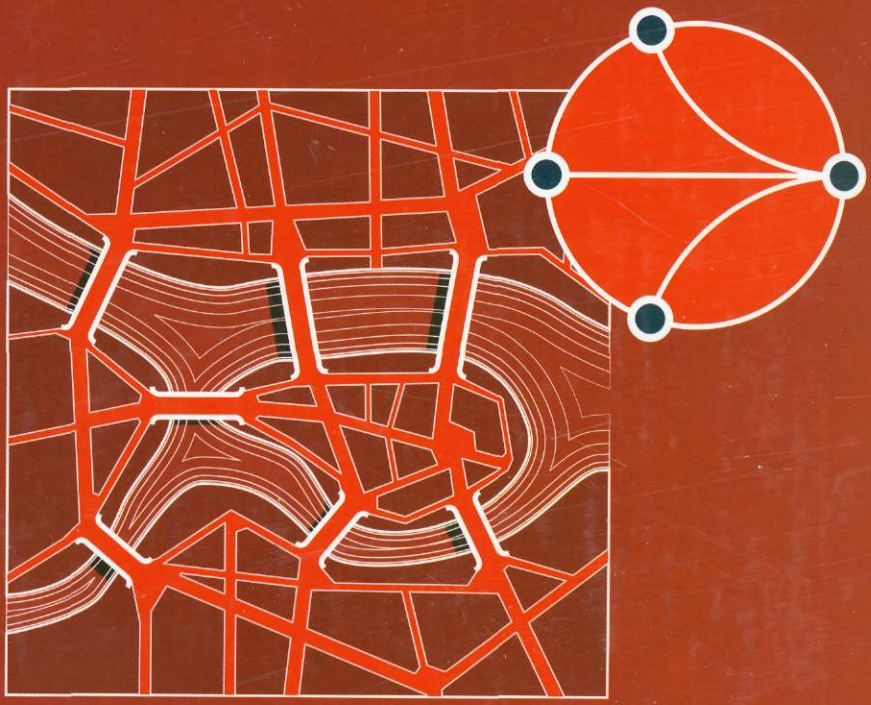
М. Ф. Бондаренко
Н. В. Білоус
А. Г. Руткас

комп'ютерна ДИСКРЕТНА МАТЕМАТИКА



М. Ф. Бондаренко
Н. В. Білоус
А. Г. Руткас

комп'ютерна ДИСКРЕТНА МАТЕМАТИКА

Рекомендовано
Міністерством освіти і науки України
як підручник для студентів
вищих навчальних закладів,
які навчаються за напрямом
«Комп'ютерні науки»

Харків
«Компанія СМІТ»
2004

УДК 519.1
Б-81

Рекомендовано Міністерством освіти і науки України
як підручник для студентів вищих навчальних закладів,
які навчаються за напрямком «Комп'ютерні науки»
(Лист №14/18.2-1530 від 15.07.2002 р.)
Видано за рахунок державних коштів. Продаж заборонено

Рецензенти:
Доктор фізико-математичних наук, професор, завідувач кафедри
«Комп'ютерна алгебра та дискретна математика»,
Одеський національний університет
Варбанець П. Д.

Доктор технічних наук, професор, Національний аерокосмічний
університет ім. Н. Е. Жуковського «ХАІ», м. Харків
Сироджа І. Б.

Б-81 ***Комп'ютерна дискретна математика***: Підручник/
М. Ф. Бондаренко, Н. В. Білоус, А. Г. Руткас.— Харків:
«Компанія СМІТ», 2004.— 480 с.

ISBN 966-8530-20-9

В підручнику викладено основні розділи дискретної математики — теорія множин, теорія відношень, математична логіка, алгебраїчні структури, автомати, алгоритми, формальні мови та граматики, теорія графів і комбінаторика.

Теоретичний матеріал проілюстровано прикладами з різних областей знань. Наведено велику кількість вправ і задач для набуття практичного досвіду.

Підручник призначено для студентів різних спеціальностей, які вивчають дискретну математику, аспірантів і спеціалістів, які використовують відповідні математичні і комп'ютерні методи.

УДК 519.1

ISBN 966-8530-20-9

ВСТУП

Основні способи подання інформації є дискретними: це слова і конструкції мов і граматик — природних і формалізованих; табличні масиви реальних даних у технічних системах та науково-природних спостереженнях; дані господарської, соціальної, демографічної, історичної статистики тощо.

Для кількісного аналізу та обчислювальних перетворень неперервних процесів доводиться їх «дискретизувати». Зрозуміло, що математичні методи обробки, аналізу та перетворення дискретної інформації необхідні у всіх галузях наукової, господарської та соціальної сферах. Зазвичай ці методи викладаються на курсах дискретної математики; іноді вживається термін «скінченна математика», або навіть «конкретна математика».

Часто для аналізу реальних систем з неперервними конструктивними елементами будуються моделі скінченної або дискретної математики. Наприклад, класична транспортна або інформаційна мережа трактується як граф із заданими пропускними здатностями або вагами гілок, а геометрична форма гілки між двома пунктами-вузлами мережі не відіграє ролі. Більш того, «неперервне» будування реальної гілки також не працює у мережній моделі: важливо, що між двома вузлами a, b мережі або немає гілки, або є гілка із заданим обмеженням $c(a, b)$ об'єму переносу речовини або інформації. В моделі досить задати числа $c(a, b)$ для кожної пари вузлів a, b. Якщо гілки немає, то $c(a, b) = 0$. Така числова модель відображення мережі ідеально пристосована для запису, збереження та перетворень у комп'ютері.

Підручник розрахований, у першу чергу, на студентів, а також читачів, які бажають вивчати математичні методи для використання їх у природних науках і комп'ютерних технологіях. У ньому відображено всі основні розділи нормативного курсу «дискретна математика» із спеціальностей комп'ютерних, фізико-математичних та інженерно-технічних напрямків. Від читача вимагається знання математики в обсязі середньої школи, і вся подальша «математична техніка» здобувається у процесі роботи з книгою та запропонованими в ній вправами.

Автори сподіваються, що деякі розділи підручника можуть бути корисними також викладачам і працюючим фахівцям, особливо при використанні рекомендованої додаткової літератури.

Структура книги, перелік викладених розділів та їх наповнення легко убачаються з вмісту. Для кращого засвоєння матеріалу в тексті наведено значну кількість ілюстрованих прикладів, а у кінці розділів — запитання та завдання. В главах з теорії графів і комбінаторики частина інформації повідомляється у формі довідок і задач. Список літератури містить джерела двох типів: загальні підручники або монографії з дискретної математики і книги з окремих розділів — логіці, мов, теорії графів, комбінаториці тощо. Для первинного вивчення весь список літератури є надлишковим, однак може виявитися корисним при поглибленому вивченні окремих розділів.

Епітет «комп'ютерна» у назві вжитий з двох причин. По-перше, постановки та математичні моделі викладених задач — теоретичних і прикладних — припускають використання комп'ютера: для символьних перетворень, для числової реалізації обчислювальних алгоритмів і т. д. Як правило, реальні зразки цих задач певне мають таку розмірність і трудомісткість, що здійснити їх аналіз та ефективний числовий розв'язок без використання комп'ютера неможливо. По-друге, відбір та викладання розділів дискретної математики у книзі виконано, враховуючи вимоги фундаментальної освіти з комп'ютерних наук, інформаційних технологій, сучасних інженерних та соціально-економічних напрямків з високим рівнем автоматизації та комп'ютеризації.

СПИСОК ПОЗНАЧЕНЬ

Множини

N — множина натуральних чисел.
Z — множина цілих чисел.
$N_{\vee}$ — множина парних чисел.
Q_{+} — множина додатних раціональних чисел.
R_{+} — множина додатних чисел.
Q — множина раціональних чисел.
R — множина дійсних чисел.
$|M|$ — потужність множини M.
$A \subseteq B$ — нестроге включення.
$A \subset B$ — строге включення.
$x \in A$ — елемент x належить множині A.
$x \notin A$ — елемент x не належить множині A.
U — універсальна множина.
$\varnothing$ — порожня множина.
2^X — множина степінь (множина всіх підмножин множини X) або булеан.
$A \cup B$ — об'єднання множин A і B.
$A \cap B$ — перетин множин A і B.
$-A$ — доповнення множини A.
$A \backslash B$ — різниця множин A і B.
σ — клас множин, що називається алгеброю.
c — континуум.

Відношення

$A \times B$ — декартів добуток множин A і B.
X^n — декартовий степінь множини X.
R^{-1} — обернене відношення до відношення R.
$S \circ R$ — композиція відношень R і S.
R^n — степінь відношення R.
$R(x)$ — переріз відношення R за елементом x.
$R(Z)$ — переріз відношення R за підмножиною Z.
Y/R — фактор-множина множини Y відносно R.
D_R — область визначення відношення R.
$\Re_R$ — область значень відношення R.
F — функціональне відношення.
$f\colon X \to Y$ — відображення.
$\sim$ — відношення еквівалентності.
$\leq$ — відношення часткового порядку.
$<$ — відношення строгого порядку.

Реляційна алгебра

σ — операція обмеження відношення.
π — операція проекції відношення.
$|><|$ — операція натурального з'єднання відношень.
$\div$ — операція ділення відношень.

Алгебраїчні структури

e — одиничний елемент.
x' — обернений елемент до елементу x.
1 — одиничний елемент відносно множення.

x^{-1} — обернений елемент до елементу x відносно множення.
0 — одиничний елемент відносно додавання.
$-x$ — обернений елемент до елементу x відносно додавання.
$\oplus_n$ — додавання за модулем n.
$\otimes_n$ — множення за модулем n.
Z_+ — множина цілих невід'ємних чисел.
φ: $A \to C$ — гомоморфізм із структури $(A, \otimes)$ у структуру $(C, \oplus)$.

Булеві функції та перетворення

$(x_1, x_2, ..., x_n)$ — двійкове слово, булевий набір або інтерпретація булевої функції f.
$\wedge$ — кон'юнкція.
$\vee$ — диз'юнкція.
$\bar{}$ — заперечення.
$\sim$ — еквіваленція.
$\to$ — імплікація.
$(B, \wedge, \vee, \bar{})$ — алгебраїчна структура — булева алгебра.
$B(\wedge, \vee, \bar{}, \to, \sim)$ — алгебра логіки.
$f(x_1, x_2, ..., x_n)$ — функція, двоїста функції $f(x_1, x_2, ..., x_n)$.
$\oplus$ — сума за модулем 2, що виключає «або».
$(B, \wedge, \oplus, 0, 1)$ — алгебра Жегалкіна.
T_0 — клас функцій, що зберігають нуль.
T_1 — клас функцій, що зберігають одиницю.
S — клас самодвоїстих функцій.
L — клас лінійних функцій.
M — клас монотонних функцій.

Математична логіка

$(\{X, I\}, \wedge, \vee, \bar{}, \to, \sim, X, I)$ — логіка висловлень.
$\frac{A_1, ..., A_n}{B}$ — B наслідок з посилання $A_1, ..., A_n$.
D — предметна область предиката.
$(\forall x)$ — квантор загальності.
$(\exists x)$ — квантор існування.
$\neg x$ — циклічне заперечення.
N_x — заперечення Лукашевича.
$I^\wedge(x)$ — узагальнене заперечення.
$J^\wedge(x)$ — характеристична функція.

Теорія графів

$G = (X, Y, f)$ — абстрактний граф.
X — множина вершин графа.
Y — множина ребер графа.
$\delta(x_i) = \deg x_i$ — степінь вершини x_i.
$d(G)$ — діаметр графа.
$r(G)$ — радіус графа.
ρ — бінарне відношення.
$\delta^+(x)$ — число дуг, що починаються у вершині x.
$\delta^-(x)$ — число дуг, що закінчуються у вершині x.
A — матриця суміжності.

B — матриця інциденцій.
$\bar{G}$ — двоїстий граф.
$\gamma(G)$ — хроматичне число або індекс.
T — дерево.
g_{nm} — число неізоморфних *n-вершинних* графів з m ребрами.
τ_n — число позначених дерев з n вершинами.
t_n — число вільних дерев з n вершинами.
T_n — число кореневих дерев.
$K(G)$ — матриця Кірхгофа.
S — матриця перерізів.
Q — матриця циклів.
K_n — повний граф з n вершинами.
K_{nm} — дводольний граф.
$\rho(G)$ — клікове число графа G.
$\alpha(G)$ — число незалежності графа G.

Мови та граматики

ε — порожній рядок.
A^* — множина всіх рядків алфавіту A, включаючи порожній рядок.
A^+ — множина всіх рядків алфавіту A, не включаючи порожній рядок.
$|\alpha|$ — довжина рядку.
a^n — n символів a поспіль у рядку символів.
$G(N, T, P, S)$ — граматика.
T — множина термінальних символів.
N — множина нетермінальних символів.
S — початковий символ граматики.
P — множина продукцій граматики.
$L(G)$ — мова, що визначена граматикою G.
$\varphi \Rightarrow \psi$ — ψ безпосередньо виводиться з φ.
$\varphi \Rightarrow^* \psi$ — ψ виводиться з φ за скінченне число кроків.
D — дерево виводу.
| — *скорочений запис продукції.*

Алгоритми

Базові функції:
$O_n(x_1, \ldots, x_n)$ — функції, що тотожно дорівнюють нулю.
$V_{n,i}(x_1, \ldots, x_n)$ — функції вибору.
$\lambda(x)$ — функції проходження.
Оператори побудови рекурсивних функцій:
S — оператор підстановки (суперпозиції).
R — оператор рекурсії.
μ — оператор мінімізації.
$T(n)$ — часова складність алгоритму.
$\Theta(g(x))$ — час роботи алгоритму з порядком зростання $g(x)$.

Автомати

λ — кінцевий маркер на вхідній стрічці автомата.
ε — порожня комірка на вхідній стрічці автомата.
$M = (S, I, O, f, g, s_0, F)$ — скінченний автомат.
$M = (S, I, O, Z, f, g, s_0, z_0, F)$ — автомат з магазинною пам'яттю.

$T = (S, I, f, s_0)$ -- машина Тьюринга.
S — скінченна множина станів автомата.
I — скінченна множина допустимих вхідних символів автомата.
O — скінченна множина допустимих вихідних символів автомата.
f — функція переходів автомата.
g — функція виходів автомата.
s_0 — початковий стан керуючого пристрою автомата.
F — множина завершальних станів автомата.
Z — скінченна множина допустимих символів пам'яті.
z_0 — початковий символ пам'яті.

Комбінаторика

A_n^k — кількість k-розміщень n-множини.
P_n — число переставлень з n елементів.
$\overline{A_n^k}$ — кількість k-розміщень (з повтореннями) n-множини.
$\underline{C_n^k}$ — кількість k-сполучень з n елементів.
$\overline{C_n^k}$ — кількість k-сполучень і повтореннями з елементів n типів.
D_n — кількість зміщень.
$D_{n,r}$ — кількість перестановок з n елементів, з яких $n - r$ змінюють свій стан, а рівно r елементів не змінюють.
S_n^k — число Стірлінга другого роду — число способів розміщення n різних предметів за k урнами.
B_n — число Белла (число способів розбиття n-множини на непорожні підмножини, що не перетинаються).
$p_k(n)$ — кількість всього розбиття числа n на k частин (натуральних доданків).
$p(n)$ — кількість всього розбиття числа n з усіма можливими значеннями числа частин k.

розділ 1

Множини

1.1. Множини. Способи задання множин

Елементи множини, способи задання множин, скінченні та нескінченні множини, упорядковані множини, парадокси

В повсякденному житті та практичній діяльності часто доводиться говорити про деякі сукупності різних об'єктів: предметів, понять, чисел, символів тощо. Наприклад, сукупність деталей механізму, аксіом геометрії, чисел натурального ряду, літер абетки. На основі інтуїтивних уявлень про подібні сукупності сформувалося математичне поняття *множини*. Значний внесок до теорії множин зробив Георг Кантор. Згодом завдяки його дослідженням теорія множин стала цілком визначеною та обґрунтованою галуззю математики, а на сьогодні вона здобула фундаментального значення. Теорія множин є підставою для всіх розділів дискретної математики та комп'ютерних наук в цілому, є однією з основ функціонального аналізу, топології, загальної алгебри. Глибокі дослідження в самій теорії множин пов'язані з основами математики.

Теорія множин разом з іншими розділами дискретної математики має безліч корисних застосувань у програмуванні. Вона використовується для побудови систем управління базами даних, під час побудови та організації роботи комп'ютерних мереж, зокрема мережі Інтернет.

Множина є настільки загальним і водночас початковим поняттям, що її строге визначення через більш прості поняття дати важко. Тому вслід за Г. Кантором ми приймаємо інтуїтивне

уявлення про ***множину*** як сукупність деяких ***елементів***, цілком визначених у випадку кожної конкретної множини.

Множини позначають великими, а елементи множин — малими латинськими буквами або малими латинськими буквами з індексами. Елементи множин часто відокремлюються фігурними дужками. Наприклад, запис $A = \{a, b, d, h\}$ означає, що множина A складається з чотирьох елементів a, b, d, h. В загальному вигляді твердження, що скінченна множина A складається з n елементів, записується так: $A = \{a_1, a_2, ..., a_n\}$. Належність елемента множині позначається символом $\in$: $a \in A$ (читають: елемент a належить множині A). В протилежному випадку позначають $a \notin A$ (читають: елемент a не належить множині A).

Далі використовуються такі загальноприйняті позначення основних числових множин.

N — ***множина натуральних чисел***, $N = \{1, 2, 3, ...\}$.

Z — ***множина цілих чисел***, $Z = \{..., -3, -2, -1, 0, 1, 2, 3, ...\}$.

Q — ***множина раціональних чисел***. Будь-яке раціональне число можна зобразити у вигляді дробу: a/b, де $a, b \in Z$, $b \neq 0$.

R — ***множина дійсних чисел***. Будь-яке дійсне число можна зобразити у вигляді нескінченного десяткового дробу $a, b_1 b_2 b_3 \ldots b_n \ldots$ із цілою частиною $a \in Z$ і $b_k \in \{0, ..., 9\}$. Множині дійсних чисел відповідає множина точок на числовій прямій.

Елементами множин можуть бути інші множини, тоді ці елементи позначатимуться великими буквами.

Приклад. $A = \{D, C\}$, $D = \{a, b\}$, $C = \{c, d, e\}$. При цьому $D \in A$, $C \in A$, але $a \notin A$ і $c \notin A$.

Приклад. $E = \{\{1, 2\}, 3\}$. Цей запис означає, що множина E містить два елементи: множину $\{1, 2\}$ і елемент 3.

> **Визначення**
>
> Множина називається ***скінченною***, якщо вона містить скінченне число елементів, і ***нескінченною***, якщо вона містить необмежене число елементів.

Приклад. Множина $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$ цифр в десятковій системі числення скінченна, а множина точок кола нескінченна.

Упорядкованою вважається така множина, в якій важливі не тільки її елементи, але і порядок їх наступності у множині. Наприклад, упорядкованою є множина, в якій кожний елемент має свій порядковий номер. Позначають упорядковану множину, як правило, або круглими, або трикутними дужками. Наприклад,

$$A = <1, 2, 3>,$$

$$A = <a_1, a_2, ..., a_n>, \; n \in N;$$

$$B = (a, b, c).$$

Приклад. Розглянемо упорядковану множину $A = (x, y)$ географічних координат довготи x та широти y. Якщо довготу і широту поміняти місцями, в результаті можна потрапити в іншу точку всесвіту.

Вказати порядковий номер для всякого дійсного числа на множині R неможливо, і порядок у R задається за допомогою порівнянь $<$ і $\leq$. Тому загальне визначення упорядкованої множини X припускає, що для всіх пар елементів з X визначено ***відношення порядку*** (див. п. 2.4)

Способи задання множин

1. Множину можна задати простим ***переліком елементів***

$$A = \{a_1, a_2, ..., a_n\}.$$

Приклад. Множину відмінників у групі позначимо O і задамо її переліком: $O = \{$Іванов, Петров, Сидоров, Кукушкіна$\}$.

Спосіб задання множини переліком її елементів не придатний для задання нескінченних множин, а у випадку скінченних множин його практично часто не можна реалізувати. Так, не можна перелічити множину риб у Тихому океані, хоча їхне число скінченне.

2. Інший спосіб задання множини складається з опису елементів ***визначеною властивістю***: $X = \{x \mid P(x)\}$, де $P(x)$ означає, що елемент x має властивість $P(x)$.

Приклад. Множину N_{10} всіх натуральних чисел, менших за 10, можна задати так: $N_{10} = \{x \mid x \in N, \; x < 10\}$.

Властивості елементів можуть бути задані не формально, а за допомогою опису на природній мові.

Приклад. Множина S студентів групи ПМ-04-1, які одержують стипендію.

Приклад. У геометрії часто доводиться мати справу з множинами, що задані своїми характеристичними властивостями. Так, коло — геометричне місце точок площини, що рівновіддалені від центра даної точки цієї площини.

3. Множина може бути задана ***рекурсивно*** вказівкою способу послідовного породження її елементів.

Приклад. Множина значень рекурсивної функції є рекурсивно-заданою множиною $F = \{\varphi_1, \varphi_2, \varphi_3, ...\}$, де $\varphi_i \in N$, $i = 1, 2, 3, ...$ Нехай $\varphi_1 = 1$, $\varphi_2 = 2$, а кожне наступне число залежить від двох попередніх таким чином:

$$\varphi_n = 3\varphi_{n-2} + \varphi_{n-1},\ n = 3, 4, ...$$

Тоді

$$\varphi_3 = 3\varphi_1 + \varphi_2 = 3 \times 1 + 2 = 5;$$

$$\varphi_4 = 3\varphi_2 + \varphi_3 = 3 \times 2 + 5 = 11 \text{ і т. д.}$$

При заданні множин можуть виникати помилки та протиріччя. Множина задана коректно, якщо для будь-якого елемента можна визначити, належить він множині чи ні.

Приклад. Визначення множини A як множини, що містить будь-які п'ять натуральних чисел, не є коректним, оскільки неможливо визначити точно елементи A. Множина всіх простих чисел визначена коректно. Для будь-якого натурального числа можна перевірити, чи є воно простим, хоча практично на це може знадобитися дуже багато часу.

Приклад. Множина всіх динозаврів, що жили на Землі, є множиною, що задана правильно. Хоча практично неможливо визначити елементи цієї множини, але теоретично ясно, що якщо тварина, яка будь-коли жила на Землі, є динозавром, то вона належить до цієї множини, у протилежному випадку — ні.

Некоректність задання множини часто пов'язана з ***протиріччям*** при перевірці належності деякого елемента множині. Наведемо два класичних приклади.

Приклад. Визначимо множину G як множину всіх множин, які не є елементами самих себе. Але тоді не можна з'ясувати, чи є сама множина G елементом множини G. Якщо так, то приходимо до протиріччя, оскільки G містить як елементи тільки множини, які не є елементами самих себе. Якщо множина G не є елементом самої себе, то тоді,

за визначенням, вона повинна бути елементом множини G, що є протиріччям.

Приклад. Єдиний перукар у місті N визначає множину K мешканців, яких він повинен голити, як сукупність всіх тих мешканців N, які не голяться самі. Але тоді для самого перукаря виходить протиріччя і при включенні його до множини K, і при віднесенні його до мешканців, які голяться самі.

Такі протиріччя називаються ***логічними парадоксами*** і вивчаються в математичній логіці.

Запитання

1. Запишіть за допомогою позначень твердження, що елемент a належить множині A, а елемент b не належить множині A.
2. Наведіть приклади множин, елементами яких є множини.
3. Наведіть приклади скінченних і нескінченних множин.
4. Яку множину називають упорядкованою?
5. Назвіть відомі вам способи задання множин. В якому випадку не можна застосувати той або інший спосіб?
6. В яких випадках множина задана некоректно?
7. Які протиріччя (парадокси) можуть виникнути при визначенні множин? Наведіть приклади.

Завдання

1. Задайте переліченням елементів такі множини:
   а) множину натуральних чисел, не більших за 7;
   б) множину букв вашого імені;
   в) множину, єдиним елементом якої є назва вашого міста;
   г) множину простих чисел між 10 і 20;
   д) множину додатних чисел, що кратні 12.
2. Задайте у вигляді $X = \{x \mid P(x)\}$ такі множини:
   а) множину натуральних чисел, не більших за 100;
   б) множину парних додатних чисел;
   в) множину натуральних чисел, що кратні 10.
3. Назвіть елементи множин:
   а) $\{x \mid x \in N,\ 3 \le x \le 12\}$;
   б) $\{x \mid x$ — десяткова цифра$\}$.
4. Визначте, елементом яких з наведених множин є 2:
   а) $\{x \mid x \in N,\ x > 1\}$;
   б) $\{x \mid x = y^2,\ y \in Z\}$;
   в) $\{2, \{2\}\}$.

1.2. Основні поняття теорії множин

Рівність множин, включення множин, універсальна і порожня множини, степінь множини

Розглянемо поняття рівності множини.

Визначення

Дві множини ***рівні***, якщо вони містять однаковий набір елементів. Позначається $A = B$. Якщо множини не рівні, це позначається $A \neq B$. Число елементів скінченної множини A позначимо через $|A|$.

Для множин A і B з нескінченним або великим числом елементів перевірка збігу наборів всіх елементів може бути важкою. Більш ефективною виявляється логічна перевірка ***двостороннього включення***. А саме, $A = B$ тоді і тільки тоді, коли з $x \in A$ виходить $x \in B$ і з $y \in B$ виходить $y \in A$.

Розглянемо приклад.

Приклад. Нехай задані множини

$A = \{1, 2, 3, 4, 5\}$;

B — множина натуральних чисел від 1 до 5;

$C = \{c \mid 1 \leq c \leq 5, c \in N\}$;

$D = \{4, 1, 5, 2, 3\}$.

Ці множини містять один набір елементів, тому $A = B = C = D$.

При заданні множин можуть бути неточності або збитковості, які необхідно усувати. Розглянемо приклади.

Приклад. Розглянемо множину A залишків, що одержуються при послідовному діленні натуральних чисел $\{3, 4, 5, 6, ...\}$ на 3: $A = \{0, 1, 2, 0, 1, 2, 0, 1, 2, 0, ...\}$. Ця множина містить всього три елементи: 0, 1, 2. Тому її можна записати у вигляді $A = \{0, 1, 2\}$.

Приклад. Нехай B — множина всіх видів шахових фігур, а C — множина всіх шахових фігур, що беруть участь в одній грі. Тоді $|B| = 6$ (пішак, тура, слон, кінь, ферзь, король), а $|C| = 32$ (16 білих і 16 чорних).

Визначення

Множина A, всі елементи якої належать множині B, називається ***підмножиною*** множини B.

Позначення. ***Нестроге включення*** позначається $A \subseteq B$, означає, що A — ***підмножина*** множини B, що, можливо, співпадає з B. ***Строге включення*** позначається $A \subset B$ і означає, що A — підмножина множини B, що не співпадає з B. Символьний вираз $A \subset B$ читають «A включено до B».

Виконання співвідношень $A \subseteq B$ і $B \subseteq A$ можливе тільки при $A = B$. І зворотно, $A = B$, якщо $A \subseteq B$ і $B \subseteq A$ водночас. Зауважимо, що іноді в літературі символом $\subset$ позначають «нестроге» включення, що допускає і рівність множин. У цьому випадку символ $\subseteq$ не використовується, а строге включення записують двома співвідношеннями $A \subset B$, $A \neq B$.

Приклад. Для множини додатних чисел R_+ використовується знак строгого включення відносно множини дійсних чисел: $R_+ \subset R$.

Приклад. Позначимо множину учнів деякого класу через X, множину відмінників у цьому класі — через Y. Тоді $Y \subseteq X$, оскільки множина відмінників у класі включена до множини учнів цього класу і теоретично може дорівнювати їй.

Визначення

Універсальною називається множина, яка містить всі можливі елементи, що зустрічаються в даній задачі. Універсальна множина позначається символом U.

Зауважимо, що універсальна множина U може бути індивідуальною для кожної окремої задачі і визначається в її умові.

Приклад. Розглянемо деяку групу студентів. Нехай A — множина юнаків групи, B — множина відмінників. У цій задачі універсальною є множина студентів групи, а множини A і B є її підмножинами: $A \subseteq U$, $B \subseteq U$.

Визначення

Порожньою називається така множина, яка не містить ніяких елементів. Така множина позначається спеціальним символом $\varnothing$.

Роль порожньої множини $\varnothing$ аналогічна ролі числа нуль. Це поняття можна використовувати для визначення насправді неіснуючої сукупності елементів (наприклад, множини зелених слонів). Більш істотним мотивом введення порожньої

множини є те, що заздалегідь не завжди відомо (або невідомо зовсім), чи існують елементи, які задовольнять характеристичну властивість кожної множини. Наприклад, множина виграшів у наступному тиражі спортлото на куплені квитки може виявитися порожньою. Порожня множина $\varnothing$ є підмножиною будь-якої множини A, $\varnothing \subseteq A$. Слід пам'ятати, що порожня множина є множиною, тому якщо деяка множина A не містить жодного елемента, то $A = \varnothing$; $|A| = 0$. Запис $A = \{\varnothing\}$ означає, що A містить один елемент — $\varnothing$, $|A| = 1$.

Таким чином, будь-яка непорожня множина A обов'язково має, як мінімум, дві підмножини — порожню множину і саму цю множину.

Визначення

Множину всіх підмножин множини X назвемо ***множиною-степенем,*** або ***булеаном множини*** X, і позначимо 2^X.

Приклад. Нехай задана множина $A = \{a, b, c\}$. Система всіх її підмножин є

$$2^A = \{\varnothing, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\},$$

так що 2^A містить 8 елементів.

Порожня множина має тільки одну підмножину — саму порожню множину, тому $2^\varnothing = \{\varnothing\}$. Для довільної множини X з n елементів кількість всіх її підмножин (тобто $|2^X|$) дорівнює 2^n:

$$|2^X| = 2^{|X|} = 2^n.$$

Запитання

1. Які множини вважаються рівними?
2. Чи можуть два елементи однієї множини бути однаковими?
3. Визначте поняття підмножини і включення множин.
4. Наведіть приклади множин A і B для випадків $A \subset B$ і $A \subseteq B$.
5. Чим відрізняється строге включення від нестрогого? Наведіть приклад.
6. Як визначається рівність множин через поняття нестрогого включення?
7. Яка множина називається універсальною?
8. Яка множина називається порожньою?
9. Запишіть відношення включення між універсальною множиною U, довільною її підмножиною A і множиною $\varnothing$.
10. Як позначається множина всіх підмножин деякої множини? Скільки елементів вона містить?

Завдання

1. Визначте, які з наведених тверджень справедливі:
   а) $|\{\varnothing\}| = 1$; в) $|\{\{\varnothing\}\}| = 2$; д) $\{x\} \subseteq \{x\}$;
   б) $\{\{\varnothing\}\} \in \{\{\{\varnothing\}\}\}$; г) $x \in \{x\}$; е) $\{x\} \in \{x\}$; ж) $\{x\} \in \{\{x\}\}$.
2. Скільки елементів містять такі множини:
   а) $\{x\}$;
   б) $\{\{x\}\}$;
   в) $\{x, \{x\}\}$;
   г) $\{\{x\}, x, \{\{x, \{x\}\}\}\}$.
3. Які з наведених тверджень правильні? Доведіть.
   а) якщо $A \subset B$ і $B \subset C$, то $A \subset C$;
   б) якщо $A \subseteq B$ і $B \subseteq A$, то $A = B$;
   в) якщо $A \subseteq B$ і $B \subseteq C$, то $A \subseteq C$.
4. Дана множина $D = \{7, 13, 25, 34, 101, 112\}$. Які з наведених множин є підмножинами множини D?
   а) $\{1, 7, 13\}$;
   б) $\{0, 1, 12\}$;
   в) $\{25, 112, 34\}$;
   г) $\{a, b, c, n\}$;
   д) $\{7, 13, 25, 34, 101, 112\}$.
   е) $\varnothing$.
5. Визначте, які з наведених множин дорівнюють одна одній:
   а) $A = \{x \mid$ існує y такий, що $x = 2y$, $y \in N\}$;
   б) $C = \{1, 2, 3\}$;
   в) $D = \{0, 2, -2, 3, -3, 4, -4, \ldots\}$;
   г) $E = \{2x \mid x \in Z\}$.
6. Побудуйте 2^A для множини A, якщо:
   а) $A = \{\{\varnothing\}\}$;
   б) $A = \{1, 2, 3, 4\}$;
   в) $A = \{$«день», «ніч»$\}$;
   г) $A = \{1, \{2, 3\}, 4\}$.
7. Скільки підмножин містить:
   а) множина днів тижня;
   б) множина місяців року.
8. Нехай задані множини S_{n-1} і S_n, такі, що $S_{n-1} = \{a_0, a_1, \ldots, a_{n-1}\}$, $S_n = \{a_0, a_1, \ldots, a_n\}$. Поясніть, як одержати з множини $2^{S_{n-1}}$ множину 2^{S_n}.
9. Складіть алгоритм, який як вхідні дані одержує дві множини і визначає, чи рівні ці множини, чи є одна з них підмножиною другої.
10. Складіть алгоритм, який як вхідні дані одержує множину і конструює список всіх можливих підмножин даної множини.

1.3. Геометрична інтерпретація множин

Діаграми Венна, круги Ейлера

Для наглядного зображення співвідношень між підмножинами універсальної множини використовуються діаграми Венна і круги Ейлера.

Побудова ***діаграми Венна*** полягає в розбитті площини на 2^n областей за допомогою n фігур. Кожна фігура на діаграмі зображує окрему множину, n — число зображуваних множин. При цьому кожна наступна фігура повинна мати одну і тільки одну загальну область-перетин з кожною з раніше побудованих фігур. Площина, на якій зображуються фігури, становить універсальну множину U. Таким чином, точки, що не належать жодній з фігур, належать тільки U. Діаграма Венна для двох множин A і B зображена на рис. 1.1.

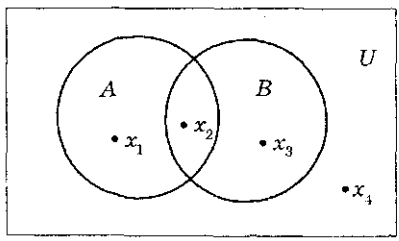


Рис. 1.1. Діаграма Венна для двох множин A і B

За допомогою діаграм Венна можна графічно показати, чи належить деякий елемент $x \in U$ розглянутим множинам, чи ні. Наприклад, на рис. 1.1 елемент x_1 належить A і не належить B, x_2 належить A і B, x_3 належить B і не належить A, x_4 не належить ні A, ні B. Будь-який елемент належить універсальній множині U.

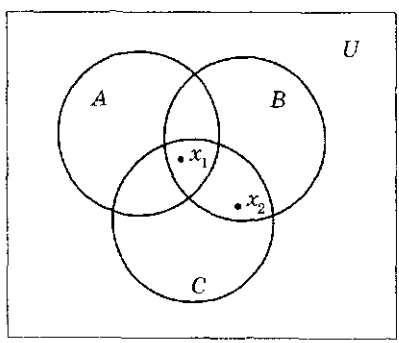


Рис. 1.2. Діаграма Венна для трьох множин A, B і C

Діаграму Венна для трьох множин A, B і C зображено на рис. 1.2, де елемент x_1 належить множинам A, B і C, x_2 належить B і C і не належить A.

Діаграму Венна для чотирьох множин A, B, C і D зображено на рис. 1.3, на якому як приклад зображено елемент x_1, що належить всім чотирьом множинам: A, B, C і D. Для ясного уявлення заштрихуємо кожну область цієї

діаграми, використовуючи більш густе штрихування там, де точки належать більшому числу множин:

- належить тільки одній з множин;
- належить тільки двом з множин;
- належить тільки трьом з множин;
- належить всім чотирьом множинам.

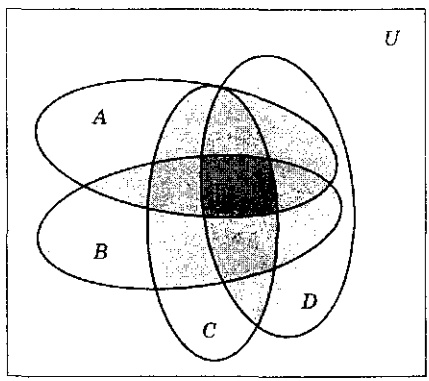
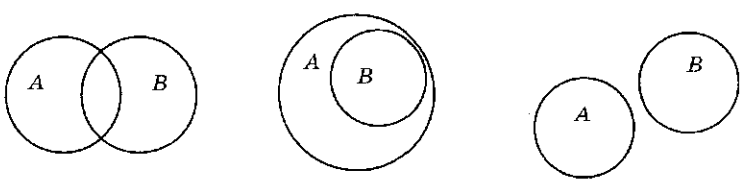


Рис. 1.3. Діаграма Венна для чотирьох множин A, B, C і D

Діаграми Венна не відображають реальні відношення включення, що встановлені між множинами, а розглядають їх у загальному випадку. Індивідуальні відношення між заданими множинами зображують за допомогою ***кругів Ейлера***. В цьому випадку множини, що не мають загальних елементів, зображують не перетинними фігурами. Відношення включення на множинах зображують, розташовуючи одну фігуру вкладеною в іншу. Розглянемо побудову кругів Ейлера на прикладі рис 1.4.



$A = \{1, 4, 6\}$;
$B = \{1, 5, 8\}$;
Загальний елемент — 1

$A = \{1, 4, 6\}$;
$B = \{1, 6\}$;
$B \subseteq A$

$A = \{1, 4, 6\}$;
$B = \{3, 5, 8\}$;
Немає загальних елементів A і B

Рис. 1.4. Зображення множин за допомогою кругів Ейлера

Запитання

1. В чому відмінність між діаграмами Венна і кругами Ейлера?
2. Спробуйте зобразити діаграму Венна для чотирьох множин, використовуючи кола. Чи можливе таке зображення?
3. Чи є фігура, зображена на рис. 1.5, діаграмою Венна для чотирьох множин? Аргументуйте відповідь.
4. Як розташовані круги Ейлера для множин, які не мають загальних елементів?
5. Як за допомогою кругів Ейлера зобразити підмножини даної множини?

Завдання

1. Визначте, яким множинам належать елементи $x_1 \ldots x_7$, що розташовані на діаграмі Венна, зображеній на рис 1.6.
2. Зобразіть такі множини у вигляді кругів Ейлера:
   а) $A = \{0, 1, 2\}$, $B = \{1, 2, 3, 4, 5\}$;
   б) $A = \{a, b, c, d, e\}$, $B = \{d, a, e\}$;
   в) N — натуральні числа, Z — цілі числа, R — дійсні числа;
   г) X — множина птахів, Y — множина звірів, Z — множина ссавців, F — множина кроликів, G — множина живих організмів, які живуть в морях і океанах.
3. Зобразіть за допомогою кругів Ейлера множину A, B, C, якщо $A \subseteq B$, $B \subseteq C$. Покажіть, що якщо $A \subseteq B$, $B \subseteq C$, то $A \subseteq C$.

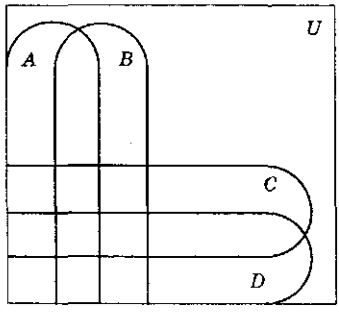


Рис. 1.5. Умова запитання 3

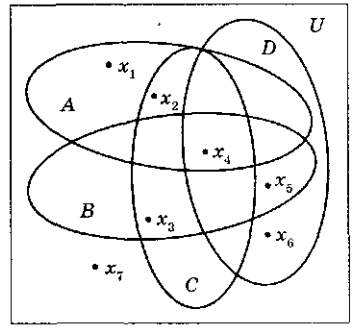


Рис. 1.6. Діаграма Венна

1.4. Операції на множинах

Об'єднання, перетин, різниця, доповнення

Для наочного зображення операцій будемо використовувати діаграми Венна, в яких круги зображують множини, що беруть участь в операції, а заштрихована частина — результат операції.

1. ***Об'єднання (сума)*** $A \cup B$ є множина, що складається з тих і тільки тих елементів, які входять або до A, або до B, або до A і B одночасно (рис. 1.7).

Приклад. Нехай дані множини $A = \{a, b, m\}$; $B = \{m, c, p\}$, тоді їх об'єднання $A \cup B = \{a, b, c, m, p\}$.

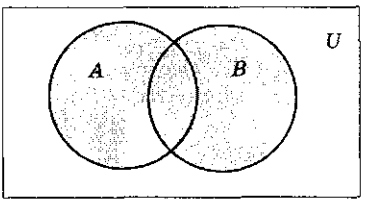


Рис. 1.7. Діаграма Венна для $A \cup B$

2. ***Перетин (добуток)*** $A \cap B$ є множина, що містить тільки елементи, які належать до A і B одночасно (рис. 1.8).

Приклад. Для множин A і B з попереднього прикладу $A \cap B = \{m\}$.

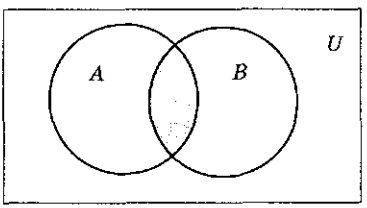


Рис. 1.8. Діаграма Венна для $A \cap B$

3. ***Різниця*** $A \backslash B$ є множина, що складається в точності з усіх елементів A, які не належать до B (рис. 1.9).

Приклад. Для вищерозглянутих тут множин A і B

$$A \backslash B = \{a, b\}.$$

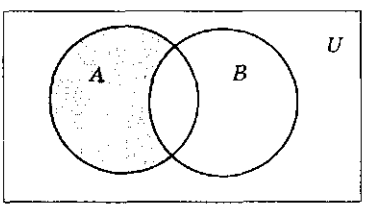


Рис. 1.9. Діаграма Венна для $A \backslash B$

4. ***Доповнення (заперечення)*** $\bar{A}$ (читається ***«не A»***) є множина $U \backslash A$ (рис. 1.10).

Різницю множин можна виразити через операції заперечення та перетину таким чином:

$$A \backslash B = A \cap \bar{B}.$$

Приклад. Виконаємо дії на множині цілих чисел $Z = \{..., -2, -1, 0, 1, 2, ...\}$ та множині $Z_- = \{..., -2, -1, 0\}$. Доповненням до множини Z_- є множина натуральних чисел $N = \{1, 2, ...\}$: $\bar{Z}_- = N$.

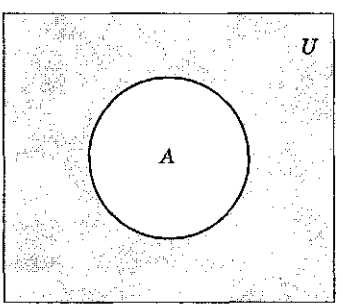


Рис. 1.10. Діаграма Венна для $\bar{A}$

Запитання

1. Дайте визначення операціям на множинах.
2. Наведіть самостійно приклади множин і виконайте кожну з чотирьох операцій.

3. При виконанні якої операції використовується універсальна множина?
4. Чи можуть деякі з цих операцій бути виражені одна через іншу? Яким чином?

Завдання

1. Для множин $A = \{1, 2, 3, 4, 5\}$, $B = \{0, 3, 6\}$ знайдіть:
   а) $A \cup B$; б) $A \cap B$; в) $A \backslash B$; г) $B \backslash A$.
2. За допомогою діаграм Венна доведіть, що
   $\overline{\overline{A}} = A$, $A \backslash B = A \cap \overline{B}$.
3. Нехай A — деяка множина. Знайдіть значення виразів:
   а) $A \cup \varnothing$; в) $A \backslash \varnothing$; д) $A \cap \varnothing$; ж) $\varnothing \backslash A$;
   б) $A \cup A$; г) $A \cup U$; е) $A \cap A$; з) $A \cap U$.
4. Знайдіть множини A і B, якщо $A \backslash B = \{1, 5, 7, 8\}$, $B \backslash A = \{2, 10\}$, $A \cap B = \{3, 6, 9\}$.
5. Нехай A, B і C — множини. Покажіть, що:
   а) $(A \cap B) \subseteq A$; г) $(A \cup B) \subseteq (A \cup B \cup C)$;
   б) $A \subseteq (A \cup B)$; д) $(A \cap B \cap C) \subseteq (A \cap B)$.
   в) $(A \backslash B) \subseteq A$;
6. Зобразіть за допомогою діаграм Венна перетин та об'єднання трьох множин.
7. Доведіть, що $A \subseteq B$ правильно тоді і тільки тоді, коли правильно $\overline{B} \subseteq \overline{A}$.
8. Які висновки можна зробити про множини A і B, якщо правильна одна з таких рівностей:
   а) $A \cup B = A$;
   б) $A \cap B = A$;
   в) $A \backslash B = A$;
   г) $A \backslash B = B \backslash A$.

1.5. Алгебра множин

Пріоритет операцій, тотожності алгебри множин, тотожні перетворення виразів

Множина 2^U всіх підмножин універсальної множини U із заданими на ньому чотирма операціями складає ***алгебру множин.***

У загальному випадку алгебру може складати будь-який клас $\mathcal{A} \subset 2^U$ підмножин універсальної множини U, замкнений відносно всіх чотирьох операцій (див. п. 4.8). Визначення алгебри, що не містить надмірних (точніше, залежних) обмежень, виглядає таким чином.

Визначення
Клас множин $\mathcal{A}$ називається ***алгеброю*** (множин), якщо:
1. $U \in \mathcal{A}$.
2. З $A, B \in \mathcal{A}$ виходить $A \cup B \in \mathcal{A}$.
3. З $A, B \in \mathcal{A}$ виходить $A \backslash B \in \mathcal{A}$.

Алгебра множин широко застосовується у програмуванні, зокрема, при роботі з різноманітними базами даних і становить основу для побудови багатьох математичних структур. Разом з тим, що алгебра множин має основоположне значення в математиці, вона дуже проста і близька до реального життя. Ми щодня застосовуємо операції та закони алгебри множин, не замислюючись над цим. Ми відраховуємо з множин задач, які потрібно розв'язати, множину розв'язаних та беремося до розв'язання решти. Із них ми, вірогідно, в першу чергу виберемо ті, які відносяться до множини легких. Ми готуємо сніданок, визначаючи перетин множини наявних продуктів з множиною продуктів, які нам подобаються. Все наше життя проходить серед множин, які якось взаємозв'язані.

Ми маємо достатньо операцій, щоб створювати складні алгебраїчні вирази. Для цього необхідно визначити, який пріоритет мають операції відносно одна до одної.

Пріоритет операцій в алгебрі множин такий:
1. $\overline{A}$.
2. $A \cap B$.
3. $A \cup B$.
4. $A \backslash B$.

Розглянемо приклад.

Приклад. Нехай треба розташувати дужки (визначити послідовність виконання операцій) у формулі:

$$E = (A \backslash B \cup \overline{A} \cap D) \backslash B.$$

З урахуванням пріоритетів це слід зробити так:

$$E = (A \backslash (B \cup ((\overline{A}) \cap D))) \backslash B.$$

В алгебрі множин $\mathcal{A}$ автоматично виконуються такі тотожності, які дозволяють віднести $\mathcal{A}$ до класу так званих ***булевих алгебр*** (див. розділ 4):

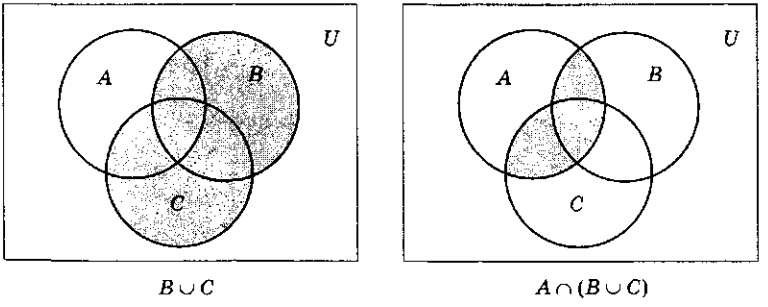
1. ***Комутативні закони***
   1.1. $A \cup B = B \cup A$. 1.2. $A \cap B = B \cap A$.
2. ***Асоціативні закони***
   2.1. $A \cup (B \cup C) = (A \cup B) \cup C$.
   2.2. $A \cap (B \cap C) = (A \cap B) \cap C$.

3. ***Дистрибутивні закони***
   3.1. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.
   3.2. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.
4. ***Властивості порожньої та універсальної множин***
   4.1. $A \cup \varnothing = A$. 4.2. $A \cap U = A$.
   4.3. $A \cup U = U$. 4.4. $A \cap \varnothing = \varnothing$.
5. ***Закони ідемпотентності***
   5.1. $A \cup A = A$. 5.2. $A \cap A = A$.
6. ***Закон інволюції***
   $\overline{\overline{A}} = A$.
7. ***Закон протиріччя***
   $A \cap \overline{A} = \varnothing$.
8. ***Закон виключеного третього***
   $A \cup \overline{A} = U$.
9. ***Закон елімінації***
   9.1. $A \cap (A \cup B) = A$. 9.2. $A \cup (A \cap B) = A$.
10. ***Закони де Моргана***
    10.1. $\overline{A \cup B} = \overline{A} \cap \overline{B}$. 10.2. $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Усі наведені тотожності можна наочно зобразити і довести, використовуючи діаграми Венна.

Приклад. Довести за допомогою діаграм Венна дистрибутивний закон 3.2. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

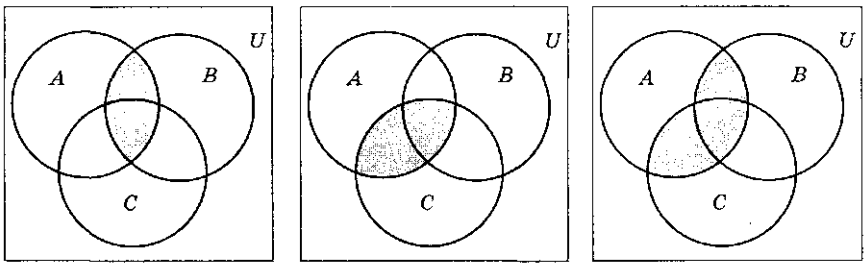
Проілюструємо на діаграмі ліву частину тотожності, виконавши спочатку об'єднання множин B і C, а потім перетин з A (рис. 1.11).



$B \cup C$ $A \cap (B \cup C)$

Рис. 1.11. Побудова діаграми Венна для $A \cap (B \cup C)$

Тепер побудуємо діаграму для правої частини тотожності — $(A \cap B) \cup (A \cap C)$ (рис. 1.12).



$(A \cap B)$ $(A \cap C)$ $(A \cap B) \cup (A \cap C)$

Рис. 1.12. Побудова діаграми Венна для $(A \cap B) \cup (A \cap C)$

Як бачимо, праві діаграми на рис. 1.11 і 1.12 співпадають, отже тотожність 3.2 справедлива.

За допомогою тотожностей алгебри множин можна здійснювати еквівалентні перетворення виразів. Розглянемо такі перетворення на прикладі.

Приклад. Спростити вираз

$\overline{(\bar{A} \cup B \cup C)} \cap (A \cap (B \cup \bar{C})) \cap \bar{B} =$ (застосуємо закон де Моргана)

$= (A \cap \bar{B} \cap \bar{C}) \cap (A \cap (B \cup \bar{C})) \cap \bar{B} =$ (асоціативність і комутативність)

$= A \cap \bar{B} \cap \bar{C} \cap A \cap \bar{B} \cap (B \cup \bar{C})) =$ (застосуємо закон ідемпотентності)

$= A \cap \bar{B} \cap \bar{C} \cap (B \cup \bar{C})) =$ (застосуємо дистрибутивний закон)

$= A \cap \bar{B} \cap \bar{C} \cap B \cup A \cap \bar{B} \cap \bar{C} \cap \bar{C} =$ (згідно з властивостями порожньої множини)

$= \varnothing \cup A \cap \bar{B} \cap \bar{C} = A \cap \bar{B} \cap \bar{C}.$

Відповідь: $A \cap \bar{B} \cap \bar{C}$.

Запитання

1. Розташуйте операції алгебри множин відповідно до їх пріоритетів.
2. Назвіть тотожності алгебри множин, запишіть відповідні формули.
3. Яким чином можна графічно зобразити та довести закони і тотожності алгебри множин? Наведіть приклад такого доведення.
4. Яким чином виконуються еквівалентні перетворення формул алгебри множин? Наведіть приклади.

Завдання

1. Доведіть за допомогою діаграм Венна:
   а) асоціативні закони;
   б) дистрибутивні закони;
   в) властивості порожньої і універсальної множин.
2. Доведіть за допомогою еквівалентних перетворень закони елімінації.
3. Виходячи з властивостей порожньої множини, за допомогою еквівалентних перетворень одержіть властивості універсальної множини.
4. Спростіть вирази:
   а) $\overline{\bar{A} \cup C} \cup (B \cup B \cap C) \cap (\bar{B} \cup \overline{B \cup C})$;
   б) $(A \cap \bar{B} \cup C) \cap \overline{(A \cup B)} \cap \bar{C}$;
   в) $A \cap ((B \cap \bar{C} \cup C \cup B) \cap C) \cap \bar{A}$);
   г) $(A \cap B \cap C) \cup (\bar{A} \cap B \cap C)$.
5. В якому відношенні знаходяться множини A і B, якщо $A \backslash B = B \backslash A = \varnothing$?
6. Чи є алгеброю клас множин, замкнених відносно:
   а) операцій $\cup$, $\cap$;
   б) операцій $\cup$, доповнення.

1.6. Нескінченні множини

Зчисленні та континуальні множини, потужність

У дискретній математиці, як і у всій математиці, в природознавстві дуже важлива роль натурального ряду чисел $N = \{1, 2, ...\}$. Хоча практично обчислення завжди виконуються зі скінченним відрізком $\{1, 2, ..., n\}$ цієї нескінченної множини, немає можливості вказати найбільше число n, яке не буде перевершене у всіх випадках життя. Тому доводиться досліджувати властивості всієї нескінченної множини чисел N. На дійсній числовій вісі R натуральні та цілі числа утворюють доволі «розріджені» підмножини N, Z (рис. 1.13), які інтуїтивно можна назвати «дискретні». Множина раціональних чисел Q, що утворюються при діленні цілих чисел m/n, щільно розташовується на дійсній вісі R: інтервал (a, b) як завгодно малої довжини $\varepsilon = b - a$ $(a \neq b)$ містить нескінченну множину різних раціональних точок. Чи можна множину Q вважати «дискретною»? Виявляється так, хоча інтуїція відмовляється це визнати.

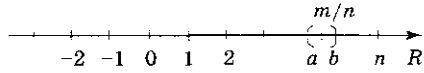


Рис. 1.13. Натуральні, цілі та раціональні числа на вісі R

Нарешті, вся дійсна вісь R — явно неперервна множина, і тут інтуїтивний висновок нас не підводить. Для строгого визначення дискретних та неперервних нескінченних множин використовуються зіставлення дискретних множин натуральному ряду чисел, а неперервних множин — відрізку $[0, 1]$ дійсної вісі. Необхідно тільки уточнити термін «зіставити». Мається на увазі встановити взаємно однозначну відповідність.

Поняття взаємно однозначної відповідності є первинним у свідомості людини при диференційованому сприйнятті предметів зовнішнього світу. Якщо зал для глядачів заповнений і немає вільних місць, нам не обов'язково рахувати кількість присутніх глядачів, вона дорівнює числу крісел у залі. Щоб зробити цей висновок, ми інтуїтивно використовуємо наявність взаємно однозначної відповідності між глядачами і кріслами. Відзначимо одну особливість: фактично у цьому випадку реалізована конкретна взаємно однозначна відповідність глядачі — крісла (кожному глядачеві відповідає одне і тільки одне визначене крісло і навпаки). Після перерви деякі глядачі можуть помінятися місцями, і конкретна відповідність стане іншою, але висновок залишиться попереднім: число глядачів дорівнює числу місць.

Визначення

Взаємно однозначною називається така ***відповідність*** між множинами A і B, при якій кожному елементу $a \in A$ відповідає один і тільки один елемент $b \in B$, і кожному елементу $b \in B$ відповідає один і тільки один елемент $a \in A$. Функція, що визначає взаємно однозначну відповідність, називається ***бієктивною функцією або бієкцією***.

Визначення

Множини A і B називаються ***еквівалентними*** або ***рівнопотужними*** ($A \sim B$), якщо між ними можна встановити взаємно однозначну відповідність.

У прикладі із заповненим залом для глядачів множина глядачів еквівалентна множині крісел. Таким чином,

еквівалентними одна одній виявляються всі скінченні множини з однаковим числом елементів n, і число n вважається ***потужністю*** цих множин.

Для нескінченної множини строге поняття потужності не вводиться, але сам термін «***потужність***» використовується для позначення властивості, загальної для всіх еквівалентних множин. Якщо дві нескінченні множини A і B еквівалентні ($A \sim B$), то рівність їх потужностей формально записується як $|A| = |B|$.

Визначення

Множина A називається ***зчисленною***, якщо вона еквівалентна натуральному ряду N ($A \sim N$). Термін «***зчисленність***» є точним замінником інтуїтивного поняття — «***дискретність***».

За допомогою бієкції $\varphi = N \to A$ можна «перелічити» всі елементи a з A, привласнивши їм індекси за правилом $\varphi(n) = a_n$. Можна записати, що $A = \{a_n,\ n = 1,\ 2,\ ...\}$. Множини парних натуральних чисел $N_ч = \{2,\ 4,\ ...,\ m,\ ...\}$, всіх натуральних чисел $N = \{1, 2, ..., n, ...\}$, цілих чисел Z і раціональних чисел Q послідовно вкладені: $N_ч \subset N \subset Z \subset Q$. Хоча для будь-яких двох з цих множин немає рівності, вони ***еквівалентні одна одній***, тобто мають однакову потужність і є зчисленними: $|N_ч| = |N| = |Z| = |Q|$. Тому відповідно до наших угод множини $N_ч$, N, Z, Q є ***дискретними***.

Дійсно, еквівалентність $N \sim N_ч$ аргументується за допомогою бієкції $\varphi(n) = 2n : 2n = m$. Множину цілих чисел Z можна «перелічити» (тобто присвоїти його елементам натуральні індекси) за правилом:

n \ m	1	2	3	4	...
1	1/1	2/1	3/1	4/1	...
2	1/2	2/2	3/2	4/2	...
3	1/3	2/3	3/3	4/3	...
4	1/4	2/4	3/4	4/4	...
...	...	...	...	...	...

Рис. 1.14. Таблиця множини Q_+

$0 = z_1;\quad 1 = z_2;\quad -1 = z_3;$

$2 = z_4;\quad -2 = z_5;\quad 3 = z_6;\ ...$

Отже $Z \sim N$.

Для доведення еквівалентності множин Q і N достатньо вказати правило нумерації множини Q_+ додатних раціональних чисел, що утворяться діленням m/n натуральних чисел m і n.

Пронумеруємо стовпці та рядки нескінченної таблиці індексами m і n відповідно (рис. 1.14).

Будемо нумерувати послідовно числа n/m вздовж пунктирних похилих стрілок, починаючи з лівого верхнього кута таблиці і переходячи після проходження кожної стрілки до сусідньої, більш довгої. При цьому пропускаються відношення m/n, чисельні значення яких зустрічалися раніше:

$q_1 = 1 = 1/1$, $q_2 = 1/2$, $q_3 = 2 = 2/1$, $q_4 = 1/3$,
$q_5 = 3 = 3/1$, $q_6 = 1/4$, $q_7 = 2/3$, $q_8 = 3/2$, $q_9 = 4 = 4/1$, ...

