13. Як формулюється екстремальна задача в теорії графів?
14. Що таке компонента зв'язності?
15. Як формулюється задача пошуку маршруту в графі?
16. Як формулюється задача пошуку маршруту з мінімальною кількістю ребер?
17. Як формулюється задача пошуку мінімального шляху у зваженому графі?
18. Що таке кістякове дерево графа?

ПЕРЕЛІК ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

1. Задання графа за допомогою матриці інцидентності, матриці суміжності та списку ребер.
2. Знаходження мінімальних кістякових дерев зважених графів.
3. Визначення ейлеровості й гамільтоновості графів.
4. Визначення планарності графів.
5. Вилучення компонент зв'язності.
6. Розв'язання задач пошуку маршруту в графі.
7. Розв'язання задач пошуку маршрутів із мінімальною кількістю ребер.
8. Знаходження мінімального шляху в зваженому графі.
9. Знаходження мінімальних кістякових дерев у зважених графах.

СПИСОК ЛІТЕРАТУРИ

1. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера. — М.: Энергоатомиздат, 1988. — 480 с.
2. *Нефедов В. Н., Осипова В. А.* Курс дискретной математики. — М.: Изд-во МАИ, 1992. — 264 с.
3. *Берж К.* Теорія графов и ее применения. — М.: Изд-во иностр. лит, 1962.
4. *Оре О.* Теория графов. — М.: Наука, 1968. — 352 с.

Навчальне видання

Бардачов Юрій Миколайович
Соколова Надія Андріївна
Ходаков Віктор Єгорович

Дискретна Математика

Оправа і титул художника *А. М. Назаренка*
Художній редактор *Г. С. Муратова*
Технічний редактор *А. І. Омоховська*
Коректори: *Л. М. Тимченко, Л. О. Зеленько*
Комп'ютерна верстка *А. А. Коркішко*

Свідоцтво про внесення до Держ. реєстру від 04.12.2000 серія ДК № 268

Підп. до друку 12.03.2002. Формат 60 × 84$^{1}/_{16}$. Папір офс. № 1. Гарнітура Times New Roman. Офс. друк. Ум. друк. арк. 16,74. Обл.-вид. арк. 20,00. Тираж 5000 пр. Вид. № 10358. Зам. № 2-60

Видавництво «Вища школа», 01054, Київ-54, вул. Гоголівська, 7

Надруковано з плівок, виготовлених у видавництві «Вища школа»,
у ВАТ «Білоцерківська книжкова фабрика», 09117, Біла Церква,
вул. Л. Курбаса, 4