Таким чином, множина Q раціональних чисел зчисленна:

$$Q = \{q_n\}, \quad n = 1, 2, \ldots.$$

Звичайно, існують нескінченні ***незчисленні*** множини, та їх потужність природно вважати більшою за $|N|$. Так, множина точок відрізка $[0, 1] = \{x \in R;\ 0 \le x \le 1\}$ *не є зчисленною* (теорема Г. Кантора). Її потужність називається ***континуум*** і позначається малою буквою c: $|[0, 1]| = c$. Сама множина $[0, 1]$ і будь-яка еквівалентна їй множина називаються ***континуальними***.

Виявляється, що на дійсній вісі R континуальними (тобто еквівалентними одна одній і відрізку $[0, 1]$) є, наприклад, множини $[a, b]$, (a, b), при будь-якому $a < b$; $(0, +\infty)$; множина $(-\infty, +\infty)$, що дорівнює R.

Континуальні також множини точок будь-якого квадрату та кругу на площині R^2, паралелепіпеду та кулі у просторі R^3 і самого простору R^3: $|R| = |R^2| = |R^3| = c$.

Завдання

1. Довести такі твердження:

   1.1. Із будь-якої нескінченної множини можна виділити зчисленну підмножину.

   1.2. Будь-яка нескінченна підмножина зчисленної множини також зчисленна.

   1.3. Сума (об'єднання) скінченної та зчисленної множин є зчисленна множина.

   1.4. Об'єднання скінченного числа зчисленних множин є зчисленна множина.

2. Яка потужність множини многочленів будь-яких степенів з цілими коефіцієнтами?

3. Дійсне число називається ***алгебраїчним***, якщо воно є коренем деякого многочлена з цілими коефіцієнтами. Всі інші числа називаються ***трансцендентними***. Яка потужність множини алгебраїчних чисел?

розділ 2

Відношення

2.1. Поняття відношення. Задання відношень

Декартів добуток множин, n-арне відношення, бінарне відношення, способи задання відношень. Окремі випадки відношень

Відношення реалізують у математичних термінах на абстрактних множинах реальні зв'язки між реальними об'єктами. Відношення застосовуються при побудові комп'ютерних баз даних, які організовані у вигляді таблиць даних. Зв'язки між групами даних у таблицях описуються мовою відношень. Самі дані обробляються і перетворюються за допомогою операцій, математично точно визначених для відношень. Такі бази даних називаються реляційними і широко застосовуються для збереження та обробки найрізноманітнішої інформації: виробничої, комерційної, статистичної тощо. Відношення також часто використовуються в програмуванні. Такі складові структури даних, як списки, дерева тощо звичайно використовуються для опису деякої множини даних разом з відношенням між елементами цієї множини.

Приклад. Деякі підприємства, які займаються дизайном інтер'єру приміщень, використовують види продукції деяких виробничих фірм, що розташовані в інших містах. Для аналізу необхідно скласти «відношення» реальних комбінацій трьох параметрів: назви фірми, місця її знаходження (місто), виду продукції, що пропонується.

Нехай відомо, що ЧПП «Orion» (Одеса) продає меблі, ТОВ «День» (Харків) продає світильники, ЧКП «Sit» (Одеса) торгує меблями та світильниками, ТОВ «House» (Харків) продає світильники та матеріали для ремонту.

В цьому відношенні беруть участь три множини:

Фірми = {ЧПП «Orion», ТОВ «День», ЧКП «Sit», ТОВ «House»} — множина фірм.

Міста = {Одеса, Харків} — множина міст.

Продукція = {меблі, світильники, матеріали для ремонту} — множина видів продукції.

Це відношення можна формально зобразити списком елементів таким чином: {(ЧПП «Orion» (Одеса) — меблі), (ТОВ «День» (Харків) — світильники), (ЧКП «Sit» (Одеса) — меблі), (ЧКП «Sit» (Одеса) — світильники), (ТОВ «House» (Харків) — світильники), (ТОВ «House» (Харків) — матеріали для ремонту)}. Ми бачимо, що одержане відношення — це множина, що складається з упорядкованих груп типу (ЧПП «Orion» (Одеса) — меблі). Перший елемент групи належить множині «Фірми», другий — множині «Міста», третій — множині «Продукція». Крім того, не всі можливі комбінації елементів цих множин належать нашому відношенню, а тільки деякі з них, наприклад, така група, як (ЧПП «Orion» (Одеса) — матеріали для ремонту) не належить нашому відношенню, оскільки ЧПП «Orion» торгує тільки меблями, а група (ЧПП «Orion» (Харків) — меблі) не належить нашому відношенню, оскільки ЧПП «Orion» знаходиться в Одесі, а не у Харкові. Виходить, що наше відношення — це деяка підмножина множини всіляких комбінацій фірм, міст, продукцій.

Для формального опису всіляких комбінацій з елементів множин, що входять до відношення, використовується поняття декартова добутку множин. Розглянемо це поняття і потім, використовуючи його, визначимо формально поняття відношення.

Визначення

Декартовим добутком множин $X_1 \times X_2 \times \ldots \times X_n$ називається множина всіх можливих упорядкованих наборів $(x_1, x_2, \ldots, x_n)$ з n елементів (які називають ***кортежами*** довжини n), в яких перший елемент належить множині X_1, другий — множині X_2, ..., n-й — множині X_n. Декартів

добуток $X \times X \times \ldots \times X$, в якому одна й та ж множина X помножується n раз сама на себе, називають ***декартовим степенем*** множини і позначають X^n. При цьому $X^1 = X$. Множину X^2 називають ***декартовим квадратом*** множини X, множину X^3 — ***декартовим кубом*** множини X.

Приклад. Нехай $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2\}$, $C = \{c_1, c_2\}$. Тоді

$$A \times B = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2), (a_3, b_1), (a_3, b_2)\}.$$

$$B \times A = \{(b_1, a_1), (b_1, a_2), (b_1, a_3), (b_2, a_1), (b_2, a_2), (b_2, a_3)\}.$$

$$A \times B \times C = \{(a_1, b_1, c_1), (a_1, b_1, c_2), (a_1, b_2, c_1), (a_1, b_2, c_2),$$
$$(a_2, b_1, c_1), (a_2, b_1, c_2), (a_2, b_2, c_1), (a_2, b_2, c_2), (a_3, b_1, c_1),$$
$$(a_3, b_1, c_2), (a_3, b_2, c_1), (a_3, b_2, c_2)\}.$$

$$B^2 = \{(b_1, b_1), (b_1, b_2), (b_2, b_1), (b_2, b_2)\}.$$

Порядок проходження пар може бути довільним, але розміщення елементів у кожній парі визначається порядком проходження множин, що перемножуються, тобто $A \times B \neq B \times A$, якщо $A \neq B$.

Визначення

n-арне відношення R на множинах $X_1, X_2, \ldots, X_n$ — це підмножина декартова добутку цих n множин: $R \subseteq X_1 \times \times X_2 \times \ldots \times X_n$.

Якщо набір елементів $(x_1, x_2, \ldots, x_n)$ належить відношенню R, то стверджують, що елементи $x_1, x_2, \ldots, x_n$ знаходяться у відношенні R. Під ***n-арним відношенням R на множині X*** розуміється підмножина n-го степеня цієї множини: $R \subset X^n$. Якщо $n = 1$, то відношення називається ***унарним***, якщо $n = 2$ — ***бінарним***. Зауважимо, що унарне відношення R на множині X — це підмножина в самому X: $R \subset X$.

Приклад. Відношенням на множинах A, B, C з попереднього прикладу є будь-яка підмножина множини $A \times B \times C$, зокрема

$$R_1 = \{(a_1, b_1, c_1), (a_2, b_1, c_1), (a_2, b_1, c_2), (a_3, b_1, c_1),$$
$$(a_3, b_1, c_2), (a_3, b_2, c_2)\};$$

$$R_2 = \{(a_2, b_2, c_1), (a_2, b_2, c_2), (a_3, b_1, c_1)\}.$$

Розглянемо окремо бінарні відношення, які є «базисними» у тому розумінні, що будь-яке n-арне відношення можна

зобразити у вигляді ланцюжка бінарних відношень, що послідовно конструюються. Цей очевидний факт є наслідком асоціативності декартова добутку множин.

Способи задання бінарних відношень

Якщо R — бінарне відношення на множинах X, Y, то факт $(x, y) \in R$ часто записується у вигляді xRy, і говорять, що елемент $x \in X$ знаходиться у відношенні R з елементом $y \in Y$.

1. Будь-яке бінарне відношення може бути задане у вигляді ***списку***, елементами якого є пари, з яких складається відношення.

Приклад. На множинах чисел $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$; $B = \{24, 25, 26\}$ побудуємо відношення «дільник», яке складається з упорядкованих пар виду (x, y), де x — дільник y, $x \in A$, $y \in B$. Позначимо це відношення через R:

$$R = \{(1, 24), (1, 25), (1, 26), (2, 24), (2, 26), (3, 24), (4, 24), (5, 25), (6, 24), (8, 24)\}.$$

2. Бінарне відношення R на множинах X і Y може бути задане за допомогою ***матриці*** ($W = W(R)$), рядки якої відповідають елементам множини X, стовпці — елементам множини Y. Якщо $n = |X|$, $m = |Y|$ — кількості елементів множин X і Y відповідно, то довільна матриця W має розмір $n \times m$. Елемент w_{ij} матриці відповідає парі $(x_i, y_j) \in A \times B$ декартова добутку множин, причому $w_{ij} = 1$, якщо $(x_i, y_j) \in R$, і $w_{ij} = 0$, якщо $(x_i, y_j) \notin R$, тобто відношення R не містить пару (x_i, y_j).

Приклад. Наступна матриця W задає відношення R «дільник» для числових множин A і B з попереднього прикладу:

$W =$

A \ B	24	25	26
1	1	1	1
2	1	0	1
3	1	0	0
4	1	0	0
5	0	1	0
6	1	0	0
7	0	0	0
8	1	0	0
9	0	0	0

.

3. Бінарне відношення R на множинах X, Y може бути задане ***графічно***. На площині зобразимо точками x_i і y_j елементи множин X і Y. Якщо пара (x_i, y_j) належить відношенню R, з'єднаємо точки x_i, y_j лінією, яка спрямована від першого елемента пари до другого. Позначивши таким чином всі пари, що належать відношенню R, одержимо фігуру, яка називається ***графом*** відношення. Спрямовані лінії, що з'єднують пари точок, називаються ***дугами***, а точки, що зображують елементи множин, — ***вершинами*** графа. Якщо бінарне відношення R задане на одній множині X ($R \subseteq X^2$), то вершинами графа будуть елементи множини X.

Приклад. Як приклад графічного задання відношення розглянемо генеалогічне дерево деякої родини, що зображене на рис. 2.1

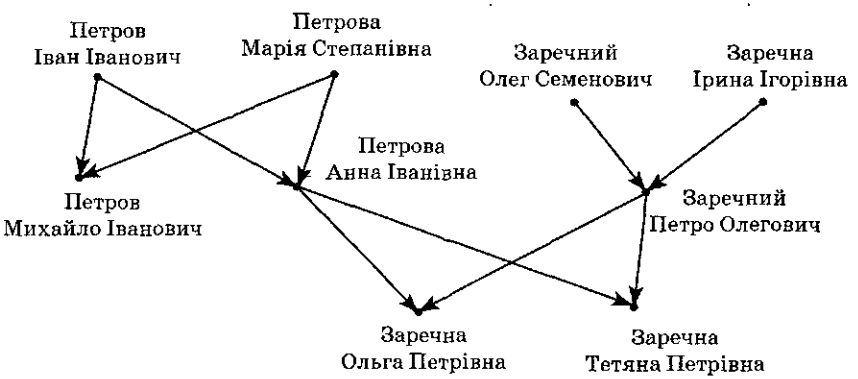


Рис. 2.1. Граф відношення «батько»

Відношення «батько» в даному випадку задане на множині, що складається з 9 осіб. Рис. 2.1 зображує граф відношення «батько». Упорядковані пари елементів множини, що належать відношенню, з'єднані дугами графа. Перший елемент такої пари є батьком, другий — його дитиною. Дуга графа спрямована від першого елемента до другого.

Розглянемо деякі часткові випадки відношень. Нехай задане бінарне відношення R на множині A: $R \subseteq A^2$. Можливий випадок, коли $R = A^2$, — таке відношення називається ***повним***. Для п'ятиелементної множини A граф повного відношення зображено на рис. 2.2,*б*. Може трапитися, що $R = \varnothing$, — таке

відношення називається ***порожнім***. При $|A| = 5$ граф зображено на рис. 2.2,*в*. Якщо відношення містить всі можливі пари виду (a, a) і не містить інших пар елементів, то таке відношення називається ***тотожним***. Граф цього відношення зображено на рис. 2.2,*а* для $|A| = 5$.

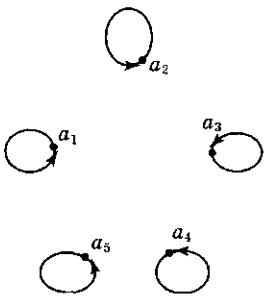


Рис. 2.2,*а*
Граф тотожного відношення

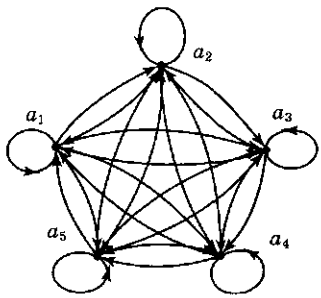


Рис. 2.2,*б*
Граф повного відношення

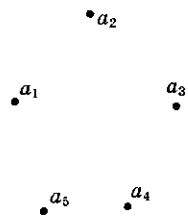


Рис. 2.2,*в*
Граф порожнього відношення

Якщо ***повне відношення*** задане за допомогою матриці, то всі елементи цієї матриці дорівнюють одиниці. Матриця ***порожнього відношення*** складається з нульових елементів.

Ми вивчили три способи задання відношень. Переліченням елементів можливо задавати n-арні відношення при $n \geq 2$. За допомогою матриці і графа зручно задавати тільки бінарні відношення.

Запитання

1. Як зв'язані теорія відношень та теорія множин?
2. Дайте визначення декартова добутку множин.
3. Нехай A — деяка множина. Що означає запис A^2, A^3, A^n?
4. Нехай A і B — множини. Поясніть, чому $A \times B \neq B \times A$.
5. Що називається відношенням? Що таке *n-арне*, бінарне, унарне відношення?
6. Назвіть способи задання відношень. Які з них застосовні для *n-арних* відношень при $n > 2$?
7. Поясніть поняття «граф», «вершина», «дуга».
8. Що таке тотожне відношення, повне відношення, порожнє відношення? Зобразіть графи цих відношень, побудуйте їх матриці.

Завдання

1. Побудуйте матрицю і граф для таких відношень, визначених на множині $A = \{1, 2, 3\}$:
   а) $\{(1, 1), (1, 2), (1, 3)\}$;
   б) $\{(1, 1), (2, 1), (2, 2), (2, 3)\}$;
   в) $\{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$;
   г) $\{(1, 3), (3, 1)\}$.

2. Побудуйте граф і список елементів для таких відношень, визначених на множині $A = \{a, b, c\}$ матрицями:

а)

	a	b	c
a	1	0	1
b	0	1	0
c	1	0	1

б)

	a	b	c
a	0	1	0
b	0	1	0
c	0	1	0

в)

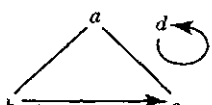
	a	b	c
a	1	1	1
b	1	0	1
c	1	1	1

г)

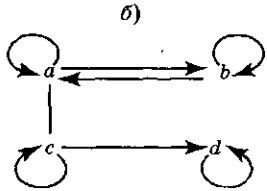
	a	b	c
a	1	0	1
b	0	1	0
c	1	0	1

3. Побудуйте матрицю і список елементів для таких відношень, що задані графічно:

а)



б)



4. Запишіть список елементів для 3-арного відношення R, що задане на множині натуральних чисел таким чином: $(a, b, c) \in R$, якщо $0 < a < b < c < 5$.

5. Запишіть список елементів для 4-арного відношення R, що задане на множині натуральних чисел таким чином: $(a, b, c, d) \in R$, якщо $abcd = 6$.

6. Задайте закон, визначальний відношення H, що зв'язує клітинки шахової дошки, розташовані на відстані одного хода конем, тобто $(x, y) \in H$, якщо кінь може перейти з клітинки x на клітинку y за один хід.

7. Доведіть, що на множині A, що складається з n елементів, може бути задано 2^{n^2} різних бінарних відношень. Скільки 3-арних відношень може бути задано на цій множині?

8. Скільки різних відношень може бути задано між множинами A і B, якщо $|A| = n$, $|B| = m$?

9. Складіть 16 різних відношень на множині $\{0, 1\}$.

10. Визначте алгоритм складання матриці відношення за заданим списком елементів і навпаки.

2.2. Операції над відношеннями

Обернене відношення, композиція відношень, степінь відношення, переріз відношення, фактор-множина

Повернемося до відношення «батько». Якщо упорядкована пара елементів (x, y) належить до цього відношення, це означає, що x є батьком y. Але якщо x є батьком y, то y є дитиною x. Із цього виходить, що існує і відношення «дитина», якому належить пара (y, x). Очевидно, що граф відношення «дитина» можна одержати з графа відношення «батько» зміною напрямку дуг, як показано на рис. 2.3, що «обертає» рис. 2.1.

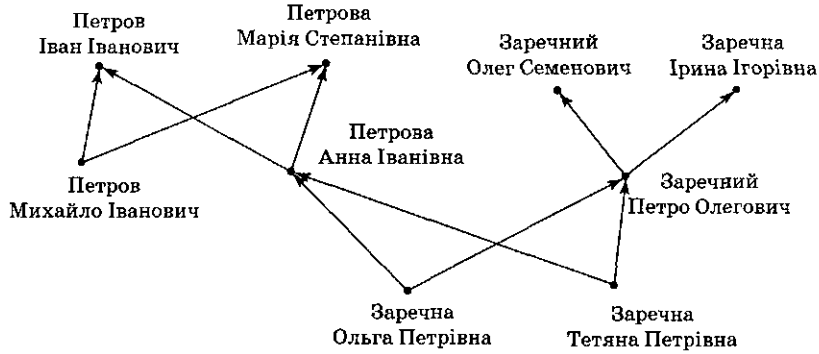


Рис 2.3. Граф відношення «дитина», оберненого до відношення «батько»

Цей приклад пояснює поняття оберненого відношення.

Визначення

Нехай R-бінарне відношення. ***Обернене*** відношення до R позначається R^{-1}. ***Упорядкована пара*** (y, x) належить R^{-1} тоді і тільки тоді, коли (x, y) належить R.

Якщо $R \subseteq X^2$, то $R^{-1} \subseteq X^2$, де X — деяка множина. Якщо $R \subseteq X \times Y$, то $R^{-1} \subseteq Y \times X$.

Якщо бінарне відношення задане на двох множинах, то граф відношення можна побудувати таким чином. Вершини графа, що відповідають елементам першої множини, розташовуються ліворуч, вершини графа, що відповідають

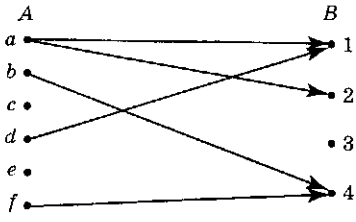


Рис. 2.4. Граф відношення $R = \{(a, 1), (a, 2), (b, 4), (d, 1), (f, 4)\}$

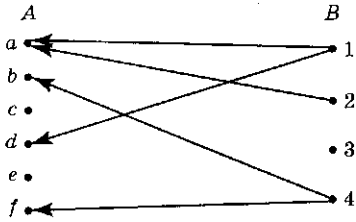


Рис. 2.5. Граф відношення $R^{-1} = \{(1, a), (2, a), (4, b), (1, d), (4, f)\}$

елементам другої множини, розташовуються праворуч. Таким чином, дуги графа спрямовані зліва направо. Розглянемо приклад. Нехай задані множини $A = \{a, b, c, d, e, f\}$, $B = \{1, 2, 3, 4\}$ і відношення $R = \{(a, 1), (a, 2), (b, 4), (d, 1), (f, 4)\}$. Граф цього відношення зображено на рис. 2.4.

Для того щоб побудувати граф відношення R^{-1}, змінимо напрямки дуг (рис. 2.5).

Тепер вивчимо спосіб одержання відношення з двох інших відношень, використовуючи операцію композиції. Нехай є множина $C = \{w, x, y, z\}$ і $B = \{1, 2, 3, 4\}$ і відношення $S = \{(1, x), (2, y), (3, x), (3, z)\}$, $S \subseteq B \times C$. Доповнимо рис. 2.4, зобразивши на ньому, крім графа відношення R, граф відношення S.

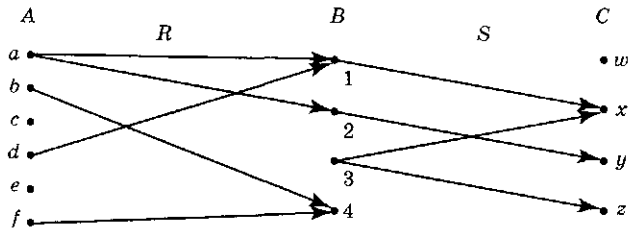


Рис. 2.6. Граф відношення R і відношення $S = \{(1, x), (2, y), (3, x), (3, z)\}$

Визначення

Нехай R і S — відношення, такі, що $R \subseteq X \times Y$, $S \subseteq Y \times Z$, де X, Y, Z — деякі множини. ***Композицією відношень*** R і S називається відношення, що складається з упорядкованих пар (x, z), $x \in X$, $z \in Z$, для яких існує елемент $y \in Y$, такий, що виконуються умови $(x, y) \in R$, $(y, z) \in S$. Композиція відношень R і S позначається $S \circ R$.

Зокрема, для відношень $R \subseteq A \times B$ і $S \subseteq B \times C$, зображених на рис. 2.6, композиція $S \circ R$ є відношення, що зображене на рис. 2.7 і є підмножиною декартова добутку $A \times C$.

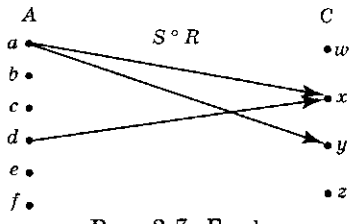


Рис. 2.7. Граф відношення $S \circ R$

Зауважимо, що для пари $(x, z) \in S \circ R$ «проміжних» елементів Y може бути кілька, однак їх кількість (якщо вона не нульова) не впливає на вид композиції $S \circ R$.

Операція композицій відношень дозволяє ввести поняття степеня бінарного відношення, що задане на одній множині.

Визначення

Нехай R — деяке відношення, визначене на множині X: $R \subseteq X \times X$. Тоді ***n*-й степінь** відношення R позначається R^n і визначається рекурсивно так:
R^0 — тотожне відношення на множині X;
$R^n = R^{n-1} \circ R$, для $n = 1, 2, \ldots$

Із визначення маємо, що $R^1 = R$, $R^2 = R \circ R$, $R^3 = R^2 \circ R$, тощо.

Приклад. Нехай відношення R на множині $A = \{a, b, c, d\}$ задане графом, зображеним на рис. 2.8.

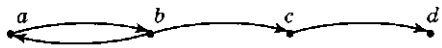


Рис. 2.8. Граф відношення R

Побудуємо степені відношення R. Квадрат відношення R^2 можна побудувати, використовуючи правило: в графі R^2 присутня дуга (x, y), якщо існує така вершина z, що правильні xRz і zRy (рис. 2.9).

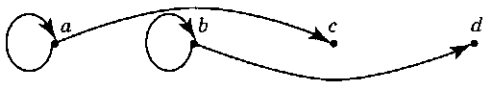


Рис. 2.9. Граф відношення R^2

Степінь R^3 можна побудувати аналогічно, використовуючи правило: в графі R^3 присутня дуга (x, y), якщо існує така вершина z, що правильні xR^2z і zRy (рис. 2.10).

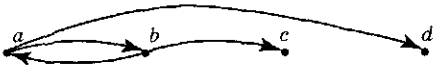


Рис 2.10. Граф відношення R^3

Побудуємо також відношення R^4. Правило побудови аналогічне: xR^4y правильне, якщо існує така вершина z, що правильні xR^3z і zRy (рис. 2.11).

Рис 2.11. Граф відношення R^4

Ми бачимо, що побудований граф R^4 не відрізняється від графа відношення R^2, тому в результаті подальших побудов відношень R^5, R^6..., ми одержимо відношення, що співпадають з R^2, R^3.

Зрозуміло, що справедливе наступне графічне трактування степеня відношення: в графі відношення R^n є дуга з x в y, якщо в графі R з вершини x у вершину y веде хоча б один маршрут довжини n (що складається з n дуг), причому одна й та ж дуга може бути включена до маршруту скільки завгодно раз.

Розглянемо ще одну операцію, яка називається перерізом бінарного відношення.

Визначення

Нехай $R \subset X \times Y$ — відношення на множинах X і Y. Якщо $x \in X$, то ***перерізом*** відношення R за x, що позначається $R(x)$, є множина $R(x) \subseteq Y$, що складається з елементів $y \in Y$, таких, що $(x, y) \in R$. Об'єднання перерізів за елементами деякої підмножини $Z \subset X$ називається ***перерізом $R(Z)$ відносно підмножини Z***. Множина, що складається з перерізів відношення $R \subseteq X \times Y$ за кожним елементом з X, називається ***фактор-множиною*** множини Y за відношенням R і позначається Y/R. Формально можна записати, що $Y/R = \{R(x), \forall x \in X\}$.

Приклад. Розглянемо перерізи відношення R на множинах $A = \{a, b, c, d, e, f\}$, $B = \{1, 2, 3, 4\}$, $R \subseteq A \times B$, що задане графом (рис. 2.4): $R = \{(a, 1), (a, 2), (b, 4), (d, 1), (f, 4)\}$. Можна одержати такі перерізи: $R(a) = \{1, 2\}$, $R(b) = \{4\}$, $R(d) = \{1\}$, $R(e) = \varnothing$, $R(f) = \{4\}$. Фактор-множина $B/R = \{\{1, 2\}, \{4\}, \{1\}, \varnothing\}$. Розглянемо

множину $D = \{d, e, f\}$, $D \subset A$. Переріз відношення R за множиною D виглядає таким чином: $R(D) = R(d) \cup R(e) \cup \cup R(f) = \{\{1\}, \{4\}, \varnothing\}$.

Крім наведених операцій, до відношень застосовні звичайні теоретико-множинні операції, як об'єднання, перетин, доповнення та різниця. Розглянемо два відношення R і S, що визначені на множинах X і Y, $R \subseteq X \times Y$, $S \subseteq X \times Y$. В результаті виконання операцій $R \cup S$, $R \cap S$ і $R \backslash S$ одержуємо множини упорядкованих пар, що є відповідно об'єднанням, перетином і різницею множин R і S. Результати виконання цих операцій також є відношеннями на множинах X і Y. Доповнення відношення R позначається $\bar{R}$ і містить всі упорядковані пари множини $X \times Y$, крім тих пар, які належать R.

Запитання

1. Що називається оберненим відношенням, і як воно позначається?
2. Як одержати граф відношення, оберненого до даного?
3. Що називається композицією відношень?
4. Нехай R — деяке відношення. Що означає запис R^2, R^3, R^n (степінь відношення розглядається відносно операції композиції відношень)? Як побудувати ці відношення, і чи завжди це можливо?
5. Дайте визначення перерізу відношення R і фактор-множини за відношенням R.
6. Назвіть теоретико-множинні операції, які застосовні до відношень. Поясніть, яким чином ці операції застосовуються до відношень.

Завдання

1. Нехай R_1 і R_2 — бінарні відношення на множині $A = \{a, b, c, d\}$, де $R_1 = \{(a, a), (a, b), (b, d)\}$, $R_2 = \{(a, d), (b, c), (b, d), (c, d)\}$:
   а) побудуйте відношення $R_1 \circ R_2$, $R_2 \circ R_1$, R_1^2, R_2^3;
   б) побудуйте перерізи відношень R_1, R_2 за елементами a, d і відносно підмножини $\{a, b\}$;
   в) побудуйте фактор-множину за відношенням R_2.
2. Знайдіть відношення R^{-1}, якщо відношення R задане таким чином:
   а) $(a, b) \in R$, якщо $a, b \in N$, $a > b$;
   б) $(a, b) \in R$, якщо $a, b \in N$, a — дільник b;
   в) A — множина країн світу; $(a, b) \in R$, якщо $a, b \in A$ і країна a межує з b.

3. Нехай A — деяка множина людей, R — відношення «батько», визначене на цій множині так, що $(a, b) \in R$, якщо a — батько b. Далі S — відношення «брат-сестра», визначене на множині A за правилом $(a, b) \in S$, якщо a і b мають одну й ту ж пару батьків. Опишіть відношення $R \circ S$, $S \circ R$, R^2, R^{-1}, $S \circ R^{-1}$. Виразіть через відношення S, R відношення C — «двоюрідний брат-сестра», де $(a, b) \in C$, якщо a — двоюрідний брат або сестра b.
4. Нехай дано множини $A = \{1, 2, 3\}$, $B = \{1, 2, 3, 4\}$ і відношення R_1, $R_2 \subseteq A \times B$: $R_1 = \{(1, 2), (2, 3), (3, 4)\}$, $R_2 = \{(1, 1), (1, 2), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (3, 4)\}$. Визначте:
   а) $R_1 \cup R_2$;
   б) $R_1 \cap R_2$;
   в) $R_1 \backslash R_2$;
   г) $R_2 \backslash R_1$.
5. Нехай A — множина студентів університету, B — множина книг у бібліотеці. Нехай задано відношення R_1, $R_2 \subseteq A \times B$, такі, що $(a, b) \in R_1$, якщо студент a згідно з навчальною програмою повинен під час навчання прочитати книгу b, і $(a, b) \in R_2$, якщо студент a під час навчання вже прочитав книгу b. Дайте словесний опис відношень, що одержуються в результаті виконання операцій:
   а) $R_1 \cup R_2$;
   б) $R_1 \cap R_2$;
   в) $R_1 \backslash R_2$;
   г) $R_2 \backslash R_1$.

2.3. Властивості бінарних відношень

Рефлексивність, антирефлексивність, симетричність, асиметричність, антисиметричність, транзитивність, антитранзитивність

Кожне бінарне відношення на множині X може мати одну або кілька з названих властивостей. Ці властивості визначають вид матриці і графа відношення.

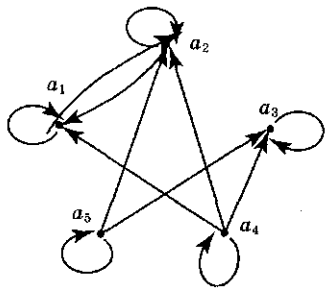
1. Рефлексивність

Відношення R на множині X називається ***рефлексивним***, якщо для будь-якого $x \in X$ має місце xRx, тобто кожний елемент $x \in X$ знаходиться у відношенні R до самого себе.

Властивість рефлексивності при заданні відношень матрицею характеризується тим, що всі діагональні елементи

матриці дорівнюють 1; при заданні відношення графом кожний елемент має петлю — дугу (x, x).

Приклад. Рефлексивним є відношення, представлене на рис. 2.12.



	a_1	a_2	a_3	a_4	a_5
a_1	1	1	0	0	0
a_2	1	1	0	0	0
a_3	0	0	1	0	0
a_4	1	1	1	1	0
a_5	0	1	1	0	1

Рис. 2.12. Приклад графа і матриці рефлексивного відношення

2. Антирефлексивність

Відношення R на множині X називається ***антирефлексивним***, якщо з x_1Rx_2 виходить, що $x_1 \neq x_2$.

Властивість антирефлексивності при заданні відношення матрицею характеризується тим, що всі діагональні елементи є нульовими. При заданні такого відношення графом жодна вершина не має петлі — немає дуг виду (x, x).

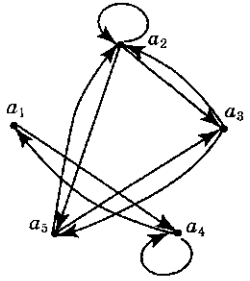
Відношення $\leq$ на множині дійсних чисел рефлексивне, відношення $<$ на множині дійсних чисел — антирефлексивне. Відношення «мати спільний дільник» на множині цілих чисел рефлексивне, відношення «бути сином» на множині людей — антирефлексивне. Відношення «бути симетричним відносно вісі X на множині точок координатної площини» не є ані рефлексивним, ані антирефлексивним: точка площини симетрична сама собі, якщо вона лежить на вісі X, і несиметрична сама собі в протилежному випадку.

3. Симетричність

Відношення R на множині X називається ***симетричним***, якщо для пари $(x_1, x_2) \in X^2$ з x_1Rx_2 виходить x_2Rx_1 (тобто для будь-якої пари R виконується або в обидві боки, або не виконується взагалі).

Матриця симетричного відношення є симетричною відносно головної діагоналі, а в графі, що задає відношення, для кожної дуги з x_i в x_k існує протилежно спрямована дуга з x_k у x_i.

Приклад. Приклад симетричного відношення зображено на рис. 2.13.



	a_1	a_2	a_3	a_4	a_5
a_1	0	0	0	1	0
a_2	0	1	1	0	1
a_3	0	1	0	0	1
a_4	1	0	0	1	0
a_5	0	1	1	0	0

Рис. 2.13. Приклад графа і матриці симетричного відношення

4. Асиметричність

Відношення R називається ***асиметричним***, якщо для пари $(x_1, x_2) \in X^2$ із x_1Rx_2 виходить, що не виконується x_2Rx_1 (тобто для будь-якої пари R або виконується в один бік, або не виконується взагалі).

5. Антисиметричність

Відношення R називається ***антисиметричним***, якщо з x_1Rx_2 і x_2Rx_1 виходить, що $x_1 = x_2$.

Приклад. Відношення «бути симетричним відносно вісі X на множині точок координатної площини» є симетричним: якщо перша точка симетрична другій, то і друга симетрична першій. Приклад антисиметричного відношення — відношення $\leq$ на множині дійсних чисел: якщо $a \leq b$ і $b \leq a$, то $a = b$. Відношення $<$ на дійсній вісі є асиметричним, оскільки якщо $a < b$, то $b < a$ не виконується.

6. Транзитивність

Відношення R називається ***транзитивним***, якщо з x_1Rx_2 і x_2Rx_3 виходить x_1Rx_3.

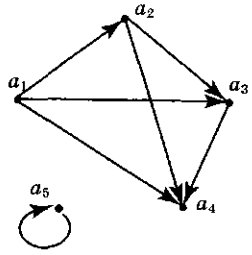
У графі, що задає транзитивне відношення R, для будь-якої пари дуг, таких, що кінець першої співпадає з початком другої, існує третя дуга, що має спільний початок з першою і спільний кінець з другою.

Приклад. Відношення $\leq$ і $<$ на множині дійсних чисел транзитивні: якщо $a \leq b$ і $b \leq c$, то $a \leq c$.

Приклад. Приклад транзитивного відношення зображено на рис. 2.14.

7. Антитранзитивність

Відношення R називається ***антитранзитивним***, якщо з x_1Rx_2 і x_2Rx_3 виходить, що не виконується x_1Rx_3.



	a_1	a_2	a_3	a_4	a_5
a_1	0	1	1	1	0
a_2	0	0	1	1	0
a_3	0	0	0	1	0
a_4	0	0	0	0	0
a_5	0	0	0	0	1

Рис. 2.14. Приклад графа і матриці транзитивного відношення

Приклад. Відношення «рівність», «бути дільником», що задані на множині цілих чисел, і відношення «жити в одному місті», що задане на множині людей, — транзитивні; відношення «бути сином» — антитранзитивне.

Запитання

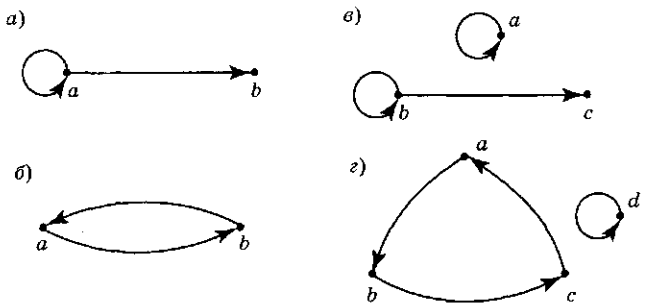
1. Чи правильне висловлювання: «Властивість відношення — це визначена умова, яка задовольняє елементи цього відношення»? Підтвердіть відповідь за допомогою прикладу.
2. Дайте визначення властивостям:
   — рефлексивності; — антирефлексивності;
   — симетричності; — асиметричності;
   — антисиметричності; — транзитивності;
   — антитранзитивності.
   Для кожного з наведених властивостей поясніть, чим граф і матриця відношень, що мають цю властивість, відрізняються від графа і матриці відношень, що не мають цієї властивості.
3. Чи може відношення мати не одну, а кілька властивостей? Наведіть приклади. Які з відомих вам властивостей можуть одночасно характеризувати відношення, а які є взаємовиключними, тобто не можуть одночасно характеризувати одне відношення?

Завдання

1. Визначте, які властивості має кожне з наведених відношень. Відношення задані на множині $A = \{1, 2, 3, 4\}$:

а) {(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)};
б) {(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)};
в) {(2, 4), (4, 2)};
г) {(1, 2), (2, 3), (3, 4)};
д) {(1, 1), (2, 2), (3, 3), (4, 4)};
е) {(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)}.

2. Визначте, які властивості має кожне з відношень, що зображені такими графами:



3. Нехай R_1 і R_2 — деякі відношення. Заповніть таблицю таким чином. У комірку таблиці помістіть «+», якщо з того, що R_1 і R_2 мають зазначену властивість, виходить, що результат операції теж має ці властивості, «−» — якщо при тому, що R_1 і R_2 мають вказану властивість, результат операції може не мати цієї властивості. Обґрунтуйте вибір кожного «+» і «−».

	$R_1 \cup R_2$	$R_1 \cap R_2$	$\overline{R_1}$	$R_1 \setminus R_2$	$R_1 \circ R_2$	$R^n, (n \in N)$	R^{-1}
Рефлексивність							
Антирефлексивність							
Симетричність							
Асиметричність							
Антисиметричність							
Транзитивність							
Антитранзитивність							

4. Визначте, які властивості мають такі відношення, що задані на деякій множині людей. Нехай $(a, b) \in R$, якщо:
а) a вище на зріст, ніж b;
б) a і b народилися в один день;
в) a є родичем b;
г) a знайомий з b.

5. Знайдіть помилку в доведенні такого твердження (це твердження не правильне).
   Твердження: Якщо деяке відношення R симетричне і транзитивне, то воно є також і рефлексивним.
   Доведення: Візьмемо пару елементів (a, b), таку, що $(a, b) \in R$, тоді і $(b, a) \in R$ (відношення симетричне). Якщо $(a, b) \in R$ і $(b, a) \in R$, то за властивістю транзитивності $(a, a) \in R$, отже відношення рефлексивне.
6. Наведіть приклад відношення, яке:
   а) є симетричним і антисиметричним;
   б) не є ані симетричним, ані антисиметричним.
7. Скільки існує відношень, що задані на множині A, $|A| = n$, які мають такі властивості:
   а) симетричність;
   б) антисиметричність;
   в) асиметричність;
   г) антирефлексивність;
   д) симетричність і рефлексивність.
8. Нехай R — рефлексивне і транзитивне відношення. Чи правильно, що $R^n = R$ для всіх $n \in N$.
9. Нехай R_1 і R_2 — деякі відношення, причому $R_1 \supseteq R_2$. Покажіть, що якщо R_1 антисиметричне, то R_2 теж антисиметричне.
10. Складіть алгоритм визначення властивостей відношення. Вхідними даними може бути матриця відношення або список елементів відношення.

2.4. Відношення еквівалентності, порядку, толерантності

Відношення еквівалентності, класи еквівалентності, шлях у графі, частковий (нестрогий) порядок, строгий порядок, лінійний порядок, порівнянні і непорівнянні елементи, відношення толерантності

Розглянемо деякі найчастіше використовувані класи бінарних відношень.

Визначення

Бінарне відношення, що має властивості рефлексивності, симетричності і транзитивності, називається ***відношенням еквівалентності*** (позначається ~).

Нехай задана множина A і відношення еквівалентності, що визначене на цій множині: $R \subseteq A \times A$. Елементи $a, b \in A$, для яких виконується aRb, називаються ***еквівалентними***. Важлива властивість будь-якого відношення еквівалентності R,

визначеного на множині A, полягає в тому, що воно розбиває множину A на неперетинні підмножини, які називаються ***класами еквівалентності***. У випадку скінченної множини A розбиття її на класи еквівалентності відбувається таким чином. Нехай на множині A задане відношення еквівалентності R. Виберемо елемент $a_1 \in A$ і утворимо клас C_1, що складається з усіх елементів $y \in A$, для яких виконується відношення a_1Ry. Клас C_1 може складатися тільки з одного елемента a_1, якщо не існує інших елементів y, таких, що a_1Ry. Зауважимо, що через рефлексивність відношення еквівалентності завжди виконується a_1Ra_1. Якщо $C_1 \neq A$, то виберемо з A елемент a_2, що не входить до класу C_1, і утворимо клас C_2, який складається з елементів $y \in A$, для яких виконується відношення a_2Ry. Якщо $(C_1 \cup C_2) \neq A$, то виберемо з A елемент a_3, що не входить до класів C_1 і C_2, і утворимо клас C_3. Будемо продовжувати побудову класів доти, доки в A не залишиться жодного елемента, що не входить до одного з класів C_i. Вийде система класів C_1, C_2, ..., C_n. Ця система класів називається системою класів еквівалентності і має такі властивості:

а) класи попарно не перетинаються;

б) будь-які два елементи з одного класу еквівалентні;

в) будь-які два елементи з різних класів не еквівалентні.

Приклад. Відношення рівності «=» на будь-якій множині чисел є відношенням еквівалентності. Відношення «вчитися в одному класі» на множині учнів школи є відношенням еквівалентності і розбиває всю множину учнів школи на окремі класи. Відношення «мати однакове ім'я» на визначеній множині людей є відношенням еквівалентності і розбиває всю множину людей на класи еквівалентності — групи людей з однаковими іменами.

Визначення

Відношення називається ***відношенням часткового порядку*** (позначається $\leq$), якщо воно:

а) рефлексивне ($a \leq a$);

б) антисиметричне (якщо $a \leq b$ і $b \leq a$, то $a = b$);

в) транзитивне (якщо $a \leq b$ і $b \leq c$, то $a \leq c$).

Якщо на множині задане відношення часткового порядку, то ця множина називається ***частково упорядкованою***.

Існує спосіб наглядного зображення скінченної частково упорядкованої множини, який називають ***діаграмою Хассе***. Кожний елемент $a \in A$ позначають точкою, і будь-яку пару точок, що відповідають елементам $a \in A$ і $b \in A$, з'єднують стрілкою, що йде з точки a в точку b, тоді і тільки тоді, коли $(a \le b)$, $(a \ne b)$, і не існує $c \in A$ такого, що $a \le c \le b$, і елемент c відрізняється і від a, і від b.

Приклад. Прикладом відношення часткового порядку є відношення включення множин, справедливе для деяких пар підмножин фіксованої множини. На рис. 2.15 зображено граф відношення включення на множині 2^A всіх підмножин триелементної множини $A = \{a, b, c\}$.

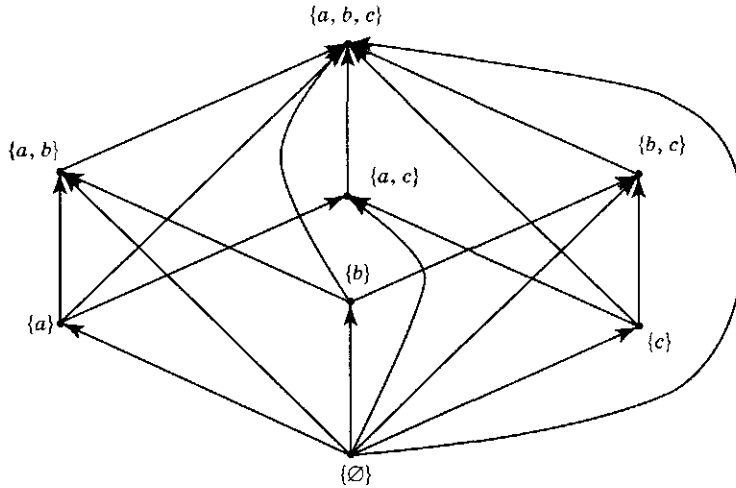


Рис. 2.15. Граф відношення включення множин

Із властивістю транзитивності пов'язане поняття шляху в графі.

Визначення

Шлях у графі відношення з вершини a до вершини b — це послідовність дуг (a, x_1), (x_1, x_2), (x_2, x_3),..., (x_{n-1}, b), $n \ge 1$. Число дуг n називається ***довжиною шляху***.

Використовуючи поняття шляху в графі, можна показати, що властивість транзитивності виконується у відношенні, що

задане графом, якщо будь-які дві вершини, між якими існує шлях, з'єднані дугою.

Часто при зображенні графа відношення порядку виключають дуги, які можна добудувати, використовуючи властивість транзитивності, тобто, якщо на графі існує шлях з a до b, то дугу (a, b) не зображують. Крім того, розташовують вершини так, щоб спрямованість всіх дуг була знизу-нагору. Тоді спрямованість дуг можна не вказувати. Одержана таким чином діаграма достатня для визначення відношення порядку. Так, відношення включення множин з графом на рис. 2.15 можна зобразити за допомогою діаграми Хассе, як зображено на рис. 2.16.

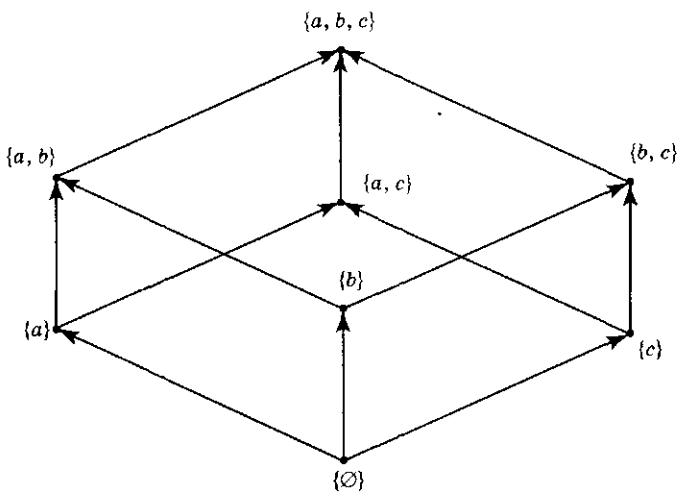


Рис. 2.16. Діаграма Хассе відношення включення множин

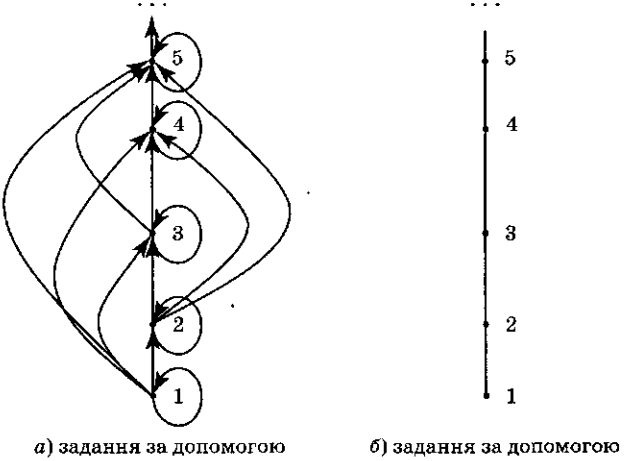
З розглянутого прикладу видно, що не всі вершини графа відношення з'єднані між собою шляхами. Наприклад, не існує шляху між $\{a, b\}$ и $\{a, c\}$.

Визначення

Елементи a і b називаються ***порівнянними*** у відношенні часткового порядку R, якщо виконується хоча б одне із співвідношень aRb або bRa. Множина A, на якій задане відношення часткового порядку R і для якої всілякі два елементи цієї множини порівнянні, називається ***лінійно упорядкованою*** або ***повністю упорядкованою***.

Повністю упорядкована множина відрізняється від частково упорядкованої тим, що в частково упорядкованій множині A можуть бути присутніми елементи, з яких можна складати непорівнянні пари.

Приклад. Множина N натуральних чисел повністю упорядкована відносно природної операції порівняння $\leq$, яка істинна на деякій підмножині R пар (a, b) декартова квадрата N^2. Будь-які два елементи $a, b \in N$ порівнянні за R, оскільки хоча б одна з нерівностей $a \leq b$, $b \leq a$ істинна. На рис. 2.17 розглянуту множину N задано за допомогою графа відношень і діаграм Хассе.



а) задання за допомогою графа відношень

б) задання за допомогою діаграми Хассе

Рис. 2.17. Лінійно упорядкована множина натуральних чисел

Відношення часткового порядку, що є рефлексивним, антисиметричним і транзитивним, називається також ***відношенням нестрогого порядку***. На відміну від нього, ***відношення строгого порядку*** (позначається $<$) визначається такими властивостями:

а) антирефлексивність (якщо $a < b$, то $a \neq b$);

б) асиметричність (якщо $a < b$, то не правильне $b < a$);

в) транзитивність (якщо $a < b$ і $b < c$, то $a < c$).

Таким чином, якщо у відношенні нестрогого порядку властивість антисиметричності замінити асиметричністю, то одержимо строгий порядок.

Визначення

Відношення називається ***відношенням толерантності***, якщо воно: а) рефлексивне; б) симетричне і в) антитранзитивне.

Толерантність зображує собою формальне уявлення інтуїтивного поняття схожості. Схожість двох об'єктів не залежить від того, в якому порядку вони порівнюються, в цьому виявляється властивість симетричності. В той же час, якщо один об'єкт схожий з другим, а другий схожий з третім, то це не означає, що перший і третій об'єкти схожі, тобто властивість транзитивності може не виконуватися.

Розглянемо відношення толерантності на прикладі популярної задачі «перетворення мухи на слона» як схожості між чотирибуквеними словами, якщо вони відрізняються однією буквою. Одержимо: муха — мура — тура — тара — кара — каре — кафе — кафр — каюр — каюк — крюк — крок — срок — сток — стон — слон.

Запитання

1. Дайте визначення відношення еквівалентності.
2. Яким чином відбувається розбиття множини на класи еквівалентності? Які елементи називаються еквівалентними?
3. Які властивості характерні для класів еквівалентності?
4. Наведіть приклади відношень еквівалентності та одержуваних при цьому класів еквівалентності.
5. Дайте визначення відношення часткового порядку. Як воно позначається? Наведіть приклад.
6. Що таке шлях у графі? Як поняття шляху застосовується для дослідження графів відношення порядку?
7. Яка множина називається лінійно упорядкованою? Що таке порівнянні і непорівнянні елементи?
8. Чим відрізняється відношення повного порядку від відношення часткового порядку? Дайте визначення відношення строгого порядку.
9. Дайте визначення відношення толерантності. Наведіть приклад.

Завдання

1. Доведіть, що будь-яке відношення еквівалентності на множині A розбиває цю множину на неперетинні підмножини — класи еквівалентності.
2. Які з наведених наборів підмножин множини $A = \{1, 2, 3, 4, 5, 6\}$ можуть бути розбиттями цієї множини на класи еквівалентності:

а) {1, 2}, {2, 3, 4}, {4, 5, 6}; в) {2, 4, 6}, {1, 3, 5};
б) {1}, {2, 3, 6}, {4, 5}; г) {1, 4, 5}, {2, 6}.

3. Нехай R_1 і R_2 — відношення еквівалентності. Визначте, чи є такі відношення відношеннями еквівалентності:
а) $R_1 \cup R_2$; б) $R_1 \cap R_2$; в) $\bar{R}_1$.
4. Нехай R_1 и R_2 — відношення часткового порядку. Визначте, чи є такі відношення відношеннями часткового порядку:
а) $R_1 \cup R_2$; б) $R_1 \cap R_2$; в) $\bar{R}_1$.
5. Визначте, які відношення еквівалентності розбивають множину на найбільше та найменше число класів еквівалентності.
6. Нехай R — відношення часткового порядку. Доведіть, що R^{-1} — теж відношення часткового порядку.
7. Які з наведених матриць задають відношення часткового порядку? Для знаходження відповіді дослідіть графи відношень.

	a	*b*	*c*	*d*
a	1	0	1	0
b	0	1	1	0
c	0	0	1	1
d	1	1	0	1

	a	*b*	*c*
a	1	0	1
b	1	1	0
c	0	0	1

	a	*b*	*c*
a	1	0	0
b	0	1	0
c	1	0	1

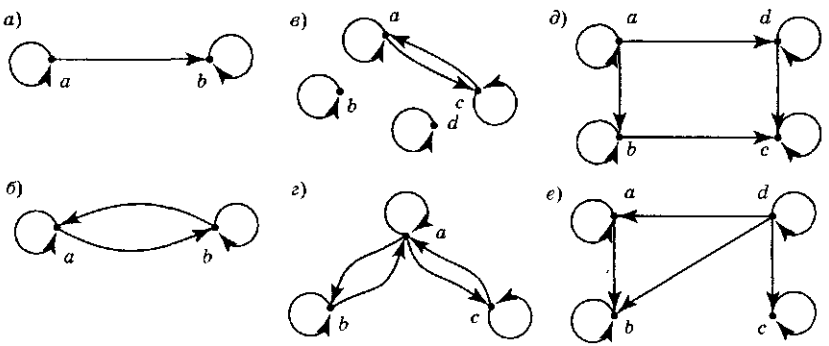
8. Які з наведених матриць задають відношення еквівалентності? Для знаходження відповіді дослідіть графи відношень.

	a	*b*	*c*	*d*
a	1	0	1	0
b	0	1	0	0
c	1	0	1	0
d	0	1	0	1

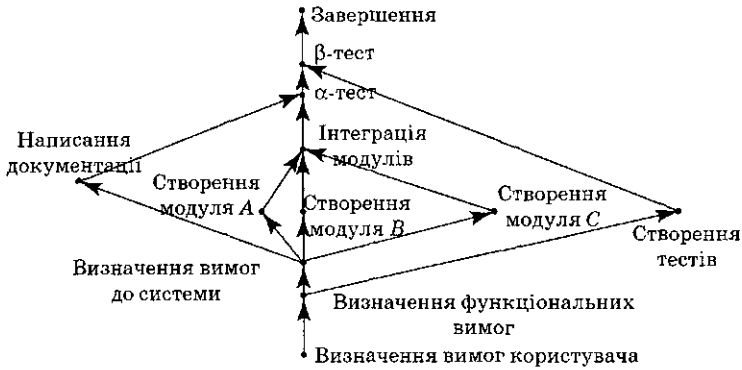
	a	*b*	*c*
a	1	1	1
b	0	1	1
c	1	1	1

	a	*b*	*c*	*d*
a	1	1	1	0
b	1	1	1	0
c	1	1	1	0
d	0	0	0	1

9. Які з наведених відношень на множині людей є відношеннями толерантності, які — відношеннями еквівалентності?
а) *a* і *b* однакового віку;
б) *a* і *b* мають спільних батьків;
в) *a* і *b* знайомі;
г) *a* і *b* розмовляють однією мовою.
10. Які з наведених відношень, задані графами, є відношеннями порядку, які — відношеннями еквівалентності?



11. Складіть можливу послідовність виконання задач для створення деякого програмного продукту, якщо діаграма, що задає частковий порядок на множині задач, така:



12. Складіть алгоритм, що визначає за заданою матрицею відношення, чи є відношення частковим або строгим порядком, еквівалентністю або толерантністю.

2.5. Функціональні відношення

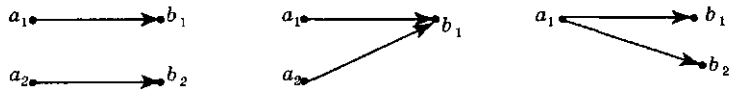
Функціональне відношення, області визначення і значень, відображення, сюр'єкція, ін'єкція, бієкція

Функціональна залежність між змінними або функція є частковим випадком відношення.

Визначення

Відношення R між множинами X і Y ($R \subseteq X \times Y$) є ***функціональним***, якщо всі його елементи (упорядковані пари) різні за першим елементом: кожному $x \in X$ або відповідає тільки один елемент $y \in Y$, такий, що xRy, або такого елемента y взагалі не існує.

Матриця функціонального відношення, що задане на скінченних множинах X і Y, містить не більше однієї одиниці в кожному рядку. Якщо функціональне відношення задано у вигляді графа, то з кожної вершини, що зображує першу координату, виходить не більше однієї дуги. Приклади функціонального і нефункціонального відношень зображено на рис. 2.18.



а) функціональне відношення *б*) функціональне відношення *в*) нефункціональне відношення

Рис. 2.18. Приклади функціонального і нефункціонального відношень

Приклад. Нехай є множина A кроликів і множина B кліток. Нехай R — відношення розміщення кроликів по клітках — «Кролики — Клітки». Для будь-якого розміщення кроликів по клітках відношення R буде функціональним, оскільки кожному кролику може відповідати тільки одна клітка. Позначимо кроликів буквами, а клітки — номерами. Нехай $A = \{a, b\}$, $B = \{1, 2, 3\}$. На рис. 2.19 зображено функціональні відношення $R_1 = \{(a, 1), (b, 3)\}$ і $R_2 = \{(a, 1), (b, 1)\}$.

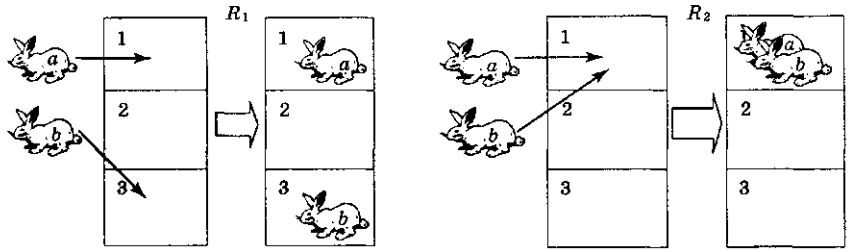


Рис. 2.19. Приклади функціональних відношень «Кролики — Клітки»

Прикладом нефункціонального відношення є відношення $R_3 = \{(a, 1), (a, 2), (b, 3)\}$. Воно може бути представлено в термінах розміщення кроликів по клітках, як показано на рис. 2.20, де кролик a займає одночасно дві клітки 1, 2, завдяки зламаній перегородці.

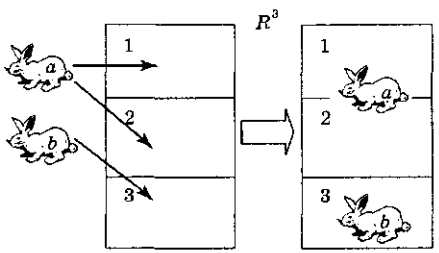


Рис. 2.20. Приклад відношення «Кролики — Клітки», що не є функцією

Визначення

Нехай R — деяке відношення, $R \subseteq X \times Y$. ***Областю визначення*** відношення R називається множина D_R, що складається з усіх елементів множини X, які зв'язані відношенням R з елементами множини Y: $D_R \subseteq X$, $D_R = \{x: \exists\, y \in Y, (x, y) \in R\}$. ***Областю значень*** відношення R називається множина $\Re_R$, що складається з усіх елементів множини Y, які зв'язані відношенням R з елементами множини X: $\Re_R \subseteq Y$, $\Re_R = \{y: \exists\, x \in X, (x, y) \in R\}$.

Визначення

Нехай F — функціональне відношення, $F \subseteq X \times Y$. Відповідність $x \to y$ від першого до другого елементу кожної пари $(x, y) \in F$ відношення F називається ***функцією*** f або ***відображенням*** f ***множини*** D_F в Y і позначається як $f: D_F \to Y$, або як $D_F \xrightarrow{f} Y$.

Множина D_F називається областю визначення або задання функції (відображення) f і позначається як $D_f (\equiv D_F)$. Стверджують також, що функція f діє з X у Y і визначена на підмножині D_f з X. Якщо $D_f = X (= D_F)$, то пишуть $f: X \to Y$ і говорять, що задано відображення $X \to Y$.

Якщо множина $A \subseteq X$, то через $f(A) = \{y \in Y: y = f(x), \forall x \in A\}$ позначається образ множини A. Множина $f(X) \subseteq Y$ називається ***образом*** або ***областю значень*** відображення f і позначається через $\Re_f = f(X)$.

Якщо множина $B \subseteq Y$, то множина $f^{-1}(B) = \{x \in X: f(x) \in B\}$ називається ***прообразом*** множини B відносно відображення f.

Запис $y = f(x)$ функціональної залежності змінної y від змінної x (що називається ***аргументом функції***) менш інформативний, ніж запис у вигляді відображення $f: D_f \to Y$, оскільки повинна доповнюватися інформацією про область визначення — підмножину припустимих значень аргументу x.

Графіком функції (відображення) $f: X \to Y$ називається сукупність «двовимірних» точок (x, y) виду $(x, f(x))$ у декартовому добутку $X \times Y$. Саме в цьому розумінні синусоїда є графіком функції $\sin x$, парабола (що охоплює вісь y) — графіком квадратичної функції x^2 тощо. Таким чином, якщо $F \subset X \times Y$ — вихідне функціональне відношення, що породжує функцію (відображення) f, то F в точності є ***графік функції*** f. Не слід змішувати

поняття «графік функції f» і «граф відношення F»: граф за допомогою дуг зі стрілками описує дію відображення f на кожному значенні аргументу x (рис. 2.21).

Види відображень

1. Функція $f\colon X \to Y$ називається ***сюр'єктивним відображенням***, якщо $\Re_f = Y$.

На графі, що зображує ***сюр'єктивне*** відображення $X \to Y$, з будь-якої вершини $x \in X$ виходить точно одна дуга, а до будь-якої вершини, що зображує елемент множини Y, заходить не менше однієї дуги (рис. 2.21).

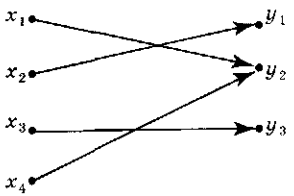


Рис. 2.21. Приклад сюр'єктивного відображення

2. Функція $f\colon X \to Y$ називається ***ін'єктивним відображенням***, якщо з $x_1 \neq x_2$ виходить $f(x_1) \neq f(x_2)$.

На графі, що зображує ін'єктивне відображення $X \to Y$, з будь-якої вершини $x \in X$ виходить точно одна дуга, а до будь-якої вершини, що зображує елемент множини Y, заходить не більше однієї дуги (рис. 2.22).

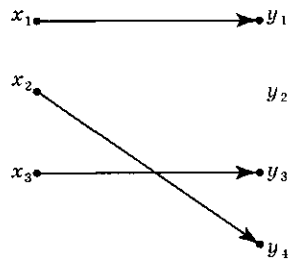


Рис. 2.22. Приклад ін'єктивного відображення

Якщо $f\colon X \to Y$ — ін'єктивне відображення і $F = \{(x,\ f(x)),\ \forall x \in X\}$ — відповідне функціональне відношення ($F \subset X \times Y$), то обернене відношення $F^{-1} = \{(y,\ x),\ \forall y \in \Re_f\}$ також є функціональним. Функція $x = f^{-1}(y)$, $f^{-1}\colon \Re_f \to X$, що задається відношенням F^{-1}, має властивості: $f^{-1}(f(x)) = x$, $\forall x \in X$; $f(f^{-1}(y)) = y$, $\forall y \in \Re_f$ і називається ***оберненою*** до функції f.

3. Функція $f\colon X \to Y$ називається ***бієктивним відображенням***, якщо вона сюр'єктивна та ін'єктивна.

На графі, що зображує бієктивне відображення $X \to Y$ скінченних множин, з будь-якої вершин $x \in X$ виходить точно одна дуга, а до будь-якої вершини $y \in Y$ заходить одна і тільки одна дуга (рис. 2.23). Таким чином, бієктивне відображення $f\colon X \to Y$ здійснює взаємно однозначне відображення між множинами X і Y, тому $X \sim Y$, $|X| = |Y|$ (див. п. 1.6).

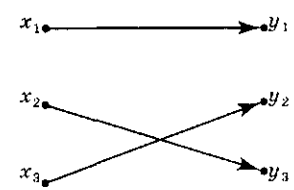


Рис. 2.23. Приклад бієктивного відображення

Властивість бієктивності забезпечує існування оберненого відображення. Це означає, що якщо f: $X \to Y$ бієктивне, то існує обернене відображення f^{-1}: $Y \to X$, причому $D_{f^{-1}} = Y$.

Проілюструємо загальне поняття функції і сюр'єктивні, ін'єктивні та бієктивні відображення на прикладах функціональних відношень типу «Кролики — Клітки». Відношення R_1, R_2 на рис. 2.19 визначають відображення f_1: $A \to B$, f_2: $A \to B$ відповідно, причому f_1 ін'єктивне, f_2 — ні. Обидва відображення не є сюр'єктивними, оскільки після розміщення кроликів залишаються вільні клітки. На схемі, що зображена на рис. 2.24, множина із шести кроликів $X = \{a, b, c, d, e, f\}$ розміщується у множині кліток $Y = \{1, 2, 3, 4\}$.

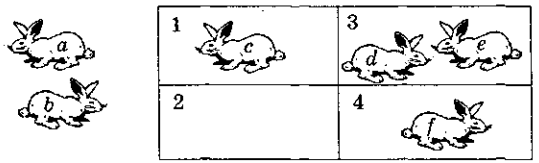


Рис. 2.24. Функція «Кролики — Клітки», що не є відображенням $X \to Y$

Ця схема розміщення відповідає відношенню R, граф якого зображено на рис. 2.25.

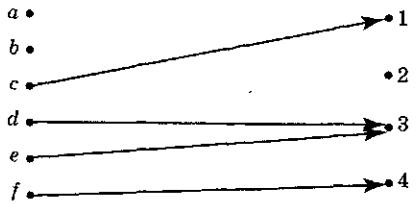


Рис. 2.25. Граф відношення $R = \{(c, 1), (d, 3), (e, 3), (f, 4)\}$

Це відношення R є функціональним, але функція f, яку задає R, визначена не на всій множині X, оскільки не всі кролики розміщені в клітках. Областю визначення функції f є множина $D_R = \{c, d, e, f\}$ і областю значень $\Re_R = \{1, 3, 4\}$.

Розглянемо два інших варіанти розміщення шести кроликів X за чотирма клітками Y, що вказані на рис. 2.26 і 2.27.

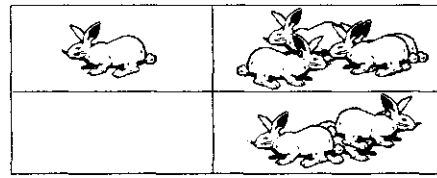


Рис. 2.26. Відображення «Кролики — Клітки»

R є сюр'єктивним відображенням, якщо всі клітки зайняті.

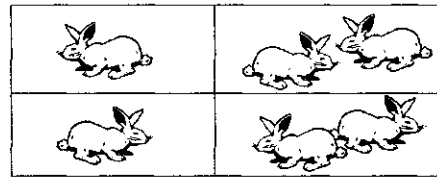


Рис. 2.27. Сюр'єктивне відображення «Кролики — Клітки»; $|X| = 6, |Y| = 4$

Для реалізації ін'єктивного розміщення шести кроликів ($|X| = 6$) необхідно збільшити кількість кліток так, щоб вона була не менше за кількість кроликів. Відповідне відображення f: $X \to Y$ буде ін'єктивним відображенням, якщо в кожній клітці розміщується не більше одного кролика, але можуть залишатися і порожні клітки, як показано на рис. 2.28.

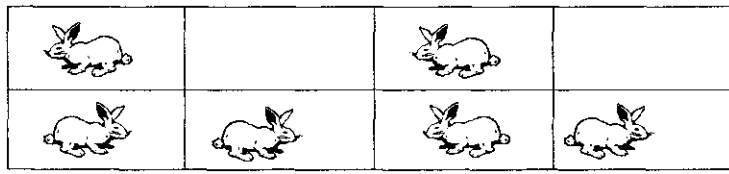


Рис. 2.28. Ін'єктивне відображення «Кролики — Клітки»; $|X| = 6, |Y| = 8$

R є бієктивним відображенням, якщо в кожній клітці знаходиться точно по одному кролику (рис. 2.29).

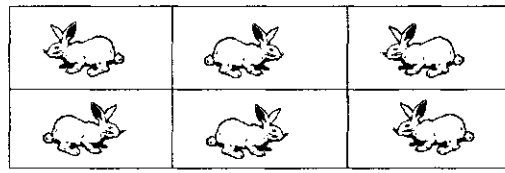


Рис. 2.29. Бієктивне відображення «Кролики — Клітки»; $|X| = 6, |Y| = 6$

Запитання

1. Яке відношення називається функціональним?
2. Що таке область визначення і область значення відношення?
3. Яке відношення $R \subset X \times Y$ задає відображення $X \to Y$?
4. Дайте визначення сюр'єктивного, ін'єктивного і бієктивного відображень. Чим характеризується граф кожного з цих відображень?
5. Який вид відображення можна назвати «багато до одного», який — «один до одного»?
6. Який вид відображення множини A на множину B ставить у відповідність кожному елементу множини A один і тільки один елемент множини B так, що кожний елемент множини B поставлено у відповідність одному і тільки одному елементу множини A?

Завдання

1. Які з наведених відношень $R \subset A^2$, що задані на множині $A = \{-20, -19, \ldots, 0, 1, \ldots, 19, 20\}$, є функціональними? Для функціональних відношень вкажіть відповідні функції:

   а) $(a, b) \in R$, якщо $a = b^2$;

   б) $(a, b) \in R$, якщо $a^2 = b$;

   в) $(a, b) \in R$, якщо $a \leq b$;

   г) $(a, b) \in R$, якщо $ab = 6$;

   д) $(a, b) \in R$, якщо $a = b^3$;

   е) $(a, b) \in R$, якщо $b = a^3$;

   ж) $(a, b) \in R$, якщо $a = b^4$;

   з) $(a, b) \in R$, якщо $a = 1/b$.

2. Які з функцій f, що знайдено при розв'язанні задачі 1, є відображеннями виду $f\colon A \to A$?

3. Нехай задано множини A, B, C і відношення $R \subseteq A \times B$ і $S \subseteq B \times C$. Покажіть на прикладах, що наведені висновки не правильні:

   а) якщо R — функціональне відношення, то $S \circ R$ — функціональне відношення;

   б) якщо R задає відображення $A \to B$, то $S \circ R$ — відображення $A \to C$;

   в) якщо S визначає сюр'єкцію $B \to C$, то $S \circ R$ — сюр'єкцію $A \to C$;

   г) якщо S визначає ін'єкцію $B \to C$, то $S \circ R$ — ін'єкцію $A \to C$.

4. Нехай задано множини A, B, C і відношення $R \subseteq A \times B$ і $S \subseteq B \times C$. Доведіть, що:

   а) якщо R і S — функціональні відношення, то $S \circ R$ — функціональне відношення;

б) якщо R і S визначають відображення $A \to B$ і $B \to C$ відповідно, то $S \circ R$ визначає відображення $A \to C$;

в) якщо R і S визначають сюр'єкції, то $S \circ R$ — також сюр'єкцію;

г) якщо R і S визначають ін'єкції, то $S \circ R$ — також ін'єкцію;

д) якщо R і S визначають бієкції, то $S \circ R$ — також бієкцію.

5. Доведіть, що якщо f — ін'єктивна функція, то існує функція f^{-1}.
6. Нехай $F \subset X \times Y$ визначає сюр'єктивну функцію. Чи визначає F^{-1} відображення $Y \to X$?
7. Знайдіть область визначення і область значень таких функцій:

   а) функція, яка кожному невід'ємному цілому числу x, $x \leq n$, ставить у відповідність його останню цифру;

   б) функція, яка рядку бітів довжиною $\leq n$ ставить у відповідність кількість одиниць у цьому рядку;

   в) функція, яка рядку бітів довжиною $\leq n$ ставить у відповідність кількість бітів, що залишилися при розбитті цього рядку на байти;

   г) функція, яка для цілого додатного числа x, $x \leq n$, знаходить найбільший квадрат, що не перевищує це число.
8. Нехай A і B — скінченні множини, що містять відповідно n і m елементів: $|A| = n$, $|B| = m$. Визначте співвідношення між n і m, якщо:

   а) існує ін'єктивне відображення з A в B;

   б) існує сюр'єктивне відображення з A в B;

   в) існує бієктивне відображення з A в B.

2.6. Реляційна модель даних

Кортеж, домен, атрибут. Реляційна алгебра та операції в ній: об'єднання, перетин, різниця, прямий добуток, обмеження, проекція, натуральне з'єднання, ділення

Одна з найбільш важливих галузей застосування комп'ютерів на сьогодні — це збереження та обробка великого обсягу інформації, що має складну внутрішню структуру. Ця інформація повинна бути представлена в пам'яті ЕОМ так, щоб її було легко здобувати, поповнювати і перетворювати. Ми розглянемо один з методів побудови баз даних,

розроблений наприкінці 60-х років і широко використовуваний зараз. Оскільки зображення інформації у вигляді таблиць є найбільш зручним та звичним для людини, його було взято за основу і показано, що будь-яке зображення даних зводиться до сукупності двовимірних таблиць. З математичної точки зору табличне зображення даних легко формулюється в термінах відношень, і тому до нього застосовано апарат теорії множин. Така модель даних називається ***реляційною*** (relation — відношення). Для роботи з ***реляційною моделлю*** було створено ***реляційну алгебру***. Кожна операція цієї алгебри використовує одну або кілька таблиць (відношень) як її операндів і продукує в результаті нову таблицю, тобто дозволяє «розрізати» та «склеювати» таблиці. Основна ідея реляційної алгебри: оскільки відношення є множинами, то засоби маніпулювання відношеннями можуть базуватися на таких традиційних теоретико-множинних операціях, як об'єднання, перетин, різниця, декартів добуток, доповнених деякими спеціальними операціями.

Розглянемо приклади реляційного зображення даних. Нехай є інформація про студентів університету, що зображена в таблиці 2.1.

Таблиця 2.1

Прізвище	Ініціали	Група
Алексєєв	І. А.	КН-01
Андрєєва	О. П.	ПМ-01
Баранов	Н. П.	ПМ-01
Бикова	Н. А.	КН-01
Волков	В. В.	ПМ-01

Ця інформація являє собою деяке відношення R_1, що задане на трьох множинах — множині прізвищ, множині ініціалів і множині груп. Відношення R_1 можна задати списком його елементів:

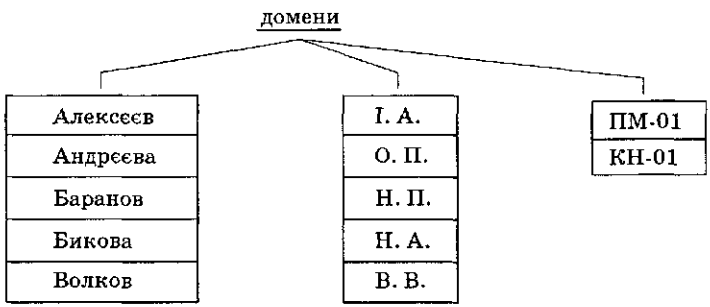
R_1 = {(Алексєєв, І. А., КН-01), (Андрєєва, О. П., ПМ-01), ..., (Волков, В. В., ПМ-01)}.

Розглянемо термінологію, що застосовується під час побудови подібних таблиць даних. Елементи відношення, як (Алексєєв, І. А., КН-01) і (Андрєєва, О. П., ПМ-01), що відповідають

рядкам таблиці, називаються ***кортежами***. Множини або області даних, на яких визначено відношення, такі, як множина прізвищ, множина ініціалів і множина груп, що відповідають стовпцям таблиці, називаються ***доменами***. Найменування стовпців таблиці називають ***атрибутами***. Відношенню привласнюють ім'я, наприклад, СТУДЕНТ 1. ***Схемою відношення*** є список атрибутів, наприклад, схемою відношення СТУДЕНТ 1 є список: (Прізвище, Ініціали, Група). Зобразимо це відношення на рис. 2.30.

Для зміни вмісту таких таблиць у базах даних використовують операції додавання, видалення кортежів і зміни значення атрибутів. Це прості операції із заповнення таблиці.

Розглянемо теоретико-множинні операції, що застосовуються для перетворення відношень у базі даних, використовуючи терміни реляційної алгебри. Відношення, до яких застосовується операція, будемо називати відношеннями-***операндами***.



СТУДЕНТ 1

Прізвище	Ініціали	Група
Алексєєв	І. А.	КН-01
Андрєєва	О. П.	ПМ-01
Баранов	Н. П.	ПМ-01
Бикова	Н. А.	КН-01
Волков	В. В.	ПМ-01

атрибути

кортежі

Рис. 2.30. Відношення СТУДЕНТ 1

Теоретико-множинні операції реляційної алгебри

Об'єднання відношень. При виконанні операції об'єднання двох відношень (∪) одержуємо відношення, що включає всі кортежі, які входять хоча б в одне з відношень-операндів.

Перетин відношень. При виконанні операції перетину двох відношень (∩) одержуємо відношення, що включає тільки ті кортежі, які входять в обидва відношення-операнда.

Різниця відношень. Відношення, яке є різницею (\) двох відношень, включає всі кортежі, що входять до відношення — перший операнд, такі, що жодне з них не входить до відношення, яке є другим операндом.

Ці операції мають сенс для відношень, що визначені на однакових доменах. Розглянемо відношення СТУДЕНТ 1 і СТУДЕНТ 2 (рис. 2.31).

СТУДЕНТ 1

Прізвище	Ініціали	Група
Алексєєв	І. А.	КН-01
Андрєєва	О. П.	ПМ-01
Баранов	Н. П.	ПМ-01
Бикова	Н. А.	КН-01
Волков	В. В.	ПМ-01

СТУДЕНТ 2

Прізвище	Ініціали	Група
Алексєєв	І. А.	КН-01
Бикова	Н. А.	КН-01
Дроздов	І. К.	КН-01
Зайцев	О. Н.	ПМ-01
Кузнєцова	Е. В.	КН-01

Рис. 2.31. Відношення СТУДЕНТ 1 і СТУДЕНТ 2

Застосувавши операції ∩, ∪ і \ до відношень СТУДЕНТ 1 і СТУДЕНТ 2, одержимо такі результати (рис. 2.32):

СТУДЕНТ 1 ∩ СТУДЕНТ 2

Прізвище	Ініціали	Група
Алексєєв	І. А.	КН-01
Бикова	Н. А.	КН-01

ВСІ СТУДЕНТИ = СТУДЕНТ 1 ∪ СТУДЕНТ 2

Прізвище	Ініціали	Група
Алексєєв	І. А.	КН-01
Андрєєва	О. П.	ПМ-01
Баранов	Н. П.	ПМ-01
Бикова	Н. А.	КН-01
Волков	В. В.	ПМ-01
Дроздов	І. К.	КН-01
Зайцев	О. Н.	ПМ-01
Кузнєцова	Е. В.	КН-01

СТУДЕНТ 1 \ СТУДЕНТ 2

Прізвище	Ініціали	Група
Андрєєва	О. П.	ПМ-01
Баранов	Н. П.	ПМ-01
Волков	В. В.	ПМ-01

Рис. 2.32. Результат виконання операцій ∩, ∪, \

Схематичне виконання операцій ∩, ∪, \ зображено на рис. 2.33.

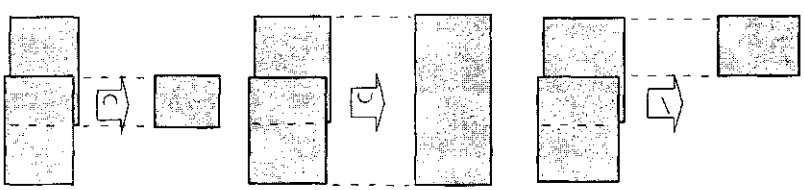


Рис. 2.33. Схема виконання операцій ∩, ∪, \

Розглянемо операцію, яка є декартовим або прямим добутком відношень.

Прямий добуток відношень. При виконанні прямого добутку (×) двох відношень одержуємо відношення, множина кортежів якого є декартовим добутком множин кортежів першого і другого операндів. Розглянемо відношення СТУДЕНТ 1 (рис. 2.30) і КУРС (рис. 2.34).

КУРС

Навч. рік	Курс
2001–2002	1
2002–2003	2

Рис. 2.34. Відношення КУРС

Виконаємо прямий добуток відношення СТУДЕНТ 1 на відношення КУРС (рис. 2.35).

СТУДЕНТ 1 × КУРС

Прізвище	Ініціали	Група	Навч. рік	Курс
Алексєєв	І. А.	КН-01	2001–2002	1
Алексєєв	І. А.	КН-01	2002–2003	2
Андрєєва	О. П.	ПМ-01	2001–2002	1
Андрєєва	О. П.	ПМ-01	2002–2003	2
Баранов	Н. П.	ПМ-01	2001–2002	1
Баранов	Н. П.	ПМ-01	2002–2003	2
Бикова	Н. А.	КН-01	2001–2002	1
Бикова	Н. А.	КН-01	2002–2003	2
Волков	В. В.	ПМ-01	2001–2002	1
Волков	В. В.	ПМ-01	2002–2003	2

Рис. 2.35. Результат виконання прямого добутку відношень СТУДЕНТ 1 і КУРС

Схематично виконання операції × зображено на рис. 2.36.

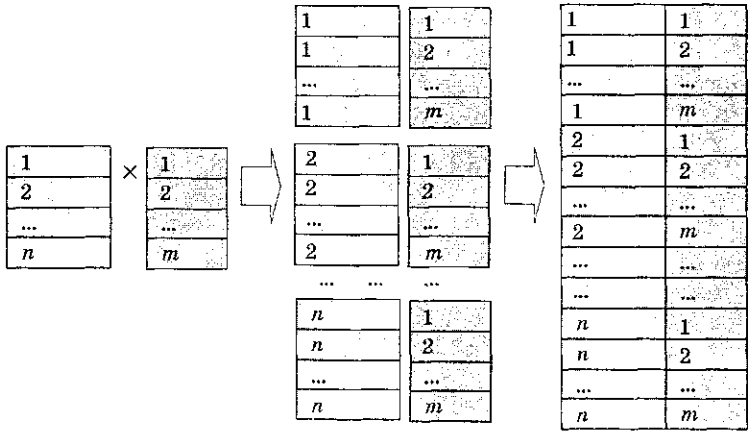


Рис. 2.36. Схема виконання операції прямого добутку відношень

Тепер перейдемо до розгляду спеціальних операцій, що застосовані у базах даних.

Спеціальні операції реляційної алгебри

Обмеження відношення. Результатом обмеження відношення (σ) за деяким атрибутом або атрибутами, є відношення, що складається в точності з тих кортежів, які задовольняють умову σ.

Виконаємо обмеження відношення ВСІ СТУДЕНТИ (рис. 2.32) за атрибутом Група = КН-01. Назвемо результат — СТУДЕНТ КН-01. В результаті одержимо відношення, що містить тільки кортежі, в яких значення атрибута Група дорівнює КН-01 (рис. 2.37).

$\sigma_{\text{Група = КН-01}}$ *(ВСІ СТУДЕНТИ) = СТУДЕНТ КН-01*

Прізвище	Ініціали	Група
Алексєєв	І. А.	КН-01
Бикова	Н. А.	КН-01
Дроздов	І. К.	КН-01
Кузнєцова	Е. В.	КН-01

Рис. 2.37. Обмеження σ за значенням КН-01 атрибута Група

Проекція відношення. Під час виконання проекції (π) відношення на заданий набір його атрибутів відношення результат виходить шляхом видалення з відношення-операнда атрибутів, не вказаних у заданому наборі.

Виконаємо проекцію відношення СТУДЕНТ 1 КУРС за атрибутами Група, Навч. рік, Курс (рис. 2.38).

$\pi_{\text{Група, Навч. рік, Курс}}$ *(СТУДЕНТ 1 КУРС)*

Група	Навч. рік	Курс
КН-01	2001–2002	1
КН-01	2002–2003	2
ПМ-01	2001–2002	1
ПМ-01	2002–2003	2

Рис. 2.38. Проекція за атрибутами Група, Навч. рік, Курс

Схематично виконання операції π зображено на рис. 2.39.

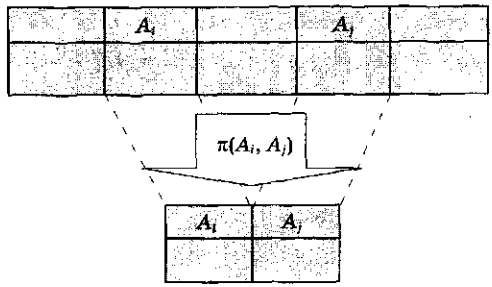


Рис. 2.39. Схема виконання операції проекції за A_i, A_j

Виконаємо проекцію відношення СТУДЕНТ КН-01 за атрибутами Прізвище, Ініціали (рис. 2.40).

$$ПРІЗВИЩЕ\ СТ\ КН\text{-}01 = \pi_{\text{Прізвище, Ініціали}}(СТУДЕНТ\ КН\text{-}01)$$

Прізвище	Ініціали
Алексєєв	І. А.
Бикова	Н. А.
Дроздов	І. К.
Кузнєцова	Е. В.

Рис. 2.40. Проекція СТУДЕНТ КН-01 за атрибутами Прізвище, Ініціали

Натуральне з'єднання відношень. У натуральному з'єднанні двох відношень ($\rhd\!\lhd$) утворюється результуюче відношення, кортежі якого є з'єднанням кортежів першого і другого відношень, якщо значення спільних атрибутів співпадає.

Схематично виконання операції $\rhd\!\lhd$ зображено на рис. 2.41.

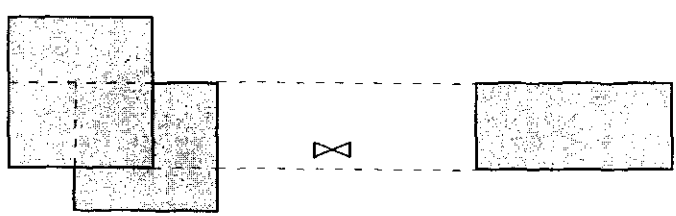


Рис. 2.41. Схема виконання операції з'єднання відношень ($\rhd\!\lhd$)

Розглянемо відношення НОМЕР (рис. 2.42).

НОМЕР

Прізвище	Ініціали	Залікова книжка №	Студ. №
Алексєєв	І. А.	11197	784567
Андрєєва	О. П.	11215	784565
Баранов	Н. П.	11213	784598
Бикова	Н. А.	11216	784588
Волков	В. В.	11217	784599
Дроздов	І. К.	11193	784611
Зайцев	О. Н.	11191	784601
Кузнєцова	Е. В.	11195	785587

Рис. 2.42. Відношення НОМЕР

Виконаємо натуральне з'єднання відношень СТУДЕНТ КН-01 і НОМЕР (рис. 2.43).

СТУДЕНТ КН-01 $\bowtie$ *НОМЕР*

Прізвище	Ініціали	Група	Залікова книжка №	Студ. №
Алексєєв	І. А.	КН-01	11197	784567
Бикова	Н. А.	КН-01	11216	784588
Дроздов	І. К.	КН-01	11193	784611
Кузнєцова	Е. В.	КН-01	11195	785587

Рис. 2.43. Природне з'єднання СТУДЕНТ КН-01 і НОМЕР

Ділення відношень. Операція ділення відношень (÷) відбувається таким чином. Відношення — дільник повинно мати набір атрибутів, що включено до набору атрибутів діленого. Результуюче відношення містить ті атрибути діленого, які не присутні в дільнику. Значення цих атрибутів беруться з тих кортежів діленого, які включають до себе кортежі дільника.

Схематично виконання операції ділення відношень зображено на рис. 2.44.

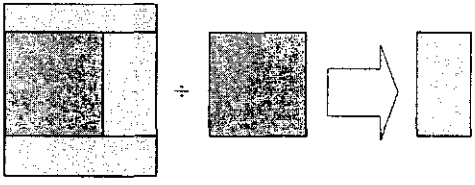


Рис. 2.44. Схема виконання операції ділення відношень (÷)

Тепер виконаємо ділення відношення НОМЕР (рис. 2.42) на відношення ПРІЗВИЩЕ СТ КН-01 (рис. 2.40), використовуючи схему рис. 2.44.

В результаті ділення ми одержуємо таблицю номерів студентів групи КН-01 (рис. 2.45).

НОМЕР ÷ ПРІЗВИЩЕ СТ КН-01

Залікова книжка №	Студ №
11197	784567
11216	784588
11193	784611
11195	785587

Рис. 2.45. Ділення НОМЕР на ПРІЗВИЩЕ СТ КН-01

Крім розглянутих нами операцій реляційної алгебри, до набору операцій бази даних звичайно включається операція привласнювання, що дозволяє зберегти результати обчислення алгебраїчних виразів, операція перейменування атрибутів, яка надає змогу коректно сформувати заголовок (схему) результуючого відношення та інші корисні операції.

Запитання

1. Що являє собою реляційний метод побудови баз даних?
2. Що називається кортежем, атрибутом, доменом? Наведіть приклад.
3. Наведіть теоретико-множинні операції, що застосовуються при роботі з реляційними базами даних. Поясніть принцип їх виконання.
4. Назвіть спеціальні реляційні операції, які вам відомі. Поясніть принцип виконання кожної з них.
5. Як реалізовані розглянуті в цьому розділі операції над відношеннями у СУБД, з якою ви працюєте?

Завдання

Повернемося до відношення, яке розглядалося у р. 1 і було складене для деякої дизайнерської фірми у м. Харкові.

Фірма-продукція-1

Назва	Місто	Продукція
ЧПП «Orion»	Одеса	меблі
ТОВ «День»	Харків	світильники
ЧКП «Sit»	Одеса	меблі
ЧКП «Sit»	Одеса	світильники
ТОВ «House»	Харків	світильники
ТОВ «House»	Харків	матеріали для ремонту

Припустимо, існує ще одне відношення, складене деякою фірмою у м. Києві.

Фірма-продукція-2

Назва	Місто	Продукція
ТОВ «Днепр»	Київ	матеріали для ремонту
ЧКП «Virta»	Київ	сантехніка
ЧКП «Virta»	Київ	плитка
ЧПП «Orion»	Одеса	меблі
ЧКП «Софія»	Київ	сантехніка
ЧКП «Софія»	Київ	плитка
ТОВ «День»	Харків	світильники

1. Виконайте операції об'єднання, перетину і різниці над відношеннями Фірма-продукція-1 і Фірма-продукція-2. Який сенс може мати застосування цих операцій? Назвіть таблицю, одержану в результаті виконання операції об'єднання Фірма-продукція-3.
2. Виконайте з'єднання відношень Фірма-продукція-3 і Фірма-телефон. Відношення Фірма-телефон таке:

Фірма-телефон

Назва	Телефон
ЧПП «Orion»	784-556, 784-557
ТОВ «День»	145-134
ЧКП «Sit»	356-777
ТОВ «House»	122-890, 126-713
ТОВ «Днепр»	555-78-12, 555-77-09
ЧКП «Virta»	233-77-77, 233-77-78
ЧКП «Софія»	550-09-09

Назвіть відношення, одержане в результаті операції з'єднання Фірма-продукція-телефон.

3. Виберіть два варіанти виконання операції проекції до відношення Фірма-продукція-телефон. За якими атрибутами ви робили проекції?
4. Виконайте обмеження відношення Фірма-продукція-телефон за умови:
   а) Продукція = меблі;
   б) Продукція = матеріали для ремонту, Місто = Харків;
   в) Місто = Одеса.
5. Як змінюється число кортежів і (або) атрибутів відношення під час виконання вивчених нами операцій. Розгляньте кожну операцію окремо.

розділ 3

Алгебраїчні структури

3.1. Алгебраїчні операції та їх властивості

Унарна операція, бінарна операція, n-арна операція, операнд, записи infix, prefix, postfix, таблиця Келі, комутативність, асоціативність, дистрибутивність, одиниця, обернений елемент, операції додавання та множення за модулем

Поняття алгебраїчної структури включає визначену множину об'єктів та операцій над цими об'єктами. Оскільки практично в будь-якій задачі обробки даних за допомогою комп'ютера виділяється множина самих даних і операції, які застосовні до цих даних, очевидно, що при цьому формуються визначені алгебраїчні структури. Ми вже знайомі з алгебраїчними структурами на множині натуральних, цілих і дійсних чисел, на множинах та відношеннях. Розглянемо такі алгебраїчні структури, як півгрупи, моноїди і групи, що, зокрема, використовуються для перетворення рядків символів і беруть участь у формуванні більш складних структур — кілець і полів. Ці поняття є базовими для загальної та лінійної алгебри, використовуються під час роботи з матрицями, при кодуванні інформації та обробці даних.

Визначення

Операцією на множині S називається функція f, яка є відображенням виду $S^n \to S$, $n \in N$, де S^n — декартів добуток $S \times S \times ... \times S$, в який S входить n разів.

У цьому визначенні є два важливих моменти. По-перше, оскільки операція є функцією, то результат застосування операції визначено однозначно. Тому даний упорядкований набір з n елементів множини S функція f переводить тільки в один елемент із S. По-друге, операція замкнена (див. п. 4.8) на S у тому розумінні, що область визначення та область значень операції лежать у S^n і S відповідно.

Стверджують, що операція $S^n \to S$ має ***порядок n*** або є ***n-арною операцією***. Частіше зустрічається ситуація, коли порядок дорівнює 1 або 2. Операції виду $S \to S$ називають ***унарними***, а операції $S^2 \to S$ називають ***бінарними***. Елементи упорядкованого набору з n елементів в області визначення S^n називають ***операндами***. Операції звичайно позначають символами, що називають ***операторами***. У випадку унарних операцій звичайно символ оператора ставлять перед або над операндом.

Приклади. Прикладами унарних операцій є операція зміни знаку (–) на множині дійсних чисел R(–2,678; – 56), операція зведення до степеня (наприклад, до квадрату) на множині R: 56^2, 7^2. В алгебрі множин прикладом унарної операції є операція доповнення множин. Бінарними операціями на множині дійсних чисел R є арифметичні операції — додавання, віднімання, множення, ділення (+, –, *, /). В алгебрі множин бінарними є операції — об'єднання ($\cup$), перетин ($\cap$), різниця ($\setminus$).

Операції записують одним з трьох способів. У першому випадку оператор ставиться між операндами (***infix***), у другому — перед операндами (***prefix***) і у третьому — після операндів (***postfix***). Розглянемо три варіанти запису бінарної операції арифметичного виразу $a + b$.

infix: $a + b$,

prefix: $+ab$,

postfix: $ab+$.

Відповідно до більшості математичних текстів ми будемо використовувати позначення *infix*. Форми запису *postfix* і *prefix* мають ту перевагу, що не потребують дужок при визначенні порядку обчислень складних виразів, і це робить їх особливо зручними для автоматичної обробки. Вони часто використовуються для представлення виразів у пам'яті комп'ютера. Ми розглянемо їх більш докладніше на прикладі *postfix*.

Алгоритм обчислення значень виразу, що записаний у формі *postfix*, виглядає так:

1) При перегляді запису зліва направо виконується перша знайдена операція, якій безпосередньо передує достатня для неї кількість операндів.
2) На місці виконаної операції і використаних для цього операндів у рядок записується результат виконання операції.
3) Повертаємося до кроку 1.

Приклад. Нехай є вираз, який у стандартній звичній для нас *infix*-формі виглядає так:

$$1 + 2 * 3 + (4 + 5 * (6 + 7)).$$

Результат переведення його до *postfix* буде таким:

$$123 * + 4567 + * + +.$$

Обчислимо тепер значення виразу, використовуючи наведений алгоритм:

$$1\ \underline{2\ 3\ *}\ +\ 4\ 5\ 6\ 7\ +\ *\ +\ +\ =\ \underline{1\ 6\ +}\ 4\ 5\ 6\ 7\ +\ *\ +\ +\ =$$
$$=\ 7\ 4\ 5\ \underline{6\ 7\ +}\ *\ +\ +\ =\ 7\ 4\ \underline{5\ 13\ *}\ +\ +\ =\ 7\ \underline{4\ 65\ +}\ +\ =$$
$$=\ \underline{7\ 69\ +}\ =\ 76.$$

Аналогічно записуються вирази з нечисловими змінними в алгебраїчній формі. Наведені пари виразів записані у формах *infix* і *postfix* відповідно:

а) $(a + b) * c + d + e * f + g$, $a\ b + c * d + e\ f * + g +$;
б) $a + (b * (c + d) + e) * f + g$, $a\ b\ c\ d + * e + f * + g +$.

Крім стандартних відомих нам операцій (наприклад, $+$, $-$, $*$), існує багато інших. Ми будемо використовувати символи $\otimes$ і $\oplus$ для позначення абстрактних бінарних операцій. Інакше кажучи, символи $\otimes$ і $\oplus$ використовуються як змінні для позначення будь-яких операцій. Бінарні операції, визначені на скінченних множинах, зручніше задавати за допомогою таблиць. Таблиця (табл. 3.1), що задає деяку бінарну операцію $\otimes$ на деякій множині A, називається ***таблицею Келі***, її рядки та стовпці нумеруються елементами множини A, а елементом таблиці, що стоїть на перетині рядку a_i і стовпця a_j, є елемент $a_k = a_i \otimes a_j$.

Таблиця 3.1.
Таблиця Келі для $\otimes$

$\otimes$	a	b	c
a	a	a	b
b	b	a	c
c	a	b	b

Приклад. Нехай операція $\otimes$ визначена на множині $\{a, b, c\}$ за допомогою таблиці 3.1.

Отже, $a \otimes b = a$, $b \otimes b = a$, $c \otimes b = b$, ...

Очевидно, що використання таблиць має велике значення, оскільки деякі операції, з якими доводиться мати справу в комп'ютерній математиці, не придатні для словесного завдання.

Наведемо важливі властивості, які можуть мати операції.

Визначення

Нехай дано множину A, на якій визначено деяку бінарну операцію $\otimes$. Якщо $a \otimes b = b \otimes a$ для всіх $a, b \in A$, то стверджують, що бінарна операція $\otimes$ на множині A має властивість — ***комутативність.***

Якщо $(a \otimes b) \otimes c = a \otimes (b \otimes c)$ для всіх $a, b, c \in A$, то стверджують, що бінарна операція $\otimes$ на множині A має властивість — ***асоціативність.***

Нехай на множині A визначено дві бінарні операції $\otimes$ і $\oplus$. Якщо для всіх $a, b, c \in A$ виконується $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$, то стверджують, що операція $\otimes$ має властивість — ***дистрибутивність*** відносно операції $\oplus$.

Зауважимо, що у визначенні асоціативності порядок операндів a, b і c збережено (операція може бути некомутативною) і використано круглі дужки, щоб вказати порядок виконання операцій. Таким чином, вираз $(a \otimes b) \otimes c$ потребує, щоб спочатку обчислювалося $a \otimes b$ і потім результат цього (скажімо, x) брав участь в операції $x \otimes c$ як перший операнд. Якщо операція асоціативна, то порядок обчислень несуттєвий і, отже, дужки не потребуються.

Приклад. Звичайна бінарна операція додавання (+) на множині дійсних чисел R комутативна і асоціативна, а операція віднімання (–) — некомутативна і неасоціативна, тобто

$$a + b = b + a, \text{ але } a - b \neq b - a;$$
$$(a + b) + c = a + (b + c), \text{ але } (a - b) - c \neq a - (b - c).$$

Крім того, на множині дійсних чисел R множення дистрибутивне відносно додавання, а додавання не дистрибутивне відносно множення, тобто

$$a * (b + c) = a * b + a * c, \; (a + b) * c = a * c + b * c,$$
$$\text{але } a + (b * c) \neq (a + b) * (a + c).$$

Для розв'язання рівнянь відносно кожної операції у множині-носії алгебраїчної структури виділяється особливий елемент, що називається одиничним елементом.

Визначення

Якщо для бінарної операції $\otimes$ на множині A існує елемент $e \in A$ такий, що для всіх $a \in A$

$$e \otimes a = a \otimes e = a,$$

тоді e називається ***одиницею*** відносно до операції $\otimes$.
Нехай $\otimes$ — операція на A з одиницею e і елементи $x, y \in A$ задовольняють рівності

$$x \otimes y = e = y \otimes x.$$

Тоді y називається ***оберненим елементом*** до x відносно до операції $\otimes$, і x називається ***оберненим елементом*** до y відносно операції $\otimes$.

Іноді розрізняють ліві та праві одиниці ($e_{\text{лів.}} \otimes a = a$ або $a \otimes e_{\text{прав.}} = a$ для будь-якого $a \in A$) і ліві та праві обернені елементи, однак у більшості випадків одиниці є двосторонніми, як у нашому визначені.

У випадках, коли бінарна операція вважається аналогічною множенню (*), одиничний елемент позначається 1, а обернений до елемента x елемент записується у вигляді x^{-1}. Коли бінарна операція вважається аналогічною додаванню (+), одиничний елемент позначається 0, а обернений до елемента x елемент записується у вигляді $-x$. Будемо також позначати обернений елемент до x як x'.

Наведемо приклади одиниць і обернених елементів. На множині дійсних чисел R правою одиницею відносно до віднімання та одиницею відносно додавання є 0, оскільки

$a - 0 = a$, але $0 - a \neq a$, якщо $a \neq 0$;

$a + 0 = a$ і $0 + a = a$ для всіх a.

В алгебрі множин для операції об'єднання $\cup$ одиничним елементом є порожня множина $\varnothing$, для операції перетину одиницею є універсальна множина U.

Для подальшого необхідно визначити операції додавання та множення за модулем n на множині цілих чисел.

Визначення

Нехай n — довільне натурально число. ***Додавання за модулем n*** цілих чисел a і b називається алгебраїчна операція, результатом якої є решта від ділення суми $a + b$ на n. ***Множенням за модулем n*** чисел a і b називається алгебраїчна операція, результатом якої є решта від ділення добутку $a * b$ на n. Ці операції (позначимо їх $\oplus_n$, $\otimes_n$) визначені на множині цілих невід'ємних чисел Z_+:

$a \oplus_n b = c$, так, що $a + b = k * n + c$, $0 \le c < n$;
$a, b, k \in Z_+$

$a \otimes_n b = d$, так, що $a * b = f * n + d$, $0 \le d < n$;
$a, b, f \in Z_+$

Областю значень цих операцій є множина цілих невід'ємних чисел, менших за n, позначимо її Z_n, $Z_n = \{0, 1, \ldots, n-1\}$.

Часто використовується позначення

$$a + b \equiv c \pmod{n},$$
$$a \times b \equiv d \pmod{n}$$

для додавання та множення за модулем n.

Приклад. Наведемо приклади додавання та множення за модулем n.

$2 \oplus_3 2 =$ Зал. $(4/3) = 1$, $\quad 2 \otimes_3 2 =$ Зал. $(4/3) = 1$,

$2 \oplus_4 2 =$ Зал. $(4/4) = 0$, $\quad 2 \otimes_4 2 =$ Зал. $(4/4) = 0$,

$7 \oplus_{10} 8 =$ Зал. $(15/10) = 5$, $\quad 7 \otimes_{10} 8 =$ Зал. $(56/10) = 6$,

$7 \oplus_{12} 8 =$ Зал. $(15/12) = 3$, $\quad 7 \otimes_{12} 8 =$ Зал. $(56/12) = 8$.

Запитання

1. Дайте визначення операції і наведіть приклади операцій, що задані на різних множинах.
2. Які з наведених дій, що задані на множині натуральних чисел N, можна назвати операціями:
   а) збільшення на 1;
   б) знаходження числа, кратного даному;
   в) додавання за модулем 5;
   г) знаходження числа, меншого, ніж дане;
   д) знаходження найбільшого спільного дільника;
   е) знаходження числа, яке при діленні на 7 дає у залишку 6.
3. Що називається порядком операції? Наведіть приклади унарної, бінарної, *n-арної* операції.

4. Що називається оператором? Наведіть приклад.
5. Що являють собою форми запису *infix*, *prefix* і *postfix*? Наведіть приклади. Чому форма запису *infix* менш зручна для автоматичної обробки?
6. Які операції можна та зручно задавати за допомогою таблиць. Чому?
7. Дайте визначення властивостей асоціативності, дистрибутивності та комутативності операції. Наведіть приклади.
8. При виконанні якої умови елемент e називається одиницею відносно деякої операції $\oplus$?
9. Що таке обернений елемент до деякого елементу x відносно деякої операції $\oplus$?
10. Чи є існування одиниці необхідною і (або) достатньою умовою для існування обернених елементів?

Завдання

1. Визначте, чи замкнені дані операції відносно даних множин. Заповніть таблицю позначеннями «З» — замкнена або «Н» — не замкнена.

	$x + y$	$x * y$	$x - y$	$\|x - y\|$	$\max(x, y)$	$\min(x, y)$	$-x$	$\|x\|$
Z (цілі числа)								
N (натуральні числа)								
$\{x \mid 0 \le x \le 10, x \in Z\}$								
$\{x \mid -5 \le x \le 5, x \in Z\}$								
$\{x \mid -10 \le x \le 0, x \in Z\}$								
$\{2x \mid 0 \le x \le 10, x \in Z\}$								

2. Визначте, чи мають зазначені операції наведені властивості. Заповніть таблицю позначеннями «Так» — має і «Ні» — не має. Операції розглядаються на множині дійсних чисел.

	$x + y$	$x * y$	$x - y$	$\|x - y\|$	$\max(x, y)$	$\min(x, y)$
Асоціативна						
Комутативна						
Має одиницю						

3. Опишіть алгоритм, який визначає, асоціативна бінарна операція або ні. Вхідними даними є таблиця операції (операція задана на скінченній множині).
4. Опишіть алгоритм, який визначає, комутативна бінарна операція $\otimes$ відносно $\oplus$ чи ні. Вхідними даними є таблиці операцій $\otimes$ і $\oplus$ (операції задані на скінченній множині).

3.2. Поняття алгебраїчної структури

Алгебраїчна структура, підструктура, гомоморфізм, ізоморфізм

Визначення

Алгебраїчною структурою (коротко — структурою) називається множина разом із заданими операціями, визначеними і замкненими (див. п. 4.8) на цій множині. Ця множина називається ***носієм*** алгебраїчної структури.

Приклад. Алгебраїчна структура з операцією додавання на множині N натуральних чисел позначається $(N, +)$.

Приклад. Множина $Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$ разом із звичайною операцією додавання $(+)$ не буде алгебраїчною структурою, оскільки результат виконання операції може не належати множині Z_7, наприклад, $6 + 3 = 9$, $9 \notin Z_7$. Але $(Z_7, \oplus_7)$ є алгебраїчною структурою, оскільки область значень операції $\oplus_7$ лежить у Z_7.

Визначення

Структура $S' = (A', \oplus')$ є ***підструктурою*** алгебраїчної структури $S = (A, \oplus)$, якщо:
1. $A' \subset A$.
2. $\oplus'$ і $\oplus$ операції одного порядку і звуження операції $\oplus$ на підмножині A' співпадає з операцією $\oplus'$ (наприклад, для бінарних операцій $a \oplus b = a \oplus' b$ для всіх $a, b \in A'$).

Очевидно, що найбільшою підструктурою структури S є сама структура S. У деяких випадках інших підструктур може не бути.

Приклади. Нехай E — множина парних натуральних чисел, тоді $(E, +)$ буде підструктурою структури $(N, +)$, де N — множина натуральних чисел.

Структура $(\{0, 1\}, *)$ є підструктурою структури $(Z, *)$, де Z — множина цілих чисел.

Відношення між алгебраїчними структурами можуть бути не тільки такі, що включають (структура — підструктура). Можливі й інші відношення, що дозволяють здійснювати перехід від структури до структури, з втратою деякої інформації або без втрати інформації.

Визначення

Нехай задано дві структури $(A, \otimes)$, $(C, \oplus)$ з операціями $\otimes$, $\oplus$ одного порядку n. Відображення $\varphi: A \to C$ називається ***гомоморфізмом*** із структури $(A, \otimes)$ у структуру $(C, \oplus)$, якщо воно переставлене з операціями у такому розумінні:

$$\varphi \cdot \otimes = \oplus \cdot \vec{\varphi},$$

де відображення $\vec{\varphi}: A^n \to C^n$ діє за правилом

$$\vec{\varphi}(a_1, a_2, \ldots, a_n) = (\varphi(a_1), \varphi(a_2), \ldots, \varphi(a_n)), \forall a_i \in A.$$

Для ***бінарних*** операцій ($n = 2$), зокрема,

$$\varphi(x \otimes y) = \varphi(x) \oplus \varphi(y) \text{ для будь-яких } x, y \in A.$$

Графічне визначення гомоморфізму для випадку бінарних операцій пояснює рис. 3.1.

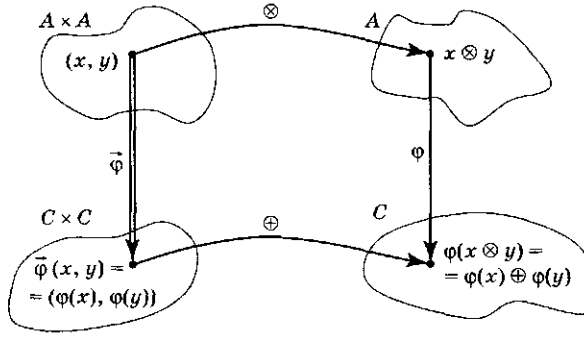


Рис. 3.1. Гомоморфізм φ: $\varphi(x \otimes y) = \varphi(x) \oplus \varphi(y)$

Якщо спростити наведену ілюстрацію, то одержимо комутативну діаграму, яка пов'язує окремі елементи множин та зображена на рис. 3.2.

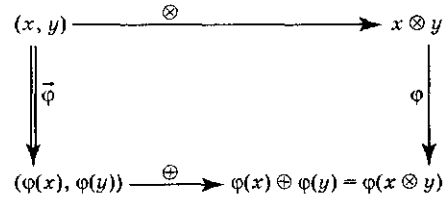


Рис. 3.2. Зв'язок між окремими елементами множин при гомоморфізмі φ

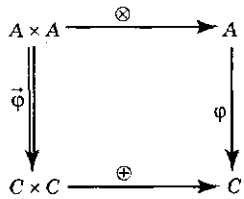


Рис. 3.3. Комутативна діаграма, що зображує гомоморфізм φ

Часто такі діаграми зображують у більш спрощеному вигляді, вказуючи тільки необхідні множини, як показано на рис. 3.3.

Подібні діаграми часто використовуються для зображення зв'язків між структурами, вони називаються ***комутативними***, оскільки показують можливість переходу до результату різними способами (за напрямком стрілок).

Приклад. Нехай задано відображення θ: $Z_+ \to Z_{10}$, що переводить будь-яке ціле невід'ємне число у решту від ділення цього числа на 10. Тоді

$$\theta(20) = 0,\ \theta(17) = 7,\ldots$$

Якщо $(Z_+, +)$ і $(Z_{10}, \oplus_{10})$ структури з операцією звичайного додавання +, що визначена на Z_+ і додаванням за модулем 10 на Z_{10}, то θ є гомоморфізмом з першої структури у другу. Наприклад,

$$\theta(24 + 38) = \theta(62) = 2,$$
$$\theta(24) \oplus_{10} \theta(38) = 4 \oplus_{10} 8 = 2.$$

Одержання даного результату двома різними способами можна проілюструвати комутативною діаграмою (рис. 3.4):

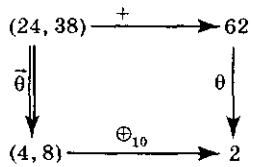


Рис. 3.4. Комутативна діаграма. Дія гомоморфізму θ з $(Z_+, +)$ в $(Z_{10}, \oplus_{10})$ для елементів 24 і 38 множини Z

В загальному випадку для гомоморфізму θ: $(Z_+, +) \to (Z_{10}, \oplus_{10})$ комутативна діаграма буде виглядати так, як це зображено на рис. 3.5.

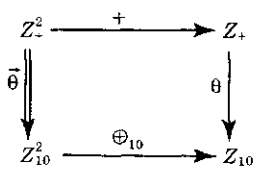


Рис. 3.5. Комутативна діаграма дії гомоморфізму θ з $(Z_+, +)$ в $(Z_{10}, \oplus_{10})$

Визначення

Гомоморфізм, який є бієкцією, називають ***ізоморфізмом***. Якщо існує ізоморфізм між двома структурами, то говорять, що вони ***ізоморфні одна одній***.

Таким чином, для будь-якого ізоморфізму φ існує обернене відображення φ^{-1}, також взаємно однозначне. Якщо існує ізоморфізм структури S у структуру Q, то існує й ізоморфізм Q у S.

Відношення ізоморфізму — це *відношення еквівалентності* на множині алгебраїчних структур, тому ізоморфізм розбиває множину всіх алгебраїчних структур на класи еквівалентності. Використовуючи ізоморфізм, можна здійснювати еквівалентні перетворення алгебраїчних структур. Якщо алгебраїчні структури S і Q *ізоморфні*, то елементи і операції Q можна перейменувати так, що Q співпадає з S. Будь-яке співвідношення у структурі S зберігається у будь-якій ізоморфній їй структурі Q. Це дозволяє, одержавши певні співвідношення у структурі S, автоматично поширити їх на всі структури, що ізоморфні S. Тому алгебраїчні структури часто розглядаються з точністю до ізоморфізму, тобто розглядаються класи еквівалентності за відношенням ізоморфізму.

Приклад. Розглянемо спосіб вимірювання довжини у дюймах та сантиметрах. Якщо додати бінарну операцію додавання, то одержимо дві структури: (inch, +), (см, +). Визначимо ізоморфізм γ: x (см) = 2,54 * x (inch).

Як показано на діаграмі (рис. 3.6), ми можемо провести обчислення (додавання) у дюймах, а потім перевести результат у сантиметри, і також можливо спочатку зобразити ті ж операнди в сантиметрах і потім провести додавання. В обох випадках буде одержано один і той же результат. Наприклад, нехай необхідно визначити довжину d деякого виробу, що складається з частин a і b. Виміривши частини a і b у дюймах, одержали, що a = 10″, b = 15″. Знайдемо d двома способами:

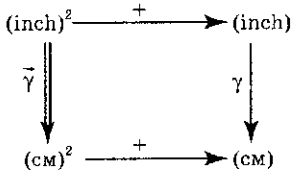


Рис. 3.6. Ізоморфізм γ з (inch, +) у (см, +)

d = 10″ + 15″ = 25″, 2,54 * 25″ = 63,5 см;

d = 10″ * 2,54 + 15″ * 2,54 = 25,4 см + 38,1 см = 63,5 см.

Для цього прикладу комутативна діаграма виглядає таким чином (рис. 3.7):

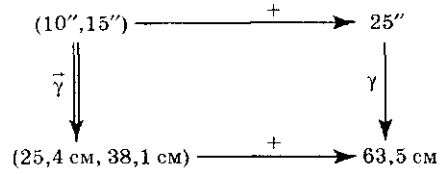


Рис. 3.7. Перетворення окремих елементів при ізоморфізмі γ з (inch, +) у (см, +)

Відображення є ізоморфізмом, оскільки γ — однозначна відповідність та існує обернене відображення γ′: x (inch) = = 0,39 * x (см).

Наведемо ще два приклади ізоморфізмів.

Приклад. Нехай Z — множина всіх цілих чисел, Z_{2N} — множина всіх парних чисел. Алгебри $(Z; +)$ і $(Z_{2N}; +)$ ізоморфні. Ізоморфізмом є відображення φ_{2N}: $n \to 2n$ для всіх $n \in Z$.

Приклад. Ізоморфізмом між алгебраїчними структурами $(R_+, *)$ і $(R, +)$, де R_+ — додатна підмножина R, є відображенням $a \to \log a$. Умова гомоморфізму $\varphi(x \otimes y) = \varphi(x) \oplus \varphi(y)$ у цьому випадку має вигляд $\log(a * b) = \log a + \log b$.

Запитання

1. Дайте визначення алгебраїчної структури.
2. Що називається підструктурою алгебраїчної структури? Яким відношенням пов'язані множини-носії алгебраїчної структури та її підструктури? Наведіть приклади підструктур.
3. Дайте визначення гомоморфізму та ізоморфізму. Чим вони відрізняються?
4. За допомогою чого здійснюється розбиття алгебраїчних структур на класи еквівалентності?
5. Наведіть приклади ізоморфних алгебраїчних структур.
6. Наведіть приклад гомоморфізму, що не є ізоморфізмом. Поясніть, чому обране вами відношення не ізоморфізм.

Завдання

1. Розглянемо алгебраїчні структури $(\{a, b, c, d\}, \oplus)$ і $(\{a, b, c\}, \otimes)$, де операції $\oplus$ і $\otimes$ задані таблицями:

$\oplus$	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

$\otimes$	a	b	c
a	a	a	c
b	a	b	c
c	c	c	b

Для кожної алгебраїчної структури визначте:

а) чи комутативна операція?

б) чи асоціативна операція?

в) чи існує одиниця для кожної операції, якщо існує, то який це елемент?

г) якщо одиниця існує, визначте, які елементи мають обернені?

2. Визначте алгебраїчну структуру $(S, \oplus)$, таку, що для будь-якої підмножини $T \subset S$ правильно, що $(T, \oplus)$ — підструктура структури $(S, \oplus)$.

3. Визначте, чи коректно задана алгебраїчна структура. Якщо так, визначте, чи комутативна операція і, якщо можливо, знайдіть одиницю і обернені елементи. Тут N — множина натуральних чисел, Z — множина цілих чисел, R_+ — додатна підмножина множини дійсних чисел:

   а) $(N, \otimes)$; $x \otimes y = x - y$;

   б) $(Z, \otimes)$; $x \otimes y = x * y - 1$;

   в) $(N, \otimes)$; $x \otimes y = max(x, y)$;

   г) $(R_+, \otimes)$; $x \otimes y = x/y$.

4. Доведіть, що:
   а) дві алгебраїчні структури не можуть бути ізоморфними, якщо множини-носії цих структур мають різні потужності;
   б) дві алгебраїчні структури можуть не бути ізоморфними, якщо множини-носії цих алгебр мають однакові потужності.

5. Нехай C — множина структур з однією бінарною операцією, $C = \{S_1, S_2, ...\}$. Визначимо відношення ~ на множині C, таке, що $S_i \sim S_j$ виконується, якщо S_i і S_j ізоморфні. Доведіть, що ~ є відношенням еквівалентності на множині C.

3.3. Найпростіші алгебраїчні структури

Півгрупа, моноїд, група, абелева група

Розглянемо основні типи алгебраїчних структур, які мають тільки одну бінарну операцію.

Визначення

Півгрупою називається алгебраїчна структура з множиною-носієм A і бінарною операцією $\otimes$: $A^2 \to A$, яка задовольняє властивості асоціативності:

$$x \otimes (y \otimes z) = (x \otimes y) \otimes z; \qquad x, y, z \in A.$$

Визначення

Моноїдом називають алгебраїчну структуру з множиною-носієм M і бінарною операцією $\otimes$: $M^2 \to M$ такою, що

1. $\otimes$ асоціативна:
   $x \otimes (y \otimes z) = (x \otimes y) \otimes z$, для всіх $x, y, z \in M$.
2. Існує $e \in M$ — одиниця відносно $\otimes$:
   $e \otimes x = x = x \otimes e$ для всіх $x \in M$.

Таким чином, моноїд — це півгрупа з одиницею.

Півгрупи і моноїди використовуються при обробці рядків символів. Розглянемо приклад.

Приклад. При обробці рядків символів вводиться операція конкатенації — злиття рядків. Позначимо її $\bullet$. Конкатенація визначається як $\alpha \bullet \beta = \alpha\beta$, де α і β — рядки символів. Візьмемо рядки: «пар», «сам», «о», «воз», «ход», «вар». Застосувавши операції конкатенації, одержуємо такі рядки:

«пар» • «о» = «паро»
«сам» • «о» = «само»
«паро» • «воз» = «паровоз»
«паро» • «ход» = «пароход»
«само» • «вар» = «самовар».

Очевидно, що ця операція асоціативна, оскільки

(«пар» • «о») • «воз» = «пар» • («о» • «воз») =
= «паровоз» і т. д.

Отже $(A^+, \bullet)$ є півгрупою, де A^+ — множина різних рядків, що складаються з букв українського алфавіту. Якщо тепер ми позначимо через A^* множину всіляких рядків, що складаються з букв українського алфавіту і порожнього рядку ε, то одержимо структуру $(A^*, \bullet)$, яка є моноїдом з одиничним елементом ε. Оскільки ε позначає порожній рядок, то можемо записати $\varepsilon = «»$.

«самовар» • «» = «» • «самовар» = «самовар» і т. д.

Визначення

Групою називають множину G з бінарною операцією $\otimes$, що замкнена в G, такою, що

1. $\otimes$ асоціативна:
   $x \otimes (y \otimes z) = (x \otimes y) \otimes z$ для всіх $x, y, z \in G$.

2. Існує елемент $e \in G$ — одиниця відносно $\otimes$:

$e \otimes x = x \otimes e = x$ для всіх $x \in G$.

3. Кожному елементу $x \in G$ відповідає обернений елемент $x' \in G$ відносно $\otimes$:

$x' \otimes x = x \otimes x' = e$ для всіх $x \in G$.

Із визначення маємо, що група — це моноїд, в якому всі елементи оборотні.

Часто до слів «група» і «моноїд» приписують термін «комутативний». Це означає, що операція у розглянутій структурі задовольняє властивість комутативності, тобто

$y \otimes x = x \otimes y$ для всіх $x, y \in M$ або G.

Комутативна група називається ***абелевою групою*** (на честь норвезького математика Абеля).

Приклади. 1. Групою є множина дійсних чисел разом з операцією додавання: $(R, +)$, підгрупою цієї групи є $(Z, +)$, де Z — множина цілих чисел. Структура $(K, +)$, де K — множина цілих чисел, що кратні k, $k \in N$, є підгрупою групи $(Z, +)$. Для цих груп одиницею є 0, обернений елемент утворюється за допомогою застосування унарної операції зміни знака «$-$». Наведені групи є абелевими групами, оскільки додавання комутативне.

2. Структура $(N, +)$, де N — множина натуральних чисел, не є групою, оскільки не існує обернених елементів і одиниці. Насправді, $(N, +)$ — півгрупа.

3. Структури $(R, *)$ і $(N, *)$ не є групами, а є моноїдами. Одиничним елементом для операції множення є 1. Обернені елементи існують на множині дійсних чисел R для всіх елементів, крім 0: не існує 0^{-1}, такого, що $0 * 0^{-1} = 1$. Таким чином, операція множення задає групу на множині дійсних чисел, крім нуля $(R\backslash\{0\}, *)$. Додатна підмножина множини дійсних чисел з операцією множення $(R_+, *)$ теж є групою — підгрупою групи $(R\backslash\{0\}, *)$. Множення комутативне, отже, ці групи є абелевими.

4. Позначимо $M_n(R)$ множину всіх квадратних матриць порядку n з елементами з множини дійсних чисел. Структура $(M_n(R), +)$ — комутативний моноїд з одиницею — нульовою матрицею. Структура $(M_n(R), *)$ — некомутативний моноїд з одиницею — одиничною матрицею.

5. Структура $(Z_n, \oplus_n)$ — група з одиницею 0 і оберненим елементом $x' = n - x$; $(Z_n, \otimes_n)$ — моноїд з одиницею 1.

Для розв'язку рівнянь необхідно існування та єдиність одиниць і обернених елементів. Доведемо ці твердження.

Твердження 1. Нехай $\otimes$ — операція на множині A й існує одиниця e відносно $\otimes$, тоді ***одиничний елемент єдиний.***

Доведення. Нехай e і e' — дві одиниці відносно $\otimes$. Тоді для будь-яких $a, b \in A$ правильне $a = e' \otimes a$, $b = b \otimes e$. Підставляючи $a = e$, $b = e'$, одержуємо $e = e' \otimes e = e'$.

Твердження 2. Нехай $\otimes$ — асоціативна операція на множині A і e — одиниця відносно $\otimes$. Тоді, якщо $x \in A$ і x має обернений елемент, то ***обернений елемент єдиний*** відносно $\otimes$.

Доведення. Припустимо, що x' і x'' — обернені елементи до x, так що

$$x \otimes x' = x' \otimes x = e,\ x \otimes x'' = x'' \otimes x = e,$$

тоді

$$x' = x' \otimes e = x' \otimes (x \otimes x'') = (x' \otimes x) \otimes x'' = e \otimes x'' = x''.$$

Всередині групи $(G, \otimes)$ можна розв'язати рівняння $a \otimes \otimes x = b$.

До рівності $a \otimes x = b$ застосуємо зліва a' — обернений до a елемент і послідовно одержуємо:

$a' \otimes (a \otimes x) = a' \otimes b$,

$(a' \otimes a) \otimes x = a' \otimes b$ ($\otimes$ асоціативна),

$e \otimes x = a' \otimes b$ (властивість обернених елементів),

$x = a' \otimes b$ (властивість одиниці); x — розв'язок.

Існування одиниці та обернених елементів відносно деякої операції накладає значне обмеження на вид таблиці Келі для цієї операції. Розглянемо групу $(Z_7, \oplus_7)$. Таблиця Келі (див. р. 3.1) для операції $\oplus_7$ виглядає таким чином:

$\oplus_7$	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

Одиницею відносно операції $\oplus_7$ є 0. Таблиця симетрична відносно діагоналі — ця умова визначає комутативність операції. Зауважимо, що кожний стовпчик таблиці містить всі елементи групи.

Твердження 3. Будь-який стовпчик таблиці Келі для операції скінченної групи містить всі елементи групи.

Доведення. Розглянемо деяку групу $(G, \otimes)$, G — скінченна множина. Припустимо, що деякий стовпчик a_i таблиці Келі для операції $\otimes$ не містить якого-небудь елемента множини G. Тоді якийсь інший елемент a_j в цьому стовпчику повинен зустрітися двічі, скажімо, у k-ому та l-ому рядках. Але тоді $a_k \otimes a_i = a_j$, $a_l \otimes a_i = a_j$ і отже,

$$a_k \otimes a_i = a_l \otimes a_i,$$

$$a_k \otimes a_i \otimes a_i' = a_l \otimes a_i \otimes a_i'$$

(помножуємо обидві частини рівності на a_i'),

$$a_k \otimes e = a_l \otimes e,$$

($a_i \otimes a_i' = e$ за визначенням оберненого елемента),

$$a_k = a_l \quad (a_i \otimes e = a_i \text{ за визначенням одиниці}).$$

Одержуємо $a_k = a_l$, що неможливе, оскільки тут a_k, a_l — елементи групи, що задають різні рядки таблиці. Таким чином, i-й стовпчик таблиці Келі є перестановкою на множині елементів групи.

Запитання

1. Дайте визначення півгрупи, моноїду і групи.
2. Наведіть приклади півгрупи, що не є моноїдом, і моноїду, що не є групою.
3. Наведіть приклади групи.
4. Яка група називається абелевою? Наведіть приклади.
5. Які властивості елементів групи роблять можливим розв'язок рівнянь усередині групи?
6. Які властивості має таблиця Келі для скінченної групи?
7. Які властивості має операція абелевої групи?

Завдання

1. Доведіть, що в групі $(G, \otimes)$ для всіх $a, b, c \in G$ виконується:
   а) якщо $a \otimes b = a \otimes c$, то $b = c$, і, якщо $b \otimes a = c \otimes a$, то $b = c$;
   б) $(a')' = a$ — обернений елемент до оберненого елемента до a дорівнює a.

2. Побудуйте таблицю Келі для групи $(Z_5, \oplus_5)$. Для кожного елемента визначте обернений.
3. Побудуйте групу з бінарною операцією *max*, результатом виконання якої є найбільший з двох операндів.
4. Нехай множина $E = \{0, -2, 2, -4, 4, ...\}$. Покажіть, що $(E, +)$ — підгрупа групи $(Z, +)$.
5. Нехай задана алгебраїчна структура $(T, \otimes)$, де операція $\otimes$ асоціативна. Нехай існує елемент $0 \in T$, такий, що для будь-якого $x \in T$ правильно $0 \otimes x = x \otimes 0 = 0$. Покажіть, що $(T, \otimes)$ не може бути групою, за винятком випадку $T = \{0\}$.
6. Нехай деякий комп'ютер використовує регістри по k біт для зображення невід'ємних цілих чисел. Єдина операція, що виконується, — додавання. Яку алгебраїчну структуру зображує цей комп'ютер, якщо:

   а) при виникненні переповнення значення старшого біта втрачається;

   б) при виникненні переповнення результатом є найбільше із зображуваних чисел.

   Яка потужність множини-носія цієї алгебраїчної структури?
7. Складіть алгоритм перевірки, чи є структура $(A, \otimes)$ групою, півгрупою або моноїдом. Вхідними даними є елементи таблиці Келі операції $\otimes$.
8. Складіть алгоритм знаходження оберненого елемента до a і розв'язку рівняння $a \otimes x = b$ відносно x у групі $(G, \otimes)$. Вхідними даними є елементи таблиці Келі операції $\otimes$.

3.4. Кільця і поля

У цьому розділі розглянуто алгебраїчні структури з двома бінарними операціями $\otimes$ і $\oplus$. Операцію $\otimes$ називають множенням, а операцію $\oplus$ — додаванням. Для $\otimes$ одиничний елемент позначається 1, а обернений до елемента x відносно $\otimes$ записується у вигляді x^{-1}. Для $\oplus$ одиничний елемент позначається 0, а обернений до елемента x відносно $\oplus$ записується у вигляді $-x$. Зрозуміло, що для різних структур ці операції визначаються по-різному, хоча часто називаються однаково. Наприклад, множення матриць, множення цілих чисел, логічне множення тощо. Розглянемо спочатку структури, що називаються кільцями. Багато математичних конструкцій, які природно виникають у лінійній алгебрі (особливо в теорії матриць), є кільцями або містять кільця як підструктури.

Визначення

Кільцем $(R, \oplus, \otimes)$ називається множина R з визначеними на неї бінарними операціями $\oplus$ і $\otimes$, такими, що

1. $\oplus$ асоціативна:
   $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ для всіх $x, y, z \in R$.
2. $\oplus$ комутативна:
   $x \oplus y = y \oplus x$ для всіх $x, y \in R$.
3. $\oplus$ має одиницю, яка називається нулем і позначається 0:
   $0 \oplus x = x$ для всіх $x \in R$.
4. Існує обернений елемент відносно $\oplus$ для кожного $x \in R$:
   $(-x) \oplus x = x \oplus (-x) = 0$ для всіх $x \in R$.
5. $\otimes$ асоціативна:
   $x \otimes (y \otimes z) = (x \otimes y) \otimes z$ для всіх $x, y, z \in R$.
6. $\otimes$ дистрибутивна відносно $\oplus$ зліва і справа:
   $x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$,
   $(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z)$ для всіх $x, y, z \in R$.

Іншими словами, якщо $(R, \oplus)$ — абелева група, $(R, \otimes)$ — півгрупа і $\otimes$ дистрибутивна за $\oplus$, то $(R, \oplus, \otimes)$ — кільце.

Будемо вважати, що кільце ***комутативне***, якщо множення $\otimes$ комутативне і є ***кільцем з одиницею***, якщо існує одиниця відносно множення. Кільце з одиницею називається ***алгеброю***. Звичайно її позначають символом 1. Легко показати, що в кільці $(R, \oplus, \otimes)$ для будь-яких $a, b \in R$ виконуються співвідношення

$$0 \otimes a = a \otimes 0 = 0,$$
$$a \otimes (-b) = (-a) \otimes b = -(a \otimes b),$$
$$(-a) \otimes (-b) = a \otimes b.$$

В кільці $(R, \oplus, \otimes)$ фактично присутня ***некомутативна*** бінарна операція віднімання «$\ominus$», визначена за правилом $a \ominus b = a \oplus (-b)$. Вона є правою оберненою відносно додавання в тому розумінні, що $(a \oplus b) \ominus b = a$. Дійсно,

$$(a \oplus b) \ominus b = (a \oplus b) \oplus (-b) = a \oplus b \oplus (-b) = a \oplus 0 = a.$$

Визначення

Поле $(R, \oplus, \otimes)$ — це комутативне кільце з одиницею 1 (що відрізняється від 0), в якому кожний елемент a (що відрізняється від 0) обернений за множенням.

Поле можна визначити як об'єднання двох абелевих груп, пов'язаних одним законом дистрибутивності.

Приклад. Розглянемо алгебраїчну структуру $(Z_n, \oplus_n, \otimes_n)$ (див. розділи 3.2, 3.3). Для випадку $n = 6$ таблиці Келі операцій $\oplus_6$ і $\otimes_6$ виглядають таким чином:

$\oplus_6$	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

$\otimes_6$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

В Z_6 існують 4 випадки, коли добуток двох ненульових елементів може давати нуль, а саме: (2, 3), (3, 4), (3, 2), (4, 3), отже $(Z_6, \oplus_6, \otimes_6)$ не є полем. Таким чином, $(Z_n, \oplus_n, \otimes_n)$ є комутативним кільцем з одиницею при будь-якому $n \in N$. Якщо p — просте число, то алгебра $(Z_p, \oplus_p, \otimes_p)$ є полем.

Приклад. $(R, *, +)$ — множина дійсних чисел із стандартними операціями множення і додавання є полем і складається з абелевих груп $(R, +)$, $(R\backslash\{0\}, *)$, операції в яких зв'язані відповідним законом дистрибутивності.

Структуру $(R, *, +)$ називають ***полем дійсних чисел***.

Запитання

1. Які властивості мають операції кільця?
2. Дайте визначення кільця, використовуючи поняття групи і півгрупи.
3. Виконання яких співвідношень можна довести, використовуючи властивості операцій кільця?
4. Які властивості мають операції поля?
5. Чим поле відрізняється від кільця?
6. Дайте визначення поля, використовуючи поняття групи.

Завдання

1. Доведіть, що в полі $(F, \oplus, \otimes)$ виконуються співвідношення:
   а) $-a = a \otimes (-1)$;
   б) $-(a + b) = (-a) + (-b)$;
   в) $-(-a) = a$;
   г) якщо $a \neq 0$, то $(a^{-1})^{-1} = a$;
   д) $(-a) \otimes (-b) = a \otimes b$;
   е) $a \otimes b = 0 \Rightarrow a = 0$ або $b = 0$.
2. Покажіть, що алгебраїчна структура $(Z_p, \oplus_p, \otimes_p)$ є полем, якщо p — просте число. Наведіть приклад.
3. Побудуйте підструктури поля дійсних чисел, що є полями, кільцями.
4. Доведіть двосторонню дистрибутивність множення $\otimes$ відносно віднімання $\ominus$ у кільці $(R, \oplus, \otimes)$:
   $a \times (b \ominus c) = (a \otimes b) \ominus (a \otimes c)$;
   $(b \ominus c) \otimes a = (b \otimes a) \ominus (c \otimes a)$.

3.5. Ґратки

Верхня та нижня грані у частково упорядкованій множині, ґратка, повна ґратка, одиниця і нуль ґратки

Розглянемо алгебраїчну структуру $(M, \leq)$ з множиною-носієм M і бінарним відношенням R ***часткового порядку***, які будемо позначати $\leq$. Нагадаємо, що два елементи $a, b \in M$ можуть бути ***порівнянними*** за відношенням $\leq$, або ***непорівнянними*** (п. 2.4).

Визначення

Верхньою гранню для пари елементів $a, b \in M$ називається елемент $c_i \in M$, такий, що $a \leq c_i$, $b \leq c_i$. ***Нижньою гранню*** для пари елементів $a, b \in M$ називається елемент $d_i \in M$, такий, що $d_i \leq a$, $d_i \leq b$.

Найменшою верхньою гранню для пари елементів $a, b \in M$ називається елемент $c \in M$, найменший з усіх верхніх граней для a, b. Це означає, що для будь-якого $c_i \in M$, такого, що $c_i \neq c$, $a \leq c_i$, $b \leq c_i$, виконується $c \leq c_i$. ***Найбільшою нижньою гранню*** для елементів $a, b \in M$ називається елемент $d \in M$ найбільший з усіх верхніх граней для a, b. Це означає, що для будь-якого $d_i \in M$, що відрізняється від d і такого, що $d_i \neq d$, $d_i \leq a$, $d_i \leq b$, виконується $d_i \leq d$.

Визначення

Частково упорядкована множина — алгебраїчна структура $(A, \leq)$, в якій кожна пара елементів має найменшу верхню і найбільшу нижню грані, називається ***ґраткою.***

Визначивши у ґратці $(A, \leq)$ бінарні операції $\wedge$ і $\vee$, де $\wedge$ — знаходження найбільшої нижньої грані і $\vee$ — знаходження найменшої верхньої грані, одержимо алгебраїчну структуру ґратки з двома бінарними операціями: $(A, \wedge, \vee)$.

Структуру скінченних частково упорядкованих множин можна зобразити за допомогою діаграм Хассе, в яких елементи зображуються точками, що розташовані на різних горизонталях, причому більші елементи з'єднуються з безпосередньо меншими за допомогою ліній, що опускаються. В результаті елемент a виявляється більше ніж елемент b тоді і тільки тоді, коли на діаграмі можна перейти від a до b за лініями, що опускаються вниз. Така діаграма є підграфом графа бінарного відношення ≥ 0, оберненого до $\leq$, причому в цьому підграфі не вказані напрямки дуг, оскільки відомо, що для кожної лінії мається на увазі напрямок зверху-вниз (порівняти рис. 2.15 і 2.16).

Приклад. Розглянемо множину B_n двійкових векторів довжини n, частково упорядковану таким чином: $v \leq w$, якщо у векторі w одиниці стоять на всіх тих місцях, на яких вони стоять у v (і, можливо, ще на деяких). Наприклад, (010) ≤ (011), а (010) і (100) не порівнянні. Зобразимо граф відношення порядку $\geq$ і діаграму для випадку $n = 3$ (рис. 3.8).

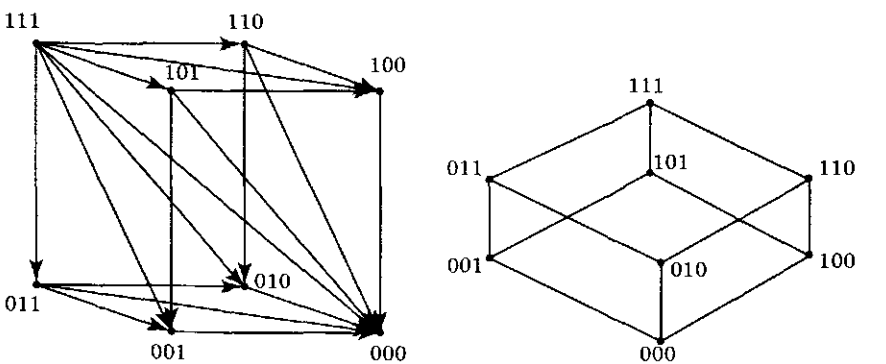


Рис. 3.8. Граф відношення порядку $\geq$ і діаграма для множини двійкових векторів B_3

Множина B_n і введений на ній частковий порядок $\leq$ утворюють ґратку. В ній найменша верхня грань $v \vee w$ — це вектор, в якому одиниці стоять на тих і тільки тих місцях, де одиниці є або в v, або в w, а найбільша нижня грань $v \wedge w$ — це вектор, в якому одиниці стоять на тих і тільки тих місцях, де є одиниці одночасно і в v і в w. Наприклад, (010) $\vee$ (100) = (110), (010) $\wedge$ (100) = (000).

Таким чином, $v \vee w$ на діаграмі визначається як найближча вершина, яка знаходиться над v і w та з якої є шлях вниз і в v і в w. Однак, якщо є шлях з v вниз до w, то $v \vee w$ буде дорівнювати v. Аналогічно, $v \wedge w$ на діаграмі визначається як найближча вершина, яка знаходиться під v і w і в яку є шлях вниз і з v і з w. Однак, якщо є шлях з v вниз у w, то $v \wedge w$ буде дорівнювати w.

Не будь-яка алгебраїчна структура, що задається діаграмою Хассе, є ґраткою.

На рис. 3.9,*а* зображено структуру, що є ґраткою, оскільки в ній кожна пара елементів має найменшу верхню і найбільшу нижню грань. Для цієї ґратки справедливі вирази: $b \wedge c = d$, $d \wedge a = d$, $e \vee g = e$, $e \vee f = a$.

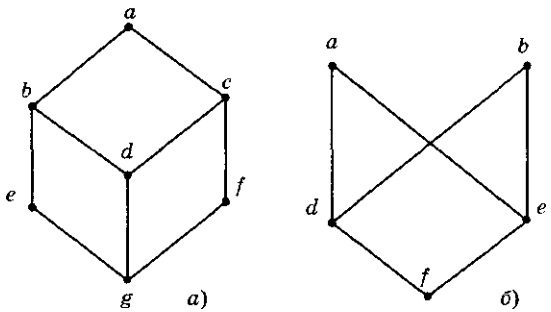


Рис. 3.9. Діаграми, що визначають відношення часткового порядку

Структура, діаграму якої зображено на рис. 3.9,*б*, не є ґраткою, оскільки пара елементів (a, b) не має верхньої грані. Крім того, пара (a, b) має три нижні грані — f, d і e, але з них неможливо обрати єдину найбільшу, оскільки d і e не порівнянні. Таким чином, пара елементів (a, b) не має найбільшої нижньої грані.

Нескладно показати, що $\wedge$ і $\vee$ асоціативні і комутативні, тому можна розглядати застосування цих операцій для будь-якої скінченної підмножини елементів ґратки.

Визначення

Ґратка, в якій найбільша нижня грань і найменша верхня грань існують для будь-якої підмножини її елементів, називається ***повною***. Через асоціативність операцій $\wedge$ і $\vee$ скінченна ґратка завжди повна. Найменша верхня грань 1 для всіх елементів повної ґратки — це максимальний елемент ґратки, що називається ***одиницею ґратки***. Найбільша нижня грань 0 для всіх елементів повної ґратки — це мінімальний елемент ґратки або ***нуль ґратки***.

Нуль і одиниця ґратки єдині, оскільки найменша верхня грань і найбільша нижня грань єдині.

В ґратах $(A, \wedge, \vee)$ виконуються такі властивості операцій $\vee$, $\wedge$ для будь-яких $a, b \in A$:

1. Операції $\wedge$, $\vee$ комутативні:
   $a \wedge b = b \wedge a$, $a \vee b = b \vee a$.

2. Операції $\wedge$, $\vee$ асоціативні:
   $(a \wedge b) \wedge c = a \wedge (b \wedge c)$, $(a \vee b) \vee c = a \vee (b \vee c)$.

3. Операції $\wedge$, $\vee$ ідемпотентні:
   $a \wedge a = a$, $a \vee a = a$.

4. Виконуються закони поглинання:
   $a \wedge (a \vee b) = a$, $a \vee (a \wedge b) = a$.

Виконання цих властивостей достатньо, щоб структура $(A, \wedge, \vee)$ була ґраткою. Якщо в ґратці присутні нуль ґратки і одиниця ґратки, то для них виконуються такі властивості:

Для будь-якого $a \in A$ правильно $a \vee 0 = a$, $a \wedge 1 = a$, $a \wedge 0 = 0$, $a \vee 1 = 1$.

Ґратка з нулем 0 і одиницею 1 називається ***ґраткою з доповненням***, якщо для кожного її елемента a є обернений елемент $\neg a$ відносно обох операцій $\wedge$ і $\vee$: для будь-якого $a \in A$ справедливо $a \vee (\neg a) = 1$, $a \wedge (\neg a) = 0$.

Операція знаходження оберненого елемента задовольняє тотожність: $\neg (\neg a) = a$ для будь-якого $a \in A$.

Ґратка називається ***дистрибутивною***, якщо операції $\wedge$ і $\vee$ дистрибутивні одна відносно іншої:

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c),\ a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

Визначення

Дистрибутивна ґратка з доповненням називається ***булевою ґраткою*** або ***булевою алгеброю*** $(A, \wedge, \vee, \neg, 0, 1)$, де $\neg$ — унарна операція знаходження оберненого елемента, $0, 1$ — нуль і одиниця ґратки.

Можна довести, що в булевій алгебрі виконуються такі тотожності, що називаються законами де Моргана:

$\neg(a \vee b) = \neg a \wedge \neg b$, $\neg(a \wedge b) = \neg a \vee \neg b$ для будь-яких $a, b \in A$.

Наведемо ще кілька прикладів ґраток.

Приклад. Будь-яка повністю упорядкована множина (див. п. 2.4), в якій для кожної пари елементів (a, b) існує максимальний з двох елемент $max(a, b)$ і мінімальний з двох елемент $min(a, b)$, є ґраткою, в якій для будь-яких $a, b \in M$ правильно $a \vee b = max(a, b)$, $a \wedge b = min(a, b)$. Така ґратка не є булевою, оскільки в ній немає обернених елементів.

Приклад. Визначимо на множині натуральних чисел N відношення часткового порядку таким чином: $a \leq b$, якщо a ділить b. Тоді $a \vee b$ — найменше спільне кратне a і b, $a \wedge b$ — найбільший спільний дільник a, b. Наприклад, $9 \vee 15 = 45$, $9 \wedge 15 = 3$, $5 \wedge 9 = 1$, $5 \vee 9 = 45$. Визначена таким чином структура $(N, \vee, \wedge)$ є ґраткою, нуль якої є число 1. Ця ґратка не є булевою, вона не має одиниці ґратки, тобто такого елемента, який був би найменшим спільним кратним для всіх пар чисел з N.

Приклад. Система всіх підмножин $\{M_i\}$ будь-якої скінченної множини X частково упорядкована за включенням, тобто $M_i \leq M_j$, якщо і тільки якщо $M_i \subseteq M_j$. Ця система є ґраткою, елементами якої є множини M_i, а операціями $\vee$, $\wedge$ — звичайні теоретико-множинні операції об'єднання та перетину: $\cup$, $\cap$. Ця ґратка завжди повна. Її одиницею є сама множина X, нулем — порожня множина. Така ґратка є булевою алгеброю.

Зауважимо, що алгебраїчний формалізм ґрат застосовується в теорії графів і у програмуванні, зокрема, у мовах програмування і аналізі типів даних.

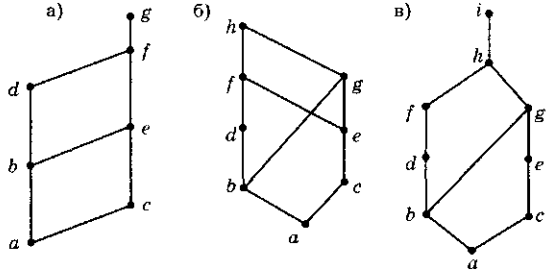
Запитання

1. Дайте визначення ґратки. Визначте найменшу верхню та найбільшу нижню грані двох елементів частково упорядкованої множини.
2. Визначте бінарні операції $\wedge$ і $\vee$.

3. Яким чином будову скінченних частково упорядкованих множин можна зобразити за допомогою діаграм?
4. Наведіть приклади діаграм, що зображують структури, які є ґратами, та структури, що не є ґратами.
5. В яких випадках скінченна упорядкована множина не є ґраткою?
6. Які елементи називаються нулем та одиницею ґратки?
7. Наведіть приклади ґрат.

Завдання

1. Визначте, чи є ґратами частково упорядковані множини, зображені такими діаграмами:



2. Визначте, чи є ґратами алгебраїчні структури у вигляді таких частково упорядкованих множин:
   а) (A = {1, 3, 6, 9, 12}, відношення порядку — «дільник»). Відношення «дільник» позначимо «|» (наприклад: 3 |6, 6 ∤ 9);
   б) (A = {1, 5, 25, 125}, відношення порядку —|);
   в) (Z — множина цілих чисел, відношення порядку — ≤);
   г) (2^A, відношення порядку ⊆), де $A = \{a, b, c\}$;
   д) (2^Z, ⊆).
3. Доведіть, що будь-яка непорожня скінченна підмножина ґратки має найменшу верхню і найбільшу нижню грані (границями можуть бути елементи ґратки, що не належать до цієї підмножини).
4. Які з наведених пар елементів є порівнянними в частково упорядкованій множині цілих додатних чисел Z^+ за відношенням «дільник» — (Z^+, |):
   а) 5, 15;
   б) 6, 9;
   в) 8, 16;
   г) 7, 7.

розділ 4

Булеві функції та перетворення

4.1. Булеві змінні і функції

Двійкові інтерпретації, істотні та фіктивні змінні

Для зображення інформації в комп'ютерах використовується двійкова система числення. Таким чином, всі операції, які виконує комп'ютер, проводяться на множині $\{0, 1\}$. Ці перетворення зручно формально зображати за допомогою апарата двійкової логіки, який був розроблений Джорджем Булем у середині XIX століття. Ця алгебраїчна структура є алгеброю і називається булевою (п. 3.5). Булева алгебра використовується при розв'язанні різних задач обробки інформації, при роботі з базами даних, в логічному програмуванні, при проектуванні інтелектуальних систем, для конструювання та аналізу роботи комп'ютерів та інших електронних пристроїв. У цій главі розглянуто основні властивості булевих функцій з аргументами з множини $\{0, 1\}$ і способи зображення булевих функцій у вигляді виразів булевої алгебри. Булева функція може мати велику кількість змінних і знаків операцій, у той час, як може існувати інше, еквівалентне зображення даної функції, що має меншу кількість змінних і операцій. У главах розділу описано методи одержання виразів з мінімальною кількістю змінних і знаків операцій.

Розглянемо двохелементну множину B, елементи якої будемо позначати через **0** і **1**: $B = \{0, 1\}$.

Визначення

Змінні, які можуть приймати значення тільки з множини B, називаються ***логічними*** або ***булевими змінними***. Самі значення 0 і 1 булевих змінних називаються ***булевими константами***.

В мовах програмування для роботи з такими змінними, як правило, вводиться спеціальний логічний (булевський) тип (наприклад, у мовах Pascal і Java — boolean, у C+ — bool). Змінна цього типу приймає два значення: true і false.

Визначення

Функція виду $y = f(x_1, x_2, \ldots, x_n)$, аргументи x_i і значення y якої належать множині B, називається n-місною булевою ***функцією***. Такі функції також називають логічними або ***перемикальними*** функціями.

Визначення

Кортеж $(x_1, x_2, \ldots, x_n)$ конкретних значень булевих змінних називається двійковим ***словом*** (***n-словом***) або ***булевим набором*** довжини n. Для булевої функції $y = f(x_1, x_2, \ldots, x_n)$ конкретне (індивідуальне) значення булевого набору $(x_1, x_2, \ldots, x_n)$ називається також інтерпретацією ***булевої функції*** f. Множина всіх двійкових слів, що позначається через B^n, називається ***n-вимірним булевим кубом*** і містить 2^n елементів-слів: $|B^n| = 2^n$.

Дійсно, для $n = 1$ є всього $2^1 = 2$ слова — (0) і (1), а за рахунок збільшення довжини слова на один символ кількість слів збільшується в два рази, оскільки додатковий символ може приймати два значення — 0 або 1. Для наочності зобразимо в одному рядку набори булевих змінних зростаючої довжини, у другому рядку — кількості різних наборів відповідної довжини:

$$(x_1),\ (x_1, x_2),\ (x_1, x_2, x_3),\ \ldots,\ (x_1, x_2, \ldots, x_n);$$
$$2,\quad 2 \cdot 2,\quad 2 \cdot 2 \cdot 2 = 2^3,\ \ldots,\ 2 \cdot 2 \cdot \ldots \cdot 2 = 2^n.$$

Покажемо, що кількість всіх можливих булевих функцій $y = f(x_1, x_2, \ldots, x_n)$ дорівнює 2^{2n}. Зауважимо, що у позначеннях п. 2.5 булева функція є відображення f: $B^n \to B$, $B = \{0, 1\}$. Позначимо n-слово одним символом $u = (x_1, x_2, \ldots, x_n)$ і пронумеруємо

всі різні слова: $\{u\} = \{u_1, u_2, ..., u_p\}$, де $p = 2^n$. Дві булеві функції $f_1(u)$ і $f_2(u)$ співпадають, якщо $f_1(u_i) = f_2(u_i)$ для всіх u_i, $i = 1, 2, ..., p = 2^n$. Оскільки $f(u)$ може мати два значення (0 або 1), то число різних функцій $f(u)$ дорівнює кількості різних булевих слів (наборів) довжини p, тобто числу $2^p = 2^{2^n}$.

Функції кількох незалежних змінних можна розглядати як функції від більшої кількості змінних. При цьому значення функції не змінюється при зміні значення цих «додаткових» змінних.

Визначення

Змінна x_i у функції $f(x_1, ..., x_{i-1}, x_i, x_{i+1}, ..., x_n)$ називається неістотною (або ***фіктивною***), якщо

$$f(x_1, ..., x_{i-1}, 0, x_{i+1}, ..., x_n) = f(x_1, ..., x_{i-1}, 1, x_{i+1}, ..., x_n)$$

при будь-яких значеннях решти змінних, тобто якщо зміна значення x_i у будь-якому наборі значень $x_1, ..., x_n$ не змінює значення функції.

В цьому випадку функція $f(x_1, ..., x_n)$ фактично залежить від $n - 1$ змінної, тобто зображує функцію $g(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)$. Стверджують, що функція g одержана з функції f видаленням фіктивної змінної, а функція f одержана з g введенням фіктивної змінної, причому ці функції за визначенням рівні.

Запитання

1. Які змінні називаються булевими або логічними змінними?
2. Дайте визначення булевої функції.
3. Що зображує область визначення і область значень булевої функції?
4. Поясніть сенс поняття «інтерпретація булевої функції».
5. Які змінні називаються неістотними або фіктивними?

Завдання

1. Скільки інтерпретацій має булева функція від трьох змінних $f(x, y, z)$? Назвіть їх.
2. Знайдіть кількість булевих функцій n змінних, що приймають значення 1 рівно на одному наборі.
3. Знайдіть кількість булевих функцій n змінних, що приймають на протилежних наборах однакові значення. Протилежними наборами називаються такі набори, які в однакових позиціях містять протилежні елементи. Наприклад: (0, 0) — (1, 1); (1, 0) — (0, 1) тощо.

4.2. Способи задання булевих функцій

Таблиця істинності, двохелементна булева алгебра, алгебра логіки, суперпозиція булевих функцій, пріоритет операцій, еквівалентність формул булевої алгебри

Булеві функції можуть бути задані такими способами:

1. За допомогою таблиці істинності (значеннями на кожній з інтерпретацій).
2. Порядковим номером, який має ця функція.
3. Аналітично (у вигляді формули).

Розглянемо кожний із зазначених способів докладніше.

4.2.1. Таблиці істинності

Таблиці, в яких кожній інтерпретації (тобто набору аргументів) функції поставлено у відповідність її значення, називаються таблицями ***істинності булевої функції.***

В таблиці істинності кожній змінній та значенню самої функції відповідає по одному стовпчику, а кожній інтерпретації — по одному рядку. Кількість рядків у таблиці відповідає кількості різних інтерпретацій функції. Булеві функції $\varphi(x)$, які залежать від однієї змінної, наведено в таблиці 4.1.

Таблиця 4.1. Булеві функції однієї змінної

x	φ_0	φ_1	φ_2	φ_3
0	0	0	1	1
1	0	1	0	1

Кожній функції відповідно до значень, що вона приймає, можна привласнити такі назви:

$\varphi_0 \equiv 0$ — функція константа 0,

$\varphi_1 = x$ — функція повторення аргументу,

$\varphi_2 = \bar{x}$ — функція інверсії або заперечення аргументу,

$\varphi_3 \equiv 1$ — функція константа 1.

Різні булеві функції двох змінних $f(x, y)$ зображено в таблиці 4.2, їх кількість дорівнює $2^{2^2} = 16$.

Таблиця 4.2. Булеві функції двох змінних

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Більшість із шістнадцяти булевих функцій $f(x, y)$ часто застосовуються на практиці. Оскільки дані функції використовуються як у математиці, так і в програмуванні, вони можуть мати різне позначення. Позначення, назви і прочитання булевих функцій від двох змінних зображено в таблиці 4.3.

Таблиця 4.3. Позначення булевих функцій двох змінних

Функція	Позначення	Назва	Інші позначення	Прочитання
1	2	3	4	5
$f_0(x, y)$	0	константа 0		будь-яке 0, константа 0
$f_1(x, y)$	$x \wedge y = xy$	кон'юнкція (логічне «і»)	·, &, &&, *, AND, I, ×, min	x і y
$f_2(x, y)$	$x \leftarrow y$	заперечення імплікації	\	x і не y
$f_3(x, y)$	x	повторення першого аргументу		як x
$f_4(x, y)$	$y \leftarrow x$	заперечення оберненої імплікації	\	не x і y
$f_5(x, y)$	y	повторення другого аргументу		як y
$f_6(x, y)$	$x \oplus y$	що виключає «або» (сума за модулем 2)	≠, <>, > <, !=, XOR	x не як y
$f_7(x, y)$	$x \vee y$	диз'юнкція (логічне «або»)	OR, АБО, +, max	x або y
$f_8(x, y)$	$x \downarrow y$	заперечення диз'юнкції (стрілка Пірса)	$x \bar{\vee} y$, $x \circ y$	не x і не y
$f_9(x, y)$	$x \sim y$	еквівалентність	⇔, =, Eqv, =	x як y
$f_{10}(x, y)$	$\bar{y}$	заперечення другого аргументу	¬y	не y

Продовження таблиці 4.3

1	2	3	4	5
$f_{11}(x, y)$	$y \to x$	обернена імплікація	$\subset$	x, якщо y (x або не y)
$f_{12}(x, y)$	$\bar{x}$	заперечення першого аргументу	$\neg x$	не x
$f_{13}(x, y)$	$x \to y$	імплікація	$\supset, \Rightarrow$, Imp	якщо x, то y (не x або y)
$f_{14}(x, y)$	$x \mid y$	заперечення кон'юнкції (штрих Шеффера)	$x \bar{\wedge} y, x \bar{\&} y$	не x або не y
$f_{15}(x, y)$	1	константа 1		будь-яке 1, константа 1

Позначення Not, And, Or, Xor, Imp, Eqv використовуються у мові програмування Basic; позначення !, &, != використовуються у мові C; позначення $\neg$, $\wedge$, $\vee$ використовуються в системі Mathcad. Для стислості у прикладах та викладеннях ми будемо опускати знак кон'юнкції і писати xy замість $x \wedge y$.

4.2.2. Номери булевих функцій та інтерпретацій

Кожній функції привласнюється порядковий номер у вигляді натурального числа, двійковий код якого зображує стовпчик значень функції у таблиці істинності. Молодшим розрядом вважається самий нижчий рядок (значення функції на інтерпретації (1, 1, ..., 1)), а старшим — самий верхній (значення функції на інтерпретації (0, 0, ..., 0)). Вказаний порядковий номер функції, як двійковий, так і десятковий, повністю визначає булеву функцію.

Кожній інтерпретації булевої функції також привласнюється свій номер — значення двійкового коду, який зображує інтерпретація. Інтерпретації, що записана у верхньому рядку таблиці істинності, привласнюється номер 0, потім йде інтерпретація номер 1 тощо. В самому нижчому рядку розташована інтерпретація за номером $2^n - 1$, де n — кількість змінних, від яких залежить булева функція.

Приклад. Знайдемо порядковий номер функції $f(x, y)$, що приймає такі значення: $f(0, 0) = 1$, $f(0, 1) = 1$, $f(1, 0) = 0$, $f(1, 1) = 1$. Побудуємо таблицю істинності для цієї функції (таблиця 4.4).

Таблиця 4.4. Таблиця істинності $f(x, y)$

x	y	$f(x, y)$
0	0	1
0	1	1
1	0	0
1	1	1

Двійковий код, що відповідає значенням цієї функції, — 1101. Переведемо двійкове число 1101_2 у десяткову систему числення. Для цього кожному розряду двійкового числа привласнимо ваговий коефіцієнт, що кратний відповідному степеню числа 2, починаючи з нижчого рядку: 2^0, 2^1, 2^2 тощо. Помноживши ваговий коефіцієнт на відповідну двійкову цифру і додавши одержані значення, знайдемо порядковий номер функції:

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13_{10}.$$

Таким чином, десятковий номер даної функції — 13, тобто розглянута функція імплікації $f_{13}(x, y) = x \to y$ (див. $f_{13}(x, y)$ в таблиці 4.3). Таким чином, функцію $f_{13}(x, y)$ можна задати за допомогою двійкового коду, що відповідає її двійковому номеру: $f_{13}(x, y)$ відповідає двійковому числу $(1101)_2$.

Приклад. Побудуємо таблицю істинності для бінарної функції з порядковим номером 14. Для цього знайдемо двійкове число, яке відповідає десятковому числу 14. Зобразивши це число як суму степенів числа 2, одержимо:

$$14_{10} = 8 + 4 + 2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 1110_2.$$

Таким чином, $f_{14}(x, y)$ відповідає двійковому числу $(1110)_2$.

Побудуємо шукану таблицю істинності, розташувавши одержане число у стовпчику значення функції таким чином, щоб молодший розряд виявився у нижчому рядку (таблиця 4.5).

Таблиця 4.5. Таблиця істинності $f_{14}(x, y)$

x	y	$f_{14}(x, y)$
0	0	1
0	1	1
1	0	1
1	1	0

4.2.3. Булеві алгебри: загальна, двохелементна і логічна

В п. 3.5 введено поняття ***булевої алгебри*** $(A, \wedge, \vee, \neg, 0, 1)$ як ***дистрибутивної ґратки з доповненням***, де $\wedge$, $\vee$ — бінарні операції, $\neg$ — унарна операція на множині-носії A, які володіють визначеними властивостями. Аналізуючи алгебраїчну форму всіх цих властивостей (див. п. 3.5) і заміняючи позначення операцій $\wedge$, $\vee$, $\neg$ на $\bullet$, $+$, $\bar{}$ відповідно, ми виділяємо такий набір ***незалежних властивостей***, які можна вважати ***аксіомами*** або ***незалежними законами*** булевої алгебри:

1. Комутативність: $a + b = b + a$, $a \bullet b = b \bullet a$.
2. Асоціативність: $a + (b + c) = (a + b) + c$,
   $a \bullet (b \bullet c) = (a \bullet b) \bullet c$.
3. Дистрибутивність: $a + (b \bullet c) = (a + b) \bullet (a + c)$,
   $a \bullet (b + c) = (a \bullet b) + (a \bullet c)$.
4. Закони для нуля, одиниці та заперечення:
   $a + 0 = a$, $a + \bar{a} = 1$, $a \bullet 1 = 1$, $a \bullet \bar{a} = 0$.

Всі інші відношення (деякі з них також називаються «законами» булевої алгебри) є наслідком зазначених вище. Зокрема, з 1–4 виходять:

5. Закони ідемпотентності: $a + a = a \bullet a = a$.
6. Інші властивості одиниці і нуля: $a + 1 = 1$, $a \bullet 0 = 0$.
7. Закони поглинання: $a \bullet (a + b) = a + (a \bullet b) = a$.
8. Закон інволюції (подвійного заперечення): $\bar{\bar{a}} = a$.
9. Закони де Моргана (подвійності): $\overline{a + b} = \bar{a} \bullet \bar{b}$;
   $\overline{a \bullet b} = \bar{a} + \bar{b}$.

Таким чином, визначення булевої алгебри з п. 3.5 виглядає так (після «алгебраїчної розшифровки» понять, що беруть у ньому участь):

> **Визначення**
>
> ***Булева алгебра*** — це алгебраїчна структура $(A, \bullet, +, \bar{}, 0, 1)$ з бінарними операціями $\bullet, +: A^2 \to A$, унарною операцією «$\bar{}$»: $A \to A$ і виділеними елементами 0, 1 в носії A, які задовольняють властивості 1–4.

Позначення операцій $\wedge$, $\vee$ замість $\bullet, +$ у спеціальних булевих алгебрах запозичене з логіки, в якій Дж. Буль вперше

виділив алгебраїчну структуру з властивостями 1–4. Носій цієї структури $B = \{0, 1\}$ складається з двох елементів.

Визначення
Алгебраїчна структура $(B, \wedge, \vee, \bar{\ })$, $B = \{0, 1\}$, де операція $\wedge$ є ***кон'юнкція*** $f_1(x, y)$, $\vee$ є ***диз'юнкція*** $f_7(x, y)$ (див. таблиці 4.2, 4.3), «$\bar{\ }$» є ***заперечення*** або ***інверсія*** $\varphi_2(x)$ (див. таблицю 4.1), називається ***двохелементною булевою алгеброю.***

Виконання законів булевої алгебри 1–4 у структурі $(B, \wedge, \vee, \bar{\ })$, а також законів 5–9 виходить з визначення операцій кон'юнкції, диз'юнкції та заперечення через таблиці істинності 4.1, 4.2 (див. п. 4.4).

Визначення
Алгеброю логіки називається двохелементна булева алгебра $(B, \wedge, \vee, \bar{\ }, \to, \sim)$, $B = \{0, 1\}$, в якій множину операцій доповнено двома бінарними операціями: ***імплікацією*** $\to (f_{13}(x, y))$ та ***еквівалентністю*** $\sim (f_9(x, y))$ (див. таблиці 4.2, 4.3).

4.2.4. Булеві формули і пріоритет операцій

Булеві функції можуть бути задані аналітично, тобто формулами.

Визначення
Формула — це вираз, що містить булеві функції та їхні суперпозиції.

Визначення
Суперпозицією називається спосіб одержання нових функцій шляхом підстановки значень одних функцій замість значень аргументів інших функцій. Точніше, ***суперпозицією*** булевої функції f_0 і функцій $f_1, \ldots, f_n$ називається функція $f(x_1, \ldots, x_m) = f_0(g_1(x_1, \ldots, x_m), \ldots, g_k(x_1, \ldots, x_m))$, де кожна з функцій $g_i(x_1, \ldots, x_m)$ співпадає з однією з функцій $f_1, \ldots, f_n$, $i \in \{1, \ldots, k\}$. При цьому деякі з функцій f_i, а значить і g_i можуть тотожно співпадати з однією із змінних x_t, $t \in \{1, \ldots, m\}$.

Приклад. Розглянемо формулу, що задає функцію $f(x, y, z)$ таким чином:

$$f(x, y, z) = (x \wedge \bar{y}) \vee z.$$

Ця формула містить функції: $g(x_1)$ — заперечення, $s(x_1, x_2)$ — кон'юнкція і $l(x_1, x_2)$ — диз'юнкція. Дану формулу можна зобразити у вигляді суперпозиції вказаних функцій таким чином:

$$f(x, y, z) = l(s(x, g(y)), z).$$

У формулі $(x \wedge \bar{y}) \vee z$ використано інфіксний запис, коли знаки функцій (операцій) розташовані між операндами. Для запису виразів у інфіксній формі необхідно визначити порядок виконання операцій, що виконується за допомогою дужок або задання пріоритету операцій.

Якщо у формулі відсутні дужки, то операції виконуються у такій послідовності: заперечення, кон'юнкція, диз'юнкція, імплікація та еквівалентність:

$$\bar{}, \wedge, \vee, \rightarrow, \sim.$$

Приклад. Попередню формулу $f(x, y, z) = (x \wedge \bar{y}) \vee z$ з урахуванням пріоритету операцій можна записати без дужок:

$$(x \wedge \bar{y}) \vee z = x \wedge \bar{y} \vee z.$$

Зворотно з урахуванням пріоритету операцій розставимо дужки:

$$x \sim z \rightarrow \bar{x} \vee z = x \sim (z \rightarrow (\bar{x} \vee z)).$$

Необхідно відзначити, що знак операції заперечення може розміщатися не тільки над окремими змінними, але й над формулами або їхніми частинами. В цьому випадку операції виконуються так, як наче б то вираз, що знаходиться під знаком заперечення, було вкладено в дужки.

Приклад. У формулі $x \rightarrow \overline{z \wedge y \vee x} \vee z$ дужки можні розставити таким чином:

$$x \rightarrow (\overline{(z \wedge y) \vee x}) \vee z).$$

На відміну від табличного завдання, зображення функції формулою не єдине. Наприклад, функцію штрих Шеффера можна зобразити за допомогою основних операцій булевої алгебри формулами:

$$f_{14} = \bar{x}_1 \vee \bar{x}_2 \quad \text{або} \quad f_{14} = \overline{x_1 x_2},$$

а функцію стрілка Пірса таким чином:

$$f_8 = \bar{x}_1 \bar{x}_2 \quad \text{або} \quad f_8 = \overline{x_1 \vee x_2}.$$

Визначення

Формули, що зображують одну й ту ж функцію, називаються ***еквівалентними або рівносильними***.

Еквівалентність формул позначається знаком рівності. Один із способів встановити еквівалентність формул складається з такого: для кожної формули будується таблиця істинності, а потім одержані таблиці порівнюються, тобто фактично для кожного набору змінних перевіряється, чи дорівнюють на ньому значення функцій. Існують й інші способи перевірки еквівалентності формул, які буде розглянуто нижче.

4.2.5. Перехід від формули до таблиці істинності функції

Побудуємо таблицю істинності для функції, що задана такою формулою:

$$f(x, y, z, t) = x \vee \overline{y}(z \vee \overline{xy}) \to t.$$

Функція залежить від чотирьох змінних і, отже, для неї є $2^4 = 16$ інтерпретацій. Для побудови таблиці істинності необхідно визначити значення функції на всіх інтерпретаціях:

$$f(0, 0, 0, 0) = 0 \vee \overline{0}(0 \vee \overline{00}) \to 0 = 0 \vee 1(0 \vee \overline{0}) \to 0 =$$
$$= 0 \vee 1(0 \vee 1) \to 0 = 0 \vee 1(1) \to 0 = 0 \vee 1 \to 0 = 1 \to 0 = 0;$$
$$f(0, 0, 0, 1) = 0 \vee \overline{0}(0 \vee \overline{00}) \to 1 = 0 \vee 1(0 \vee \overline{0}) \to 1 =$$
$$= 0 \vee 1(0 \vee 1) \to 1 = 0 \vee 1(1) \to 1 = 0 \vee 1 \to 1 = 1 \to 1 = 1;$$
$$f(0, 0, 1, 0) = 0 \vee \overline{0}(1 \vee \overline{00}) \to 0 = 0 \vee 1(1 \vee \overline{0}) \to 0 =$$
$$= 0 \vee 1(1 \vee 1) \to 0 = 0 \vee 1(1) \to 0 = 0 \vee 1 \to 0 = 1 \to 0 = 0;$$
$$\ldots$$
$$f(1, 1, 1, 1) = 1 \vee \overline{1}(1 \vee \overline{11}) \to 1 = 1 \vee 0(1 \vee \overline{1}) \to 1 =$$
$$= 1 \vee 0(1 \vee 0) \to 1 = 1 \vee 0(1) \to 1 = 1 \vee 0 \to 1 = 1 \to 1 = 1.$$

Розмістимо в таблиці істинності інтерпретації в порядку збільшення відповідних їм двійкових номерів і запишемо одержане значення функції на кожній інтерпретації. Одержимо таблицю істинності функції $f(x, y, z, t) = x \vee \overline{y}\ (z \vee \overline{xy}) \to t$ (таблиця 4.6).

Таблиця 4.6. Таблиця істинності $f(x, y, z, t)$

x	y	z	t	$f(x, y, z, t)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1

Продовження таблиці 4.6

x	y	z	t	$f(x, y, z, t)$
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Запитання

1. Назвіть способи завдання булевих функцій.
2. Яким чином будується таблиця істинності булевої функції?
3. Назвіть основні булеві функції від двох змінних.
4. Яким чином визначається номер булевої функції? Номер інтерпретації?
5. Дайте визначення суперпозиції булевих функцій.
6. Який пріоритет визначений для операцій алгебри логіки? Для якої цілі служить пріоритет операцій?
7. Які формули називаються еквівалентними?
8. Яким чином здійснюється перехід від формули до таблиці істинності функції?

Завдання

1. Опустіть максимально можливе число дужок у формулі з урахуванням пріоритету виконання операцій:
   а) $(((x \sim y) \sim ((((x \wedge z) \wedge t) \vee (\bar{x})) \to y)) \vee y)$;
   б) $((x \to z) \to (((y \sim (\bar{z})) \wedge t)) \vee ((\bar{t} \wedge x) \sim y))$;
   в) $(y \wedge (\bar{z}) \vee (((\bar{x}) \to z)) \vee ((\bar{t} \wedge y)) \vee ((\bar{y}) \sim (t \vee x)))$.
2. Визначте інтерпретації, на яких виконуються співвідношення:
   а) $x \wedge (x \to y) \to (x \to z) = 0$;
   б) $(x \to y) \wedge (t \to x) \vee \bar{t} = 1$;
   в) $(\bar{x} \downarrow y) \wedge (x \sim y) = 1$.
3. Побудуйте таблиці істинності їх функцій та визначте їхній порядковий номер:
   а) $f(x, y) = (x \to y) \wedge (y \to x)$;
   б) $f(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$;
   в) $f(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$;
   г) $f(x, y, z) = (x \wedge \bar{y}) \vee (x \wedge \bar{z}) \vee (y \wedge \bar{z})$;

д) $f(x, y) = x \mid (x \mid y)$;
е) $f(x, y) = (x \mid y) \mid (x \mid y)$.

4. Перевірте за допомогою таблиць істинності, чи справедливі такі співвідношення:
а) $x \vee (y \sim z) = (x \vee y) \sim (x \vee z)$;
б) $x \to (y \sim z) = (x \to y) \sim (x \to z)$;
в) $x \wedge (y \sim z) = (x \wedge y) \sim (x \wedge z)$;
г) $x \to (y \vee z) = (x \to y) \vee (x \to z)$;
д) $x \to (y \wedge z) = (x \to y) \wedge (x \to z)$;
е) $x \oplus (y \wedge z) = (x \oplus y) \wedge (x \oplus z)$;
ж) $x \to (y \to z) = (x \to y) \to (x \to z)$.

4.3. Двоїстість

Двоїсті та самодвоїсті булеві функції, принцип двоїстості

В основі булевої алгебри, як і деяких інших алгебраїчних структурах, лежить принцип двоїстості.

Визначення
Функція $f^*(x_1, \ldots, x_n)$ називається ***двоїстою*** до функції $f(x_1, \ldots, x_n)$, якщо

$$f^*(x_1, \ldots, x_n) = \overline{f}(\overline{x}_1, \ldots, \overline{x}_n). \qquad (4.1)$$

Доведемо, що $(f^*)^* = f$, для цього за формулою 4.1 знайдемо функцію, що двоїста f^* і, використовуючи закон подвійного заперечення $f = \overline{\overline{f}}$ (докладніше буде розглянуто нижче, в п. 4.4), одержимо:

$$(f^*(x_1, \ldots, x_n))^* = (\overline{f}(\overline{x}_1, \ldots, \overline{x}_n))^* = (\overline{\overline{f}}(\overline{\overline{x}}_1, \ldots, \overline{\overline{x}}_n)) =$$
$$= f(x_1, \ldots, x_n).$$

Таким чином, відношення двоїстості між функціями симетрично. Із визначення двоїстості маємо, що для будь-якої функції двоїста їй визначається однозначно.

Визначення
Функція, двоїста сама собі, тобто $f = f^*$, називається ***самодвоїстою.***

Сумістимо таблиці істинності булевої функції $f(x, y)$ і двоїстої функції $f^*(x, y)$, враховуючи у стовпчику значень f^* рівність $f(\overline{0}, \overline{0}) = \overline{f}(1, 1)$, $f(\overline{1}, \overline{0}) = \overline{f}(0, 1)$ тощо. В стовпцях f і f^* сумісної таблиці 4.7 кожне значення функції $f(x, y)$ дорівнює

запереченню симетричного йому значення функції $f^*(x, y)$ відносно горизонтальної лінії середини таблиці (в таблиці 4.7 симетричні значення з'єднані стрілками).

Таблиця 4.7. Порівняння двоїстих функцій

x	y	$f(x, y)$	$f^*(x, y)$
0	0	$f(0, 0)$	$\bar{f}(1, 1)$
0	1	$f(0, 1)$	$\bar{f}(1, 0)$
1	0	$f(1, 0)$	$\bar{f}(0, 1)$
1	1	$f(1, 1)$	$\bar{f}(0, 0)$

Таким чином, щоб побудувати таблицю істинності функції, що двоїста даній, необхідно побудувати таблицю істинності заданої функції, кожне значення булевої функції замінити на протилежне і записати одержаний стовпчик у зворотній послідовності.

Приклад. Знайти функцію, яка двоїста функції $f(x, y, z)$, якщо відомо, що $f(x, y, z) = 1$ тільки на інтерпретаціях (001), (011), (111).

Для стовпця значень f генеруємо набір протилежних (інверсних) значень (10101110). Записавши його у зворотній послідовності, одержимо таким чином стовпчик значення двоїстої функції f^* (таблиця 4.8).

Таблиця 4.8. Таблиця істинності двоїстих функцій

x	y	z	$f(x, y, z)$	$f^*(x, y, z)$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

Розглянемо таблицю істинності самодвоїстої функції (таблиця 4.9).

Таблиця 4.9. Самодвоїста функція

x	y	$f(x, y) = f^*(x, y)$
0	0	$f(0, 0) = \overline{f}(1, 1)$
0	1	$f(0, 1) = \overline{f}(1, 0)$
1	0	$f(1, 0) = \overline{f}(0, 1)$
1	1	$f(1, 1) = \overline{f}(0, 0)$

Проведемо лінію симетрії посередині таблиці. Як видно з таблиці 4.9, кожне значення функції дорівнює запереченню симетричного йому значення. Таким чином, за таблицею істинності функції завжди можна визначити, є дана функція самодвоїстою чи ні.

Приклад. Функції $f(x, y, z)$ і $g(x, y, z)$ задані таблицями істинності (таблиця 4.10). Визначити, чи є дані функції самодвоїстими.

Таблиця 4.10. Функції $f(x, y, z)$ и $g(x, y, z)$

x	y	z	$f(x, y, z)$	$g(x, y, z)$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

З таблиці 4.10 видно, що кожне значення функції $f(x, y, z)$ є запереченням симетричного йому значення. Отже, функція $f(x, y, z)$ є самодвоїстою. Для функції $g(x, y, z)$ є значення функції, що не дорівнюють запереченню симетричних їм значень, наприклад: $g(0, 0, 0) = 0$, а симетричне йому значення $g(1, 1, 1) = 0$. Отже, функція $g(x, y, z)$ не є самодвоїстою.

Значення самодвоїстої функції повністю визначаються її значеннями на половині інтерпретацій. Всього маємо 2^n інтерпретацій функції n змінних. Половина даної кількості буде становити $2^n/2 = 2^{n-1}$. Таким чином, кількість самодвоїстих функцій від n змінних дорівнює $2^{2^{n-1}}$.

Нехай функція F задана як суперпозиція функцій f_0 і функцій $f_1, \ldots, f_n$: $F = f_0(f_1, \ldots, f_n)$. Функцію F^*, що двоїста F, можна одержати, замінивши в формулі F функції $f_0; f_1, \ldots, f_n$ на двоїсті до них $f_0^*; f_1^*, \ldots, f_n^*$.

Дійсно, якщо внутрішні функції f_i, $i = \overline{1, n}$ залежать від m аргументів $x_1, \ldots, x_m$, то, за визначенням, для результуючої функції $F = F(x_1, \ldots, x_m)$ двоїста F^* має вигляд:

$$F^* = \bar{F}(\bar{x}_1, \ldots, \bar{x}_m) = \bar{f}_0(f_1(\bar{x}_1, \ldots, \bar{x}_m), \ldots, f_n(\bar{x}_1, \ldots, \bar{x}_m)),$$

причому

$$f_i(\bar{x}_1, \ldots, \bar{x}_m) = \bar{f}_i(\bar{x}_1, \ldots, \bar{x}_m) = \bar{\bar{f}}_i^*(x_1, \ldots, x_m).$$

З іншого боку,

$$f_0^*(f_1^*, \ldots, f_n^*) = \bar{f}_0(\bar{f}_1^*, \ldots, \bar{f}_n^*) = \bar{f}_0(f_1(\bar{x}_1, \ldots, \bar{x}_m), \ldots, f_n(\bar{x}_1, \ldots, \bar{x}_m)).$$

Таким чином,

$$F^* = f_0^*(f_1^*, \ldots, f_n^*).$$

Методом математичної індукції це правило переходу до двоїстої функції поширюється на «багатоступінчасті» суперпозиції булевих функцій, коли «внутрішні» функції $f_1, \ldots, f_n$, в свою чергу, є суперпозиціями інших функцій тощо.

Вкажемо функції, що двоїсті до «елементарних» функцій логіки $\wedge$, $\vee$, $\bar{}$, константа 0, константа 1:

$f(x, y) = x \wedge y;$ $\qquad f^*(x, y) = \overline{\bar{x} \wedge \bar{y}} = x \vee y;$

$f(x) = \bar{x};$ $\qquad f^*(x) = \bar{\bar{\bar{x}}} = \bar{x} = f(x);$

$f(x) = 0;$ $\qquad f^*(x) = \bar{0} = 1 = \overline{f(x)}.$

Оскільки відношення двоїстості симетрично, можна сказати, що кон'юнкція двоїста диз'юнкції, диз'юнкція двоїста кон'юнкції, функція «0» двоїста функції «1», і навпаки, а заперечення — самодвоїста функція.

Звідси виходить так званий ***принцип двоїстості***, що вказує правило одержання двоїстих формул у булевій алгебрі: «Для того щоб одержати двоїсту формулу булевої алгебри, необхідно замінити в ній всі кон'юнкції на диз'юнкції, диз'юнкції на кон'юнкції, 0 на 1, 1 на 0 і використувати дужки, де необхідно, щоб порядок використання операцій залишився попереднім».

Приклад. Знайти функцію, двоїсту функції $f = x \vee \bar{y}z \vee 0$.

Скористаємося наведеним вище правилом одержання двоїстих функцій:

$$f^* = (x \vee (\bar{y}z) \vee 0)^* = x \wedge (\bar{y} \vee z) \wedge 1.$$

Наслідок. *Якщо функції рівні, то і двоїсті їм функції також рівні*:

якщо $f_1 = f_2$, то $f_1^* = f_2^*$.

Запитання

1. Дайте визначення двоїстої функції.
2. Які функції називаються самодвоїстими?
3. Яким чином формується таблиця істинності двоїстої функції?
4. Визначте, використовуючи таблицю істинності булевої функції, чи є вона самодвоїстою чи ні.
5. Сформулюйте принцип двоїстості.
6. Яким чином можна аналітичним шляхом одержати із заданої формулою функції двоїсту до неї?

Завдання

1. Визначте кількість наборів, на яких самодвоїста функція n змінних дорівнює 1.
2. Знайти двоїсті формули до таких функцій:
   а) $(x \wedge (y \vee z)) \vee \bar{x} \wedge \bar{y}$;
   б) $xy \vee yz \vee xz$;
   в) $x\bar{y} \vee x \vee y \vee zt$.
3. Визначте, чи є такі функції самодвоїстими:
   а) $f(x, y) = (\bar{x} \vee y) \wedge (\bar{y} \vee x)$;
   б) $f(x, y, z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$;
   в) $f(x, y, z) = (\bar{x} \wedge \bar{y}) \vee (x \wedge z) \vee (\bar{y} \wedge \bar{z})$;
   г) $f(x, y, z) = (x \wedge \bar{y}) \vee (x \wedge \bar{z}) \vee (y \wedge \bar{z})$;
   д) $f(x, y) = x \vee \overline{(x \wedge y)}$.

4.4. Закони булевої алгебри

У цій главі буде перевірено виконання законів булевої алгебри 1–4, а також 5–9 з п. 4.2.3 для двохелементної алгебраїчної структури $(B, \wedge, \vee, \bar{\ })$, де $\wedge$, $\vee$, $\bar{\ }$ є операції кон'юнкції f_1, диз'юнкції f_7 та заперечення φ_2, що задані таблицями істинності (таблиці 4.1, 4.2). В подальшому, якщо не оговорене

протилежне, ми будемо називати двохелементну булеву алгебру $(B, \wedge, \vee, \bar{})$ просто «булевою алгеброю».

1. ***Комутативність кон'юнкції та диз'юнкції***

$$x \vee y = y \vee x; \quad x \wedge y = y \wedge x.$$

Таблиця 4.11. Таблиці істинності

x	y	$x \vee y$	$y \vee x$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Сумістимо таблиці істинності для обох частин першої рівності (таблиця 4.11).

Стовпці $x \vee y$ и $y \vee x$ у таблицях істинності містять однакові значення, що доводить комутативність операції диз'юнкції. Знайдемо тотожність, двоїсту даній, для чого замінимо всі функції на двоїсті їм:

$$(x \vee y)^* = (y \vee x)^*, \quad x \wedge y = y \wedge x.$$

2. ***Асоціативність кон'юнкції та диз'юнкції***

$$x \vee (y \vee z) = (x \vee y) \vee z; \quad x \wedge (y \wedge z) = (x \wedge y) \wedge z.$$

Створимо таблиці істинності для лівої та правої частин другої рівності (таблиця 4.12).

Таблиця 4.12. Доведення асоціативності кон'юнкції

x	y	z	$y \wedge z$	$x \wedge (y \wedge z)$	$x \wedge y$	$(x \wedge y) \wedge z$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	1	0
1	1	1	1	1	1	1

Стовпчики, що відповідають лівій та правій частинам другої рівності, містять однакові значення, що доводить справедливість тотожності $x \wedge (y \wedge z) = (x \wedge y) \wedge z$. Тотожність, що виражає асоціативність диз'юнкції, двоїста доведеній, оскільки може бути одержана з неї шляхом заміни всіх функцій на двоїсті їм, і, отже, вона теж правильна.

3. ***Дистрибутивність кон'юнкції та диз'юнкції відносно одна одній***

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z);$$

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z).$$

Доведемо дистрибутивність кон'юнкції щодо диз'юнкції, використовуючи таблицю істинності (таблиця 4.13).

Таблиця 4.13. Дистрибутивність кон'юнкції щодо диз'юнкції

x	y	z	$y \vee z$	$x \wedge (y \vee z)$	$x \wedge y$	$x \wedge z$	$(x \wedge y) \vee (x \wedge z)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Стовпчики, які відповідають лівій та правій частинам першої рівності, містять однакові значення, що і доводить справедливість рівності $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$. Двоїста тотожність виражає дистрибутивність диз'юнкції відносно кон'юнкції.

4. ***Ідемпотентність кон'юнкції та диз'юнкції***

$$x \vee x = x; \quad x \wedge x = x.$$

Побудуємо таблиці істинності функцій з лівих частин рівності (таблиця 4.14).

В таблиці значення всіх стовпців однакові і співпадають із значенням змінної x, що і доводить обидві тотожності. Дані тотожності так, як і розглянуті вище, є двоїстими одна одній.

Таблиця 4.14. Закон ідемпотентності

x	$x \vee x$	$x \wedge x$
0	0	0
1	1	1

5. ***Закон виключеного третього***

$$x \vee \bar{x} = 1.$$

Доведемо цей закон, використовуючи таблицю істинності (таблиця 4.15).

Таблиця 4.15. Закон виключеного третього

x	$\bar{x}$	$x \vee \bar{x}$
0	1	1
1	0	1

Стовпчик таблиці істинності, який зображує ліву частину тотожності, що доводиться, дорівнює константі одиниці, що і треба було довести.

6. ***Закон протиріччя***

$$x \wedge \bar{x} = 0.$$

Цей закон є двоїстим до доведеного вище закону виключеного третього.

7. ***Тотожності з константами***

$$x \vee 0 = x; \quad x \wedge 1 = x; \quad x \vee 1 = 1; \quad x \wedge 0 = 0.$$

Побудуємо таблиці істинності для лівих частин тотожностей (таблиця 4.16).

Таблиця 4.16. Таблиці істинності тотожностей

x	$x \vee 0$	$x \wedge 1$	$x \vee 1$	$x \wedge 0$
0	0	0	1	0
1	1	1	1	0

Одержана таблиця доводить справедливість даних тотожностей.

8. ***Закони елімінації***

$$x \wedge (x \vee y) = x; \quad x \vee (x \wedge y) = x.$$

Доведемо цей закон аналітично, використовуючи тотожності з константами і дистрибутивний закон:

$$x \wedge (x \vee y) = (x \vee 0) \wedge (x \vee y) = x \vee (0 \wedge y) = x \vee 0 = x;$$

$$x \vee (x \wedge y) = (x \wedge 1) \vee (x \wedge y) = x \wedge (1 \vee y) = x \wedge 1 = x.$$

9. ***Закон подвійного заперечення***

$$\bar{\bar{x}} = x.$$

Побудуємо відповідну таблицю істинності (таблиця 4.17).

Одержана таблиця істинності доводить справедливість тотожності.

Таблиця 4.17. Закон подвійного заперечення

x	$\bar{x}$	$\bar{\bar{x}}$
0	1	0
1	0	1

Наслідок. *Якщо до деякої частини A формули F булевої алгебри операція заперечення застосована більше одного разу, то можна видалити будь-яке парне число даних операцій і значення формули F не зміниться.*

10. ***Закони де Моргана***

$$\overline{x \vee y} = \bar{x} \wedge \bar{y}; \quad \overline{x \wedge y} = \bar{x} \vee \bar{y}.$$

Доведемо першу з наведених тотожностей за допомогою таблиці істинності (таблиця 4.18).

Таблиця 4.18. Доведення закону де Моргана

x	y	$x \vee y$	$\overline{x \vee y}$	$\bar{x}$	$\bar{y}$	$\bar{x} \wedge \bar{y}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Побудована таблиця істинності доводить справедливість тотожності. Другий закон де Моргана є двоїстим першому і, отже, також є правильним.

Запитання

1. Назвіть основні закони булевої алгебри.
2. Якими способами можна довести закони булевої алгебри?
3. Яким чином принцип двоїстості застосовано до законів булевої алгебри?
4. Сформулюйте і запишіть тотожності для кожного з десяти законів булевої алгебри.

Завдання

1. Спростіть за допомогою законів логіки Буля нижченаведені вирази. Потім за допомогою таблиць істинності порівняйте одержані вирази з вихідними:

а) $(x \vee (\bar{t} \wedge y)) \wedge ((\bar{x} \wedge (\bar{y} \vee t)) \vee z)) \vee \bar{z} \vee (x \vee (y \wedge \bar{t}))$;

б) $((x \vee z) \wedge (x \vee t)) \wedge (((z \vee (z \wedge y)) \wedge \bar{z}) \vee \bar{x})$;

в) $(\bar{y} \vee t) \wedge ((\bar{x} \wedge z) \vee (x \wedge z) \vee (\bar{t} \wedge \bar{z}) \vee (x \wedge \bar{z})) \wedge (y \vee t)$;

г) $(x \vee \bar{z}) \wedge (\bar{x} \vee \bar{y}) \wedge (\bar{y} \vee z) \wedge (\bar{x} \vee y) \wedge (y \vee z)$;

д) $(x \wedge z) \vee ((y \vee \bar{t}) \wedge (\bar{x} \vee \bar{t}) \wedge (t \vee y) \wedge (\bar{x} \vee t) \vee (x \wedge \bar{z}))$;

е) $((\bar{y} \vee \bar{z}) \wedge (x \vee y)) \vee (t \wedge \bar{z}) \vee (((\bar{y} \wedge \bar{x}) \vee z) \wedge (x \vee y))$;

ж)$(x \wedge \bar{z}) \vee (\bar{x} \wedge \bar{y}) \vee (y \wedge z) \vee (\bar{x} \wedge y) \vee (z \wedge \bar{y})$;

з) $((x \vee (z \vee (y \wedge z))) \wedge (\overline{z \wedge t}) \wedge (z \wedge \bar{t})) \wedge (z \vee (\bar{t} \wedge \bar{z}) \vee t)$;

і) $((x \vee \bar{x}) \wedge (\bar{y} \vee \bar{t}) \wedge (\bar{y} \vee \bar{z}) \wedge (\bar{z} \vee t)) \vee ((\bar{y} \vee z) \wedge (z \vee t))$;

к) $(x \vee \bar{z}) \wedge ((\bar{x} \wedge t) \vee (y \wedge t) \vee (\bar{x} \wedge \bar{t}) \vee (y \wedge \bar{t})) \wedge (x \vee z)$.

4.5. Диз'юнктивні та кон'юктивні розкладання булевих функцій

Теореми розкладання, елементарні кон'юнкція і диз'юнкція, конституенти нуля та одиниці, нормальні форми

Серед множин еквівалентних формул, що зображують обрану булеву функцію f, виділяється одна формула, яка називається ***досконалою нормальною формою*** функції f. Вона має регламентовану логічну структуру і однозначно визначається за функцією f (п. 4.6), а її побудова заснована на рекурентному застосуванні теорем про спеціальні розкладання булевої функції за змінними, які наводяться в цьому розділі.

Для спрощення математичних викладень введемо двійковий параметр σ і позначення x^σ таким чином:

$$x, \sigma \in B = \{0, 1\},$$

$$x^\sigma = \begin{cases} \bar{x}, & \text{если } \sigma = 0, \\ x, & \text{если } \sigma = 1. \end{cases}$$

Можемо зробити висновок, що

$$x^\sigma = \begin{cases} 1, & x = \sigma, \\ 0, & x \neq \sigma. \end{cases}$$

Безпосередньо перевіряється, що бінарна функція $f(x, \sigma) = x^\sigma$ зображується формулою:

$$x^\sigma = x\,\sigma \vee \bar{x}\,\bar{\sigma}. \qquad (4.2)$$

Теорема 1. *Про диз'юнктивне розкладання булевої функції $f(x_1, x_2, \ldots, x_n)$ за k змінними*

Будь-яку булеву функцію $f(x_1, x_2, \ldots, x_n)$ можна зобразити в такій формі:

$$f(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) =$$

$$= \bigvee_{(\sigma_1, \sigma_2, \ldots, \sigma_k)} x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_k^{\sigma_k} \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_k, x_{k+1}, \ldots, x_n). \qquad (4.3)$$

Запис $\bigvee_{(\sigma_1, \sigma_2, \ldots, \sigma_k)}$ означає багатократну диз'юнкцію, яка береться за всіма можливими наборами значень $(\sigma_1, \sigma_2, \ldots, \sigma_k)$ при будь-якому k $(1 \le k \le n)$.

□ *Доведення.* Необхідно перевірити, що на довільній інтерпретації $(a_1, a_2, \ldots a_n)$ ліва і права частини рівності (4.3) приймають однакові значення. Підставивши значення $a_1, a_2, \ldots a_n$ замість $x_1, x_2, \ldots, x_n$, одержимо:

$$f(a_1, a_2, \ldots a_n) =$$

$$= \bigvee_{(\sigma_1, \sigma_2, \ldots, \sigma_k)} a_1^{\sigma_1} \wedge a_2^{\sigma_2} \wedge \ldots \wedge a_k^{\sigma_k} \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_k, a_{k+1}, \ldots, a_n).$$

За формулою (4.2) можемо записати, що

$$a_i^{\sigma_i} = \begin{cases} 1, & a_i = \sigma_i \\ 0, & a_i \ne \sigma_i \end{cases}.$$

Кон'юнкція $a_1^{\sigma_1} \wedge a_2^{\sigma_2} \wedge \ldots \wedge a_k^{\sigma_k}$ дорівнює нулю, якщо хоча б один елемент кон'юнкції $a_i^{\sigma_i}$ дорівнює нулю, яка відбудеться, якщо $a_i \ne \sigma_i$. Якщо ж $(a_1, a_2, \ldots a_k)$ співпадає з $(\sigma_1, \sigma_2, \ldots, \sigma_k)$, то $a_1^{\sigma_1} \wedge a_2^{\sigma_2} \wedge \ldots \wedge a_k^{\sigma_k} = 1$.

Звідси виходить, що

$$a_1^{\sigma_1} \wedge a_2^{\sigma_2} \wedge \ldots \wedge a_k^{\sigma_k} \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_k, a_{k+1}, \ldots, a_n) =$$
$$= 0 \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_k, a_{k+1}, \ldots, a_n) = 0$$

при $(a_1, a_2, \ldots a_k) \ne (\sigma_1, \sigma_2, \ldots, \sigma_k)$.

Якщо $(a_1, \ldots a_k) = (\sigma_1, \ldots, \sigma_k)$, то

$$a_1^{\sigma_1} \wedge a_2^{\sigma_2} \wedge \ldots \wedge a_k^{\sigma_k} \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_k, a_{k+1}, \ldots, a_n) =$$
$$= 1 \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_k, a_{k+1}, \ldots, a_n) = f(a_1, a_2, \ldots a_n).$$

Виходячи з того, запишемо значення правої частини тотожності (4.3):

$$\bigvee_{(\sigma_1, \sigma_2, \ldots, \sigma_k)} a_1^{\sigma_1} \wedge a_2^{\sigma_2} \wedge \ldots \wedge a_k^{\sigma_k} \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_k, a_{k+1}, \ldots, a_n) =$$

(розпишемо багатократну диз'юнкцію $\bigvee$ на наборах $(\sigma_1, \sigma_2, \ldots, \sigma_k)$)

$$= 0 \vee \ldots \vee 0 \vee f(a_1, a_2, \ldots a_n) \vee 0 \vee \ldots \vee 0 = f(a_1, a_2, \ldots a_n).$$

Отже, значення правої частини тотожності (4.3) співпадає зі значенням лівої частини на довільній інтерпретації. Теорему доведено. ■

Приклад. Запишемо диз'юнктивне розкладання функції $f(x, y, z, t) = \overline{(x \wedge y \vee \bar{z})} \wedge t$ за змінними x, z.

Розв'язок. Скористаємося теоремою про розкладання:

$$f(x, y, z, t) = \bigvee_{(\sigma_1, \sigma_2)} z_1^{\sigma_1} \wedge z_2^{\sigma_2} \wedge f(\sigma_1, y, \sigma_2, t) =$$

$$= \bar{x} \wedge \bar{z} \wedge f(0, y, 0, t) \vee \bar{x} \wedge z \wedge f(0, y, 1, t) \vee$$

$$\vee x \wedge \bar{z} \wedge f(1, y, 0, t) \vee x \wedge z \wedge f(1, y, 1, t).$$

Обчислимо:

$$f(0, y, 0, t) = \overline{(0 \wedge y \vee 0)} \wedge t = 0,$$

$$f(0, y, 1, t) = \overline{(0 \wedge y \vee 1)} \wedge t = t,$$

$$f(1, y, 0, t) = \overline{(1 \wedge y \vee 0)} \wedge t = 0,$$

$$f(1, y, 1, t) = \overline{(1 \wedge y \vee 1)} \wedge t = \bar{y} \wedge t.$$

Підставимо одержані значення $f(0, y, 0, t)$, $f(0, y, 1, t)$, $f(1, y, 0, t)$, $f(1, y, 1, t)$ у формулу диз'юнктивного розкладання за змінними x, z:

$$f(x, y, z, t) = \bar{x} \wedge \bar{z} \wedge 0 \vee \bar{x} \wedge z \wedge t \vee x \wedge \bar{z} \wedge 0 \vee x \wedge z \wedge (\bar{y} \wedge t) =$$

$$= \bar{x} \wedge z \wedge t \vee x \wedge z \wedge \bar{y} \wedge t.$$

Розглянемо наслідки з теореми 1.

Наслідок 1. *Диз'юнктивне розкладання булевої функції* $f(x_1, x_2, \ldots, x_n)$ за *одною змінною.*

Будь-яку булеву функцію $f(x_1, x_2, \ldots, x_n)$ можна зобразити у такій формі:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{\sigma_i} x_i^{\sigma_i} \wedge f(x_1, x_2, \ldots, x_{i-1}, \sigma_i, x_{i+1}, \ldots, x_n). \quad (4.4)$$

Запис означає, що диз'юнкція береться за всіма значеннями σ_i, тобто 0 і 1. Запис $f(x_1, x_2, \ldots, x_{i-1}, \sigma_i, x_{i+1}, \ldots, x_n)$ позначає значення функції на наборі $x_1, x_2, \ldots, x_n$, де замість значення змінної x_i підставлено σ_i.

Приклад. Розглянемо функцію $f(x, y, z, t) = \overline{(xy \vee \bar{z})}t$. Отримати диз'юнктивне розкладання цієї функції за змінною x.

Розв'язок. Скористаємося наслідком теореми про розкладання:

$$f(x, y, z, t) = \bar{x} \wedge f(0, y, z, t) \vee x \wedge f(1, y, z, t).$$

Обчислимо значення функції $f(x, y, z, t)$ при $x = 0$ і $x = 1$:

$$f(0, y, z, t) = \overline{(0 \wedge y \vee \bar{z})}t = zt;$$

$$f(1, y, z, t) = \overline{(1 \wedge y \vee \bar{z})}t = \bar{y} \wedge z \wedge t.$$

Підставимо одержані значення $f(0, y, z, t)$ і $f(1, y, z, t)$ у формулу диз'юнктивного розкладання:

$$f(x, y, z, t) = \bar{x} \wedge f(0, y, z, t) \vee x \wedge f(1, y, z, t) = \bar{x}zt \vee x\bar{y}zt.$$

Наслідок 2. *Диз'юнктивне розкладання булевої функції* $f(x_1, x_2, \ldots, x_n)$ *за всіма* n *змінними.*

Будь-яку булеву функцію $f(x_1, x_2, \ldots, x_n) \neq 0$ можна зобразити у такій формі:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{\substack{(\sigma_1, \sigma_2, \ldots, \sigma_n) \\ f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1}} x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_n^{\sigma_n}. \qquad (4.5)$$

Запис $\bigvee_{\substack{(\sigma_1, \sigma_2, \ldots, \sigma_n) \\ f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1}}$ означає, що диз'юнкція береться за всіма наборами значень $(\sigma_1, \sigma_2, \ldots, \sigma_n)$, на яких $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$.

Дійсно, запишемо співвідношення (4.3) для випадку $k = n$:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{(\sigma_1, \sigma_2, \ldots, \sigma_n)} x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_n^{\sigma_n} \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_n).$$

Зауважимо таке:

якщо $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 0$,

то $x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_n^{\sigma_n} \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 0$;

якщо $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$,

то $x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_n^{\sigma_n} \wedge f(\sigma_1, \sigma_2, \ldots, \sigma_n) = x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_n^{\sigma_n}$.

Тому диз'юнктивне розкладання за всіма змінними містить тільки кон'юнкції, що відповідають наборам $(\sigma_1, \sigma_2, \ldots, \sigma_n)$, для яких $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$.

Приклад. Розглянемо функцію $f(x, y, z) = xy \vee \bar{z}$. Отримати диз'юнктивне розкладання цієї функції за всіма змінними.

Розв'язок. Визначимо значення функції на кожній з інтерпретацій:

$$f(0, 0, 0) = 0 \wedge 0 \vee \bar{0} = 0 \vee 1 = 1,$$

$$f(0, 0, 1) = 0 \wedge 0 \vee \bar{1} = 0 \vee 0 = 0,$$

$$f(0, 1, 0) = 0 \wedge 1 \vee \bar{0} = 0 \vee 1 = 1,$$

$$f(0, 1, 1) = 0 \wedge 1 \vee \bar{1} = 0 \vee 0 = 0,$$
$$f(1, 0, 0) = 1 \wedge 0 \vee \bar{0} = 0 \vee 1 = 1,$$
$$f(1, 0, 1) = 1 \wedge 0 \vee \bar{1} = 0 \vee 0 = 0,$$
$$f(1, 1, 0) = 1 \wedge 1 \vee \bar{0} = 1 \vee 1 = 1,$$
$$f(1, 1, 1) = 1 \wedge 1 \vee \bar{1} = 1 \vee 0 = 1.$$

Використовуючи формулу (4.5), одержимо:

$$f(x, y, z) = x^0 y^0 z^0 \vee x^0 y^1 z^0 \vee x^1 y^0 z^0 \vee x^1 y^1 z^0 \vee x^1 y^1 z^1 =$$
$$= \bar{x}\,\bar{y}\,\bar{z} \vee \bar{x}\,y\,\bar{z} \vee x\,\bar{y}\,\bar{z} \vee x\,y\,\bar{z} \vee x\,y\,z.$$

Визначення

Елементарною кон'юнкцією називається кон'юнкція будь-якого числа булевих змінних, що взяті із запереченням або без нього, в якій кожна змінна зустрічається не більше одного разу. Елементарною кон'юнкцією, що містить нуль змінних, будемо вважати константу 1.

Приклад. Елементарними кон'юнкціями для функції від однієї змінної можуть бути y, $\bar{z}$, для функції від двох змінних — $\bar{x} \wedge y$, $x \wedge \bar{z}$, для функції від трьох змінних — $x \wedge \bar{y} \wedge z$, $x \wedge \bar{y} \wedge \bar{z}$, $x \wedge y \wedge z$ та інші.

Визначення

Диз'юнктивною нормальною формою (ДНФ) називається формула, що зображена у вигляді диз'юнкції елементарних кон'юнкцій.

Визначення

Елементарна кон'юнкція $x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_n^{\sigma_n}$ називається ***конституентою одиниці (мінтермом)*** функції $f(x_1, x_2, \ldots, x_n)$, якщо $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$, тобто інтерпретація, яка обертає в одиницю дану елементарну кон'юнкцію, обертає в одиницю і функцію f.

Конституента одиниці має такі властивості:

1. Конституента одиниці дорівнює одиниці тільки на відповідній їй інтерпретації.

2. Значення конституенти одиниці однозначно визначається номером відповідної інтерпретації.

3. Кон'юнкція будь-якого числа різних конституент одиниці функції дорівнює нулю.

Приклад. Елементарна кон'юнкція $x \wedge \bar{y}$ є конституентою одиниці функції двох змінних $f(x, y)$ на інтерпретації (1, 0), оскільки $x \wedge \bar{y} = x^1 \wedge y^0$ і $x \wedge \bar{y} = 1$. Елементарна кон'юнкція $x \wedge y \wedge z$ є конституентою одиниці функції трьох змінних $f(x, y, z)$ на інтерпретації (1, 1, 1), оскільки $x \wedge y \wedge z = x^1 \wedge y^1 \wedge z^1$ і $x \wedge y \wedge z = 1$.

Визначення

Досконалою диз'юнктивною нормальною формою (ДДНФ) булевої функції називається формула, що зображена у вигляді диз'юнкції конституент одиниці даної функції.

ДДНФ функції є результатом диз'юнктивного розкладання функції за всіма змінними і відповідає формулі (4.5). Як видно з визначення, ДДНФ функції містить тільки $\wedge$, $\vee$, $\bar{}$, отже, застосувавши до ДДНФ принцип двоїстості, можна одержати двоїсте зображення, яке називається ***кон'юнктивним розкладанням.***

Теорема 2. *Про кон'юнктивне розкладання булевої функції $f(x_1, x_2, \ldots, x_n)$ за k змінними*

Будь-яку булеву функцію $f(x_1, x_2, \ldots, x_n)$ можна зобразити у такій формі:

$$f(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n) =$$

$$= \bigwedge_{(\sigma_1, \sigma_2, \ldots, \sigma_k)} x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee \ldots \vee x_k^{\bar{\sigma}_k} \vee f(\sigma_1, \sigma_2, \ldots, \sigma_k, x_{k+1}, \ldots, x_n). \quad (4.6)$$

Запис $\bigwedge_{(\sigma_1, \sigma_2, \ldots, \sigma_k)}$ означає, що кон'юнкція береться за всіма можливими наборами значень $(\sigma_1, \sigma_2, \ldots, \sigma_k)$.

□ *Доведення.* Доведемо теорему, використовуючи принцип двоїстості і тотожність (4.3). Запишемо диз'юнктивне розкладання функції $f^*(x_1, x_2, \ldots, x_n)$, яка двоїста функції $f(x_1, x_2, \ldots, x_n)$.

$$f^*(x_1, x_2, \ldots, x_n) =$$

$$= \bigvee_{(\sigma_1, \sigma_2, \ldots, \sigma_k)} x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_k^{\sigma_k} \wedge f^*(\sigma_1, \sigma_2, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

Тотожність, двоїста до останньої, має вигляд:

$$f^{**}(x_1, x_2, \ldots, x_n) =$$

$$= \bigwedge_{(\sigma_1, \sigma_2, \ldots, \sigma_k)} (x_1^{\sigma_1})^* \vee (x_2^{\sigma_2})^* \vee \ldots \vee (x_k^{\sigma_k})^* \vee f^{**}(\sigma_1, \sigma_2, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

В лівій частині рівності одержимо $f(x_1, x_2, \ldots, x_n)$. У правій частині перейдемо до двоїстих виразів для $x_i^{\sigma_i}$. Для цього запишемо формулу (4.2) і знайдемо двоїсту до неї шляхом заміни всіх вхідних до неї функцій на двоїсті:

$$(x_i^{\sigma_i})^* = (x_i\sigma_i \vee \bar{x}_i\bar{\sigma}_i)^* = (x_i \vee \sigma_i)(\bar{x}_i \vee \bar{\sigma}_i) =$$

$$= x_i\bar{\sigma}_i \vee \bar{x}_i\sigma_i = x_i(\bar{\sigma}_i) \vee \bar{x}_i(\bar{\sigma}_i) = x_i^{\bar{\sigma}_i}.$$

Таким чином, рівність з f^{**} приймає форму (4.6), що і треба було довести. ■

Приклад. Розглянемо кон'юнктивне розкладання функції $f(x, y, z, t) = (xy \vee \bar{z})t$ за змінними x, t.

Розв'язок. За формулою (4.6) маємо:

$$f(x, y, z, t) = (x^{\bar{0}} \vee t^{\bar{0}} \vee f(0, y, z, 0))(x^{\bar{0}} \vee t^{\bar{1}} \vee$$

$$\vee f(0, y, z, 1))(x^{\bar{1}} \vee t^{\bar{0}} \vee f(1, y, z, 0))(x^{\bar{1}} \vee t^{\bar{1}} \vee f(1, y, z, 1)) =$$

$$= (x \vee t \vee f(0, y, z, 0))(x \vee \bar{t} \vee f(0, y, z, 1))(\bar{x} \vee t \vee$$

$$\vee f(1, y, z, 0))(\bar{x} \vee \bar{t} \vee f(1, y, z, 1)).$$

Використовуючи формули

$$f(0, y, z, 0) = (0 \wedge y \vee \bar{z}) \wedge 0 = 0,$$

$$f(0, y, z, 1) = (0 \wedge y \vee \bar{z}) \wedge 1 = \bar{z},$$

$$f(1, y, z, 0) = (1 \wedge y \vee \bar{z}) \wedge 0 = 0,$$

$$f(1, y, z, 1) = (1 \wedge y \vee \bar{z}) \wedge 1 = y \vee \bar{z},$$

які індукує функція f при спеціальних значеннях змінних x, t, одержимо шукане кон'юнктивне розкладання:

$$f(x, y, z, t) = (x \vee t \vee 0)(x \vee \bar{t} \vee \bar{z})(\bar{x} \vee t \vee 0)(\bar{x} \vee \bar{t} \vee (y \vee \bar{z})) =$$

$$= (x \vee t)(x \vee \bar{t} \vee \bar{z})(\bar{x} \vee t)(\bar{x} \vee \bar{t} \vee y \vee \bar{z}).$$

Наслідок 3. *Кон'юнктивне розкладання булевої функції* $f(x_1, x_2, \ldots, x_n)$ за *однією змінною.*

Будь-яку булеву функцію $f(x_1, x_2, \ldots, x_n)$ можна зобразити у такій формі:

$$f(x_1, x_2, \ldots, x_n) = \bigwedge_{\sigma_i} x_i^{\bar{\sigma}_i} \vee f(x_1, x_2, \ldots x_{i-1}, \sigma_i, x_{i+1}, \ldots, x_n). \quad (4.7)$$

Запис $\bigwedge\limits_{\sigma_i}$ означає, що кон'юнкція береться за двома можливими значеннями σ_i, тобто 0 і 1. Індекс i є фіксованим.

Приклад. Одержати кон'юнктивне розкладання функції $f(x, y, z) = xy \vee \bar{z}$ за змінною x.

Розв'язок. Скористаємося попереднім наслідком:

$$f(x, y, z) = (x^0 \vee f(0, y, z)) \wedge (x^1 \vee f(1, y, z)) =$$

$$= (x \vee f(0, y, z)) \wedge (\bar{x} \vee f(1, y, z)) =$$

$$= (x \vee 0 \cdot y \vee \bar{z})(\bar{x} \vee 1 \cdot y \vee \bar{z}) = (x \vee \bar{z})(\bar{x} \vee y \vee \bar{z}).$$

Наслідок 4. *Кон'юнктивне розкладання булевої функції* $f(x_1, x_2, \ldots, x_n)$ *за всіма змінними.*

Будь-яку булеву функцію $f(x_1, x_2, \ldots, x_n) \neq 1$ можна зобразити у такій формі:

$$f(x_1, x_2, \ldots, x_n) = \bigwedge_{\substack{(\sigma_1, \sigma_2, \ldots, \sigma_n) \\ f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 0}} x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee \ldots \vee x_n^{\bar{\sigma}_n}. \qquad (4.8)$$

У правій частині кон'юнкція береться за всіма наборами значень $(\sigma_1, \sigma_2, \ldots, \sigma_k)$, на яких $f(x_1, x_2, \ldots, x_n) = 0$.

Розглянемо, яким чином формула (4.8) одержана з формули (4.6). При $k = n$

$$f(x_1, x_2, \ldots, x_n) = \bigwedge_{(\sigma_1, \sigma_2, \ldots, \sigma_n)} x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee \ldots \vee x_n^{\bar{\sigma}_n} \vee f(\sigma_1, \sigma_2, \ldots, \sigma_n).$$

Якщо $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 0$, то $x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee \ldots \vee x_n^{\bar{\sigma}_n} \vee f(\sigma_1, \sigma_2, \ldots, \sigma_n) = x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee \ldots \vee x_n^{\bar{\sigma}_n}$,

якщо $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$, то $x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee \ldots \vee x_n^{\bar{\sigma}_n} \vee f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$.

Тому кон'юнктивне розкладання за всіма змінними містить тільки диз'юнкції, що відповідають наборам $(\sigma_1, \sigma_2, \ldots, \sigma_n)$, для яких $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 0$.

Приклад. Одержати кон'юнктивне розкладання функції $f(x, y, z) = xy \vee \bar{z}$ за всіма змінними.

Розв'язок. Визначимо значення функції на кожній з інтерпретацій:

$$f(0, 0, 0) = 0 \wedge 0 \vee \bar{0} = 0 \vee 1 = 1,$$

$$f(0, 0, 1) = 0 \wedge 0 \vee \bar{1} = 0 \vee 0 = 0,$$

$$f(0, 1, 0) = 0 \wedge 1 \vee \bar{0} = 0 \vee 1 = 1,$$

$$f(0, 1, 1) = 0 \wedge 1 \vee \bar{1} = 0 \vee 0 = 0,$$
$$f(1, 0, 0) = 1 \wedge 0 \vee \bar{0} = 0 \vee 1 = 1,$$
$$f(1, 0, 1) = 1 \wedge 0 \vee \bar{1} = 0 \vee 0 = 0,$$
$$f(1, 1, 0) = 1 \wedge 1 \vee \bar{0} = 1 \vee 1 = 1,$$
$$f(1, 1, 1) = 1 \wedge 1 \vee \bar{1} = 1 \vee 0 = 1.$$

За формулою (4.8) одержуємо

$$f(x, y, z) = (x^1 \vee y^1 \vee z^0)(x^1 \vee y^0 \vee z^0)(x^0 \vee y^1 \vee z^0) =$$
$$= (x \vee y \vee \bar{z})(x \vee \bar{y} \vee \bar{z})(\bar{x} \vee y \vee \bar{z}).$$

Визначення

Елементарною диз'юнкцією називається диз'юнкція будь-якого числа булевих змінних, що взяті із запереченням або без нього, в якій кожна змінна зустрічається не більше одного разу. Елементарною диз'юнкцією, що містить нуль змінних, будемо вважати константу 0.

Приклад. Елементарними диз'юнкціями від однієї, двох та більше змінних є: y, $\bar{z}$, $\bar{x} \vee y$, $x \vee \bar{z}$, $\bar{x} \vee y \vee z$, $x \vee \bar{y} \vee \bar{z}$, $x \vee y \vee z$ і т. п.

Визначення

Кон'юнктивною нормальною формою (КНФ) називається формула, що зображена у вигляді кон'юнкції елементарних диз'юнкцій.

Визначення

Елементарна диз'юнкція $x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee \ldots \vee x_n^{\bar{\sigma}_n}$ називається ***конституентою нуля (макстермом)*** функції $f(x_1, x_2, \ldots, x_n)$, якщо $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 0$, тобто інтерпретація, яка обертає в нуль дану елементарну диз'юнкцію, обертає в нуль і функцію f.

Конституента нуля має такі властивості:

1. Конституента нуля дорівнює нулю тільки на відповідній їй інтерпретації.
2. Конституента нуля однозначно визначається номером відповідної їй інтерпретації.
3. Диз'юнкція будь-якого числа різних конституент нуля функції дорівнює одиниці.

Приклад. Елементарна диз'юнкція $x \vee \bar{y}$ є конституентою нуля функції двох змінних $f(x, y)$ на інтерпретації (0, 1), оскільки $x \vee \bar{y} = x^1 \vee y^0 = x^0 \vee y^1$, отже, на інтерпретації $(x, y) = (0, 1)$ виконано рівність $x \vee \bar{y} = 0$. Елементарна диз'юнкція $x \vee y \vee z$ є конституентою нуля функції трьох змінних $f(x, y, z)$ на інтерпретації (0, 0, 0). Дійсно, $x \vee y \vee z = x^1 \vee y^1 \vee z^1 = x^0 \vee y^0 \vee z^0$, отже, диз'юнкція $x \vee y \vee z$ дорівнює нулю на інтерпретації $(0, 0, 0) = (x, y, z)$.

Визначення

Досконалою кон'юнктивною нормальною формою (ДКНФ) функції називається формула, що зображена у вигляді кон'юнкції конституент нуля даної функції.

ДКНФ функції є результатом кон'юнктивного розкладання функції за всіма змінними і відповідає формулі (4.8). З аналізу формул (4.5) і (4.8), відповідних ДДНФ і ДКНФ функції, можна зробити такі висновки:

1. Для кожної булевої функції $f(x_1, x_2, \ldots, x_n)$, що не є константою нуль, існує зображення у вигляді ДДНФ.
2. Для кожної булевої функції $f(x_1, x_2, \ldots, x_n)$, що не є константою одиниця, існує зображення у вигляді ДКНФ.
3. Дві різні булеві функції не можуть мати однакові ДДНФ або ДКНФ.
4. Для кожної булевої функції $f(x_1, x_2, \ldots, x_n)$ існує зображення у вигляді формули булевої алгебри, що містить тільки операції диз'юнкції, кон'юнкції та заперечення.

Запитання

1. Запишіть формули диз'юнктивного розкладання булевих функцій від n змінних за k змінними ($k < n$) за всіма n змінними, за однією змінною.
2. Запишіть формули кон'юнктивного розкладання булевих функцій від n змінних за k змінними ($k < n$) за всіма n змінними, за однією змінною.
3. Дайте визначення таких понять: елементарна кон'юнкція, елементарна диз'юнкція, конституента одиниці, конституента нуля. Яким чином ці поняття пов'язані з диз'юнктивним і кон'юнктивним розкладанням булевих функцій?
4. Які властивості мають конституенти одиниці та конституенти нуля?

5. Сформулюйте визначення понять нормальних та досконалих нормальних форм булевих функцій. Поясніть їх зв'язок з диз'юнктивним та кон'юнктивним розкладанням булевих функцій.

Завдання

1. Випишіть конституенти одиниці булевої функції $f(x_1, x_2, x_3, x_4)$ з такого списку елементарних кон'юнкцій:

   а) $x_1\bar{x}_2\bar{x}_3$; б) $x_1\bar{x}_2\bar{x}_1$; в) $x_4x_2\bar{x}_3x_1$; г) x_1.

2. Випишіть конституенти нуля булевої функції $f(x_1, x_2, x_3, x_4)$ з такого списку елементарних диз'юнкцій:

   а) $x_1 \vee \bar{x}_3 \vee x_5$; б) $x_4 \vee x_2 \vee \bar{x}_3$; в) $\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4$; г) $\bar{x}_1$.

3. Знайти диз'юнктивне розкладання таких функцій за змінними x, z:

   а) $(yx \vee x\bar{z})(x \vee \bar{y}z(z \vee \bar{x}y))$;
   б) $((x \vee (z \vee yz)))(z \vee \bar{x}\,\bar{z} \vee y)$;
   в) $((x \vee \bar{y})(\bar{y} \vee \bar{z}\,)(\bar{z} \vee x)) \vee ((\bar{y} \vee z))$;
   г) $(x \vee \bar{z})(\bar{x}t \vee yt \vee \bar{x}\bar{t} \vee y\bar{t})(x \vee z)$;
   д) $(xy \vee x\bar{y}) \vee ((\bar{x} \vee y)(z \vee \bar{t}\,)(\bar{x} \vee \bar{y})(t \vee z))$.

4. Знайти кон'юнктивне розкладання таких функцій за змінними x, z:

   а) $(xz \vee y)(x\bar{y} \vee \bar{x}\,\bar{y} \vee \bar{z})(x \vee \bar{y})$;
   б) $((y \vee z)(t \vee yz)) \vee \bar{t}\,\bar{x} \vee ((z \vee y)(\bar{t} \vee \bar{z}))$;
   в) $yt \vee ((z \vee \bar{t})(x \vee z)(\bar{t} \vee \bar{z})(x \vee \bar{z})) \vee \bar{y}t$;
   г) $(\bar{z} \vee t)(t \vee x) \vee (\bar{z} \vee \bar{x})(\bar{z} \vee \bar{t})(\bar{t} \vee z)$;
   д) $x\bar{t} \vee ((\bar{z}\,\bar{y}) \vee t)(z \vee y)) \vee ((\bar{t} \vee \bar{z})(z \vee y))$;
   е) $(((t \vee \bar{t}\,z)\bar{t}) \vee \bar{y})((y \vee t)(y \vee x))$;
   ж)$((z \vee \bar{x})(\bar{x} \vee \bar{t})(x \vee z)(\bar{y} \vee x)) \vee y\bar{t} \vee yt$;
   з) $(t \vee \bar{x}\bar{t} \vee x)(y \vee (t \vee tz))\bar{z}\bar{x}t$;
   і) $\bar{z}\,\bar{y} \vee tz \vee \bar{y}z \vee t\bar{z} \vee y\bar{t}$.

4.6. Нормальні форми зображення булевих функцій

Алгоритми переходу від таблиць істинності булевих функцій до ДДНФ/ДКНФ і навпаки, алгоритми переходу від довільної формули до ДКНФ і ДДНФ

Для кожної інтерпретації функції існують єдині відповідні їй конституента одиниці та конституента нуля. Тому різних конституент одиниці та нуля для функції n змінних $f(x_1, x_2, \ldots, x_n)$ існує стільки ж, скільки й інтерпретацій цієї

функції — 2^n. Полічимо, скільки існує різних ДДНФ і ДКНФ для булевих функцій n змінних. ДДНФ функції n змінних можна зобразити у вигляді:

$$f(x_1, x_2, \ldots, x_n) = f_0 \wedge M_0 \vee f_1 \wedge M_1 \vee \ldots \vee f_{2^n-1} \wedge M_{2^n-1},$$

де f_i — значення функції $f(x_1, x_2, \ldots, x_n)$ на i-й інтерпретації;
M_i — конституента одиниці, що відповідає i-й інтерпретації.

Оскільки $M_0, M_1, \ldots, M_{2^n-1}$ однакові для всіх функцій n змінних, то вигляд ДДНФ залежить від набору 2^n булевих констант $(f_0, f_1, \ldots, f_{2^n-1})$. З цього маємо, що кількість різних ДДНФ дорівнює кількості упорядкованих наборів 2^n булевих констант і дорівнює 2^{2^n} (п. 4.1). Кількість різних булевих функцій від n змінних також становить 2^{2^n} (п. 4.1). Отже, для кожної булевої функції існує єдина ДДНФ. Аналогічно доводиться, що різних ДКНФ функції від n змінних існує 2^{2^n} і, отже, кожній булевій функції відповідає єдина ДКНФ.

Приклад. Запишемо конституенти одиниці та нуля, що відповідають інтерпретаціям функцій трьох змінних (таблиця 4.19).

Таблиця 4.19. Конституенти нуля та одиниці функцій трьох змінних

Номер інтерпретації	Інтерпретація			Конституента одиниці	Конституента нуля
	x_1	x_2	x_3		
0	0	0	0	$\bar{x}\bar{y}\bar{z}$	$x \vee y \vee z$
1	0	0	1	$\bar{x}\bar{y}z$	$x \vee y \vee \bar{z}$
3	0	1	1	$\bar{x}yz$	$x \vee \bar{y} \vee \bar{z}$
4	1	0	0	$x\bar{y}\bar{z}$	$\bar{x} \vee y \vee z$
5	1	0	1	$x\bar{y}z$	$\bar{x} \vee y \vee \bar{z}$
6	1	1	0	$xy\bar{z}$	$\bar{x} \vee \bar{y} \vee z$
7	1	1	1	xyz	$\bar{x} \vee \bar{y} \vee \bar{z}$

Алгоритм переходу від таблиці істинності булевої функції до ДДНФ

1. Виділити всі інтерпретації $(\sigma_1, \sigma_2, \ldots, \sigma_n)$, на яких значення функції дорівнює одиниці.

2. Записати конституенти одиниці виду $x_1^{\sigma_1} \wedge x_2^{\sigma_2} \wedge \ldots \wedge x_n^{\sigma_n}$, що відповідають відзначеним інтерпретаціям.

3. Одержати ДДНФ функції за допомогою з'єднання операцією диз'юнкції записаних конституент одиниці.

Алгоритм переходу від таблиці істинності булевої функції до ДКНФ

1. Виділити всі інтерпретації $(\sigma_1, \sigma_2, \ldots, \sigma_n)$, на яких значення функції дорівнює нулю.

2. Записати конституенти нуля виду $x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee \ldots \vee x_n^{\bar{\sigma}_n}$, що відповідають виділеним інтерпретаціям.

3. Записавши кон'юнкцію конституент нуля, одержати ДКНФ функції.

Приклад. Одержати ДДНФ для функцій $f_{13}(x, y)$ і $f_8(x, y)$ (таблиці 4.20, 4.21).

Розв'язок. Побудуємо ДДНФ заданих функцій, використовуючи вищеописаний алгоритм. На першому кроці побудуємо таблицю істинності булевої функції $f_{13}(x, y)$ (таблиця 4.20).

Таблиця 4.20. Функція $f_{13}(x, y)$

x	y	$f_{13}(x, y)$
0	0	1
0	1	1
1	0	0
1	1	1

Для одержання ДДНФ відзначимо інтерпретації, на яких $f_{13}(x, y) = 1$, запишемо відповідні конституенти одиниці, об'єднавши їх знаком диз'юнкції (див. формулу (4.5)):

$$f_{13}(x, y) = x^0 y^0 \vee x^0 y^1 \vee x^1 y^1 = \bar{x}\,\bar{y} \vee \bar{x}y \vee xy.$$

Перейдемо до ДДНФ функції $f_8(x, y)$ з таблицею істинності (таблиця 4.21).

Таблиця 4.21. Функція $f_8(x, y)$

x	y	$f_8(x, y)$
0	0	1
0	1	0
1	0	0
1	1	0

Дана функція дорівнює одиниці тільки на одній інтерпретації, яка відповідає конституенті одиниці, отже:

$$f_8(x, y) = x^0 y^0 = \bar{x}\,\bar{y}.$$

Приклад. Одержати ДКНФ для функцій $f_{13}(x, y)$ і $f_8(x, y)$.

Розв'язок. Функція f_{13} дорівнює нулю на єдиному наборі (1, 0) (див. таблицю 4.20). Записавши відповідну до цього набору конституенту нуля, одержимо ДКНФ функції f_{13} згідно з алгоритмом:

$$f_{13}(x, y) = x^{\bar{1}} \vee y^{\bar{0}} = x^0 \vee y^1 = \bar{x} \vee y.$$

Для одержання ДКНФ функції $f_8(x, y)$ випишемо всі три конституенти нуля (див. таблицю 4.21), з'єднавши їх знаком кон'юнкції:

$$f_8(x, y) = (x^{\bar{0}} \vee y^{\bar{1}})\,(x^{\bar{1}} \vee y^{\bar{0}})\,(x^{\bar{1}} \vee y^{\bar{1}}) = (x \vee \bar{y})\,(\bar{x} \vee y)\,(\bar{x} \vee \bar{y}).$$

Кількість інтерпретацій, на яких функція дорівнює одиниці, дорівнює кількості конституент одиниці у ДДНФ цієї функції. Аналогічно кількість інтерпретацій, на яких функція дорівнює нулю, дорівнює кількості конституент нуля у ДКНФ цієї функції. Тому для функцій, що дорівнюють нулю на більш ніж половині інтерпретацій, вигідніше будувати ДДНФ, ніж ДКНФ. Дійсно, в даному випадку ДДНФ буде містити менше символів змінних і знаків операцій, ніж ДКНФ. І навпаки, для функцій, що дорівнюють одиниці на більшості інтерпретацій, вигідніше будувати ДКНФ.

Одержання таблиці істинності функції, що задана ДДНФ або ДКНФ, зображує процедуру, обернену розглянутій вище.

Приклад. Для функції $f(x, y, z) = x\,y\,\bar{z} \vee \bar{x}\,y\,z \vee \bar{x}\,\bar{y}\,\bar{z}$, що задана ДДНФ, побудувати її таблицю істинності.

Розв'язок. Дана функція містить три конституенти одиниці — $x \wedge y \wedge \bar{z}$, $\bar{x} \wedge y \wedge z$, $\bar{x} \wedge \bar{y} \wedge \bar{z}$, яким відповідають інтерпретації (1, 1, 0), (0, 1, 1), (0, 0, 0). На даних інтерпретаціях функція дорівнює одиниці, на решті — нулю. Запишемо вказані значення функції у стовпчик $f(x, y, z)$ (таблиця 4.22). Стовпці x, y, z, $f(x, y, z)$ утворюють таблицю істинності функції f.

Приклад. Для функції $g(x, y, z) = (x \vee y \vee \bar{z})(\bar{x} \vee y \vee z)(\bar{x} \vee \bar{y} \vee \bar{z})$, що задана ДКНФ, побудувати її таблицю істинності.

Розв'язок. Скористаємося вищеописаним алгоритмом. Конституентам нуля $x \vee y \vee \bar{z}$, $\bar{x} \vee y \vee z$, $\bar{x} \vee \bar{y} \vee \bar{z}$ даної функції

відповідають інтерпретації (0, 0, 1), (1, 0, 0), (1, 1, 1). В наведених інтерпретаціях функція дорівнює нулю, в решті — одиниці. Записуємо дані значення функції у стовпчик $g(x, y, z)$ (таблиця 4.22). Стовпці x, y, z, $g(x, y, z)$ створюють таблицю істинності функції g.

Таблиця 4.22. Таблиця істинності $f(x, y, z)$ и $g(x, y, z)$

x	y	z	$f(x, y, z)$	$g(x, y, z)$
0	0	0	1	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	0

Алгоритм переходу від довільної формули алгебри логіки до ДДНФ

1. Виключити константи, використовуючи закони дій з константами.

2. Опустити знаки заперечення безпосередньо на змінні, використовуючи закони де Моргана.

3. Використовуючи дистрибутивний закон, розкрити дужки. До одержаних елементарних кон'юнкцій застосувати закони ідемпотентності й протиріччя, спростити їх і звести подібні. Результатом виконання вказаних дій є одержання ДНФ булевої функції.

4. Побудувати конституенти одиниці функції введенням у кожну елементарну кон'юнкцію відсутніх змінних, використовуючи закон виключеного третього.

5. За допомогою дистрибутивного закону розкрити дужки і звести подібні, використовуючи закон ідемпотентності. Одержана формула відповідає ДДНФ функції.

Алгоритм переходу від довільної формули алгебри логіки до ДКНФ

1. Виключити константи, використовуючи закони дій з константами.

2. Опустити знаки заперечення безпосередньо на змінні, використовуючи закони де Моргана.

3. За допомогою використання дистрибутивного закону, звести функцію до виду кон'юнкції елементарних диз'юнкцій. До одержаних елементарних диз'юнкцій застосувати закони ідемпотентності й виключеного третього, спростити їх і звести подібні. Результатом виконання вказаних дій є одержання КНФ булевої функції.

4. Побудувати конституенти нуля функції введенням у кожну елементарну диз'юнкцію відсутніх змінних, використовуючи закон протиріччя.

5. За допомогою дистрибутивного закону звести функцію до виду кон'юнкції конституент нуля і спростити формулу, використовуючи закон ідемпотентності. Одержана формула є ДКНФ функції.

Приклад. Побудувати ДДНФ функції

$$f(x, y, z) = xy \vee \overline{(x(\bar{y} \vee z) \vee yz)}.$$

Розв'язок. Скористаємося передостаннім алгоритмом. Опускаємо заперечення на змінні, використовуючи закони де Моргана:

$$xy \vee \overline{(x(\bar{y} \vee z) \vee yz)} = xy \vee \overline{(x(\bar{y} \vee z))}\,\overline{(yz)} =$$
$$= xy \vee (\bar{x} \vee \overline{(\bar{y} \vee z)})(\bar{y} \vee \bar{z}) = xy \vee (\bar{x} \vee (y\bar{z}))(\bar{y} \vee \bar{z}).$$

Побудуємо ДНФ, використовуючи дистрибутивний закон, закони ідемпотентності і протиріччя:

$$xy \vee (\bar{x} \vee (y\bar{z}))(\bar{y} \vee \bar{z}) = xy \vee (\bar{x}\,\bar{y} \vee y\bar{z}\,\bar{y} \vee \bar{x}\,\bar{z} \vee y\bar{z}\,\bar{z}) =$$
$$= xy \vee \bar{x}\,\bar{y} \vee 0 \vee \bar{x}\,\bar{z} \vee y\bar{z} = xy \vee \bar{x}\,\bar{y} \vee \bar{x}\,\bar{z} \vee y\bar{z}.$$

Дана функція залежить від трьох змінних, тому до елементарних кон'юнкцій необхідно ввести відсутні змінні, використовуючи закон виключеного третього:

$$xy \vee \bar{x}\,\bar{y} \vee \bar{x}\,\bar{z} \vee y\bar{z} = xy(z \vee \bar{z}) \vee \bar{x}\,\bar{y}(z \vee \bar{z}) \vee \bar{x}(y \vee \bar{y})\,\bar{z} \vee (x \vee \bar{x})y\bar{z}.$$

Використовуючи дистрибутивний закон, розкриємо дужки і зведемо подібні для одержання ДДНФ:

$$xyz \vee xy\bar{z} \vee \bar{x}\,\bar{y}z \vee \bar{x}\,\bar{y}\,\bar{z} \vee \bar{x}y\bar{z} \vee \bar{x}\,\bar{y}\,\bar{z} \vee xy\bar{z} \vee \bar{x}y\bar{z} =$$
$$= xyz \vee xy\bar{z} \vee \bar{x}\,\bar{y}z \vee \bar{x}\,\bar{y}\,\bar{z} \vee \bar{x}y\bar{z}.$$

Одержана ДДНФ заданої функції:

$$f(x, y, z) = xyz \vee xy\bar{z} \vee \bar{x}\,\bar{y}z \vee \bar{x}\,\bar{y}\,\bar{z} \vee \bar{x}y\bar{z}.$$

Приклад. Побудувати ДКНФ функції

$$f(x, y, z) = xy \vee \overline{(x(\bar{y} \vee z) \vee yz)}.$$

Розв'язок. Скористаємося останнім алгоритмом. Як у попередньому прикладі, опустимо заперечення безпосередньо на змінні і, використовуючи закони де Моргана, одержимо:

$$f(x, y, z) = xy \vee (\bar{x} \vee (y\bar{z}))(\bar{y} \vee \bar{z}).$$

Побудуємо КНФ, використовуючи дистрибутивний закон, закони ідемпотентності й виключеного третього:

$$xy \vee (\bar{x} \vee (y\bar{z}))(\bar{y} \vee \bar{z}) = xy \vee (\bar{x} \vee y)(\bar{x} \vee \bar{z})(\bar{y} \vee \bar{z}) =$$

$$= (x \vee (\bar{x} \vee y)(\bar{x} \vee \bar{z})(\bar{y} \vee \bar{z}))(y \vee (\bar{x} \vee y)(\bar{x} \vee \bar{z})(\bar{y} \vee \bar{z})) =$$

$$= (x \vee \bar{x} \vee y)(x \vee \bar{x} \vee \bar{z})(x \vee \bar{y} \vee \bar{z})(y \vee \bar{x} \vee y)(y \vee \bar{x} \vee \bar{z})(y \vee \bar{y} \vee \bar{z}) =$$

$$= 1 \cdot 1 \cdot (x \vee \bar{y} \vee \bar{z})(y \vee \bar{x})(y \vee \bar{x} \vee \bar{z}) \cdot 1 = (x \vee \bar{y} \vee \bar{z})(y \vee \bar{x})(y \vee \bar{x} \vee \bar{z}).$$

Дана функція залежить від трьох змінних, отже до елементарної диз'юнкції $(y \vee \bar{x})$ необхідно ввести відсутню змінну z, використовуючи закон протиріччя. Після чого, використовуючи дистрибутивний закон, слід звести функцію до виду кон'юнкції конституент нуля:

$$(x \vee \bar{y} \vee \bar{z})(y \vee \bar{x})(y \vee \bar{x} \vee \bar{z}) = (x \vee \bar{y} \vee \bar{z})(\bar{x} \vee y \vee z\bar{z})(\bar{x} \vee y \vee \bar{z}) =$$

$$= (x \vee \bar{y} \vee \bar{z})(\bar{x} \vee y \vee z)(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee y \vee \bar{z}).$$

Після зведення подібних за допомогою закону ідемпотентності одержуємо ДКНФ:

$$f(x, y, z) = (x \vee \bar{y} \vee \bar{z})(\bar{x} \vee y \vee z)(\bar{x} \vee y \vee \bar{z}).$$

Запитання

1. Скільки існує конституент одиниці та нуля для функції n змінних?
2. Яким чином для заданої інтерпретації булевої функції будуються відповідні їм конституенти одиниці, нуля?
3. Опишіть алгоритми переходу від таблиці істинності булевої функції до ДДНФ і ДКНФ.
4. Поясніть алгоритм переходу від ДДНФ булевої функції до таблиці істинності.
5. Опишіть алгоритм переходу від ДКНФ булевої функції до таблиці істинності.
6. Дайте порівняльну характеристику алгоритмів переходу від довільної формули булевої функції до ДДНФ і ДКНФ.

Завдання

1. Знайти ДНФ функції, що задана формулою $(x(\bar{x} \to y)) \to y$.
2. Одержати КНФ формули $(\overline{x \vee y})(x \to y)$.
3. Побудувати таблиці істинності для функцій, що задані ДКНФ:
   а) $f_1(x, y, z) = (\bar{x} \vee y \vee \bar{z})(x \vee \bar{y} \vee z)(x \vee y \vee \bar{z})$;
   б) $f_2(x, y, z) = (\bar{x} \vee y \vee z)(x \vee \bar{y} \vee \bar{z})(\bar{x} \vee \bar{y} \vee z)$;
   в) $f_3(x, y, z) = (\bar{x} \vee \bar{y} \vee \bar{z})(x \vee y \vee z)(\bar{x} \vee y \vee \bar{z})$.
4. Побудувати таблиці істинності для функцій, що задані ДДНФ:
   а) $f(x, y, z) = x\,\bar{y}\,\bar{z} \vee \bar{x}\,y\,\bar{z} \vee x\,\bar{y}\,z$;
   б) $f(x, y, z) = x\,y\,z \vee x\,\bar{y}\,\bar{z} \vee \bar{x}\,\bar{y}\,z$;
   в) $f(x, y, z) = x\,y\,\bar{z} \vee \bar{x}\,y\,z \vee x\,\bar{y}\,z$;
   г) $f(x, y, z) = \bar{x}\,y\,z \vee x\,\bar{y}\,\bar{z} \vee \bar{x}\,\bar{y}\,z$.
5. Знайдіть ДДНФ таких функцій:
   а) $f_1(x, y, z) = (1, 1, 0, 1, 0, 0, 0, 0)$, тут функція f_1 задана стовпцем значень з таблиці істинності;
   б) $f_2(x, y, z) = x \wedge y \wedge z$;
   в) $f_3(x, y, z) = x \oplus y \oplus z$;
   г) $f_4(x, y, z) = x \vee yz$;
   д) $f_5(x, y, z) = (x \wedge y) \to z$.
6. Отримайте ДКНФ таких функцій:
   а) $f_1(x, y, z) = (1, 1, 0, 1, 0, 0, 0, 0)$, тут функція f_1 задана стовпцем значень з таблиці істинності;
   б) $f_2(x, y, z) = x \vee y \vee z$;
   в) $f_3(x, y, z) = x \oplus y \oplus z$;
   г) $f_4(x, y, z) = x \vee \bar{y}z \vee \bar{z}$;
   д) $f_{201}(x, y, z)$, тут вункція задана порядковим номером.
7. Доведіть справедливість таких тотожностей, використовуючи закони булевої алгебри:
   а) $((x \mid y) \mid (x \sim y)) \mid ((z \oplus t) \to (t \leftarrow z)) = ((y \to z) \to (x \leftarrow z)) \downarrow \downarrow ((x \mid t) \mid (t \to y))$;
   б) $((x \wedge \bar{z}) \downarrow (y \leftarrow z)) \wedge ((x \mid t) \leftarrow (y \wedge t)) = ((x \mid y) \mid (x \oplus y)) \to \to ((z \oplus t) \wedge (t \to z))$;
   в) $((x \downarrow y) \vee (x \oplus y)) \leftarrow ((z \leftarrow t) \downarrow (z \sim t)) = ((z \to x) \wedge (z \to y)) \to \to ((x \downarrow t) \vee (y \downarrow t))$;
   г) $((x \sim y) \leftarrow (x \downarrow y)) \downarrow ((z \sim t) \downarrow (z \leftarrow t)) = ((z \leftarrow x) \downarrow \downarrow (z \leftarrow y)) \mid ((x \downarrow t) \downarrow (y \downarrow t))$;
   д) $((x \wedge y) \vee (x \oplus y)) \leftarrow ((t \leftarrow z) \downarrow (t \sim z)) = ((x \to z) \wedge (y \to z)) \to \to ((x \mid t) \mid (y \mid t))$;
   е) $((x \vee y) \leftarrow (x \oplus y)) \vee ((z \leftarrow t) \downarrow (z \sim t)) = ((z \leftarrow x) \downarrow \downarrow (z \leftarrow y)) \wedge ((x \vee t) \leftarrow (y \downarrow t))$;
   ж) $((t \to y) \to (\bar{z} \leftarrow y)) \downarrow ((z \vee x) \mid (t \to x)) = ((\bar{z} \mid t) \mid (z \oplus t)) \mid ((x \sim y) \to \to (\bar{x} \leftarrow y))$.

4.7. Алгебра Жегалкіна. Лінійні функції

Структура і тотожності алгебри Жегалкіна, зображення диз'юнкції та заперечення поліномом Жегалкіна, лінійність булевих функцій

Визначення

Алгебра $(B, \wedge, \oplus, 0, 1)$, що утворена множиною $B = \{0, 1\}$ разом з операціями $\wedge$ (кон'юнкції), $\oplus$ (XOR — от eXclusive OR, сума за модулем 2) і константами 0, 1, називається ***алгеброю Жегалкіна***.

Приклад. Формула $(x \oplus y \oplus z) \wedge (x \oplus z \oplus 1) \oplus x \wedge y \oplus 1$, де x, y, z — булеві змінні, є прикладом формули алгебри Жегалкіна, тому що вона містить операції кон'юнкції і XOR.

В алгебрі Жегалкіна операція кон'юнкції повністю ідентична множенню, а операція XOR зображує додавання за модулем для скінченних множин.

4.7.1. Тотожності алгебри Жегалкіна

Нагадаємо властивості операції кон'юнкції (див. п. 4.4):

1) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ — асоціативність.

2) $x \wedge y = y \wedge x$ — комутативність.

3) $x \wedge x = x$ — ідемпотентність.

4) $x \wedge 0 = 0$, $x \wedge 1 = x$ — дії з константами.

Властивості операції XOR (додавання за модулем 2):

5) $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ — асоціативність операції XOR.

Для доведення будуємо таблицю істинності (таблиця 4.23).

Таблиця 4.23. Доведення асоціативності операції XOR

x	y	z	$y \oplus z$	$x \oplus (y \oplus z)$	$x \oplus y$	$(x \oplus y) \oplus z$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	1	1	1
0	1	1	0	0	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	0	0	0
1	1	1	0	1	0	1

Ліва та права частини тотожності мають однакові таблиці істинності, таким чином, тотожність 5 доведена.

6) $x \oplus y = y \oplus x$ — комутативність операції XOR виходить з таблиці істинності (таблиця 4.24).

Таблиця 4.24. Доведення тотожності 6

x	y	$x \oplus y$	$y \oplus x$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

7) $x \oplus x = 0$ — закон зведення подібних доданків.

Як видно з таблиці істинності для операції XOR (таблиця 4.25), якщо операнди однакові, то функція дорівнює нулю, а якщо операнди різні, то значення функції — одиниця.

Таблиця 4.25. Доведення тотожностей 7 і 8

x	$x \oplus x$	$x \oplus 0$
0	0	0
1	0	1

8) $x \oplus 0 = x$ — операція з константою 0 (таблиця 4.25).

9) $x(y \oplus z) = xy \oplus xz$ — дистрибутивність $\wedge$ відносно $\oplus$ виходить з таблиці істинності (таблиця 4.26).

Таблиця 4.26. Дистрибутивності $\wedge$ відносно $\oplus$

x	y	z	$y \oplus z$	$x(y \oplus z)$	xy	xz	$xy \oplus xz$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	0	0	1	1	0

Операція XOR (сума за модулем 2) відіграє важливу роль у програмуванні, при обробці та кодуванні інформації. Вона має важливу властивість — наявність оберненого елемента x' для кожного $x \in \{0, 1\}$ (операції диз'юнкції і кон'юнкції такої властивості не мають). У цій алгебрі кожний елемент є оберненим до самого себе: $x' = x$. Це дозволяє розв'язувати рівняння шляхом додавання до обох частин однакових елементів. Наприклад, рівняння виду $x \oplus a = b$ розв'язується так:

$x \oplus a \oplus a = b \oplus a;$
$x \oplus 0 = b \oplus a;$
$x = b \oplus a.$

Можливість розв'язку подібних рівнянь, що відсутня у булевій алгебрі, обумовила широке застосування операції XOR, зокрема, при кодуванні інформації.

Алгебра Жегалкіна є кільцем з одиницею 1 (див. п. 3.4). Алгебра Жегалкіна не є ґраткою, оскільки в ній не виконуються всі необхідні для цього властивості, наприклад, ідемпотентність обох операцій (п. 3.5).

Будь-яка булева функція може бути зображена через операції кон'юнкції, диз'юнкції, заперечення (п. 4.5). Для доведення зображення будь-якої булевої функції формулою алгебри Жегалкіна достатньо виразити диз'юнкцію і заперечення через кон'юнкцію і XOR — операції алгебри Жегалкіна.

Зображення заперечення в алгебрі Жегалкіна формулою $\bar{x} = x \oplus 1$ виходить з таблиці істинності (таблиця 4.27):

Таблиця 4.27. Операція заперечення

x	$\bar{x}$	$x \oplus 1$
0	1	1
1	0	0

Зображення диз'юнкції в алгебрі Жегалкіна реалізується формулою $x \vee y = xy \oplus x \oplus y$.

Доведемо цю формулу аналітично:

$$x \vee y = \overline{\overline{x \vee y}} = \overline{\bar{x} \wedge \bar{y}} = \overline{(x \oplus y)(y \oplus 1)} = (x \oplus 1)(y \oplus 1) \oplus 1 =$$
$$= xy \oplus y \oplus x \oplus 1 \oplus 1 = xy \oplus y \oplus x.$$

Таким чином, будь-яка логічна функція може бути зображена формулою в алгебрі Жегалкіна.

4.7.2. Поліном Жегалкіна

Серед всіх еквівалентних зображень функції в алгебрі Жегалкіна виділяється особливий вид формул, що називаються ***поліном Жегалкіна***.

Визначення

Поліномом Жегалкіна називається скінченна сума за модулем 2 попарно різних елементарних кон'юнкцій над множиною змінних $\{x_1, x_2, \ldots, x_n\}$. Кількість змінних,

що входять до елементарної кон'юнкції, називається ***рангом елементарної кон'юнкції.***
Кількість попарно різних елементарних кон'юнкцій у поліномі називається ***довжиною полінома.***

Зображення у вигляді поліному існує та єдине для кожної булевої функції.

Для побудови поліному Жегалкіна функції, що задана деякою формулою алгебри Жегалкіна, необхідно розкрити всі дужки в даній формулі за законом дистрибутивності і виконати всі можливі спрощення з використанням законів дій з константами, ідемпотентності і зведення подібних доданків.

Приклад. Зобразити поліномами Жегалкіна логічні функції імплікацію ($\to$) і еквівалентність ($\sim$).

Розв'язок. Спочатку запишемо ДДНФ даних функцій, потім виразимо операції диз'юнкції та заперечення через операції кон'юнкції та XOR. Одержавши формулу алгебри Жегалкіна, користуючись описаним вище правилом, одержимо поліном Жегалкіна для кожної з даних функцій:

$$x \to y = \bar{x} \vee y = (x \oplus 1) \vee y = (x \oplus 1)y \oplus (x \oplus 1) \oplus y =$$
$$= xy \oplus y \oplus x \oplus 1 \oplus y = xy \oplus x \oplus 1;$$
$$x \sim y = x\,y \vee \bar{x}\,\bar{y} = x\,y\,\bar{x}\,\bar{y} \oplus x\,y \oplus \bar{x}\,\bar{y} = x\,y \oplus \bar{x}\,\bar{y} =$$
$$= xy \oplus (x \oplus 1)(y \oplus 1) = xy \oplus xy \oplus x \oplus y \oplus 1 = x \oplus y \oplus 1.$$

За видом полінома Жегалкіна визначається така важлива властивість булевих функцій, як лінійність.

Визначення

Булева функція називається ***лінійною***, якщо її поліном Жегалкіна не містить кон'юнкцій змінних.

Чи є лінійними операції булевої алгебри? Заперечення — лінійна функція, оскільки її поліном Жегалкіна $x \oplus 1$ не містить кон'юнкцій змінних. Диз'юнкція — нелінійна функція, оскільки її поліном Жегалкіна $x \oplus y \oplus xy$ містить кон'юнкцію змінних x і y.

Приклад. Визначити, чи лінійні функції імплікації ($\to$) і еквівалентності ($\sim$).

Розв'язок. Проаналізуємо структуру формул, що виведені у попередньому прикладі. З нього видно, що імплікація ($\to$) є нелінійною функцією, а еквівалентність ($\sim$) — функція лінійна.

Приклад. Дослідити на лінійність функцію $f(x, y, z) = (x \vee y) \to \bar{z}$.

Розв'язок. Побудуємо поліном Жегалкіна функції $f(x, y, z)$, використовуючи такі тотожності: $x \to y = \bar{x} \vee y$, $x \vee y = xy \oplus x \oplus y$, $\bar{x} = x \oplus 1$:

$$f(x, y, z) = (x \vee y) \to \bar{z} = (\overline{x \vee y}) \vee \bar{z} = (\overline{x \vee y})\bar{z} \oplus (\overline{x \vee y}) \oplus \bar{z} =$$
$$= (xy \oplus x \oplus y \oplus 1)(z \oplus 1) \oplus (xy \oplus x \oplus y \oplus 1) \oplus z \oplus 1 =$$
$$= xyz \oplus xz \oplus yz \oplus 1 \wedge z \oplus xy \wedge 1 \oplus x \wedge 1 \oplus y \wedge 1 \oplus 1 \wedge 1 \oplus$$
$$\oplus xy \oplus x \oplus y \oplus 1 \oplus z \oplus 1 = xyz \oplus xz \oplus yz \oplus z \oplus xy \oplus x \oplus$$
$$\oplus y \oplus 1 \oplus xy \oplus x \oplus y \oplus 1 \oplus z \oplus 1 = xyz \oplus xz \oplus yz \oplus 1.$$

Функція $f(x, y, z) = (x \vee y) \to \bar{z}$ не є лінійною, оскільки її поліном Жегалкіна містить кон'юнкції змінних.

Теорема

Для будь-якої булевої функції існує єдиний поліном Жегалкіна.

□ *Доведення.* Доведемо існування полінома. Для будь-якої функції існує зображення у вигляді формул булевої алгебри (п. 4.5). Від формул булевої алгебри завжди можна перейти до формул алгебри Жегалкіна, а потім до поліному Жегалкіна, використовуючи тотожності зображення операцій диз'юнкції та заперечення в алгебрі Жегалкіна, а також правило побудови полінома. Таким чином, для будь-якої булевої функції існує поліном Жегалкіна.

Підрахуємо кількість різних поліномів Жегалкіна від n змінних. Спочатку знайдемо кількість різних елементарних кон'юнкцій без заперечень від n змінних. У загальному випадку таку кон'юнкцію можна записати у такому вигляді:

$$(x_1 \vee \sigma_1) \wedge (x_2 \vee \sigma_2) \wedge \ldots \wedge (x_n \vee \sigma_n),$$

де $\sigma_1, \sigma_2, \ldots, \sigma_n$ — булеві константи, причому $\sigma_i = 0$, якщо x_i присутнє в кон'юнкції і $\sigma_i = 1$, якщо x_i не присутнє в кон'юнкції. Набір $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ повністю визначає елементарну кон'юнкцію, отже кількість таких наборів $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ дорівнює кількості елементарних кон'юнкцій без заперечень від n змінних. Кількість двійкових наборів (кодів) довжини n дорівнює 2^n, що було доведено в п. 4.1. Тому кількість елементарних кон'юнкцій від n змінних без заперечень дорівнює 2^n.

Поліном Жегалкіна, за визначенням, є сумою за модулем 2 таких елементарних кон'юнкцій. Перенумеруємо всі можливі елементарні кон'юнкції без заперечень від n змінних і позначимо їх K_1, K_2, ..., K_{2^n}. Запишемо поліном Жегалкіна у вигляді:

$$K_1 \wedge \sigma_1 \oplus K_2 \wedge \sigma_2 \oplus \ldots \oplus K_{2^n} \wedge \sigma_{2^n},$$

де σ_1, σ_2,..., σ_{2^n} — булеві константи, причому $\sigma_i = 1$, якщо кон'юнкція K_i присутня у поліномі і $\sigma_i = 0$, якщо кон'юнкція K_i не присутня у поліномі. Набір (σ_1, σ_2, ..., σ_{2^n}) повністю визначає вид полінома, отже кількість таких наборів довжини 2^n дорівнює кількості поліномів Жегалкіна від n змінних, тобто дорівнює 2^{2^n} (п. 4.1). Кількість різних булевих функцій від n змінних також дорівнює 2^{2^n} (п. 4.1). Оскільки одному поліному не можуть відповідати дві різні функції, кожній булевій функції відповідає єдиний поліном Жегалкіна. Теорему доведено. ■

На цій теоремі засновано метод невизначених коефіцієнтів, який можна використовувати для знаходження полінома Жегалкіна заданої функції. Розглянемо цей метод на прикладі.

Приклад. Побудувати поліном Жегалкіна для функції $f_{13}(x, y)$ — імплікації, використовуючи метод невизначених коефіцієнтів.

Розв'язок. Запишемо поліном для даної функції у вигляді суми за модулем 2 всіх можливих елементарних кон'юнкцій для x, y з невизначеними коефіцієнтами:

$$f_{13}(x, y) = x \to y = a_1xy \oplus a_2x \oplus a_3y \oplus a_4,$$

де коефіцієнти a_1, a_2, a_3, a_4 приймають значення з множини {0, 1} і визначають присутність або відсутність елементарної кон'юнкції в поліномі. Шукаємо послідовно значення коефіцієнтів, підставляючи значення змінних і функції на різних інтерпретаціях:

$$f_{13}(0, 0) = 0 \to 0 = 1,$$

$$1 = a_1 \wedge 0 \wedge 0 \oplus a_2 \wedge 0 \oplus a_3 \wedge 0 \oplus a_4 = a_4.$$

Отже, $a_4 = 1$.

Далі

$$f_{13}(0, 1) = 0 \to 1 = 1,$$

$$1 = a_1 \wedge 0 \wedge 1 \oplus a_2 \wedge 0 \oplus a_3 \wedge 1 \oplus 1 = a_3 \oplus 1,$$

звідки маємо, що $a_3 = 0$.

На такій інтерпретації

$$f_{13}(1, 0) = 1 \to 0 = 0,$$

$$0 = a_1 \wedge 1 \wedge 0 \oplus a_2 \wedge 1 \oplus a_3 \wedge 0 \oplus 1 = a_2 \oplus 1,$$

звідки маємо, що $a_2 = 1$.

Нарешті,

$$f_{13}(1, 1) = 1 \to 1 = 1,$$

$$1 = a_1 \wedge 1 \wedge 1 \oplus 1 \wedge 1 \oplus 0 \wedge 1 \oplus 1 = a_1 \oplus 1 \oplus 1 = a_1.$$

Підставивши одержані значення коефіцієнтів a_1, a_2, a_3, a_4, одержуємо поліном Жегалкіна для функції f_{13}:

$$x \to y = a_1xy \oplus a_2x \oplus a_3y \oplus a_4 =$$

$$= 1 \wedge xy \oplus 1 \wedge x \oplus 0 \wedge y \oplus 1 = xy \oplus x \oplus 1.$$

Запитання

1. Запишіть структуру алгебри Жегалкіна.
2. Назвіть основні закони алгебри Жегалкіна.
3. Запишіть тотожності, що дозволяють виразити основні операції булевої алгебри в алгебрі Жегалкіна.
4. Дайте визначення поняттю поліному Жегалкіна.
5. Які булеві функції називаються лінійними?
6. Скільки різних поліномів Жегалкіна можна побудувати для довільної булевої функції?
7. Сформулюйте правило побудови полінома Жегалкіна.

Завдання

1. Представити у вигляді поліному Жегалкіна такі логічні функції:
   а) $(xz \vee y)(x\bar{y} \vee \bar{x}\,\bar{y} \vee \bar{z})(x \vee \bar{y})$;
   б) $((y \vee z)(t \vee y\bar{z})) \vee t\,\bar{x} \vee ((z \vee y)(\bar{t} \vee \bar{z}))$;
   в) $yt \vee ((z \vee \bar{t})(x \vee z)(\bar{t} \vee \bar{z})(x \vee \bar{z})) \vee \bar{y}t$;
   г) $(\bar{z} \vee t)(t \vee x) \vee (\bar{z} \vee \bar{x})(\bar{z} \vee \bar{t})(\bar{t} \vee z)$;
   д) $x\bar{t} \vee ((\bar{z}\,\bar{y}) \vee t)(z \vee y)) \vee ((\bar{t} \vee \bar{z})(z \vee y))$;
   е) $(((t \vee \bar{t}\,z)\bar{t}) \vee \bar{y})((y \vee t)(y \vee x))$;
   ж) $((z \vee \bar{x})(\bar{x} \vee \bar{t})(x \vee z)(\bar{y} \vee x)) \vee y\bar{t} \vee yt$;
   з) $(t \vee \bar{x}\,\bar{t} \vee x)(y \vee (t \vee tz))\bar{z}\bar{x}t$;
   і) $\bar{z}\,\bar{y} \vee tz \vee \bar{y}z \vee t\bar{z} \vee y\bar{t}$.
2. Дослідити на лінійність такі булеві функції:
   а) $((y \vee z) \leftarrow (y \oplus z)) \vee ((x \leftarrow t) \downarrow (x \sim t))$;
   б) $((z \to t) \mid (z \oplus t)) \mid ((x \sim y) \to (x \wedge y))$;
   в) $((z \mid t) \mid (\bar{z} \sim t)) \to ((x \oplus y) \wedge (y \to x))$;

г) $((t \leftarrow x) \vee (t \sim x)) \leftarrow ((z \leftarrow y) \downarrow (\bar{z} \oplus y))$;
д) $((z \rightarrow t) \leftarrow (z \sim t)) \vee ((x \wedge y) \downarrow (x \oplus y))$;
е) $((z \vee t) \mid (z \sim t)) \mid ((\bar{x} \oplus \bar{y}) \rightarrow (x \leftarrow y))$;
ж) $((y \mid z) \mid (y \sim z)) \rightarrow ((x \oplus t) \wedge (x \rightarrow t))$;
з) $((y \downarrow \bar{t}) \vee (\bar{y} \oplus t)) \leftarrow ((x \leftarrow z) \downarrow (x \sim z))$;
і) $(x \oplus \bar{y}) \leftarrow (y \wedge x)) \downarrow ((\bar{z} \sim \bar{t}) \downarrow (t \leftarrow z))$;
к) $((x \downarrow y) \vee (\bar{x} \sim y)) \leftarrow ((z \leftarrow t) \downarrow (z \sim t))$;
л) $((z \rightarrow x) \leftarrow (x \oplus \bar{z})) \wedge ((t \leftarrow y) \downarrow (y \sim t))$;
м) $((z \mid \bar{y}) \mid (z \sim \bar{y})) \mid ((\bar{x} \oplus \bar{t}) \rightarrow (\bar{x} \leftarrow \bar{t}))$;
н) $((z \downarrow \bar{y}) \vee (z \oplus \bar{y})) \leftarrow ((\bar{t} \leftarrow \bar{x}) \downarrow (\bar{t} \sim \bar{x}))$.

3. За допомогою методу невизначених коефіцієнтів побудувати поліном Жегалкіна для таких функцій:
а) $(yx \vee x\bar{z})(x \vee \bar{y}z(z \vee \bar{x}y))$;
б) $((x \vee (z \vee yz)))(z \vee \bar{x}\,\bar{z} \vee y)$;
в) $((x \vee \bar{y})(\bar{y} \vee \bar{z})(\bar{z} \vee x)) \vee ((\bar{y} \vee z))$;
г) $(x \vee z)(\bar{x}t \vee yt \vee \overline{xt} \vee y\bar{t})(x \vee z)$;
д) $(xy \vee x\bar{y}) \vee ((\bar{x} \vee y)(z \vee \bar{t})(\bar{x} \vee \bar{y})(t \vee z))$.

4.8. Повнота та замкненість

Замкнені класи булевих функцій, функціонально повна система булевих функцій, теорема про одночасну повноту

Будь-яка логічна функція може бути зображена за допомогою операцій булевої алгебри (п. 4.5) або алгебри Жегалкіна (п. 4.7). Природно виникають запитання: «Чи існують інші множини операцій, за допомогою яких можна визначити будь-яку булеву функцію?», «Які властивості вони мають?». Введемо позначення: P_2 — множина всіх можливих булевих функцій, що залежать від будь-якого числа змінних.

Визначення

Замкненням множини Σ ***булевих функцій*** називається множина $[\Sigma]$, що складається з функцій, які можна одержати суперпозицією функцій з Σ. Якщо $\Sigma = [\Sigma]$, то множина булевих функцій Σ називається ***замкненим класом***. Іншими словами можна сказати, що множина Σ називається замкненим класом, якщо будь-яка суперпозиція функцій з Σ також належить Σ.

Визначення

Система булевих функцій $\Sigma = \{f_1, f_2, \ldots, f_n\}$ називається ***функціонально повною***, якщо $[\Sigma] = P_2$.

З іншого боку, якщо існують булеві функції, що не належать [Σ], то система Σ не є функціонально повною.

Теорема. *Про одночасну повноту*

Нехай дано дві системи булевих функцій Σ_1 *та* Σ_2, *причому система* Σ_1 *— функціонально повна. Якщо кожна функція із системи* Σ_1 *зображена у вигляді суперпозиції функцій із системи* Σ_2, *то система функцій* Σ_2 *також є функціонально повною.*

□ *Доведення.* Нехай $f \in P_2$ — деяка булева функція. Зобразимо її у вигляді суперпозиції функцій з Σ_1. Виразимо кожну функцію одержаної формули через функції системи Σ_2, що за умовами теореми можливо. Одержана формула зображує функцію f і містить тільки функції з Σ_2. Отже, система Σ_2 є функціонально повною, що і треба було довести. ■

Приклад. Довести, що системи $\{\vee, \bar{}\}$, $\{\wedge, \bar{}\}$ — функціонально повні, якщо відомо, що система $\{\wedge, \vee, \bar{}\}$ є такою.

Розв'язок. Виразимо функцію $\wedge$ через функції системи $\{\vee, \bar{}\}$:

$$x \wedge y = \overline{\overline{x \wedge y}} = \overline{\bar{x} \vee \bar{y}}.$$

Отже, всі функції повної системи $\{\wedge, \vee, \bar{}\}$ можна зобразити за допомогою функцій системи $\{\vee, \bar{}\}$. Таким чином, система $\{\vee, \bar{}\}$ є функціонально повною.

Аналогічно доводиться повнота системи функцій $\{\wedge, \bar{}\}$, оскільки функцію $\vee$ можна виразити через функції системи $\{\wedge, \bar{}\}$ таким чином:

$$x \vee y = \overline{\overline{x \vee y}} = \overline{\bar{x} \wedge \bar{y}}.$$

Твердження доведено.

Запитання

1. Як визначається замкнення множини булевих функцій?
2. Визначте поняття замкненого класу булевих функцій.
3. Яка система булевих функцій називається функціонально повною?
4. Назвіть основні замкнені класи булевих функцій.
5. Сформулюйте теорему про повноту двох систем булевих функцій.

Завдання

1. Довести повноту таких систем функцій шляхом зведення їх до відомих повних класів:
   а) $\{x \to y, \bar{x}\}$;
   б) $\{x \downarrow y\}$;

в) $\{x \to y,\ x \oplus y\}$;
г) $\{x \vee y,\ x \oplus y,\ 1\}$;
д) $\{x \sim y,\ 1,\ x \vee y\}$.

2. Перевірити на повноту такі класи функцій:
а) $\{x \wedge y,\ x \oplus y \oplus 1\}$;
б) $\{0,\ 1,\ (x \wedge y) \vee z\}$;
в) $\{0,\ 1,\ x \sim y\}$;
г) $\{x \wedge y \oplus x,\ x \sim y,\ 0\}$;
д) $\{x \vee y,\ x \wedge y \oplus x \wedge z\}$;
е) $\{\overline{x},\ (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)\}$;
ж) $\{x \oplus y,\ x \vee y \vee \overline{z}\}$.

4.9. Функції, що зберігають нуль та одиницю. Монотонні функції

Відношення порядку для інтерпретацій, ознаки монотонності функції

Визначення

Булева функція $f(x_1, x_2, \ldots, x_n)$ називається ***функцією, що зберігає 0***, якщо на нульовому наборі вона дорівнює 0: $f(0, 0, \ldots, 0) = 0$.

Визначення

Булева функція $f(x_1, x_2, \ldots, x_n)$ називається ***функцією, що зберігає 1***, якщо на одиничному наборі вона дорівнює 1: $f(1, 1, \ldots, 1) = 1$.

Функції $x \wedge y$ і $x \vee y$ зберігають 0, оскільки $0 \wedge 0 = 0$ і $0 \vee 0 = 0$. Крім того, дані функції також зберігають 1, оскільки $1 \wedge 1 = 1$ і $1 \vee 1 = 1$. Функція $\overline{x}$ не зберігає 0 і не зберігає 1, оскільки $\overline{0} = 1$, $\overline{1} = 0$.

Приклад. Визначте, чи зберігає 0 та 1 функція $f(x, y, z) = x \vee \overline{y}\,\overline{z}$.

Розв'язок. Перевіримо значення даної функції на нульовому та одиничному наборах.

$$f(0, 0, 0) = 0 \vee \overline{0} \wedge \overline{0} = 0 \vee 1 \wedge 1 = 0 \vee 1 = 1,$$
$$f(1, 1, 1) = 1 \vee \overline{1} \wedge \overline{1} = 1 \vee 0 \wedge 0 = 1 \vee 0 = 1.$$

Отже, дана функція зберігає 1 і не зберігає 0.

Розглянемо важливий клас булевих функцій — монотонні булеві функції. Для цього введемо ***відношення порядку***

(див. п. 2.4.), яке будемо позначати символом $\leq$, ***для наборів булевих констант*** (інтерпретацій) $\alpha = (a_1, a_2, \ldots, a_n)$, $\beta = (b_1, b_2, \ldots, b_n)$ таким чином:

$$\alpha \leq \beta, \quad \text{якщо } a_i \leq b_i \text{ для всіх } i = 1, \ldots, n.$$

Якщо хоча б для однієї пари (a_i, b_i) відношення $a_i \leq b_i$ не виконується, то відповідні їм набори α і β у відношенні порядку не беруть участі, тобто є непорівнянними.

Приклад. Визначити відношення порядку для інтерпретацій функції одної та двох змінних.

Розв'язок. Для функції одної змінної $f(x)$ маємо два набори змінних: (0) і (1). Відношення порядку встановлюється таким чином:

$$(0) \leq (0),\ (0) \leq (1),\ (1) \leq (1).$$

Тут всі пари інтерпретацій порівнянні.

Для функції двох змінних $f(x, y)$ маємо чотири інтерпретації (0, 0), (0, 1), (1, 0) і (1, 1), для яких відношення порядку встановлюється таким чином:

$$(0, 0) \leq (0, 0);\ (0, 0) \leq (0, 1);\ (0, 0) \leq (1, 0);$$

$$(0, 0) \leq (1, 1);\ (0, 1) \leq (0, 1);\ (0, 1) \leq (1, 1);$$

$$(1, 0) \leq (1, 0);\ (1, 0) \leq (1, 1);\ (1, 1) \leq (1, 1).$$

Непорівнянними наборами є, наприклад, (0, 1) і (1, 0).

Визначення

Булева функція f називається ***монотонною***, якщо для будь-яких пар наборів значень змінних $(a_1, \ldots, a_n)$ і $(b_1, \ldots, b_n)$, для яких виконується відношення $(a_1, \ldots, a_n) \leq (b_1, \ldots, b_n)$, правильна і нерівність $f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n)$.

Для перевірки функції на монотонність необхідно дослідити виконання нерівності $f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n)$ для всіх пар наборів $(a_1, \ldots, a_n) \leq (b_1, \ldots, b_n)$, крім випадку $(a_1, \ldots, a_n) = (b_1, \ldots, b_n)$, у якому значення функцій співпадають.

Приклад. Дослідити на монотонність функції $f(x, y) = x \wedge y$, $g(x, y) = x \oplus y$.

Розв'язок. Для функції $f(x, y)$ запишемо всі набори значень змінних, для яких виконується відношення порядку, визначимо значення функції на даних наборах і порівняємо їх:

$(0, 0) \le (0, 1)$, $f(0, 0) = 0$, $f(0, 1) = 0$, $f(0, 0) \le f(0, 1)$.
$(0, 0) \le (1, 0)$, $f(0, 0) = 0$, $f(1, 0) = 0$, $f(0, 0) \le f(1, 0)$.
$(0, 0) \le (1, 1)$, $f(0, 0) = 0$, $f(1, 1) = 1$, $f(0, 0) \le f(1, 1)$.
$(0, 1) \le (1, 1)$, $f(0, 1) = 0$, $f(1, 1) = 1$, $f(0, 1) \le f(1, 1)$.
$(1, 0) \le (1, 1)$, $f(1, 0) = 0$, $f(1, 1) = 1$, $f(1, 0) \le f(1, 1)$.

Висновок: функція $f(x, y) = x \wedge y$ є монотонною.

Аналогічно проведемо дослідження функції $g(x, y)$.

$(0, 0) \le (0, 1)$, $g(0, 0) = 0$, $g(0, 1) = 1$, $g(0, 0) \le g(0, 1)$.
$(0, 0) \le (1, 0)$, $g(0, 0) = 0$, $g(1, 0) = 1$, $g(0, 0) \le g(1, 0)$.
$(0, 0) \le (1, 1)$, $g(0, 0) = 0$, $g(1, 1) = 0$, $g(0, 0) \le g(1, 1)$.
$(0, 1) \le (1, 1)$, $g(0, 1) = 1$, $g(1, 1) = 0$, $g(0, 1) \not\le g(1, 1)$.

Висновок: функція $g(x, y) = x \oplus y$ не є монотонною.

Теорема

Булева функція, відмінна від констант 0 і 1, є монотонною, якщо і тільки якщо вона припускає зображення формулою булевої алгебри без заперечень.

□ *Доведення.* Нехай деяка формула булевої алгебри без заперечень зображує деяку функцію f від n змінних. Перетворимо дану формулу на ДНФ, розкривши дужки за допомогою дистрибутивного закону. Одержана ДНФ також не буде містити заперечень. Припустимо, що функція f немонотонна, тоді на якійсь інтерпретації $\alpha \le \beta$ значення функції $f(\alpha) \not\le f(\beta)$, тобто $f(\alpha) = 1$, $f(\beta) = 0$. Рівність $f(\alpha) = 1$ означає, що ДНФ функції f містить елементарну кон'юнкцію, яка дорівнює 1 на наборі α. Оскільки дана кон'юнкція не містить заперечень, її можна позначити через $x_1 \wedge x_2 \wedge \ldots \wedge x_k$. Запишемо набір значень α як $(a_1, a_2, \ldots, a_k, a_{k+1}, \ldots, a_n)$. Оскільки $x_1 \wedge x_2 \wedge \ldots \wedge x_k = 1$ на наборі α, то $a_1 \wedge a_2 \wedge \ldots \wedge a_k = 1$, і, отже, $a_1 = a_2 = \ldots = a_k = 1$. Дослідимо тепер набір $\beta = (b_1, b_2, \ldots, b_k, b_{k+1}, \ldots, b_n)$. Спочатку було зроблене припущення, що $\alpha \le \beta$, отже, $b_1 = b_2 = \ldots = b_k = 1$. Тоді кон'юнкція $x_1 \wedge x_2 \wedge \ldots \wedge x_k$ при підстановці значень з набору β дорівнює одиниці, звідки виходить, що $f(\beta) = 1$. Однак з початку доведення було зроблене припущення $f(\beta) = 0$. Значить, припущення помилково і функція f монотонна.

Нехай тепер f — монотонна функція від n змінних. Зобразимо її у ДНФ. Припустимо, що дана ДНФ містить елементарну кон'юнкцію із запереченням. Позначимо її $\bar{x}_1 \wedge K$, де K —

елементарна кон'юнкція змінних $x_2, \ldots, x_k$. Розглянемо деякий набір $\alpha_1 = (0, a_2, \ldots, a_k, a_{k+1}, \ldots, a_n)$, на якому $\bar{x}_1 \wedge K = 1$, а отже, і $f(\alpha_1) = 1$. Розглянемо набір $\alpha_2 = (1, a_2, \ldots, a_k, a_{k+1}, \ldots, a_n)$. Оскільки $\alpha_1 \leq \alpha_2$ і функція f монотонна, то $f(\alpha_2) = 1$. Останнє твердження буде справедливе тільки тоді, коли ДНФ функції f містить елементарну кон'юнкцію $x_1 \wedge K$, яку набір α_2 обертає на одиницю. Таким чином, ДНФ функції f можна записати у такому вигляді: $\bar{x}_1 \wedge K \vee x_1 \wedge K$. Застосувавши дистрибутивний закон і закон виключеного третього, одержимо:

$$\bar{x}_1 \wedge K \vee x_1 \wedge K = (\bar{x}_1 \vee x_1) \wedge K = K.$$

Таким чином, змінна із запереченням завжди може бути видалена з ДНФ монотонної функції. Теорему доведено. ■

Приклад. Визначити, чи є функція $f(x, y, z, t) = (\bar{x} \vee \bar{y}) \to (z \vee t)$ монотонною.

Розв'язок. Виразимо $f(x, y, z, t)$ через елементарні функції булевої алгебри:

$$(\bar{x} \vee \bar{y}) \to (z \vee t) = \overline{(\bar{x} \vee \bar{y})} \vee (z \vee t) = xy \vee z \vee t.$$

Одержана формула булевої алгебри не містить заперечень, отже функція $f(x, y, z, t)$ є монотонною.

Запитання

1. Дайте визначення булевих функцій, що зберігають нуль та одиницю.
2. Як за таблицею істинності функції визначити, чи зберігає вона нуль (одиницю)?
3. Назвіть функції двох змінних, які зберігають нуль (одиницю).
4. В чому полягає монотонність булевих функцій?
5. Як визначається відношення порядку для пар наборів булевих констант?
6. Якщо функція f зображена формулою булевої алгебри із запереченнями, то чи правильно, що f немонотонна? Обґрунтуйте відповідь.
7. Чи можна за видом ДНФ булевої функції стверджувати про її монотонність?

Завдання

1. Довести монотонність таких функцій:
   а) $x \wedge (y \vee z)$; в) $x \vee y \vee z$;
   б) $x \vee (y \wedge z)$; г) $x \wedge y \wedge z$.
2. Дослідити такі функції на монотонність:

а) $yx \oplus y$; г) $x \to (y \to x)$;
б) $xy \oplus y \oplus x$; д) $x \to (x \to y)$;
в) $x \sim y$; е) $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$.

3. Довести, що функція, двоїста монотонній, сама є монотонною.

4.10. Теореми Поста про повноту

Класи Поста, критерій повноти, нескоротність та слабка повнота системи функцій

Класами Поста називаються такі п'ять множин булевих функцій:

T_0 — клас функцій, що зберігають нуль;
T_1 — клас функцій, що зберігають одиницю;
S — клас самодвоїстих функцій;
M — клас монотонних функцій;
L — клас лінійних функцій.

Булева функція $f = f(x_1, \ldots, x_n)$ може належати одному або кільком класам Поста. Більш того, ***функція, що проектує***, $f(x_1, \ldots, x_i \ldots, x_n) = x_i$ належить усім п'ятьом класам (переконайтесь у цьому). Однак існують функції, що не входять до жодного класу, наприклад, штрих Шеффера $f_{14}(x, y) = \overline{x \wedge y}$. Дійсно, поліном Жегалкіна $\overline{x \wedge y} = xy \oplus 1$ тут нелінійний, так що $f_{14} \notin L$. Той факт, що функція f_{14} не міститься в жодному з класів T_0, T_1, S, M, відразу вбачається з таблиці істинності для f_{14} (див. таблицю 4.2).

Теорема 1

Кожний з класів Поста замкнений.

Це означає, що будь-яка суперпозиція функцій з одного й того ж класу Поста призводить до функції того ж класу. Дійсно, якщо f, $g_i \in T_0$ і

$$\varphi(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)), \text{ то}$$

$$\varphi(0, \ldots, 0) = f(g_1(0, \ldots, 0), \ldots, g_m(0, \ldots, 0)) = f(0, \ldots, 0) = 0;$$

тому $\varphi \in T_0$. Аналогічно перевіряється, що суперпозиція булевих функцій не порушує властивості самодвоїстості, монотонності, збереження нуля, лінійності.

Виявляється, для довільної системи $\Sigma = \{f\}$ булевих функцій повнота Σ означає, що ***доповнення кожного з класів Поста***

містить хоча б одну функцію f системи Σ. Відповідний критерій повноти, що належить Посту, ми наводимо без доведення.

Теорема 2. *Критерій повноти Поста*

Система Σ булевих функцій повна, якщо і тільки якщо вона містить хоча б одну функцію, що не зберігає нуль, хоча б одну функцію, що не зберігає одиницю, хоча б одну несамодвоїсту функцію, хоча б одну немонотонну функцію і хоча б одну нелінійну функцію.

Зауважимо, що одна й та ж функція може зображати у функціонально повній системі одну або кілька необхідних властивостей. Тому мінімальне число булевих функцій у функціонально повному наборі дорівнює одиниці. Одна функція може мати всі п'ять необхідних властивостей.

Визначення

Повна система булевих функцій називається ***нескоротною***, якщо з неї не можна виключити жодної булевої функції без утрати властивості повноти.

Наслідок. *Максимальна кількість булевих функцій у нескоротному функціонально повному наборі дорівнює чотирьом.*

Приклад. Визначити за допомогою теореми Поста, чи є система $\{\bar{}, \to\}$ функціонально повною?

Розв'язок. Заперечення зображується поліномом Жегалкіна: $\bar{x} = x \oplus 1$, отже функція заперечення лінійна.

За таблицею істинності 4.28 визначаємо, що функція заперечення самодвоїста, не зберігає 0, не зберігає 1, немонотонна. Імплікація є нелінійною функцією (приклад п. 4.7).

Таблиця 4.28. $f(x) = \bar{x}$

x	$\bar{x}$
0	1
1	0

Таблиця 4.29. $f(x) = x \to y$

x	y	$x \to y$
0	0	1
0	1	1
1	0	0
1	1	1

За таблицею істинності 4.29 визначаємо, що імплікація несамодвоїста, не зберігає 0, зберігає 1, немонотонна. Побудуємо таблицю 4.30, в якій відзначимо знаком «+» наявність

відповідних властивостей у функції — заперечення та імплікації, а знаком «-» — відсутність.

У кожному рядку таблиці присутній знак «-». Отже, для кожного класу Поста в даній системі є хоча б одна функція, яка цьому класу Поста не належить. За теоремою Поста така система булевих функцій є функціонально повною.

Таблиця 4.30. Властивості досліджуваного набору

Назва властивості	$\bar{x}$	$x \to y$
Лінійність	+	-
Монотонність	-	-
Зберігання 0	-	-
Зберігання 1	-	+
Самодвоїстість	+	-

В таблиці 4.31 наведено результати дослідження з належності до класів Поста кожної булевої функції двох змінних. Знак «+» означає, що функція належить класу, а знак «-» — означає, що функція не належить класу.

Таблиця 4.31. Належність булевих функцій двох змінних класам Поста

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
		$f=0$	$x \vee y$	$\overline{x \to y}$	x	$\overline{x \leftarrow y}$	y	$x \oplus y$	$x \vee y$	$x \to y$	$x \sim y$	$\bar{y}$	$x \leftarrow y$	$\bar{x}$	$x \to y$	$x \mid y$	$f=1$
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
L: Лінійність		+	-	-	+	-	+	+	-	-	+	+	-	+	-	-	+
M: Монотонність		+	+	-	+	-	+	-	+	-	-	-	-	-	-	-	+
T_0: Зберігання 0		+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-
T_1: Зберігання 1		-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+
S: Самодвоїстість		-	-	-	+	-	+	-	-	-	-	+	-	+	-	-	-

На практиці часто припускається, що постійні булеві функції-константи 0 і 1 задані заздалегідь. У такому випадку до набору функцій Σ фактично додають функції-константи 0 і 1.

Визначення

Систему булевих функцій Σ називають ***функціонально повною в слабкому розумінні***, якщо будь-яка логічна функція може бути зображена суперпозицією функцій з множини, що одержана шляхом додавання до множини Σ констант 0 і 1.

З теореми Поста маємо, що для функціональної повноти системи функції Σ у слабкому розумінні достатньо, щоб у Σ містилися нелінійна та немонотонна функції, оскільки решта необхідних властивостей присутня у констант (константа одиниці не зберігає 0, константа нуля не зберігає 1, і обидві ці функції несамодвоїсті).

При синтезі логічних схем кожній функції відповідає логічний елемент визначеного типа, а функціональна повнота означає можливість реалізувати будь-яку булеву функцію за допомогою елементів з відповідної «повної» системи типів. Константи 0 і 1 не потребують спеціальних елементів для реалізації, тому часто їх вважають даними. В такому випадку систему типів логічних елементів досліджують на функціональну повноту в слабкому розумінні.

Запитання

1. Сформулюйте теорему про замкнені класи булевих функцій.
2. Що зображує суперпозиція несамодвоїстої функції та заперечення?
3. Яку функцію можна одержати за допомогою суперпозиції немонотонної функції та констант 0 і 1?
4. Яка функція може бути зображена за допомогою суперпозиції нелінійної функції, заперечення та констант 0 і 1?
5. Сформулюйте теорему Поста.
6. Яка повна система булевих функцій називається нескоротною?
7. Сформулюйте наслідок з теореми Поста.
8. Дайте визначення функціонально повної у слабкому розумінні системи булевих функцій. Наявності яких функцій достатньо для виконання умови функціонально повної у слабкому розумінні системи?

Завдання

1. За допомогою теореми Поста перевірити з повноти набори булевих функцій:
   а) $\{x \vee y, \neg x\}$;
   б) $\{x \to y, \neg x\}$;

в) $\{x \wedge y \oplus x,\ 1\}$;
г) $\{x \oplus y,\ x \sim (y \wedge z)\}$;
д) $\{x \wedge y,\ x \oplus y,\ x \sim (x \vee y)\}$;
е) $\{x \sim y,\ 1,\ x \vee y\}$;
ж) $\{x \wedge y,\ x \oplus y \oplus 1\}$;
з) $\{0,\ 1,\ (x \wedge y) \vee z\}$;
і) $\{0,\ 1,\ x \sim y\}$;
к) $\{x \wedge y \oplus x,\ x \sim y,\ 0\}$;
л) $\{x \vee y,\ x \wedge y \oplus x \wedge z\}$;
м) $\{\bar{x},\ (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)\}$;
н) $\{1,\ \bar{x},\ x \oplus y\}$;
о) $\{x \oplus y,\ x \vee y \vee \bar{z}\}$.

2. Довести функціональну повноту кожної з функцій: штрих Шеффера або стрілка Пірса.

4.11. Мінімізація булевих функцій

Основні поняття. Метод карт Карно (діаграм Вейча), частково визначені функції, метод Квайна — Мак-Класкі, метод Порецького — Блейка

4.11.1. Основні поняття

Пошук найбільш простої логічної формули булевої функції має велике значення при формуванні запитів до баз даних, у логічному програмуванні, в інтелектуальних системах.

Задача мінімізації складається з пошуку найпростішої, згідно з обраним ***критерієм мінімізації***, формули. Критерії можуть бути різними, наприклад: кількість змінних у формулі, кількість знаків кон'юнкції та диз'юнкції або комбінація подібних критеріїв.

Нижче розглянуто мінімізацію на множині ДНФ і КНФ кількості символів змінних та операцій, що містяться у формулі. Така задача називається ***канонічною задачею мінімізації***. Мінімальні форми, що одержані в результаті її розв'язку, називаються мінімальними ДНФ і КНФ.

Визначення

Імплікантою деякої функції f називається функція g, така, що на всіх інтерпретаціях, на яких g дорівнює одиниці, f теж дорівнює одиниці.

Мінтерми функції є її імплікантами, елементарні кон'юнкції, що входять до складу ДНФ функції, також є її імплікантами.

Визначення
Множина S, що складається з імплікант функції f, називається ***покриттям*** (або ***повною системою імплікант***) функції f, якщо кожне одиничне значення функції f покривається одиницею хоча б одної імпліканти з множини S.

Набір імплікант, складових ДНФ функції, є її покриттям. Набір всіх конституент одиниці функції, що входять до її ДДНФ, є покриттям даної функції.

Будь-яку елементарну кон'юнкцію A, що входить до складу елементарної кон'юнкції B і містить менше змінних, ніж кон'юнкція B, називають ***власною частиною*** кон'юнкції B, і вважають, що кон'юнкція A ***покриває*** кон'юнкцію B.

Визначення
Простою імплікантою функції f називається така кон'юнкція-імпліканта, що ніяка її власна частина не є імплікантою даної функції.

Множина всіх простих імплікант функції становить ***покриття*** даної функції.

Визначення
Диз'юнкція всіх простих імплікант функції називається її ***скороченою ДНФ***.

Визначення
Диз'юнктивним ядром булевої функції f називається така множина її простих імплікант, яка створює покриття f, але після видалення будь-якої імпліканти втрачає цю властивість, тобто перестає бути повною системою імплікант.

Визначення
Тупиковою ДНФ називається ДНФ даної булевої функції, що складається тільки з простих імплікант.

На відміну від скороченої ДНФ, тупикова ДНФ може не містити деякі з простих імплікант функції. Кожна булева функція має єдину скорочену ДНФ і може мати декілька тупикових ДНФ. У кожну з тупикових ДНФ входять все імпліканти диз'юнктивного ядра даної функції.

Визначення
Мінімальною ДНФ (МДНФ) даної булевої функції називається одна з її тупикових ДНФ, якій відповідає найменше значення критерію мінімізації ДНФ.

Для знаходження множини простих імплікант функції, що задана СДНФ, використовуються такі очевидні перетворення формул булевої алгебри, які традиційно називаються «операціями».

Операція ***неповного диз'юнктивного склеювання***:

$$Ax \vee A\bar{x} = A \vee Ax \vee A\bar{x}.$$

Операція ***диз'юнктивного поглинання***:

$$A \vee Ax = A.$$

Виконання двох зазначених операцій послідовно зображує виконання операції ***повного диз'юнктивного склеювання***:

$$Ax \vee A\bar{x} = A.$$

Тут A — деяка елементарна кон'юнкція змінних, x — булева змінна.

Приклад. Нехай є функція f, що задана ДДНФ:

$$f(x, y, z) = xyz \vee \bar{x}\, y\, z \vee \bar{x}\, \bar{y}\, z \vee \bar{x}\, \bar{y}\, \bar{z}.$$

Виконаємо операції повного склеювання таким чином:

$$f(x, y, z) = x\, y\, z \vee \bar{x}\, y\, z \vee \bar{x}\, \bar{y}\, z \vee \bar{x}\, \bar{y}\, \bar{z} =$$
$$= (x\, y\, z \vee \bar{x}\, y\, z) \vee (\bar{x}\, y\, z \vee \bar{x}\, \bar{y}\, z) \vee (\bar{x}\, \bar{y}\, z \vee \bar{x}\, \bar{y}\, \bar{z}) =$$
$$= y\, z \vee \bar{x}\, z \vee \bar{x}\, \bar{y}.$$

Виконаємо тепер операції склеювання іншим способом:

$$f(x, y, z) = (x\, y\, z \vee \bar{x}\, y\, z) \vee (\bar{x}\, \bar{y}\, z \vee \bar{x}\, \bar{y}\, \bar{z}) = y\, z \vee \bar{x}\, \bar{y}.$$

В обох випадках одержані тупикові ДНФ функції $f(x, y, z)$. Друга тупикова ДНФ простіша за першу, оскільки містить меншу кількість символів змінних і знаків операцій. Побудуємо таблицю істинності функції $f(x, y, z)$ та її трьох імплікант, що входять до складу ДНФ (таблиця 4.32).

Таблиця 4.32. Покриття одиничних значень функції

x	y	z	f	$y\,z$	$\bar{x}\,z$	$\bar{x}\,\bar{y}$
0	0	0	1	0	0	1
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	1	1	1	1	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	1	1	0	0

На тих інтерпретаціях, на яких імпліканта дорівнює одиниці, функція також дорівнює одиниці. Таким чином, кожна імпліканта «покриває» деякі одиничні значення функції (в таблиці це показано стрілками). Для аналізу різних зображень булевої функції через КНФ і одержання мінімальних КНФ природно трансформувати викладені вище поняття та операції склеювання за принципом двоїстості. Двоїсті поняття відповідно називаються термінами: ***імпліцента***, ***проста імпліцента***, ***повна система імпліцент***, ***скорочена КНФ***, ***тупикова КНФ***, ***мінімальна КНФ***. Використовувані для мінімізації КНФ операції склеювання мають такий вигляд:

Операція ***неповного кон'юнктивного склеювання***:

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

Операція ***кон'юнктивного поглинання***:

$$A(A \vee x) = A.$$

Операція ***повного кон'юнктивного склеювання***:

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

Тут A — деяка елементарна диз'юнкція, x — булева змінна.

4.11.2. Мінімізація булевих функцій методом карт Карно (діаграм Вейча)

Ціллю мінімізації є знаходження мінімальної з тупикових ДНФ (КНФ), тобто знаходження мінімального покриття даної функції. Для цього необхідно побудувати всі можливі тупикові ДНФ (КНФ), використовуючи операції склеювання та поглинання для даної функції. Методика Карно і Вейча дозволяє виконати зазначені операції графічно.

Карта Карно для ДНФ (діаграма Вейча — для КНФ) є аналогом таблиці істинності, зображеній у спеціальній формі. Значення змінних розташовані у заголовках рядків і стовпців карти. Кожній конституенті одиниці функції відповідає одна клітка (комірка) таблиці. Нуль або одиниця в клітці визначає значення функції на даній інтерпретації. Значення змінних розташовані так, щоб сусідні (що мають спільну межу) рядки і стовпці таблиці відрізнялись значенням тільки однієї змінної. Тому запис інтерпретацій двох змінних у порядку (0, 0), (0, 1), (1, 0), (1, 1) неприпустимий, оскільки сусідні набори

(0, 1) і (1, 0) відрізняються значеннями обох змінних. У картах Карно інтерпретації двох змінних розташовуються у такій послідовності: (0, 0), (0, 1), (1, 1), (1, 0). У цій послідовності перша та остання інтерпретації також відрізняються значенням тільки однієї змінної, тому перший і останній рядки (стовпці) таблиці вважаються сусідніми (протилежні границі таблиці вважаються співпадаючими). При такому розташуванні мінтерми, до яких застосовна операція склеювання, розташовуються у сусідніх клітках карти, і склеювання проводиться графічно за допомогою об'єднання кліток у групи.

Карти Карно для функцій двох змінних мають вид таблиці 2 × 2, де стовпці відповідають значенням першої змінної, а рядки — значенням другої змінної. На рис. 4.1 зображено структури карти Карно, в клітках вказано відповідні мінтерми у виді формул з абстрактними змінними.

y \ x	0	1
0	$\bar{x}\bar{y}$	$x\bar{y}$
1	$\bar{x}y$	xy

Рис. 4.1. Структура карти Карно для двох змінних

Структура карти Карно для функцій трьох змінних має вид таблиці 2 × 4, де стовпці відповідають всіляким наборам значень перших двох змінних, а рядки — значенням 0 і 1 третьої змінної (рис. 4.2). В клітках структури вказано відповідні мінтерми з абстрактними змінними.

z \ xy	00	01	11	10
0	$\bar{x}\bar{y}\bar{z}$	$\bar{x}y\bar{z}$	$xy\bar{z}$	$x\bar{y}\bar{z}$
1	$\bar{x}\bar{y}z$	$\bar{x}yz$	xyz	$x\bar{y}z$

Рис. 4.2. Структура карти Карно для функції трьох змінних

Для конкретної булевої функції карта Карно заповнюється таким чином. У клітках, що відповідні інтерпретаціям, на яких функція дорівнює одиниці, записують одиниці. Ці клітки відповідають конституентам одиниці, що присутні у ДДНФ функції. В інші клітки записують нулі.

Приклад. Побудувати карту Карно для функції

$$f(x, y, z) = \bar{x}\,\bar{y}\,\bar{z} \vee \bar{x}\,y\,z \vee \bar{x}\,\bar{y}\,z \vee xyz.$$

Розв'язок. Побудова карти Карно для заданої функції: поміщаємо одиниці до кліток, що відповідають мінтер-

мам, які присутні в даній ДДНФ, і нулі — в решту кліток (рис. 4.3).

z \ xy	00	01	11	10
0	1	0	0	0
1	1	1	1	0

Рис. 4.3. Карта Карно для функції
$f(x, y, z) = \bar{x}\,\bar{y}\,\bar{z} \vee \bar{x}\,y\,z \vee \bar{x}\,\bar{y}\,z \vee x\,y\,z$

До конституентів одиниці, що відповідають будь-яким двом сусіднім кліткам, можна застосувати операцію склеювання, оскільки вони відрізняються тільки однією змінною. Зауважимо, що на карті Карно для функції трьох змінних кожна клітка має три сусідні клітки, на карті Карно для функції чотирьох змінних — чотири, на карті Карно для функції п'яти змінних — п'ять тощт. При цьому для кліток на карті Карно функції трьох змінних, розташованих у крайньому правому або у крайньому лівому стовпцях, сусідніми є клітки, що розташовані безпосередньо зліва (або справа), вище (або нижче) у сусідньому рядку, і крайня ліва (або права) у тому ж рядку. Як приклад на рис. 4.4 відзначено клітки, сусідні з клітками A і B.

z \ xy	00	01	11	10
0		—		
1	—	A	—	

z \ xy	00	01	11	10
0				—
1	—		—	B

Рис. 4.4. Клітки, сусідні з клітками A і B

Правило склеювання кліток і запису мінімальної ДНФ

1. Будується карта Карно, що відповідає даній функції.

2. Клітки об'єднуються у групи. що позначають операції склеювання. В об'єднанні беруть участь тільки сусідні клітки, в яких знаходяться одиниці.

3. В групу можні об'єднати тільки кількість кліток, що дорівнює 2^n, $n = 1, 2, 3, \ldots$. При цьому група може мати тільки прямокутну або квадратну форму.

4. Задача склеювання полягає у знаходженні набору максимальних груп кліток. Максимальна група — це група, яка не входить цілком у жодну іншу групу і відповідає простій імпліканті функції. Кількість груп у такому наборі повинна

бути мінімальною, оскільки така група відповідає мінімальній тупиковій ДНФ. Кожна одиниця карти Карно повинна входити хоча б до однієї групи, що забезпечує покриття функції одержаним *набором імплікант*.

5. Кожна група кліток, що одержана після склеювання, відповідає тій імпліканті функції, реальні змінні якої мають однакове значення для всіх кліток групи. Змінні беруться без заперечення, якщо їм відповідають одиничні значення, і із запереченням — в іншому випадку.

6. Диз'юнкція всіх одержаних простих імплікант зображує результат мінімізації формули і є *мінімально* ДНФ.

Оскільки при мінімізації на множині ДНФ у склеюванні беруть участь тільки клітки, що містять одиниці, нулі в картах Карно, як правило, не вказують і мають на увазі, що порожні клітки містять нулі.

Приклад. Знайти мінімальну ДНФ функції з попереднього прикладу.

Розв'язок. Опускаючи нулі на рис. 4.3, записуємо вихідну карту Карно у такому вигляді (рис. 4.5).

z \ xy	00	01	11	10
0	1			
1	1	1	1	

Рис. 4.5. Карта Карно функції $f(x, y, z)$

Графічно склеювання позначається об'єднанням сусідніх одиниць таблиці у групи, результатом чого є дві імпліканти A і B (рис. 4.6).

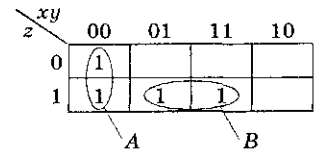


Рис. 4.6. Знаходження імплікант функції $f(x, y, z)$

Імпліканта $A = \bar{x}\,\bar{y}\,\bar{z} \vee \bar{x}\,\bar{y}\,z = \bar{x}\,\bar{y}\,(\bar{z} \vee z) = \bar{x}\,\bar{y}$.

Імпліканта $B = \bar{x}\,y\,z \vee x\,y\,z = (\bar{x} \vee x)yz = yz$.

Таким чином, одержимо мінімальну ДНФ:

$$f(x, y, z) = A \vee B = \bar{x}\,\bar{y} \vee yz.$$

Приклад. Знайти МДНФ для функції

$$f(x, y, z) = \bar{x}y\bar{z} \vee xy\bar{z} \vee \bar{x}\bar{y}z \vee xyz.$$

Розв'язок. Побудуємо карту Карно для заданої функції (рис. 4.7).

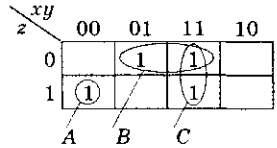


Рис. 4.7. Карта Карно для $f(x, y, z) = \bar{x}\,y\,\bar{z} \vee x\,y\,\bar{z} \vee \bar{x}\,\bar{y}\,z \vee x\,y\,z$

Мінімальна ДНФ $f(x, y, z) = A \vee B \vee C = \bar{x}\bar{y}\ z \vee y\bar{z} \vee xy.$

Приклад. Знайти МДНФ такої функції:

$$f(x, y, z) = x\ y\ z \vee x\ y\bar{z} \vee x\bar{y}\ z \vee x\bar{y}\bar{z} \vee \bar{x}\ y\bar{z} \vee \bar{x}\bar{y}\ z \vee \bar{x}\bar{y}\bar{z}.$$

Розв'язок. Побудуємо відповідну карту Карно (рис. 4.8).
Мінімальна ДНФ $f(x, y, z) = A \vee B \vee C = \bar{y} \vee \bar{z} \vee x.$

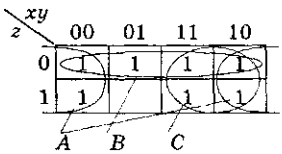


Рис. 4.8. Карта Карно для функції $f(x, y, z)$

Розглянемо застосування карт Карно для мінімізації функцій, що залежать від чотирьох змінних. Карта Карно для функції чотирьох змінних має розмір 4×4. Кожна клітка має чотири сусідніх. Правила склеювання кліток і запису результуючої формули залишаються попередніми. Відмінність полягає в тому, що сусідніми необхідно вважати не тільки крайній правий та крайній лівий стовпці, але також крайній верхній і крайній нижній рядки.

Приклад. Побудувати мінімальну ДНФ для функції

$$f(x, y, z, t) = x\ y\ z\bar{t} \vee x\bar{y}\ z\ t \vee \bar{x}\ y\ z\ t \vee$$
$$\vee\ \bar{x}\bar{y}\ z\ t \vee x\bar{y}\bar{z}\ t \vee \bar{x}\bar{y}\bar{z}\ t \vee x\bar{y}\bar{z}\bar{t} \vee \bar{x}\bar{y}\bar{z}\bar{t}.$$

Розв'язок. Побудуємо відповідну карту Карно (рис. 4.9).
Запишемо мінімальну ДНФ, поєднавши диз'юнкцією прості імпліканти A, B, C, D відповідно:

$$f(x, y, z, t) = \bar{y}\bar{z} \vee \bar{y}\ t \vee \bar{x}\ z\ t \vee x\ y\ z\bar{t}.$$

Карта Карно для функції п'яти змінних зображує у просторі двошаровий паралелепіпед, де кожний шар відповідає карті Карно від перших чотирьох змінних функції. Перший шар зображує на площині всілякі інтерпретації, при яких п'ята змінна дорівнює нулю, а другий — всілякі інтерпретації, при яких п'ята змінна дорівнює одиниці. Кожна клітка на карті Карно для функції п'яти змінних має п'ять сусідніх кліток: чотири — на своєму шарові карти і п'ята — на сусідній, тобто клітку, яка співпаде з даною, якщо розташувати шари карти один поверх

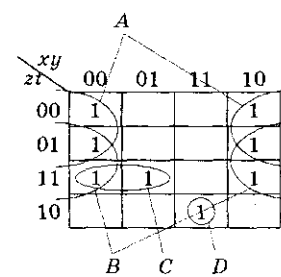


Рис. 4.9. Карта Карно для $f(x, y, z, t)$

іншого. Об'єднання кліток у групи на кожному шарі карті здійснюється як і у випадку чотирьох змінних. Крім того, можна склеювати дві однакові групи, що знаходяться на різних шарах карти, розташування яких співпаде, якщо помістити один шар карти поверх іншого.

Приклад. Побудувати мінімальну ДНФ для функції

$$f(x, y, z, t, w) =$$
$$= \bar{x}\,\bar{y}\,\bar{z}\,\bar{t}\,\bar{w} \vee \bar{x}\,y\,\bar{z}\,\bar{t}\,\bar{w} \vee x\,y\,\bar{z}\,\bar{t}\,\bar{w} \vee x\,\bar{y}\,\bar{z}\,\bar{t}\,\bar{w} \vee x\,y\,\bar{z}\,\bar{t}\,w \vee$$
$$\vee \bar{x}\,y\,\bar{z}\,\bar{t}\,w \vee \bar{x}\,\bar{y}\,z\,t\,\bar{w} \vee x\,y\,z\,t\,\bar{w} \vee x\,y\,z\,\bar{t}\,\bar{w} \vee x\,\bar{y}\,z\,\bar{t}\,\bar{w}.$$

Розв'язок. Побудуємо карту Карно для даної функції (рис. 4.10).

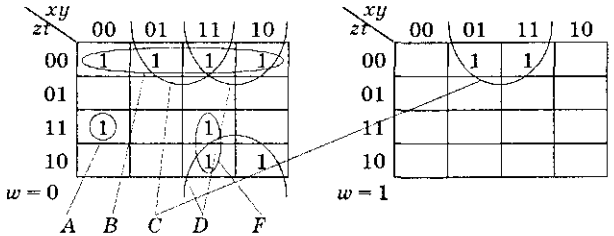


Рис. 4.10. Карта Карно для функції $f(x, y, z, t, w)$

Прості імпліканти, що входять до мінімальної ДНФ:

$A = \bar{x}\,\bar{y}\,z\,t\,\bar{w}$, $\quad B = \bar{z}\,\bar{t}\,\bar{w}$,
$C = y\,\bar{z}\,\bar{t}$, $\quad D = x\,\bar{t}\,\bar{w}$,
$F = x\,y\,z\,\bar{w}$.

Запишемо мінімальну ДНФ, поєднавши одержані імпліканти знаками диз'юнкції:

$$f(x, y, z, t, w) = \bar{x}\,\bar{y}\,z\,t\,\bar{w} \vee \bar{z}\,\bar{t}\,\bar{w} \vee y\,\bar{z}\,\bar{t} \vee x\,\bar{t}\,\bar{w} \vee x\,y\,z\,\bar{w}.$$

Мінімізація на множині КНФ

Для мінімізації булевої функції на множині КНФ використовуються діаграми Вейча. Їх побудова аналогічна картам Карно. На карті позначаються клітки, що відповідають інтерпретаціям, на яких функція дорівнює нулю. Після цього проводиться склеювання кліток, що містять нулі і формування мінімальної КНФ. Склеювання кліток здійснюється за тими ж правилами, що й при диз'юнктивній мінімізації. Кожна група

кліток, що одержана в результаті склеювання, відповідає диз'юнкції тільки тих змінних, які мають однакове значення для всіх кліток групи. Змінні беруться без заперечення, якщо їм відповідає нульове значення, і із запереченням — в іншому випадку. Кон'юнкція одержаних елементарних диз'юнкцій є результатом мінімізації формули.

Приклад. Одержати мінімальну КНФ для функції, що задана ДКНФ таким чином:

$$f(x, y, z, t, w) =$$
$$= (x \vee y \vee z \vee t \vee w)\,(x \vee y \vee z \vee t \vee \bar{w})\,(x \vee y \vee \bar{z} \vee$$
$$\vee\, t \vee w)\,(x \vee y \vee \bar{z} \vee \bar{t} \vee w) \wedge (x \vee \bar{y} \vee \bar{z} \vee t \vee w)(x \vee \bar{y} \vee \bar{z} \vee$$
$$\vee\, \bar{t} \vee w)(\bar{x} \vee y \vee z \vee t \vee w)(\bar{x} \vee y \vee z \vee t \vee \bar{w})(\bar{x} \vee \bar{y} \vee \bar{z} \vee t \vee w) \wedge$$
$$\wedge\, (\bar{x} \vee y \vee \bar{z} \vee \bar{t} \vee \bar{w}).$$

Розв'язок. Дана функція обертається на нуль на таких наборах: (0, 0, 0, 0, 0), (0, 0, 0, 0, 1), (0, 0, 1, 0, 0), (0, 0, 1, 1, 0), (0, 1, 1, 0, 0), (0, 1, 1, 1, 0), (1, 0, 0, 0, 0), (1, 0, 0, 0, 1), (1, 1, 1, 0, 0), (1, 0, 1, 1, 1). Побудуємо карту Карно для цієї функції (рис. 4.11).

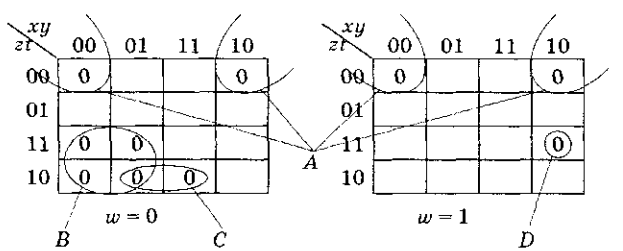


Рис. 4.11. Карта Карно для функції $f(x, y, z, t, w)$

Запишемо мінімальну КНФ, поєднавши знаками кон'юнкції елементарні диз'юнкції:

$$f(x, y, z, t, w) = A \wedge B \wedge C \wedge D =$$
$$= (y \vee z \vee t)(x \vee \bar{z} \vee w)(\bar{y} \vee \bar{z} \vee t \vee w)(\bar{x} \vee y \vee \bar{z} \vee \bar{t} \vee \bar{w}).$$

Мінімізація частково визначених функцій

Якщо для розв'язку задачі використовуються не всі набори вхідних даних, то припустиме будь-яке значення функції на невикористовуваних інтерпретаціях і така функція називається

частково визначеною. Іншими словами, функція не визначена на вказаних інтерпретаціях. При мінімізації такі функції довизначаються так, щоб одержати найбільш економічну мінімальну ДНФ (КНФ).

Приклад. Функція $f(x, y, z, t)$ дорівнює одиниці на наборах (0, 0, 1, 0), (0, 1, 1, 0), (1, 0, 1, 0), (1, 0, 0, 0) і не визначена, якщо $xy = 1$. Необхідно побудувати мінімальну ДНФ.

Розв'язок. Складемо карту Карно для заданої функції (рис. 4.12).

Мінімальна ДНФ

$$f(x, y, z, t) = A \vee B = z\bar{t} \vee x\bar{t}.$$

Метод карт Карно не є формальним і складний у реалізації. Крім того, він не зручний у застосуванні до функцій з кількістю змінних більше шести. Інший метод мінімізації, який буде розглянутий нижче, придатний для програмної реалізації і може застосовуватися при великій кількості змінних.

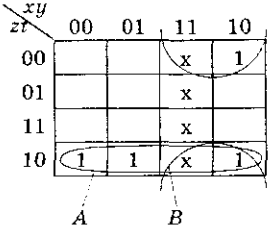


Рис. 4.12. Карта Карно для частково визначеної функції

4.11.3. Мінімізація функцій методом Квайна — Мак-Класкі

Метод мінімізації Квайна — Мак-Класкі також реалізує перехід від ДДНФ до мінімальної ДНФ з використовуванням операцій склеювання та поглинання. Він був запропонований В. Квайном, а потім удосконалений Мак-Класкі.

Алгоритм Квайна складається з таких кроків:

1. Записати ДДНФ заданої функції.
2. Виконати всі можливі операції неповного диз'юнктивного склеювання. Результуюча формула є диз'юнкцією всіх можливих імплікант даної функції.
3. Виконати всі можливі операції диз'юнктивного поглинання. Результуюча формула є скороченою ДНФ даної функції.
4. Скласти імплікантну таблицю і знайти диз'юнктивне ядро.

5. Спростити імплікантну таблицю за допомогою видалення рядків, що відповідають імплікантам диз'юнктивного ядра, і стовпців, що відповідають тим конституентам одиниці, які покриваються імплікантами ядра.
6. Знайти всі тупикові ДНФ даної функції.
7. Знайти мінімальну ДНФ.

Якщо при мінімізації деякої функції виходить, що спрощена імплікантна таблиця порожня, то тупикова ДНФ цієї функції є мінімальною і складається тільки з імплікант диз'юнктивного ядра. Якщо ж спрощена імплікантна таблиця не порожня, то в кожній тупиковій ДНФ функції, крім імплікант ядра, присутні і деякі прості імпліканти, що не входять до диз'юнктивного ядра. Набір даних імплікант визначає різниці між тупиковими ДНФ.

Приклад. Знайти методом Квайна мінімальну ДНФ функції

$$f(x, y, z) = x\,y\,z \vee x\,\bar{y}\,\bar{z} \vee \bar{x}\,y\,z \vee \bar{x}\,\bar{y}\,z \vee \bar{x}\,\bar{y}\,\bar{z}.$$

Розв'язок. Виконаємо всі можливі операції диз'юнктивного склеювання і поглинання:

$$x\,y\,z \vee \bar{x}\,y\,z = y\,z,$$
$$x\,\bar{y}\,\bar{z} \vee \bar{x}\,\bar{y}\,\bar{z} = \bar{y}\,\bar{z},$$
$$\bar{x}\,y\,z \vee \bar{x}\,\bar{y}\,z = \bar{x}\,z,$$
$$\bar{x}\,\bar{y}\,z \vee \bar{x}\,\bar{y}\,\bar{z} = \bar{x}\,\bar{y}.$$

Одержимо таку формулу:

$$f(x, y, z) = x\,y\,z \vee x\,\bar{y}\,\bar{z} \vee \bar{x}\,y\,z \vee \bar{x}\,\bar{y}\,z \vee \bar{x}\,\bar{y}\,\bar{z} =$$
$$= y\,z \vee \bar{y}\,\bar{z} \vee \bar{x}\,z \vee \bar{x}\,\bar{y}.$$

Зробивши попарне порівняння всіх елементарних кон'юнкцій, що входять у дану формулу, приходимо до висновку, що одержати інші елементарні кон'юнкції за допомогою операції склеювання в даній формулі неможливо. Таким чином, одержано диз'юнкцію всіх можливих імплікант даної функції:

$$f(x, y, z) = y\,z \vee \bar{y}\,\bar{z} \vee \bar{x}\,z \vee \bar{x}\,\bar{y}.$$

Одержана формула є скороченою ДНФ даної функції.

Складемо імплікантну таблицю (таблиця 4.33). Її рядки задаються простими імплікантами, а стовпці — конституентами

одиниці функції. Зірочкою відзначається кожна клітка таблиці, для якої імпліканта з рядка є власною частиною конституенти із стовпця.

Таблиця 4.33. Імплікантна таблиця функції $f(x, y, z)$

	$x\,y\,z$	$x\bar{y}\bar{z}$	$\bar{x}\,y\,z$	$\bar{x}\bar{y}\,z$	$\bar{x}\bar{y}\bar{z}$
$y\,z$	*		*		
$\bar{y}\bar{z}$		*			*
$\bar{x}\,z$			*	*	
$\bar{x}\bar{y}$				*	*

Знайдемо диз'юнктивне ядро. До нього входить кожна проста імпліканта, яка є єдиною у покритті будь-якої конституенти одиниці. В імплікантній таблиці по одному знаку «*» містять стовпці, що відповідають конституентам одиниці xyz і $x\,\bar{y}\,\bar{z}$, напроти простих імплікант yz і $\bar{y}\bar{z}$. Ці прості імпліканти складають диз'юнктивне ядро. Складемо спрощену імплікантну таблицю. Для цього з імплікантної таблиці викреслюємо рядки, що відповідають імплікантам диз'юнктивного ядра, і стовпці, що відповідають конституентам одиниці, які покриті імплікантами ядра (таблиця 4.34).

Таблиця 4.34. Одержання спрощеної імплікантної таблиці

	$x\,y\,z$	$x\bar{y}\bar{z}$	$\bar{x}\,y\,z$	$\bar{x}\bar{y}\,z$	$\bar{x}\bar{y}\bar{z}$
$y\,z$	⊛		*		
$\bar{y}\bar{z}$		⊛			*
$\bar{x}\,z$			*	*	
$\bar{x}\bar{y}$				*	*

В даному випадку імпліканти ядра покривають всі конституенти одиниці, крім однієї, тому спрощена імплікантна таблиця має такий вигляд (таблиця 4.35):

Таблиця 4.35. Спрощена імплікантна таблиця

	$\bar{x}\bar{y}\,z$
$\bar{x}\,z$	*
$\bar{x}\bar{y}$	*

Із спрощеної імплікантної таблиці знаходимо, що тупикові ДНФ даної функції, крім диз'юнктивного ядра, включають або імпліканту $\bar{x}\,z$, або $\bar{x}\,\bar{y}$. Таким чином, одержуємо дві тупикові ДНФ для даної функції:

ДНФ1: $f(x, y, z) = y\,z \vee \bar{y}\,\bar{z} \vee \bar{x}\,z$;

ДНФ2: $f(x, y, z) = y\,z \vee \bar{y}\,\bar{z} \vee \bar{x}\,\bar{y}$.

Вказані ДНФ містять по 6 символів змінних, а також однакову кількість знаків операцій диз'юнкції (2) і кон'юнкції (3), тому виберемо як мінімальну ДНФ1, що містить менше знаків операцій заперечення.

В методі Квайна для функції більшого числа змінних перелічення варіантів склеювання і множини тупикових ДНФ є значною складністю. Удосконалення, що введене Мак-Класкі, спрощує запис мінімізаційної формули. Згідно з його пропозицією конституенти одиниці записуються у вигляді двійкового коду — номери даної конституенти. Вказаний номер відповідає інтерпретації, на якій конституента дорівнює одиниці. Всі останні імпліканти, що одержані в результаті застосування операцій неповного склеювання, також записуються у вигляді двійкових кодів. У позиціях коду, що відповідають реальним змінним імпліканти, поміщається набір значень змінних, який обертає імпліканту на одиницю. В позиціях фіктивних змінних імпліканти ставиться прочерк. Таким чином, операції склеювання і поглинання проводяться не з самими імплікантами, а з їх двійковими кодами, що робить алгоритм простим у програмній реалізації.

Крім того, множина кодів імплікант ділиться на групи за кількістю одиниць, що в них утримуються. Оскільки операції неповного склеювання застосовні до імплікант, коди яких відрізняються значенням тільки в одній позиції, в склеюванні можуть брати участь тільки імпліканти, що належать до сусідніх груп, номери яких відрізняються на одиницю. Крім того, виконання операцій склеювання здійснюється покроково. На першому кроці здійснюються всі можливі склеювання конституент одиниці, на другому кроці здійснюється склеювання на множині імплікант, що одержані на першому кроці, тощо. Ділення на групи та кроки дозволяє істотно скоротити кількість перевірок пар імплікант на

можливість здійснення склеювання і робить алгоритм більш ефективним.

При виконанні операцій поглинання з формули видаляються всі імпліканти, що не є простими. Результуюча формула зображує диз'юнкцію простих імплікант функції — її скорочену ДНФ. Імпліканти, що брали участь хоча б в одному склеюванні, не є простими. Отже, замість операцій поглинання можна видалити з формули всі імпліканти, що брали участь хоча б в одному склеюванні, з однаковим результатом. Тому при виконанні операцій склеювання одержані імпліканти позначаються. Після виконання всіх можливих склеювань позначені імпліканти видаляються. Це дає той же результат, що і при виконанні операцій поглинання.

Формальний метод знаходження множини всіх тупикових ДНФ функції був запропонований С. Петриком. У цьому методі кожній з простих імплікант функції, що не входять до диз'юнктивного ядра, приписується буквене позначення. Нагадаємо, що пошук простих імплікант проводиться у спрощеній імплікантній таблиці. За допомогою цієї таблиці записується логічна формула покриття конституент одиниці простими імплікантами. Для цього для кожної конституенти одиниці складається диз'юнкція літерних позначень всіх простих імплікант, які її покривають. Далі записується формула, що зображує кон'юнкцію одержаних елементарних диз'юнкцій. В формулі розкриваються всі дужки, після чого вона здобуває вид диз'юнкції елементарних кон'юнкцій. Кожна елементарна кон'юнкція в одержаній формулі відповідає одній з тупикових ДНФ функції. Для побудови такої ДНФ необхідно записати імпліканти, що відповідають літерним позначенням, які входять до складу даної елементарної кон'юнкції, й імпліканти диз'юнктивного ядра функції.

Приклад. Знайти за допомогою метода Квайна — Мак-Класкі мінімальну ДНФ функції $f(x, y, z, t)$, що задана такою ДДНФ:

$$f(x, y, z, t) = \bar{x}\,\bar{y}\,\bar{z}\,\bar{t} \vee \bar{x}\,\bar{y}\,\bar{z}\,t \vee \bar{x}\,\bar{y}\,z\,\bar{t} \vee \bar{x}\,\bar{y}\,z\,t \vee \bar{x}\,y\,\bar{z}\,t \vee$$
$$\vee\, \bar{x}\,y\,z\,t \vee x\,\bar{y}\,\bar{z}\,\bar{t} \vee x\,\bar{y}\,z\,\bar{t} \vee x\,y\,\bar{z}\,\bar{t} \vee x\,\bar{y}\,z\,t \vee x\,y\,\bar{z}\,t.$$

Розв'язок. У таблиці 4.36 запишемо конституенти одиниці даної функції у вигляді двійкових кодів у другий стовпчик. У першому стовпці запишемо десяткові номери інтерпретацій.

Таблиця 4.36. Двійкові коди конституент одиниці функції

Десяткові номери інтерпретацій	Двійкові коди конституент одиниці	Конституенти одиниці функції $f(x, y, z, t)$
0	0000	$\bar{x}\bar{y}\bar{z}\bar{t}$
1	0001	$\bar{x}\bar{y}\bar{z}t$
2	0010	$\bar{x}\bar{y}z\bar{t}$
3	0011	$\bar{x}\bar{y}zt$
5	0101	$\bar{x}y\bar{z}t$
7	0111	$\bar{x}yzt$
8	1000	$x\bar{y}\bar{z}\bar{t}$
10	1010	$x\bar{y}z\bar{t}$
11	1011	$x\bar{y}zt$
12	1100	$xy\bar{z}\bar{t}$
13	1101	$xy\bar{z}t$

Відповідно до методу здійснимо такі кроки:

1. Згрупуємо двійкові коди імплікант з однаковою кількістю одиниць. Назвемо число одиниць m індексом групи. Упорядкуємо групи у порядку зростання m.
2. Починаючи з $m = 0$, зробимо порівняння кожного двійкового коду в групі з індексом m з кожним кодом з групи з індексом $m + 1$. Якщо порівнювані двійкові коди різні тільки в одному розряді, то у наступний стовпчик таблиці запишемо відповідний їм двійковий код з порожньою позначкою «-» на місці зазначеного розряду. Напроти кожного нового коду запишемо номери кодів двох імплікант, що брали участь у порівнянні, і у наступному стовпчику ці імпліканти позначимо знаком «V», оскільки вони не є простими імплікантами. Всі коди, які залишилися непозначеними знаком «V», відповідають простим імплікантам, тому позначимо їх знаком «X».
3. Якщо серед знов одержаних імплікант є однакові, то з них для подальшого використання залишаємо тільки одну.
4. Повторюємо кроки 1–3 доти, доки існує можливість одержувати нові коди імплікант.

Результат виконання описаних кроків наведено у таблиці 4.37. Спочатку заповнюємо три стовпця нульового

циклу: m (десятковий індекс групи), двійковий код імпліканти, номер імпліканти. Імпліканти ділимо на групи за значенням m.

Таблиця 4.37. Одержання простих імплікант функції $f(x, y, z, t)$ методом Квайна — Мак-Класкі

ЦИКЛ 0				ЦИКЛ 1				ЦИКЛ 2			
m	Код імпл.	№ імпл.	Вид імпл.	m	Код	№	Вид	m	Код	№	Вид
0	0000	0	V	0	000-	0, 1	V	0	00--	(0, 1), (2, 3)	X
1	0001	1	V	0	00-0	0, 2	V	0	~~00--~~	~~(0, 2), (1, 3)~~	
1	0010	2	V	0	-000	0, 8	V	0	-0-0	(0, 2), (8, 10)	X
1	1000	8	V	1	00-1	1, 3	V	0	~~-0-0~~	~~(0, 8), (2, 10)~~	
2	0011	3	V	1	0-01	1, 5	V	1	0--1	(1, 3), (5, 7)	X
2	0101	5	V	1	001-	2, 3	V	1	~~0--1~~	~~(1, 5), (3, 7)~~	
2	1010	10	V	1	-010	2, 10	V	1	-01-	(2, 3), (10, 11)	X
2	1100	12	V	1	10-0	8, 10	V	1	~~-01-~~	~~(2, 10), (3, 11)~~	
3	0111	7	V	1	1-00	8, 12	X				
3	1011	11	V	2	0-11	3, 7	V				
3	1101	13	V	2	-011	3, 11	V				
				2	01-1	5, 7	V				
				2	-101	5, 13	X				
				2	101-	10, 11	V				
				2	110-	12, 13	X				

Потім здійснимо порівняння першої імпліканти з групи $m = 0$ (імпліканта 0000) з першою імплікантою з групи $m = 1$ (імпліканта 0001). Вони відрізняються тільки в одному розряді, тому робимо їх склеювання та одержуємо нову імпліканту з кодом 000-. Записуємо даний код у стовпчик коду імпліканти циклу 1, а напроти імплікант 0000 і 0001 ставимо позначки «V», оскільки вони не є простими. Їх номери вказуємо поряд з кодом 000-. Потім порівнюємо імпліканту 0000 з другою імплікантою з групи $m = 1$ — імплікантою 0010 тощо.

Коли порівняння всіх кодів імплікант стовпця «цикл 0» завершено, аналізуємо стовпчик виду імпліканти. Напроти всіх імплікант ставимо позначення «V», оскільки в циклі 0 кожна імпліканта припускає склеювання і не є простою.

Далі розділяємо на групи $m = 0$, $m = 1$ і $m = 2$ коди імплікант циклу 1. Потім попарно порівнюємо ці імпліканти й одержуємо стовпчик коду імплікант «цикл 2». Коди, що містять знаки «-», можуть утворювати нові імпліканти, тільки якщо вони містять знаки «-» в одних і тих же розрядах. Закінчивши порівняння, виділяємо прості імпліканти у стовпці «вид» «циклу 1», які не мають позначки «V», і позначаємо їх позначкою «X». У стовпці «код» «циклу 2» є однакові імпліканти, тому довільно викреслюємо один з однакових рядків, щоб кожна імпліканта зустрічалася тільки один раз. Порівнюючи коди імплікант у «циклі 2», приходимо до висновку, що методом склеювання з них неможливо одержати нові імпліканти. Всі коди циклу 2 відповідають *простим імплікантам*, отже, побудову таблиці завершено.

Для знаходження тупикових ДНФ побудуємо імплікантну таблицю, в рядках якої розташовуємо двійкові коди, що відповідають простим імплікантам, а в стовпцях — коди, що відповідають конституентам одиниці. Якщо двійковий код рядку є частиною коду стовпця (позиції із знаком «-» не порівнюються), то у відповідну клітку таблиці записуємо знак «*» (таблиця 4.38).

Таблиця 4.38. Імплікантна таблиця функції f(x, y, z, t)

	0000	0001	0010	0011	0101	0111	1000	1010	1011	1100	1101
1-00							*			*	
-101					*						*
110-										*	*
00--	*	*	*	*							
-0-0	*		*				*	*			
0--1		*		*	*	*					
-01-			*	*				*	*		

Відзначимо стовпці таблиці, які містять по одному знаку «*». Відповідні їм прості імпліканти є *диз'юнктивними ядрами*. Викреслюємо рядки таблиці, що відповідають ядрам, і стовпці, покриті ядрами, як зображено у таблиці 4.39.

Таблиця 4.39. Одержання спрощеної імплікантної таблиці

	0000	0001	0010	0011	0101	0111	1000	1010	1011	1100	1101
1-00							*			*	
-101					*						*
110-										*	*
00--	*	*	*	*							
-0-0	*		*				*	*			
0--1		*		*	*	⊛					
-01-			*	*				*	⊛		

Одержуємо спрощену імплікантну таблицю (таблиця 4.40).

Таблиця 4.40. Спрощена імплікантна таблиця

		0000	1000	1100	1101
A	1-00		*	*	
B	-101				*
C	110-			*	*
D	00--	*			
E	-0-0	*	*		

Тепер знайдемо всі тупикові ДНФ функції $f(x, y, z, t)$ і оберемо з них мінімальну. Згідно з методом Петрика кожній з імплікант таблиці приписуємо літерне позначення і записуємо формулу покриття конституент одиниці простими імплікантами. Конституента одиниці з кодом 0000 може бути покрита імплікантою D або імплікантою E, тобто диз'юнкції: $D \vee E$. Конституента одиниці з кодом 1000 може бути покрита імплікантою A або імплікантою E: $A \vee E$. Аналогічно 1100 може бути покрита $A \vee C$, а 1101 — диз'юнкцією $B \vee C$. Таким чином, покриття конституент одиниці спрощеної імплікантної таблиці функції $f(x, y, z, t)$ простими імплікантами можна записати у вигляді формули покриття:

$$(D \vee E)(A \vee E)(A \vee C)(B \vee C).$$

Якщо в цій формулі розкрити дужки, то отримаємо символьне зображення всіх можливих наборів простих імплікант для тупикових ДНФ, що не включає диз'юнктивні ядра:

$(D \vee E)(A \vee E)(A \vee C)(B \vee C) = (DA \vee E)(AB \vee C) =$
$= DAB \vee EAB \vee DAC \vee EC.$

В одержаній формулі кожна кон'юнкція буквених позначень відповідає набору імплікант у деякій тупиковій ДНФ, до якої обов'язково входять також диз'юнктивні ядра. Для того щоб одержати мінімальну ДНФ, необхідно вибрати набір з мінімальною кількістю імплікант (у даному прикладі це EC) і додати імпліканти ядра (0--1, -01-). Одержуємо мінімальну ДНФ вихідної функції f:

$$\text{МДНФ } f(x, y, z, t) = \bar{y}\,\bar{t} \vee x\,y\,\bar{z} \vee \bar{x}\,t \vee \bar{y}\,z.$$

Таким чином, за допомогою метода Квайна — Мак-Класкі одержана мінімальна ДНФ функції $f(x, y, z, t)$, яка містить всього лише 4 елементарні кон'юнкції замість 11 конституент одиниці ДДНФ.

Недолік методу мінімізації булевих функцій Квайна — Мак-Класкі полягає у необхідності записувати ДДНФ функції, яка вже при семи змінних може містити більше ста конституент одиниці. Наступний метод дозволяє здійснювати диз'юнктивну мінімізацію, використовуючи як вихідну довільну ДНФ функції.

4.11.4. Мінімізація функцій методом Порецького — Блейка

Метод мінімізації Порецького — Блейка реалізує перехід від довільної ДНФ функції до скороченої ДНФ за допомогою операцій узагальненого склеювання і поглинання.

Операція ***узагальненого склеювання***:

$$Ax \vee B\bar{x} = Ax \vee B\bar{x} \vee AB.$$

Доведемо справедливість даної тотожності:

$$Ax \vee B\bar{x} = Ax(1 \vee B) \vee B\bar{x}(1 \vee A) = Ax \vee ABx \vee B\bar{x} \vee AB\bar{x} =$$
$$= Ax \vee B\bar{x} \vee AB(x \vee \bar{x}) = Ax \vee B\bar{x} \vee AB.$$

Метод Порецького — Блейка полягає у застосуванні різних операцій узагальненого склеювання до ДНФ функції. Потім у одержаній формулі здійснюються всі можливі операції поглинання.

Приклад. Знайти скорочену ДНФ за методом Порецького — Блейка для функції $f(x, y, z) = xy\bar{z} \vee xz \vee \bar{x}y$.

Розв'язок. Здійснимо операції узагальненого склеювання:

$$xy\bar{z} \vee xz = xy \vee xy\bar{z} \vee xz, \qquad \text{тут } A = xy,\ B = x;$$

$$xy\bar{z} \vee \bar{x}y = y\bar{z} \vee xy\bar{z} \vee \bar{x}y, \qquad \text{тут } A = y\bar{z},\ B = y;$$

$$xz \vee \bar{x}y = yz \vee xz \vee \bar{x}y, \qquad \text{тут } A = z,\ B = y.$$

Очевидно, диз'юнкція лівих частин співпадає з функцією f.

До знову одержаних імплікант може бути застосована операція повного диз'юнктивного склеювання (п. 4.11.1):

$$y\bar{z} \vee yz = y, \quad xy \vee \bar{x}y = y.$$

Таким чином, вихідна формула прийме такий вигляд:

$$f(x, y, z) = xy\bar{z} \vee xz \vee \bar{x}y = (xy \vee \bar{x}y) \vee xy\bar{z} \vee xz \vee (yz \vee y\bar{z}) =$$
$$= y \vee xy\bar{z} \vee xz \vee y = y(x\bar{z} \vee 1) \vee xz = xy \vee z.$$

Отже, $f = xy \vee z$ є скорочена ДНФ.

Запитання

1. В чому полягає задача мінімізації булевих функцій? У чому особливість її канонічної форми?
2. Дайте визначення поняттю імпліканти булевої функції.
3. Що зображує повна система імплікант?
4. Яка імпліканта називається простою?
5. Дайте визначення скороченої, тупикової і мінімальної диз'юнктивних нормальних форм?
6. Що називають диз'юнктивним ядром булевої функції?
7. Для яких цілей служить імплікантна таблиця?
8. Запишіть формули операцій диз'юнктивного склеювання і поглинання.
9. Запишіть формули операцій кон'юнктивного склеювання і поглинання.
10. Що зображує карта Карно (діаграма Вейча)?
11. Сформулюйте правило склеювання кліток і запису мінімальної ДНФ за методом карт Карно.
12. В чому відмінність застосування діаграм Вейча для мінімізації на множині КНФ від карт Карно?
13. Яким чином здійснюється мінімізація частково визначених функцій?
14. Назвіть основні кроки алгоритму мінімізації Квайна.

15. Які модифікації запропонував внести Мак-Класкі у метод мінімізації Квайна?
16. Для чого призначений метод Петрика?
17. В чому полягає недолік методу мінімізації булевих функцій Квайна — Мак-Класкі?
18. В чому суть методу мінімізації булевих функцій Порецького — Блейка?
19. Запишіть формулу операції узагальненого склеювання.
20. Дайте порівняльну характеристику методів мінімізації Порецького — Блейка і Квайна — Мак-Класкі.

Завдання

1. Побудувати карти Карно для таких функцій:
   а) $f(x, y, z, t) = xyzt \vee \bar{x}yzt \vee \bar{x}\bar{y}\bar{z}\bar{t} \vee xy\bar{z}\bar{t}$;
   б) $f(x, y, z, t) = x\bar{y}zt \vee \bar{x}y\bar{z}t \vee xyz\bar{t} \vee \bar{x}yz\bar{t}$;
   в) $f(x, y, z, t) = \bar{x}yz\bar{t} \vee x\bar{y}zt \vee x\bar{y}z\bar{t} \vee xyz\bar{t}$;
   г) $f(x, y, z, t) = xy\bar{z}t \vee \bar{x}yz\bar{t} \vee \bar{x}\bar{y}zt \vee \bar{x}y\bar{z}t$;
   д) $f(x, y, z, t) = \bar{x}yzt \vee \bar{x}\bar{y}zt \vee xyzt \vee x\bar{y}\bar{z}\bar{t}$;
   е) $f(x, y, z, t) = (x \vee y \vee \bar{z} \vee t)(\bar{x} \vee y \vee z \vee \bar{t})(\bar{x} \vee \bar{y} \vee \bar{z} \vee t)(x \vee \bar{y} \vee z \vee t)(x \vee \bar{y} \vee z \vee \bar{t})$.
2. Знайти мінімальні ДНФ функцій, що задані такими картами Карно:

а)

$zt \backslash xy$	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	1

в) $v = 0$

$xy \backslash zt$	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	0	1	1	0
10	0	1	1	0

б)

$zt \backslash xy$	00	01	11	10
00	1	1	1	1
01	1	1	1	0
11	0	1	0	0
10	1	0	0	1

г) $v = 1$

$xy \backslash zt$	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

3. Знайти мінімальні КНФ для функцій, що задані такими діаграмами Вейча:

а)

$zt \backslash xy$	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	1	1	1	1
10	0	0	1	1

б)

$zt \backslash xy$	00	01	11	10
00	0	1	0	1
01	0	1	1	0
11	0	0	0	0
10	1	0	0	1

4. Знайти мінімальні ДНФ частково визначених функцій, що задані такими діаграмами Вейча:

а)

zt \ xy	00	01	11	10
00	-	0	0	-
01	1	-	1	1
11	0	0	-	0
10	-	0	-	-

б)

zt \ xy	00	01	11	10
00	1	-	1	1
01	-	1	-	-
11	0	-	0	0
10	1	0	0	1

5. Знайти мінімальні КНФ частково визначених функцій, заданих такими картами Карно:

а)

zt \ xy	00	01	11	10
00	1	1	1	-
01	0	0	1	1
11	-	-	1	-
10	1	1	1	-

б)

zt \ xy	00	01	11	10
00	-	0	1	1
01	1	1	-	-
11	0	-	0	0
10	1	1	0	-

6. Знайти мінімальні ДНФ функцій методом Квайна — Мак-Класкі. ДДНФ функцій задані номерами конституент одиниці таким чином:
   а) $f(x, y, z, t) = \{0, 1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 15\}$;
   б) $f(x, y, z, t) = \{0, 1, 2, 3, 4, 5, 7, 8, 10, 11, 13, 14\}$;
   в) $f(x, y, z, t) = \{0, 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15\}$;
   г) $f(x, y, z, t) = \{1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 14, 15\}$;
   д) $f(x, y, z, t) = \{1, 3, 4, 5, 6, 8, 9, 10, 12, 13, 15\}$;
   е) $f(x, y, z, t) = \{2, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14\}$.

7. Знайти скорочені ДНФ методом Порецького — Блейка для таких функцій:
   а) $f(x, y, z, t) = yzt \vee \bar{x}zt \vee \bar{x}\bar{y}\bar{t} \vee xy\bar{z}$;
   б) $f(x, y, z, t) = x\bar{y}z \vee \bar{x}\bar{z} \vee \bar{x}\bar{y}\bar{z}\bar{t} \vee \bar{x}y$;
   в) $f(x, y, z, t) = \bar{x}\bar{t} \vee \bar{x}zt \vee x\bar{y}z \vee xyz\bar{t}$;
   г) $f(x, y, z, t) = xy\bar{z} \vee \bar{z}\bar{t} \vee \bar{x}zt \vee \bar{y}t$;
   д) $f(x, y, z, t) = xy\bar{z} \vee \bar{x}z\bar{t} \vee \bar{x}\bar{y}\bar{z}t$;
   е) $f(x, y, z, t) = \bar{y}zt \vee \bar{x}t \vee xyzt \vee x\bar{y}\bar{t}$.

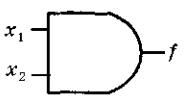
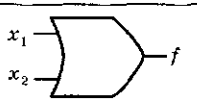
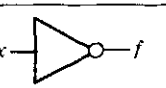
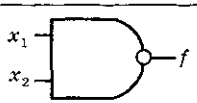
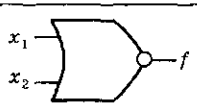
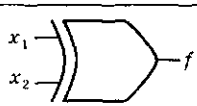
4.12. Логічні схеми

Основні логічні елементи, повнота набору логічних елементів, аналіз та синтез логічних схем

Логічні схеми у комп'ютерах та інших електронних пристроях оперують з наборами вхідних та вихідних даних, що складаються з нулів та одиниць. Булева алгебра і булеві

функції зображують математичний апарат для роботи з такими даними і використовуються для аналізу та синтезу ***логічних схем*** (***ланцюгів***). Основою побудови логічних схем є набір ***логічних елементів***. Кожний логічний елемент реалізує деяку булеву функцію. Його входи відповідають булевим змінним, а вихід — значенню функції. Графічні символи і назви найчастіше використовуваних логічних елементів зображено у таблиці 4.41.

Таблиця 4.41. Основні логічні елементи

Назва	Символ логічного елемента	Булева функція
І		$f = x_1 \wedge x_2$ кон'юнкція
АБО		$f = x_1 \vee x_2$ диз'юнкція
НЕ		$f = \bar{x}$ заперечення
І-НЕ		$\overline{f = x_1 \wedge x_2}$ штрих Шеффера
АБО-НЕ		$\overline{f = x_1 \vee x_2}$ стрілка Пірса
ВИКЛЮЧНЕ АБО		$f = x_1 \oplus x_2$ XOR

Набір логічних елементів ***повний***, якщо з його допомогою можна реалізувати будь-яку булеву функцію. Для повноти такого набору необхідно і достатньо, щоб відповідний йому набір булевих функцій був повний. Як базовий набір елементів для розв'язку конкретної задачі обирають такий набір, за допомогою якого легше всього реалізувати необхідні в даній задачі функції.

Логічна схема будується з набору базових елементів і зображує суперпозицію даних елементів так само, як формула є суперпозицією базових функцій булевої алгебри.

Приклад. Побудувати логічний ланцюг, що реалізує функцію

$$f(x, y, z) = (x \vee y)z.$$

Розв'язок. Використовуючи логічні елементи «І» й «АБО», формуємо суперпозицію, що відповідає даній функції. Одержаний логічний ланцюг зображено на рис. 4.13.

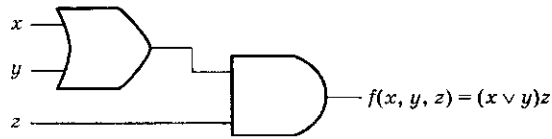


Рис. 4.13. Логічний ланцюг, що реалізує функцію $f(x, y, z) = (x \vee y)z$

Логічні елементи, що реалізують операції з властивостями асоціативності і комутативності, на схемах можуть зображатися з числом входів більше двох, що означає багатократне застосування даної операції. Логічний елемент «АБО» має три входи і реалізує функцію $y = x_1 \vee x_2 \vee x_3$ (рис. 4.14).

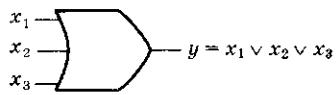


Рис. 4.14. Елемент «АБО», що має три входи

Під час дослідження логічних ланцюгів виникають дві основні задачі: ***аналіз*** та ***синтез***. Аналіз логічного ланцюга полягає у побудові булевої функції, яку реалізує даний логічний пристрій. Для цього визначається значення вихідного сигналу на всіх наборах вхідних даних і складається таблиця істинності функції. Використовуючи таблицю істинності і правила побудови ДДНФ або ДКНФ, можна побудувати формулу, що відповідає даній логічній функції. З іншого боку, використовуючи логічну схему, можна спочатку побудувати формулу, що відповідає шуканій функції, а потім, використовуючи одержану формулу, побудувати таблицю істинності функції. За даною логічною схемою формулу можна побудувати, записавши

суперпозицію булевих функцій, що відповідає схемній суперпозиції логічних елементів.

Задача синтезу полягає у побудові логічного ланцюга для булевої функції, що задана таблицею або за допомогою формули. Використовуючи правила побудови ДДНФ і ДКНФ, можна перейти від таблиці істинності функції до відповідної формули, а потім реалізувати змінні та операції формули логічним ланцюгом.

Вартість логічного ланцюга залежить від його складності. Оскільки економічно рентабельно робити логічні ланцюги мінімальної вартості, булеві функції, за якими здійснюється побудова ланцюгів, повинні бути попередньо мінімізовані. Тому перед побудовою логічного ланцюга необхідно одержати мінімальне зображення функції.

Приклад. Записати булеву функцію, яку реалізує логічний ланцюг, зображений на рис. 4.15. Побудувати мінімальний ланцюг, що реалізує дану функцію.

Розв'язок. Будемо послідовно подавати на вхід схеми інтерпретації функції і визначати її значення на кожній з них (рис. 4.15).

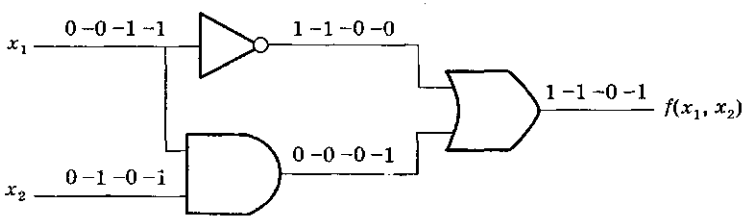


Рис. 4.15. Інтерпретації і значення функції, що реалізована логічним ланцюгом

Побудуємо таблицю істинності шуканої функції (таблиця 4.42).

Таблиця 4.42. Таблиця істинності $f(x_1, x_2)$

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

За таблицею побудуємо ДКНФ даної функції:

$$f(x_1, x_2) = \bar{x}_1 \vee x_2.$$

Одержана формула є мінімальною КНФ даної функції, оскільки містить тільки одну конституенту нуля. Скористаємося другим способом і запишемо формулу, що відповідає суперпозиції логічних елементів, зображеній на вихідній схемі:

$$f(x_1, x_2) = \bar{x}_1 \vee x_1 x_2.$$

Ця формула еквівалентна одержаній раніше, однак є більш складною. Використовуючи мінімальну формулу для функції $f(x_1, x_2)$, можна побудувати мінімальний логічний ланцюг, що реалізує дану булеву функцію (рис. 4.16).

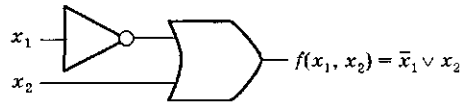


Рис. 4.16. Логічний ланцюг, що реалізує функцію $f(x_1, x_2) = \bar{x}_1 \vee x_2$

Таким чином, мінімізація булевої функції дозволила скоротити вихідну логічну схему на елемент «І».

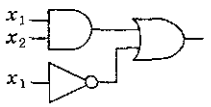
Запитання

1. Що є основою для побудови логічних схем?
2. Як булева алгебра зв'язана з проектуванням логічних ланцюгів?
3. Нарисуйте графічні позначення основних логічних елементів.
4. Який набір логічних елементів називається повним?
5. Якому набору логічних елементів слід віддати перевагу під час розв'язання конкретної задачі?
6. Охарактеризуйте основні задачі, що виникають при дослідженні логічних ланцюгів.
7. У чому полягає аналіз логічних ланцюгів?
8. Яким чином здійснюється синтез логічних ланцюгів?
9. Яку операцію слід здійснити над булевою функцією перед її реалізацією у вигляді логічного ланцюга?

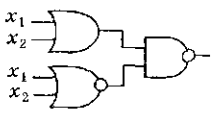
Завдання

1. Провести аналіз таких логічних ланцюгів:

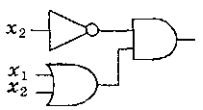
а)



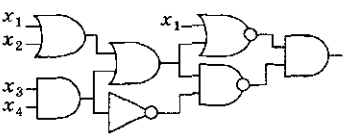
в)



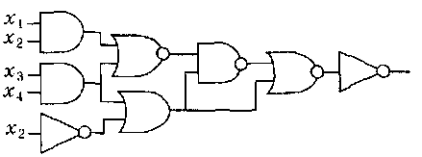
б)



г)



д)



2. Здійснити синтез логічних схем для таких булевих функцій:
   а) $x \to (x \to y)$;
   б) $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$;
   в) $xy \oplus y \oplus x$;
   г) $\bar{z}\,\bar{y} \vee tz \vee \bar{y}z \vee t\bar{z}$;
   д) $(x \wedge y) \to z$;
   е) $x\ y\ z \vee x\ \bar{y}\ \bar{z} \vee \bar{x}\ \bar{y}\ z$.

розділ 5

Математична логіка

5.1. Історія і задачі математичної логіки

Як наука логіка виникла ще у IV ст. до н. е. в працях старогрецького філософа Аристотеля, основоположника формальної *логіки*. Перший його твір, що дійшов до нас, спеціально присвячений логіці, — «Аналітики» (384–322 р. до н. е.). Аристотелю належить відкриття формального характеру логічного висновку, яке полягає в тому, що в наших міркуваннях одні пропозиції виходять з інших через певний зв'язок між їх формою, структурою незалежно від їх змісту. Протягом багатьох століть логіка майже не розвивалася. Справжній прогрес було досягнуто тільки у XIX ст., коли у логіці стали застосовувати математичні методи.

Ідею математизації логіки висловив ще у XVII ст. великий німецький вчений Лейбниць. Він сформулював задачу створення нової логіки, яка була б «мистецтвом обчислення». В цій логіці, за думкою Лейбниця, кожному поняттю відповідав би символ, а міркування мали б вигляд обчислень. Часи Лейбниця були епохою, коли аксіоматична геометрія стародавніх греків переживала новий розквіт. Математика зображувала науку, в якій умовивід відігравав більш важливу роль порівняно з іншими науками. Всі математичні теореми опиралися на точні доведення, засновані на висновку наслідків із загальноприйнятих математичних аксіом або постулатів. Тому не дивно, що вчені при аналізі математичних міркувань

відкрили величезну кількість схем (способів) побудови умовиводів. Лейбниць вирішив так сформулювати правила математичного доведення, щоб при їх застосуванні не приходилося більше думати про змістовний сенс математичних виразів. Для цього необхідно створити обчислення, в якому природні, змістовні доведення були б замінені формальними обчисленнями і тим самим стали б предметом математики. Таке обчислення, розуміється, припускає спеціальну символіку, в якій можуть бути зображені аксіоми, теореми та визначення математики.

Тільки у середині XIX ст. ірландський математик Дж. Буль частково втілив у життя ідею Лейбниця. В його роботах «Математичний аналіз», «Закони мислення», а також у роботах Де Моргана було закладено основи булевої алгебри і алгебри логіки, які розглядалися у розділі 4. За допомогою алгебри логіки можна описувати міркування та «обчислювати» їх результати, якщо символам змінних поставити у відповідність деякі твердження, що називаються висловленнями. Таке застосування алгебри логіки одержало назву ***логіки висловлень*** і згодом було поширено додаванням змінних, що приймають значення із множини понять, що сформувало ***логіку предикатів***.

У зв'язку з тим, що в математичній логіці прийнята символічна мова, вона також називається символічною ***логікою***. В роботах Пеано, Пірса та Шредера також створювалася й удосконалювалася математична символіка для законів мислення. Ці роботи вселяли віру у безмежні можливості формалізації. Зусиллями таких видатних вчених, як Рассел, Уайтхед, Гільберт, Бернайс, Гедель і Черч, формалізація у рамках сучасних логічних обчислень досягла високого рівня. При спробах реалізувати практично ідею формалізації виникли труднощі логічного характеру, які виявилося неможливо здолати засобами класичної формальної логіки. Ці труднощі остаточно не усунуті і по цей час, але спроби їх здолання дали могутній поштовх розвиткові нових розділів логіки — неокласичних, модальних, інтуїційних.

Широке застосування комп'ютерів зажадало відповідного математичного забезпечення: розробки спеціальних мов для баз даних і для уявлення знань, поглибленого логічного аналізу природних мов. Ці проблеми і прагнення змоделювати на комп'ютері великий клас інтелектуальних процедур

(ідеї штучного інтелекту) поставили перед логічною наукою нові задачі. Результати розв'язку зазначених задач знаходять нині все нові і різноманітні застосування у багатьох галузях науки і техніки. Математична логіка займається формалізацією деякої області людського мислення, у тому числі з ціллю надання можливості написання програми для обчислювальної машини, яка в цьому розумінні придбає здатність міркувати.

Запитання

1. Хто є засновником формальної логіки?
2. Яким чином відбувався розвиток логіки як науки?
3. Що зображує математична логіка?
4. Назвіть можливі галузі застосування математичної логіки.

Завдання

1. Які висновки будуть правомірними при такому посиланні: *«Якщо студент не знає логіку висловлень, то не зможе розв'язати задану логічну задачу»?*

   1.1. Студент розв'язав задану логічну задачу. Отже, студент знає логіку висловлень.

   1.2. Студент не знає логіку висловлень. Отже, він не розв'яже задану задачу.

   1.3. Студент знає логіку висловлень. Отже, він розв'яже задану задачу.

   1.4. Студент не розв'яже задану логічну задачу. Отже, він не знає логіку висловлень.

5.2. Поняття логіки висловлень

Висловлення, істиннісне значення, атом, логічні зв'язки, правильно побудована формула, інтерпретація висловлення, пріоритет і ранг операцій, тавтологія, тотожно хибна формула, незагальнозначуща формула

В природних мовах інформація передається за допомогою слів, об'єднаних у речення. Формальна логіка займається аналізом речень, звертаючи основну увагу на форму і відволікаючись від змісту.

Математична логіка вивчає та модулює тільки оповідальні речення. Наказові, окличні та питальні речення знаходяться поза сферою розгляду. Скорочені речення виключаються тому,

що можуть мати подвійне значення. Наприклад, на питання *«чи не спізнилися ви на лекцію?»* скорочена відповідь *«так»* незрозуміла. В цьому випадку у відповіді необхідно більш повне твердження, яке могло б бути формалізоване. Для формалізації підійде речення: *«Я спізнився на лекцію»*. Математична логіка не вивчає внутрішню структуру і зміст речень, а обмежується розглядянням їх істиннісних властивостей, тобто чи зображують вони істину або хибність. У зв'язку з цим із розглядання виключені оповідальні речення, для яких неможливо визначити, істинне воно або хибне. Очевидно, що таке зображення природної мови не є повним, але воно необхідне для застосування логіки висловлень.

Визначення

Висловлення — це оповідальне речення, про яке можна сказати, істинне воно або хибне, але не те й інше одночасно.

Визначення

Істина або хибність, приписана деякому висловленню, називається ***істиннісним значенням*** цього висловлення. Позначається: «Істина» — *I*, *T* (True) або 1, «Хибність» — *X*, *F* (False) або 0.

Приклад. Визначити, які з даних речень є висловленнями: *«Волга впадає в Чорне море»*, *«Волга впадає в Каспійське море»*, *«Який сьогодні день?»*, *«Відстань від Землі до Сонця дорівнює 150 000 000 км»*.

Розв'язок. Перші два речення є висловленнями, причому перше є хибним висловленням, а друге — істинним. Третє речення висловленням не є (за визначенням), оскільки воно не оповідальне. Четверте речення також не є висловленням, тому що його істинність або хибність залежить від потрібної точності.

Оповідальні речення бувають простими та складними. Складні речення, як правило, складаються з простих речень, поєднаних сполучниками. Кожне просте речення є самостійним твердженням, і воно вже не може бути розбите на більш дрібні речення. Ці прості речення та сполучники є елементами словника, необхідного для формалізації природної мови за допомогою логіки висловлень.

Визначення

Атомами (елементарними висловленнями) називаються висловлення, які відповідають простим оповідальним реченням, тобто не мають складових частин.

Як символи для позначення атомів використовуються великі букви латинського алфавіту A, B, C, ... або великі букви з індексами. Кожна буква у міркуванні повинна позначати одне і тільки одне елементарне висловлення.

Визначення

Логіка ***висловлень*** — це алгебраїчна структура $(\{X, I\}, \wedge, \vee, \bar{}, \to, \sim, X, I)$ з носієм — двійковою множиною $\{X$: «Хибність», I: «Істина»$\}$, операціями — ***логічними зв'язками*** ($\wedge$ — кон'юнкція, $\vee$ — диз'юнкція, $\bar{}$ — заперечення, $\to$ — імплікація, $\sim$ — еквівалентність) і константами: X — хибність і I — істина.

В розділі 4.2.3 глави 4 було визначено *алгебру логіки*, як алгебраїчну структуру $(B, \wedge, \vee, \bar{}, \to, \sim, 0, 1)$, що створена двійковою множиною $B = \{0, 1\}$ разом з операціями кон'юнкції, диз'юнкції, заперечення, імплікації, еквівалентності і константами 0 і 1. Оскільки операції алгебри логіки і логіки висловлень однакові, а між множинами-носіями даних алгебраїчних структур $\{0, 1\}$ і $\{X, I\}$ можна провести взаємно однозначну відповідність, приходимо до висновку, що вказані алгебраїчні структури ізоморфні (див. п. 3.2). Тому всі твердження, що зроблені у главі 4 відносно алгебри логіки, справедливі і для логіки висловлень, зокрема, комутативний, асоціативний та інші закони з п. 4.2.

Складні речення природної мови складаються з простих речень і службових слів («якщо», «і», «то», «або» і тощо). В граматиці зазначені службові слова називаються зв'язками (або сполучниками), оскільки вони об'єднують прості речення в одне складне. Наприклад, два речення «*Ми поїдемо влітку до Криму*» і «*Ми поїдемо влітку в гори*» можна об'єднати зв'язкою «або» в одне складне речення «*Ми поїдемо влітку до Криму, або ми поїдемо влітку в гори*». Тут зв'язку «або» не можна приєднати ані до першого, ані до другого простого речення, вона обслуговує одночасно обидва простих речення і тому називається ***бінарною***. Розглянемо зворот «*неправильно, що...*»,

який вживається з ціллю заперечення. Наприклад, у реченні *«Неправильно, що жителів у Києві менше, ніж у Харкові»* відбувається заперечення речення *«В Києві менше жителів, ніж у Харкові»*. Зв'язка «неправильно, що ...» є унарною, тому що застосовується до одного речення. Крім розглянутих, у природній мові існують зв'язки :«якщо..., то...», «чи», «і», «або», «ні... ні...», «...тоді і тільки тоді, коли...» й ін.

Операції логіки висловлень — логічні зв'язки — розглядаються як формальні позначення зв'язок, що їм відповідають, природної мови (таблиця 6.1).

Таблиця 5.1. Логічні зв'язки в логіці висловлень

Назва	Позначення	Аналоги природної мови
еквівалентність	~, ≡, ↔	еквівалентно, рівносильно, «тоді і тільки тоді», якщо і тільки якщо
імплікація	→, ⊃	тягне, «якщо ..., то», «тільки якщо»
кон'юнкція	∧, &	і
диз'юнкція	∨	або, «або одне з двох... або обидва»
заперечення	¬, ¯	не, «неправильно, що»

Операції ∧, ∨, →, ~ є бінарними логічними зв'язками, на відміну від операції ¬, яка є унарною. Користуючись введеними логічними зв'язками, можна з елементарних висловлень будувати складні висловлення, що називаються формулами або ***молекулами***.

Визначення

В логіці висловлень правильно ***побудована формула визначається*** рекурсивно таким чином:

1. Атом є формула.
2. Якщо A і B — формули, то $(A \wedge B)$, $(A \vee B)$, $(A \to B)$, $(A \sim B)$ і $\neg A$ — також формули.
3. Ніяких формул, крім породжених вказаними вище правилами, не існує.

Формули логіки висловлень, що відповідають складним висловленням, приймають значення I або X залежно від значень елементарних висловлень, з яких вони побудовані, і логічних зв'язок.

Визначення

Приписування істиннісних значень атомам, з яких побудоване висловлення, називається інтерпретацією ***висловлення***.

Для висловлення, що містить n атомів, можна скласти 2^n інтерпретацій, як і для n-місної булевої функції.

Поряд з висловленнями, істиннісне значення яких незалежне від ситуації, можна вважати однозначно визначеним, як, наприклад, «$2 \times 2 = 4$» $= I$, існують висловлення, які можуть приймати різні значення. Наприклад, висловленню «*Завтра буде дощ*» можна надавати значення і «Істина», і «Хибність» залежно від конкретної ситуації.

Формули логіки висловлень можна задавати таблицями істинності подібно до булевих функцій. Наведемо таблицю істинності для логічних зв'язок логіки висловлень (таблиця 5.2).

Таблиця 5.2. Таблиця істинності логічних зв'язок

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \to B$	$A \sim B$
X	X	I	X	X	I	I
X	I	I	X	I	I	X
I	X	X	X	I	X	X
I	I	X	I	I	I	I

Розглянемо приклади і вирази природної мови, які відповідають логічним зв'язкам.

Заперечення. ***Заперечення*** $\neg A$ істинне тоді і тільки тоді, коли A хибне. Ця унарна операція відповідає запереченню у звичайній мові, яке може мати різні синтаксичні вирази, наприклад, речення «*Неправильно, що* у Івана є час» рівнозначне реченню «У Івана *немає* часу».

Кон'юнкція. Висловлення $A \wedge B$, що називається ***кон'юнкцією*** A і B, істинне тоді і тільки тоді, коли істинні обидва висловлення A і B. Ця логічна операція відповідає у природній мові зв'язці «і», що з'єднує два речення.

Приклад. Записати у вигляді формули логіки висловлень і визначити істиннісне значення таких висловлень:

I — «6 *ділиться на* 3, *і* 10 *більше* 5»;

II — «6 *ділиться на* 3, *і* 7 *більше* 10».

Розв'язок. Виділимо атоми. Їх три:

A — «6 ділиться на 3», B — «10 більше 5»,

C — «7 більше 10».

Тоді висловлення I буде відповідати формулі $A \wedge B$, висловлення II — формулі $A \wedge C$. Будемо вважати, що висловлення A і B істинні, а висловлення C хибне. Використовуючи істиннісні значення висловлень A, B, C, визначимо значення висловлень I і II:

$$A \wedge B = I \wedge I = I; \quad A \wedge C = I \wedge X = X.$$

Диз'юнкція. Висловлення $A \vee B$, що називається ***диз'юнкцією*** A і B, хибне тоді і тільки тоді, коли хибні обидва висловлення A і B.

Ця логічна операція відповідає поєднанню висловлень природної мови за допомогою зв'язки «або», що вжита у розумінні «або, що не виключає»: «правильне A, або правильне B, або обидва висловлення правильні».

Приклад. Записати у вигляді формули логіки висловлень і визначити істиннісне значення таких висловлень:

I — «5 + 2 = 10 або 5 × 2 = 10»,

II — «6 − 3 = 2 або 3 × 2 = 5».

Розв'язок. Виділимо атоми:

A — «5 + 2 = 10»; C — «6 − 3 = 2»;

B — «5 × 2 = 10»; D — «3 × 2 = 5».

Тоді висловлення I буде відповідати формулі $A \vee B$, висловлення II — формулі $C \vee D$. Висловлення B істинне, а висловлення A, C і D хибні, тому:

$$A \vee B = X \vee I = I; \quad C \vee D = X \vee X = X.$$

Імплікація. Висловлення $A \to B$, що називається ***імплікацією*** (умовним реченням), хибне тоді і тільки тоді, коли A істинне, а B хибне.

В імплікації $A \to B$ висловлення A називається засновком (***умовою***, ***антецедентом***), B — наслідком (***висновком***, ***консеквентом***). Причинно-наслідковий зв'язок між A і B, що виражається імплікацією, на природній мові описується такими зворотами: «якщо A, то B»,«A є достатньою підставою для B», «B, тому що A», «B, за умови виконання A», «A тягне B» тощо.

Використовувані в точних науках поняття достатньої та необхідної умов можна формально виразити за допомогою імплікації. Саме для двох фактів (подій) A і B висловлення $A \to B$ означає, що «*A є достатньою умовою для B*» і одночасно, що «*B є необхідною умовою для A*». Необхідність B для A виражається також у формі «*B тільки, якщо A*». Твердження «*A є необхідною і достатньою умовою для B*» еквівалентне подвійній імплікації $A \leftrightarrow B$, або

$$(A \to B) \wedge (B \to A) \equiv A \leftrightarrow B.$$

Розглянемо кілька прикладів.

У висловленні «*Якщо число n — парне (A), то n ділиться на 4 (B)*» умова A буде необхідною, але недостатньою. Жодне непарне число на 4 не ділиться, і в той же час є парні числа, які не діляться на 4. Отже, правильна імплікація $B \to A$, вихідне висловлення $A \to B$ хибне.

У висловленні «*Якщо йде дощ (A), то на небі хмари (B)*» умова A буде достатньою, але не необхідною. Існують випадки, коли на небі є хмари, але дощу немає. Тут правильне вихідне висловлення $A \to B$.

У висловленні «*Якщо геометрична фігура — квадрат (A), то вона — рівнобічний прямокутник (B)*» умова A буде і необхідною і достатньою для виконання B: $A \leftrightarrow B$.

Приклад. Записати у вигляді формули логіки висловлень і побудувати таблицю істинності висловлення «*Якщо йде дощ, то над моєю головою відкрита парасолька*».

Розв'язок. Введемо атоми:

A — «йде дощ»;

B — «над моєю головою відкрита парасолька».

Тоді висловлення «Якщо йде дощ, то над моєю головою відкрита парасолька» буде відповідати формулі $A \to B$. Результати інтерпретації цього висловлення наведено у таблиці 5.3.

Таблиця 5.3. Таблиця істинності висловлення

A	B	$A \to B$	Результат
X	X	I	залишуся сухим
I	X	X	змокну
X	I	I	залишуся сухим
I	I	I	залишуся сухим

Другий рядок таблиці 5.3 вказує на відсутність причинно-наслідкового зв'язку між подіями A і B.

Еквівалентність (еквіваленція). Якщо A і B — висловлення, то висловлення $A \sim B$ істинне тоді і тільки тоді, коли A і B або обидва істинні, або обидва хибні.

Ця операція відповідає у природній мові зворотам: «...тоді і тільки тоді, коли...», «для того щоб..., необхідно і достатньо...». Наприклад, «*Вивчення дискретної математики буде успішним тоді і тільки тоді, коли буде освоєна математична логіка*».

Використовуючи таблицю істинності еквівалентності, можні довести, що вираз $A \sim B$ еквівалентний виразу $(A \to B) \wedge (B \to A)$. Таким чином, логічна еквівалентність зображує імплікацію в обох напрямках, тому вираз «*A істинне тоді і тільки тоді, коли B істинне*» означає, що «*A* тягне *B*, і *B* тягне *A*».

Заміняючи імплікацію її записом у вигляді ДКНФ ($A \to B = \neg A \vee B$), одержимо:

$$A \sim B = (A \to B) \wedge (B \to A) = (\neg A \vee B) \wedge (\neg B \vee A),$$

$$A \sim B = (\neg A \vee B) \wedge (\neg B \vee A). \qquad (5.1)$$

Формула (5.1) дає ДКНФ для еквівалентності. За принципом двоїстості (п. 4.3) ДДНФ для еквівалентності має вигляд:

$$A \sim B = A \wedge B \vee \neg A \wedge \neg B. \qquad (5.2)$$

Приклад. Записати у вигляді формули логіки висловлень і визначити істиннісне значення висловлень:

I — «*Для того щоб* $2 \times 2 = 4$, *необхідно і достатньо, щоб* $2 + 2 = 4$»;

II — «$2 \times 2 = 5$ *рівносильно тому, що* $3 \times 3 = 8$».

Розв'язок. Введемо позначення атомів:

A — $2 \times 2 = 4$; B — $3 \times 3 = 8$;

C — $2 + 2 = 4$; D — $2 \times 2 = 5$.

Висловлення I відповідає формулі $A \sim C$, висловлення II — формулі $D \sim B$. Будемо вважати, що атоми A і C істинні, а атоми B і D — хибні, і визначимо істиннісні значення складних висловлень:

$$A \sim C = I \sim I = I; \quad D \sim B = X \sim X = I.$$

Прочитання формул складних висловлень може бути неоднозначним, якщо не ввести дужки, що вказують, в якому порядку зв'язуються між собою символи. Деякі дужки можна

опустити, увівши послідовність виконання або пріоритет операцій таким же чином, як для операцій алгебри логіки:

$$\neg,\ \wedge,\ \vee,\ \rightarrow,\ \sim.$$

Наприклад, такі вирази без дужок дорівнюють формулам з дужками:

$$A \rightarrow B \wedge C = A \rightarrow (B \wedge C);$$

$$C \sim A \wedge B \rightarrow C = C \sim ((A \wedge B) \rightarrow C).$$

Будь-якій формулі логіки висловлень можна поставити у відповідність деяке складне висловлення природної мови і навпаки, «правильні» складні речення можна записати у вигляді формули логіки висловлень. Аналіз складного речення необхідно починати з визначення такого факту: чи є воно скороченим варіантом більш розповсюдженого складного речення? Скорочений варіант слід замінити повним варіантом речення. Далі виділити прості речення та взяти їх в дужки, залишаючи поза дужками службові слова, що поєднають прості речення. Процес взяття у дужки повторюється доти, доки цілком усе складне речення не виявиться взятим у дужки. Після цього сполучники та звороти природної мови замінюються відповідними логічними зв'язками, а прості речення — атомарними формулами.

Пример. Записати у вигляді формули логіки висловлень таке речення: «*Оскільки я ліг пізно спати, я проспав і через це не пішов на пару*».

Розв'язок. Виділимо прості речення у цьому складному реченні та візьмемо їх у дужки, залишаючи службові слова поза їх межами:

«(Оскільки (я ліг пізно спати), (я проспав)) і через це не (пішов на пару)».

Всі три речення зв'язані службовими словами, що виражають логічні відношення. Крім цього, перед третім простим реченням стоїть частка «*не*», що відповідає логічній операції «заперечення». Третє просте речення не є повним, оскільки розділяє спільний підмет «*я*» з другим простим реченням. Доповнимо третє речення відсутнім підметом і введемо атоми P, Q, S таким чином:

P — «Я ліг пізно спати»;

Q — «Я проспав»;

S — «Я пішов на пару».

Замінимо прості речення символами атомів, а службові слова — логічними зв'язками, одержимо формулу логіки висловлень:

$$(P \to Q) \to \neg S.$$

Приклад. Побудувати формулу і таблицю істинності для висловлень: «*Якщо студент не підготувався до іспиту або йому попався складний білет, то він не складе іспит на позитивну оцінку*». Визначити, в яких випадках це висловлення виявиться хибним.

Розв'язок. Виділимо прості висловлення і послідовність їх поєднання службовими словами за допомогою дужок: «Якщо ((студент не підготувався до іспиту) або (йому попався складний білет)), то (він не складе іспит на позитивну оцінку)». Позначимо атоми:

A — «Студент підготувався до іспиту»;

B — «Студенту попався складний білет»;

C — «Студент складе іспит на позитивну оцінку».

Одержана формула має вигляд:

$$(\neg A \vee B) \to \neg C.$$

Побудуємо відповідну таблицю істинності (таблиця 5.4).

Таблиця 5.4. Таблиця істинності $(\neg A \vee B) \to \neg C$

A	B	C	$\neg A$	$\neg A \vee B$	$\neg C$	$\neg A \vee B \to \neg C$
X	X	X	I	I	I	I
X	X	I	I	I	X	X
X	I	X	I	I	I	I
X	I	I	I	I	X	X
I	X	X	X	X	I	I
I	X	I	X	X	X	I
I	I	X	X	I	I	I
I	I	I	X	I	X	X

З таблиці ми бачимо, що існують три інтерпретації: (X, X, I), (X, I, I), (I, I, I), на яких вихідне твердження виявляється хибним. Інтерпретація (X, X, I), означає, що студент не підготувався до іспиту, але одержав нескладний білет, і йому вдалося скласти іспит на позитивну оцінку. У випадку (I, I, I) студенту попався важкий білет, але він підготовлений до цього іспиту і склав іспит на позитивну оцінку. В інтерпретації (X, I, I) студент не підготувався до іспиту, йому попався важкий білет, але він все одно склав іспит на позитивну оцінку.

Виходячи з прийнятих формул логіки висловлень істиннісних значень, формули розділяються на тотожно істинні, тотожно хибні та незагальнозначущі.

Визначення

Формула називається ***тотожно істинною*** (***тавтологією*** або ***загальнозначущою***), якщо вона приймає значення «Істина» на всіх інтерпретаціях (наборах значень змінних). Формула називається ***тотожно хибною*** (***суперечливою або нездійсненною***), якщо вона приймає значення «Хибність» на всіх інтерпретаціях. Формула називається ***незагальнозначущою***, ***нейтральною*** або ***несуперечливою***, якщо вона на одних інтерпретаціях приймає значення «Істина», а на інших — «Хибність». Усі формули, які не належать до суперечливих, утворюють множину *здійсненних* формул.

Визначення

Міркування називається ***правильним***, якщо воно виражається тотожно істинною формулою.

Таким чином, перевірити правильність міркування можна, побудувавши відповідну до нього формулу і визначивши, чи є вона тотожно істинною.

Довести, що формула є тавтологією, можна двома способами:

1. Побудувати таблицю істинності цієї формули. Тоді, якщо у таблиці істинності на всіх інтерпретаціях функція приймає значення «Істина», то відповідне до формули міркування є тавтологією.
2. Застосувавши до формули тотожні перетворення, звести її за допомогою тотожних перетворень до виду одного з логічних законів. Якщо в результаті перетворень одержимо значення «Істина», то формула — тавтологія.

Запитання

1. Який вид речень моделює формальна логіка?
2. Наведіть приклади речень, які не розглядаються у формальній логіці.
3. Дайте визначення поняттю «висловлення».
4. Що мають на увазі під істиннісним значенням висловлення?
5. Які висловлення називаються атомами?
6. Дайте визначення логіки висловлень.

7. Що у логіці висловлень називають логічними зв'язками, наведіть їх.
8. Покажіть, що алгебра логіки і логіка висловлень ізоморфні. Який з цього маємо висновок?
9. Дайте визначення правильно побудованої формули.
10. Наведіть приклади формул логіки висловлень, що містять будь-які логічні зв'язки, і відповідних до них речень природної мови.
11. Сформулюйте алгоритм запису складного речення природної мови у вигляді формули логіки висловлень.
12. Назвіть види функцій логіки висловлень з точки зору прийнятих ними істиннісних значень.

Завдання

1. Чи є такі формули загальнозначущими, суперечливими або несуперечливими:
   а) $\neg(\neg A) \to A$;
   б) $(A \to B) \to (B \to A)$;
   в) $(A \wedge (B \to A)) \to A$;
   г) $(A \vee \neg B) \vee (\neg A \vee B)$.
2. Розставити різними способами дужки у таких формулах:
   а) $\neg A \vee \neg B \wedge C$;
   б) $A \to B \to C \to D$.
3. Виключити якомога більше число дужок у формулі:
   а) $(\neg((A) \vee (C))) \vee (B)$;
   б) $(((A) \to (B)) \to (C)) \vee ((A) \to ((B) \to (C)))$;
   в) $((B) \sim (\neg(C))) \vee (((A) \to (A)) \to ((B) \vee (D)))$;
   г) $\neg((\neg((A) \vee (B))) \neg (C)) \to ((\neg((C) \to (D))) \vee E))$;
   д) $(\neg((B) \sim (C))) \wedge ((\neg(E)) \vee (\neg(A)))$;
   е) $\{(\neg((A) \to (B))) \vee (\neg\{(C) \vee (D))) \wedge \neg(F)\}$.
4. Побудуйте складні висловлення з використанням тільки зазначених операцій:
   а) еквівалентність;
   б) імплікація і кон'юнкція;
   в) заперечення, кон'юнкція і диз'юнкція.
5. Доведіть, що заперечення висловлення «A є достатня та необхідна умова для B» еквівалентне висловленню «$\neg A$ є достатня і необхідна умова для $\neg B$».
6. Побудуйте висловлення, еквівалентне $A \vee B$, використовуючи тільки операції заперечення і кон'юнкції.
7. Побудуйте складне висловлення, еквівалентне $A \vee B$, використовуючи тільки операції кон'юнкції і заперечення.
8. Побудуйте складне висловлення, еквівалентне $A \wedge B$, використовуючи тільки операції диз'юнкції і заперечення.
9. Побудуйте два складних висловлення, еквівалентних $A \to B$, використовуючи тільки:

а) операції диз'юнкції і заперечення;
б) заперечення і кон'юнкції.

10. Використовуючи тотожності, спростіть формули логіки висловлень:
а) $\neg (A \vee B \vee C)\ (A\ (B \vee \neg C)) \wedge \neg B$;
б) $(A \vee B) \wedge \neg C \vee A \vee \neg C \vee B \vee A$.

5.3. Дедуктивні висновки у логіці висловлень

Логічний наслідок та його властивості, аксіоми, доведення, правила дедуктивних висновків

Найважливішою характеристикою логічного висновку є відношення сумісності між його засновком та висновком. У логіці правила висновку використовуються, щоб виводити одні істинні речення з інших істинних речень.

Визначення

Висловлення B є ***логічним наслідком*** висловлення A, якщо формула $A \to B$ є *тотожно істинною*. Це визначення може бути узагальнено на випадок довільного числа засновків таким чином: висловлення B називається ***логічним наслідком висловлень*** $A_1\ A_2$, ..., A_n, ***якщо*** $A_1 \wedge A_2 \wedge \ldots \wedge A_n \to B$ — ***тотожно істинна формула.***

Приклад. Показати, що висловлення $(A \wedge B) \vee \neg C$ є логічним наслідком висловлення $A \wedge \neg C$.

Розв'язок. Достатньо впевнитися, що формула $(A \wedge \neg C) \to ((A \wedge B) \vee \neg C)$ є загальнозначущою. Використаємо тотожності логіки висловлень для еквівалентних перетворень, враховуючи, що $x \to y = f_{13}(x, y) = \bar{x} \vee y$ (див. таблицю 4.3 п. 4.2).

$$(A \wedge \neg C) \to ((A \wedge B) \vee \neg C) = \neg (A \wedge \neg C) \vee ((A \wedge B) \vee \neg C) =$$
$$= \neg A \vee C \vee (A \wedge B) \vee \neg C = \neg A \vee (A \wedge B) \vee \neg C \vee C =$$
$$= \neg A \vee (A \wedge B) \vee I = I.$$

Таким чином, доведено загальнозначущість формули.

Визначення

Дедуктивним висновком називається висновок формули B з формули A, заснований на тому, що B є логічним наслідком A.

Приклад. Довести правильність міркування за дедукцією: *«Резолюція ухвалюється, якщо і тільки якщо за неї голосує більшість депутатів. За резолюцію не проголосувала більшість депутатів, тому резолюція не ухвалюється».*

Розв'язок. Засновками у цьому висновку є висловлення «Резолюція приймається, якщо і тільки якщо за неї голосує більшість депутатів» і «За резолюцію не проголосувала більшість депутатів», а висновком — «Резолюція не приймається». Введемо такі атоми:

P — «за резолюцію проголосувала більшість депутатів»;

Q — «резолюція приймається».

Тоді засновки і висновок позначимо відповідно через $P \sim Q$, $\neg P$, $\neg Q$, приєднаємо за допомогою імплікації до кон'юнкції засновків $(P \sim Q) \wedge (\neg P)$ висновок $(\neg Q)$ і перевіримо, наприклад, за допомогою тотожних перетворень, чи є імплікація $((P \sim Q) \wedge (\neg P)) \to (\neg Q)$ логічним законом.

$((P \sim Q) \wedge (\neg P)) \to (\neg Q) = ((\neg P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P)) \to$ $\to (\neg Q) =$ — виразили еквівалентність через кон'юнкцію, диз'юнкцію та заперечення (див. (5.1)).

$((\neg P \vee Q) \wedge (\neg P) \wedge (P \vee \neg Q)) \to (\neg Q) =$ — застосували комутативний закон.

$((\neg P) \wedge (P \vee \neg Q)) \to (\neg Q) =$ — використовували закон поглинання.

$((\neg P \wedge P) \vee (\neg P \wedge \neg Q)) \to (\neg Q) =$ — застосували дистрибутивний закон.

$(\neg P \wedge \neg Q) \to (\neg Q) =$ — використовували закон суперечності.

$\neg(\neg P \wedge \neg Q) \vee (\neg Q) =$ — виразили імплікацію через заперечення та диз'юнкцію. Тепер розкриємо дужки і застосуємо закон виключеного третього:

$$P \vee Q \vee \neg Q = I.$$

Одержано істинне висловлення. Отже, висновок виходить з засновків, і задане міркування задовольняє визначення дедуктивного висновку. Істинність висновку в дедуктивному висновку гарантується істинністю засновків.

Твердження 1. Висловлення B є логічним наслідком висловлення A, якщо висловлення $A \to \neg B$ є тотожно хибним.

Твердження 2. Висловлення B є логічним наслідком висловлення A, якщо на всіх інтерпретаціях, на яких A істинне, B теж істинне.

Тотожна істинність або хибність засновку імплікації дозволяє зробити висновок про істинність або хибність наслідку.

Твердження 3. Якщо висловлення B є логічним наслідком висловлення A і висловлення A — тотожно істинне висловлення, висловлення B також є тотожно істинним.

Твердження 4. Якщо висловлення A є тотожно хибним, то для будь-якого висловлення B правильно, що $A \to B$.

При створенні математичної логіки переслідувалася ціль побудови формальної мови для математичних міркувань і доведень. У математиці і «чистій» логіці доводять теореми, тобто виводять наслідки з певних припущень. Припущення називаються ***аксіомами*** або ***гіпотезами***, при цьому передбачається, що вони тотожно істинні у всій розглянутій теорії. ***Доведення*** являє собою логічний висновок списку висловлень. Додавання висловлення у список доведення можливе, якщо дане висловлення є наслідком висловлень, внесених до цього списку раніше, або якщо воно є аксіомою чи гіпотезою. Теорема вважається доведеною, якщо твердження теореми записане у список доведення, тобто якщо встановлено, що твердження теореми є логічним наслідком введених аксіом.

Правила для дедуктивного висновку будуються на підставі загальнозначущих формул логіки висловлень виду $A \to B$. Ці правила часто записують як правила формального висновку у такому вигляді:

$$\frac{A_1, \ldots, A_n}{B}.$$

Тут A_1, ..., A_n — засновки висновку, а B — наслідок. Тавтологія, що відповідає такому правилу, — $A_1 \wedge A_2 \wedge \ldots \wedge A_n \to B$. Найбільш часто використовувані правила дедуктивного висновку наведено у таблиці 5.5.

Таблиця 5.5. Правила дедуктивних висновків логіки висловлень

Правило дедуктивного висновку	Тавтологія	Назва правила
1	2	3
$\frac{A}{A \vee B}$	$A \to (A \vee B)$	Правило введення диз'юнкції
$\frac{A,\ B}{A \wedge B}$	$((A) \wedge (B)) \to (A \wedge B)$	Правило введення кон'юнкції

Продовження таблиці 5.5

1	2	3
$\dfrac{A \vee B,\ \neg A}{B}$	$(A \vee B) \wedge \neg A \to B$	Правило видалення диз'юнкції (Диз'юнктивний силогізм)
$\dfrac{A \wedge B}{A}$	$(A \wedge B) \to A$	Правило видалення кон'юнкції
$\dfrac{A \to B}{\neg B \to \neg A}$	$(A \to B) \to (\neg B \to \neg A)$	Правило контрапозиції імплікації
$\dfrac{A \to B,\ A}{B}$	$(A \wedge (A \to B)) \to B$	Правило відділення (Modus Ponens)
$\dfrac{\neg B,\ A \to B}{\neg A}$	$(\neg B \wedge (A \to B)) \to \neg A$	Від'ємна форма правила відділення (Modus Tollens)
$\dfrac{A \to B,\ B \to R}{A \to R}$	$((A \to B) \wedge (B \to R)) \to \to (A \to R)$	Гіпотетичний силогізм

З усіх правил, наведених у таблиці 5.5, найбільш часто використовується правило відділення. ***Правило відділення*** має такий логічний сенс: якщо засновок правильний, то правильний і наслідок з нього. Наведемо приклади міркувань за допомогою правила відділення:

«Якщо студент не вивчив теорію, то він не виконає завдання. Студент не вивчив теорію. Отже, студент не виконає завдання».

«Якщо студент одержав п'ять, значить, він розв'язав задачу. Студент одержав п'ять. Отже, студент розв'язав задачу».

Приклад. Дано істинне висловлення *«Якщо n ділиться на* 9, *то n ділиться на* 3». Нехай також відомо, що *«n ділиться на* 9». Який висновок можна зробити, виходячи з цих двох висловлень?

Розв'язок. Введемо атомарні висловлення:

A — «n ділиться на 9»;

B — «n ділиться на 3».

Висловлення «Якщо n ділиться на 9, то n ділиться на 3» можна зобразити у вигляді формули $A \to B$. З одночасного

виконання засновків $A \to B$ і A можемо зробити висновок B за правилом відділення: «*n* ділиться на 3».

Приклад. Визначте тип правила дедуктивного висновку, яке було використане у такому міркуванні: «*Температура повітря* +1 °С, *і йде дощ. Отже температура повітря* +1 °С».

Розв'язок. Введемо атоми:

A — «Температура повітря +1 °С»;

B — «Йде дощ».

Висловлення «Температура повітря +1 °С, і йде дощ» можна зобразити у вигляді формули $A \wedge B$, а одержаний висновок «температура повітря +1 °С» є висловлення A. Очевидно, що висновок зроблено відповідно до правила видалення кон'юнкції (див. табл. 5.5).

Запитання

1. В чому полягає відмінність дедуктивних висновків від недедуктивних?
2. Дайте визначення логічного наслідку одного (кількох) висловлень.
3. Яким чином будується дедуктивний висновок?
4. Дайте стислу характеристику основних правил дедуктивного висновку.

Завдання

1. Якщо конгрес відмовляється прийняти нові закони, то страйк не буде закінчено, якщо він не триває більше року і президент фірми не йде у відставку. Чи закінчиться страйк, якщо конгрес відмовляється діяти і страйк тільки почався? Побудуйте логічний висновок і одержіть відповідь.
2. Доведіть такі твердження: «З тотожно хибної формули логічно виходить будь-яка формула».
3. Доведіть твердження логіки висловлень: «Тотожно істинна формула логічно виходить з будь-якої формули».

5.4. Обчислення висловлень

Мова, аксіоми і правила висновку, повнота та несуперечність, правила відділення і підстановки, теорема дедукції та її наслідок, доведення методом від супротивного

Для логічного аналізу необхідно створити сукупність правил визначення істинності або хибності висловлень.

Довести те, що деяка формула логіки висловлень є тавтологією, можна, використовуючи таблицю істинності, еквівалентні перетворення формули, а також за допомогою дедуктивного

висновку. На базі логіки висловлень створено ***формальну систему — обчислення висловлень***, яка дозволяє за допомогою правил дедуктивного висновку перевірити, чи є задана формула загальнозначущою, а також одержати загальнозначущі формули логіки висловлень.

Насправді, існують різні формалізації логіки висловлень, тобто ***різні обчислення висловлень*** або за іншою термінологією ***різні формальні системи***. Термін ***формальний*** означає, що об'єкти розглядаються без інтерпретації їх змісту, значення або сенсу. Операції над об'єктами проводяться за строгими формальними правилами, не розглядається значення сенсу проведених операцій.

Обчислення висловлень містить мову, систему аксіом і правила висновку.

Мова обчислення висловлень складається з правильно побудованих формул логіки висловлень.

Аксіомами обчислення висловлень є деяка множина загальнозначущих формул логіки висловлень.

Правила висновку дозволяють одержувати нові формули, які є істинними за умови істинності всіх засновків, що входять до правила.

Теорема 1

Теореми обчислення висловлень є тотожно істинними формулами.

□ *Доведення.* Теореми — це формули, які є логічним наслідком множини аксіом даного обчислення. Аксіоми обчислення висловлень є тотожно істинними формулами, а логічні наслідки тотожно істинних формул також є тотожно істинними (твердження 3, п. 5.3). Таким чином, теореми обчислення висловлень є тотожно істинними формулами, що і треба було довести. ■

Системи аксіом обчислення висловлень підбираються таким чином, щоб обчислення мало властивість *повноти*.

Повнота обчислення висловлень полягає в тому, що в даній системі є достатня кількість аксіом для того, щоб вивести будь-яку формулу логіки висловлень, яка є тотожно істинною.

Крім того, обчислення висловлень має властивість несуперечності.

Теорема 2. *Несуперечність обчислення висловлень*

Не існує формули A такої, що формули A і $\neg A$ є теоремами цього обчислення.

□ *Доведення.* Скористаємося методом від супротивного. Нехай правильно, що A і $\neg A$ одночасно є теореми цього обчислення. За законом суперечності хоча б одна з них не є загальнозначущою. З іншого боку, за теоремою 1 всі формули обчислення висловлень є загальнозначущими. Одержана суперечність доводить теорему. ■

Якщо жодну з аксіом системи обчислення висловлень не можна вивести з решти, застосовуючи правила висновку даної системи, то говорять, що система аксіом ***незалежна***. Аксіоми обчислення висловлень підбираються таким чином, щоб вони були незалежні.

Крім правила відділення (Modus Ponens, п. 5.3), у обчисленні висловлень часто використовується так зване правило підстановки.

Правило підстановки

Нехай F_1 і F_2 — формули логіки висловлень, A — атомарна формула. Якщо $F_1(A)$ — формула, виведена в обчисленні висловлень, що містить атом A, то $F_1(B)$ — виведена формула, одержана заміною всіх входжень A у формулі F_1 на формулу F_2.

Правило підстановки записується так:

$$\frac{F_1(A \| F_2)}{F_1(B)}.$$

Правило підстановки виражає той факт, що якщо у тотожно істинній формулі всі входження будь-якого атома замінити на деяку формулу, то одержаний вираз залишиться тотожно істинним.

Приклад. Використовуючи правило підстановки і комутативний закон для диз'юнкції, довести загальнозначущість такої формули:

$$A \vee B \wedge C \sim B \wedge C \vee A.$$

Розв'язок. Запишемо тотожність, що відповідає комутативному закону для диз'юнкції:

$$A \vee D \sim D \vee A.$$

Визначимо підстановку — атомарну формулу D замінимо на $(B \wedge C)$:

$$A \vee (B \wedge C) \sim (B \wedge C) \vee A.$$

Опустивши дужки згідно з пріоритетом операцій, переконуємося в істинності вихідної формули.

Часто правило підстановки не згадують, а використовують його як очевидний факт, правильний для тотожностей. У цьому випадку аксіоми обчислення висловлень називають ***схемами аксіом***, підкреслюючи те, що кожний атомарний символ у них може бути замінений на деяку формулу.

Опишемо дві формальні системи обчислення висловлень.

Система S_1

I. Мова складається з правильно побудованих формул логіки висловлень, що містять операції $\{\neg, \rightarrow\}$. Алфавіт мови співпадає з алфавітом логіки висловлень і містить, крім символів перелічених логічних операцій, символи дужок і символи для позначення висловлень.

II. Аксіоми:

1. $A \rightarrow (B \rightarrow A)$;
2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$;
3. $(\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B)$.

III. Правила висновку:

1. Правило відділення (Modus Ponens);
2. Правило підстановки.

Система S_2

I. Мова складається з правильно побудованих формул логіки висловлень, що містять операції $\{\wedge, \vee, \neg, \rightarrow\}$. Алфавіт мови співпадає з алфавітом логіки висловлень і містить, крім символів перелічених логічних операцій, символи дужок і символи для позначення висловлень.

II. Аксіоми:

1. $A \rightarrow (B \rightarrow A)$;
2. $(A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$;
3. $(A \wedge B) \rightarrow A$;
4. $(A \wedge B) \rightarrow B$;
5. $A \rightarrow (B \rightarrow (A \wedge B))$;
6. $A \rightarrow (A \vee B)$;
7. $B \rightarrow (A \vee B)$;

8. $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$;
9. $(A \to B) \to ((A \to \neg B) \to \neg A)$;
10. $\neg\neg A \to A$.

III. Правила висновку:

1. Правило відділення (Modus Ponens);
2. Правило підстановки.

Приклад. Довести вивідність формули $A \to A$ в системі S_1. Процедуру доведення проведемо покроково:

1. $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$ — підстановка в аксіому 2 $(A \to A)$ замість B, A замість C.

2. $A \to ((A \to A) \to A)$ — підстановка в аксіому 1 $(A \to A)$ замість B.

3. $(A \to (A \to A)) \to (A \to A)$ — за правилом 1, кроки 1, 2.

4. $A \to (A \to A)$ — підстановка в аксіому 1 A замість B.

5. $A \to A$ — за правилом 1, кроки 3, 4.

Доведення проведено.

Часто у математичних міркуваннях істинність твердження B доводять за припущенням істинності твердження A, після чого приходять до висновку, що правильне твердження «якщо A, то B». Такий прийом доведення є правильним і сформульований у такій теоремі. Спочатку введемо символ $\vdash$ тавтології.

Теорема 3. *Теорема дедукції*

Якщо $A_1, \ldots, A_n, C \vdash B$, *то* $A_1, \ldots, A_n \vdash C \to B$. *Зокрема, якщо* $A \vdash B$, то $\vdash A \to B$.

Теорема 4. *Наслідок з теореми дедукції*

Із засновків $A \to B$, $B \to C$ *виводимо* $A \to C$:

$$A \to B, B \to C \vdash A \to C.$$

Приклад. Довести, що формула $(\neg A \to \neg B) \to ((\neg A \to B) \to A)$ виведена в системі S_2.

1. $(\neg A \to B) \to (\neg A \to \neg B) \to \neg\neg A)$ — підстановка в аксіому 9 $\neg A$ замість A.

2. $(\neg A \to B), (\neg A \to \neg B) \vdash \neg\neg A$ — теорема 3, крок 1.

3. $\neg\neg A \vdash A$ — аксіома 10.

4. $(\neg A \to B), (\neg A \to \neg B) \vdash A$ — теорема 4, крок 2, 3.

5. $(\neg A \to \neg B) \vdash (\neg A \to B) \to A$ — теорема 3, крок 4.

6. $\vdash (\neg A \to \neg B) \to ((\neg A \to B) \to A)$ — теорема 3, крок 5.

Доведення проведено.

Всі теореми обчислення висловлень зв'язані з розв'язком такої задачі: «чи випливає це твердження з деякої сукупності інших тверджень?» Тобто, доводиться тотожна істинність формули виду $A \to B$, що можна здійснити кількома методами. Замість прямого логічного висновку формули B з формули A часто виявляється зручним довести суперечність формули $A \to \neg B$, тим самим довівши істинність формули $A \to B$. В формулі $A \to \neg B$ присутнє заперечення наслідку $\neg B$, тому такий метод доведення називається ***доведенням від супротивного***.

Дві схеми доведення методом від супротивного

1. $(A \wedge \neg B) \to (C \wedge \neg C) \equiv A \to B$ — якщо з припущення, що A — правильно, а B — неправильно, виходять два суперечних один одному висловлень, то це означає, що з A виходить B (див. твердження 1, п. 5.3).

2. $\neg B \to \neg A \equiv A \to B$ — якщо з припущення, що B — неправильно, виходить, що A неправильно, то це означає, що з A виходить B. Таким чином, довівши істинність лівої частини однієї з наведених схем, доводять істинність висловлення $A \to B$.

Запитання

1. Що являє обчислення висловлень?
2. Поясніть поняття мови, аксіом і правил висновку обчислення висловлень.
3. Що є теоремами обчислення висловлень?
4. В чому полягає повнота і несуперечність обчислення висловлень?
5. Дайте визначення незалежній системі аксіом.
6. Назвіть правила висновку, які найбільш часто застосовуються під час побудови обчислення висловлень?
7. Сформулюйте теорему дедукції та її наслідок.
8. В чому полягає метод доведення від супротивного?
9. Дайте порівняльну характеристику двом схемам доведення від супротивного.

Завдання

1. Нехай A — «дверний замок зламаний», B — «вхідні двері відкриті». Випишіть відповідний логічний висновок за правилом Modus Ponens.

2. Перевірте правильність таких висновків:

а) $\dfrac{A \to B,\ \neg A \to B}{B}$; в) $\dfrac{A \to B,\ C \to \neg B}{A \to \neg C}$;

б) $\dfrac{A \to B,\ \neg B \to \neg C}{C \to A}$; г) $\dfrac{A \to B,\ \neg A \to C}{B \to C}$.

5.5. Логіка предикатів

Порядок предиката, область визначення предиката, терм, предметні змінні та константи

Нагадаємо, що формалізація природної мови у логіці висловлень здійснюється розбиванням «мовних повідомлень» на неподільні оповідальні речення (атоми) та їх змістове об'єднання за допомогою зв'язок, причому внутрішня структура атомів не враховується. Однак у природній мові існує велика кількість умовиводів, які не можуть бути формалізовані описаним способом. Розглянемо класичний умовивід:

«Кожна людина смертна.

Оскільки Сократ людина, то він смертний».

Інтуїтивно зазначений логічний висновок представляється коректним. Введемо такі атоми:

A — «кожна людина смертна»;

B — «Сократ — людина»;

C — «Сократ смертний».

Тоді вихідний умовивід буде відповідати формулі логіки висловлень $A \wedge B \to C$.

Зведемо дану формулу до нормальної форми:

$$A \wedge B \to C = \neg(A \wedge B) \vee C = \neg A \vee \neg B \vee C.$$

На інтерпретації (1,1,0) одержана формула дорівнює нулю, отже, ця формула не є загальнозначущою, тобто у рамках логіки висловлень C не є логічним наслідком A і B. Така обмеженість можливостей нашої формалізації зв'язана з тим, що в атомі A не враховується внутрішня змістова особливість узагальнення «кожний». Це викликає необхідність удосконалити логіку висловлень з тим, щоб вона повніше пояснювала здібності людини робити логічні висновки. Для цього до ***логіки предикатів*** введено додаткові, нові порівняно з логікою висловлень логічні поняття, а саме: ***терм***, ***предикат*** і ***квантор***.

Визначення

Визначено деякий ***предикат***, якщо:
а) задана деяка (довільна) множина M, що називається областю визначення предиката (предметна область);
б) фіксована множина $\{1, 0\}$, що називається областю значень;
в) вказане правило, за допомогою якого кожному елементу, що взятий з предметної області, ставиться у відповідність один з двох елементів з області значень.

Поняття предиката є частковим випадком поняття функції, для якої чітко фіксована область значень.

Назва предикат походить від англійського слова predicate, що означає висловлення або присудок. Предикатом частіше всього позначають властивість або дію, виражену у висловленні присудком, а об'єкти і суб'єкти цієї дії, а також інші члени речення є аргументами даного предиката. Для позначення предиката часто обирають слово, що відбиває його змістове значення, або заголовну букву латинського або іншого алфавіту.

Визначення

Предикат P, що має n аргументів, називається ***n-місним предикатом***, позначається $P(x_1, x_2, ..., x_n)$.

Визначення

Кількість аргументів предиката $P(x_1, x_2, ..., x_n)$ називається його ***порядком***.

Так, наприклад, висловлення «x — дійсне число» можна зобразити одномісним предикатом, «y менше z» — двомісним предикатом, а «x і y батьки z» — трьохмісним предикатом. Якщо x, y і z заміщені конкретними значеннями (об'єктами), то предикат переходить у висловлення, яке розглядається як *нульмісний предикат*. Наприклад: «Терм і квантор — поняття логіки предикатів». Таким чином, якщо кількість аргументів предиката P $(x_1, x_2, ..., x_n)$ n змінних дорівнює нулю, то предикат є висловленням; якщо $n = 1$, то предикат відповідає властивості; якщо $n = 2$, то предикат є бінарним відношенням; якщо $n = 3$, то предикат — тернарне відношення.

Приклад. Зобразити у вигляді предикатів висловлення: «x ділиться на 13», «x ділиться на y», «x — просте число».

Розв'язок. Оберемо як назву предикатів дії або властивості цих речень: ДІЛИТЬСЯ, ПРОСТЕ. Тоді задані висловлення можна записати у вигляді предикатів таким чином: ДІЛИТЬСЯ(x, 13), ДІЛИТЬСЯ(x, y), ПРОСТЕ(x). Тут перший і третій предикати є одномісними і кожний виражає деяку властивість числа x; другий предикат — двомісний і виражає бінарне відношення подільності на множині чисел.

У логіці предикатів існує поняття ***функціонального символу***. Наприклад: мінус (x, y) — функціональний символ «$x - y$»; батько(x) — функціональний символ «батько людини x». Якщо функціональний символ має n аргументів, то він називається ***n-місним функціональним символом***, наприклад: мінус(x, y) — двохмісний функціональний символ. Індивідуальний символ або константа може розглядатися як функціональний символ без аргументів.

Отже, для побудови атомів логіки предикатів дозволяється використовувати такі типи символів:

1. ***Індивідуальні символи*** або ***константи***, які звичайно є іменами об'єктів, наприклад: Сократ, 13.
2. ***Символи предметних змінних***, за які звичайно беруться букви латинського алфавіту, можливо, з індексами, наприклад: x, y, z.
3. ***Функціональні символи*** — рядкові букви латинського алфавіту або осмислені слова з рядкових букв, наприклад: мінус, батько.
4. ***Предикати*** — великі букви або осмислені слова з великих букв, наприклад: P, Q, ДІЛИТЬСЯ, БІЛЬШЕ, ПРОСТЕ.

Визначення

Аргументи предиката називаються ***термами***. ***Терм*** визначається рекурсивно таким чином:

1. Константа є терм.
2. Змінна є терм.
3. Якщо f є n-місним функціональним символом, а t_1, t_2,..., t_n — терми, то $f(t_1, t_2, ..., t_n)$ є терм.
4. Ніяких термів, крім породжених за допомогою вказаних вище правил, не існує.

Визначення

Терми приймають значення із заздалегідь визначеної множина, яка називається ***предметною областю*** M.

Визначення

Терми-константи і терми-змінні називаються ***предметними константами*** і ***предметними змінними***.

Приклад. Зобразити у вигляді предикатів такі речення:

1) «Студенти складають сесію».

2) «Число $x + 1$ більше числа x».

3) «Брат Марини».

Розв'язок. 1) Речення «Студенти складають сесію» може приймати значення «Істина» або «Хибність», тому його можна зобразити у вигляді предиката. У внутрішній структурі цього речення можна виділити присудок «складають», підмет «студенти» і доповнення «сесію». Останні можна розглядати як предметні константи. Таким чином, одержуємо нульмісний предикат СКЛАДАТИ(студенти, сесію).

2) Присудком у цьому реченні є слово «більше». Зобразимо підмет «$x + 1$» і доповнення «x» у вигляді термів. Причому терм «$x + 1$» має внутрішню структуру, оскільки його можна зобразити за допомогою функціонального символу плюс(x, 1). Тоді вихідне речення прийме вигляд двомісного предиката: БІЛЬШЕ(плюс(x, 1), x). Тут x — предметна змінна, а 1 — константа.

3) Речення «брат Марини» не можна зобразити у вигляді предиката, оскільки його значенням є не «Істина» або «Хибність», а деякий елемент предметної області, яка відповідає множині людей.

Приклад. Перекласти на природну мову такі висловлення логіки предикатів:

1) *ДОРІВНЮВАТИ(x, 5).*

2) *ЗНАТИ(папа (Вася), математика).*

Розв'язок. 1) Предикат ДОРІВНЮВАТИ(x, 5) відповідає твердженню «x дорівнює 5» природної мови. Тут 5 — константа, x — предметна змінна.

2) У висловленні ЗНАТИ(папа(Вася), математика) функціональний символ «папа(x)» приймає значення з множини людей, що відповідає відношенню «бути батьком x». Тому вираз папа(Вася) слід інтерпретувати як «Васін папа». Таким чином, предикат ЗНАТИ(папа(Вася), математика) відповідає реченню «папа у Васі знає математику» природної мови. Тут «Вася» і «математика» є константами, а x — предметна змінна.

Запитання

1. Дайте визначення поняттю предикат.
2. Що називається порядком предиката?
3. Наведіть приклади n-місних предикатів.
4. Назвіть способи визначення предикатів.
5. Наведіть приклади функціональних символів.
6. Який функціональний символ називають n-місним?
7. Дайте визначення терма.
8. Що розуміють під предметною областю?
9. Дайте визначення понять предметна змінна і предметна константа. Наведіть приклади.

Завдання

1. З наведених нижче речень випишіть окремо висловлення, окремо — предикати:
   а) $x > 0$; в) про нього щось говорять;
   б) $2 + 3 = 6$; г) x брат y;
   д) протилежні боки A і B паралелограма рівні;
   е) кожне явище x має свою причину y.
2. В наведених одномісних предикатах зробіть можливі підстановки змінної x так, щоб одержати істинні висловлення. Які з них припускають одну, а які — багато підстановок?
   а) x — найвища горна вершина у світі;
   б) x — представник діалектичної логіки;
   в) $x + 7 = 15$; г) x — логічна зв'язка;
   д) x — видатний античний логік.
3. Змінні функції «$x > y$» приймають значення на множині $\{1, 2, 3\}$; B_1, B_2 — предикати, що задаються цією функцією відповідно при алфавітному і зворотному йому порядках. Встановіть:
   а) область визначення предикатів B_1 і B_2;
   б) значення істинності $B_1(2, 3)$ і $B_2(2, 3)$.

4. Скільки різних предикатів визначає висловлення «$x + y = z$», якщо M_x, M_y і M_z — множини значень змінних x, y, z:
   а) $M_x = M_y = M_z = \{1, 2\}$;
   б) $M_x = \{1\}$, $M_y = \{1, 2\}$, $M_z = \{2, 3\}$?
5. Визначте, чи еквівалентні такі предикати:
   а) $x^2 = 1$ і $x = 1$; б) $x^2 = x$ і $x = 1$.

5.6. Квантори

Квантор загальності, квантор існування, зв'язана та вільна змінна, зменшення порядку n-місних предикатів

При визначенні істиннісного значення предиката неабиякий інтерес становить питання: чи є він істинним при будь-якому значенні предметної змінної або чи існує хоча б одне значення змінної, при якому цей предикат істинний. Наприклад, твердження «*Всі прості числа мають два дільника*» можна формалізувати за допомогою предиката МАТИ_ДВА_ДІЛЬНИКА(x), який є істинним для всіх x у предметній області простих чисел. Твердження «*Існують натуральні числа, які не діляться на 2*» означає, що предикат ДІЛИТЬСЯ_НА_2(x) істинний не для всіх x у предметній області натуральних чисел.

Визначення

Нехай $P(x)$ — предикат, визначений на M. Висловлення «для всіх $x \in M$, $P(x)$ істинне» позначається $\forall x\ P(x)$. Знак $\forall$ називається ***квантором загальності***.

Квантори керують областю значення змінної, наступної за символом квантора. Якщо застосовується квантор загальності, то ми говоримо, що висловлення істинне для всіх x з деякої множини.

Визначення

Висловлення «існує таке $x \in M$, що $P(x)$ істинне» позначається $\exists x\ P(x)$, де знак $\exists$ називається ***квантором існування***.

Квантор існування застосовується, коли треба вказати, що існує хоча б одне значення змінної, для якого істинне це висловлення.

В логіці предикатів (або першого порядку) існує таке обмеження: не можна застосовувати квантори до предикатів.

Наприклад, не можна записати $\forall P\ P(x)$. Однак такі операції здійсненні у логіках більш високих порядків.

Визначення

Перехід від $P(x)$ до $\forall x\ P(x)$ або $\exists x\ P(x)$ називається ***зв'язуванням*** змінної x, а сама змінна x у цьому випадку — ***зв'язаною.***

Визначення

Змінна, не зв'язана ніяким квантором, називається ***вільною.***

Від того, чи є змінна зв'язаною або вільною, залежить значення предиката. Вільна змінна — це предметна змінна, яка може приймати різні значення з множини M, і значення предиката $P(x)$ залежить від значення змінної x. Навпаки, вираз $\forall x\ P(x)$ не залежить від змінної x і при заданих P і M має визначене значення; тут x — зв'язана змінна.

Зв'язані змінні зустрічаються не тільки у логіці. Наприклад, у виразах $\sum_{x=1}^{10} f(x)$ або $\int_a^b f(x)dx$ змінна x зв'язана і дані вирази при фіксованих a, b і f мають визначені значення, не залежні від будь-якого значення x.

Приклад. Записати у вигляді предикатів з кванторами такі висловлення: «*Всі студенти складають іспити*», «*Деякі студенти складають іспити на відмінно*».

Розв'язок. Введемо предикати: P — «складати іспити» і Q — «складати іспити на відмінно». Предметна область даних предикатів являє множину студентів. Тоді вихідні вирази набудуть вигляду:

$$\forall(x)\ P(x) \quad \text{і} \quad \exists(x)\ Q(x).$$

Приклад. Розглядаючи як предметну область множину дійсних чисел, записати у вигляді виразу логіки предикатів математичні твердження:

«*Для всіх x правильно, що* $(x-1)^2 = x^2 - 2x + 1$»;

«*Існує число, квадрат якого дорівнює* 4».

Розв'язок. Введемо предикат ДОРІВНЮЄ(x, y), який істинний у тому випадку, якщо значення змінної x дорівнює значенню y. Тоді, використовуючи квантори, можна записати:

$\forall x$ ДОРІВНЮЄ$((x-1)^2, (x^2-2x+1))$;

$\exists x$ ДОРІВНЮЄ$(x^2, 4)$.

Застосування кванторів до багатомісних предикатів зменшує кількість вільних змінних, від яких залежить цей предикат. Нехай $A(x, y)$ — деякий двомісний предикат, визначений на довільній множині M. Квантор загальності і квантор існування можна застосувати до неї як для змінної x, так і для змінної y:

$$\forall x\, A(x, y); \quad \forall y\, A(x, y); \quad \exists x\, A(x, y); \quad \exists y\, A(x, y).$$

Всі чотири наведені вирази є записами одномісних предикатів від відповідної вільної змінної. Так, $\forall x\, A(x, y)$ — одномісний предикат від змінної y: $\forall x\, A(x, y) = F(y)$. Предикат F істинний точно для таких елементів $b \in M$, для яких предикат $A(x, b)$ істинний на всіх значеннях аргументу x. Якщо зобразити множину значень істинності предиката $A(x, y)$ у вигляді матриці, то предикат $F(y) = \forall x\, A(x, y)$ істинний для таких $y = b$, для яких стовпчик аргументу $y = b$ містить виключно букву I. Для ілюстрації нижче наведено матрицю предиката (таблиця 5.6), що визначений на множині M з п'яти елементів a_1, a_2, a_3, a_4, a_5, і матриці одномісних предикатів (таблиці 5.7–5.10), що одержані з вихідного за допомогою застосування кванторів.

Подібним чином $\forall y\, A(x, y) = C(x)$ — одномісний предикат від x, що істинний для таких $a \in M$, для яких рядок аргументу $x = a$ містить тільки букву I. Предикат $\exists x\, A(x, y) = H(y)$ істинний для таких $a \in M$, для яких відповідний стовпчик $y = b$ містить, щонайменше, один раз букву I. Нарешті, предикат $\exists y\, A(x, y)$ істинний для таких $x = a \in M$, для яких у відповідному рядку $x = a$ зустрічається буква I.

Таблиця 5.6. Предикат $A(x, y)$

X \ Y	a_1	a_2	a_3	a_4	a_5
a_1	I	I	X	I	X
a_2	X	I	X	I	X
a_3	I	I	X	I	I
a_4	X	I	X	I	I
a_5	I	I	X	I	I

Таблиця 5.7. Предикат $\forall x\, A(x, y)$

y	$\forall x\, A(x, y)$
a_1	X
a_2	I
a_3	X
a_4	I
a_5	X

Таблиця 5.8.
Предикат $\exists x\, A(x, y)$

y	$\exists x\, A(x, y)$
a_1	I
a_2	I
a_3	X
a_4	I
a_5	I

Таблиця 5.9.
Предикат $\forall y\, A(x, y)$

x	$\forall y\, A(x, y)$
a_1	X
a_2	X
a_3	X
a_4	X
a_5	X

Таблиця 5.10.
Предикат $\exists y\, A(x, y)$

x	$\exists y\, A(x, y)$
a_1	I
a_2	I
a_3	I
a_4	I
a_5	I

Таким чином, застосування квантора за однією із змінних двомісного предиката перетворює його на одномісний. У випадку трьохмісних предикатів застосування квантора призводить до двомісного предикату. Аналогічно, для n-місних предикатів застосування квантора за будь-якою змінною перетворює предикат на $(n - 1)$-місний, тобто зменшує його порядок на одиницю.

Квантор загальності можна інтерпретувати як узагальнення кон'юнкції, а квантор існування — як узагальнення диз'юнкції. Насправді, якщо область визначення M предиката P скінченна, наприклад, $M = \{a_1, a_2, \ldots, a_n\}$, то висловлення $\forall x\, P(x)$ еквівалентне кон'юнкції $P(a_1) \wedge P(a_2) \wedge \ldots \wedge P(a_n)$, а висловлення $\exists x\, P(x)$ — диз'юнкції $P(a_1) \vee P(a_2) \vee \ldots \vee P(a_n)$.

Як приклад розглянемо предикат $P(x)$, який означає «x — непарне число» і визначений на області $M = \{a, b, c\}$. Висловлення $\forall x\, P(x)$ означає: «a — непарне число, і b — непарне число, і c — непарне число»; а висловлення $\exists x\, P(x)$ означає те ж, що і диз'юнкція «a — непарне число, або b — непарне число, або c — непарне число».

Запитання

1. Що розуміють під квантором загальності?
2. Дайте визначення поняттю квантор існування.
3. Які змінні називаються зв'язаними, а які — вільними?
4. Поясніть на прикладах призначення зв'язаних змінних.
5. До яких наслідків призводить застосування квантора за однією із змінних n-місного предиката?
6. Скількома різними способами може бути застосований квантор існування до n-місного предикату?
7. Коли предикат $\forall x\, A(x, y)$ приймає значення «Істина»?

Завдання

1. Запишіть такі висловлення, використовуючи знаки кванторів:
   а) існує число x таке, що $x + 1 = 5$;
   б) яким би не було число y, $y + 0 = y$;
   в) будь-яке число або додатне, або від'ємне, або дорівнює нулю.

2. Вкажіть вільні та зв'язані входження кожної із змінних у таких формулах:
   а) $\forall x\ P(x, y) \wedge \forall y\ Q(y)$;
   б) $\forall x\ (P(x) \to P(y))$;
   в) $\forall x\ (P(x) \to Q(y)) \vee \exists y\ R(x, y)$;
   г) $\forall x\ [(P(x) \to Q(y)) \vee \exists y\ R(x, y)]$.

3. Нехай x і y — будь-які люди, $Q(x, y)$ означає «x батько y». Наведені висловлення сформулюйте природною мовою, визначивши їх значення істинності:
   а) $\forall x\ \exists y\ Q(x, y)$; в) $\forall y\ \exists x\ Q(x, y)$; д) $\forall x\ \forall y\ Q(x, y)$;
   б) $\exists x\ \forall y\ Q(x, y)$; г) $\exists y\ \forall x\ Q(x, y)$; е) $\exists x\ \exists y\ Q(x, y)$.

4. Нехай $N(x)$ — «x — натуральне число», $C(x)$ — «x — ціле число», $P(x)$ — «x — просте число», $E(x)$ — «x — парне число», $O(x)$ — «x — непарне число», $D(x, y)$ — «y ділиться на x». Сформулюйте природною мовою наведені висловлення, встановивши їх значення істинності:
   а) $P(z)$;
   б) $E(2) \wedge P(2)$;
   в) $\forall x\ (D(2, x) \to E(x))$;
   г) $\exists x\ (E(x) \wedge D(x, 6))$;
   д) $\forall x\ [P(x) \to \exists y\ (E(y) \wedge D(x, y))]$;
   е) $\forall x\ (N(x) \to C(x))$;
   ж) $\exists x\ (N(x) \to C(x))$;
   з) $\forall x\ (C(x) \to N(x))$;
   і) $\forall x\ \forall y\ [O(x) \to (P(y) \to D(x, y))]$;
   к) $\forall x\ [C(x) \to (E(x) \vee \neg E(x))]$;
   л) $\exists x\ \forall y\ [(C(x) \wedge C(y)) \to D(x, y)]$;
   м) $\forall x\ \forall y\ [(E(x) \wedge O(x)) \to \neg D(x, y)]$.

5. Предикат $P(x, y)$ задано в предметній області $D = \{a, b\}$ такою матрицею:

x	a	a	b	b
y	a	b	a	b
$P(x, y)$	0	1	1	1

   Яка з нижченаведених формул визначає цей предикат?
   а) $\forall x\ P(x, a)$; в) $\exists y\ \forall x\ P(x, y)$; д) $\forall y\ \forall x\ \neg P(x, y)$.
   б) $\forall y\ P(a, y)$; г) $\forall y\ \forall x\ P(x, y)$;

5.7. Формули у логіці предикатів

Елементарна формула, правильно побудовані формули, область дії квантора, інтерпретація формул логіки предикатів, загальнозначущі та суперечливі формули, логічний наслідок

Використовуючи поняття предиката, квантора і терма, можна визначити поняття формули у логіці предикатів.

Визначення

Якщо P — n-місний предикат і t_1, ..., t_n — терми, то $P(t_1, ..., t_n)$ називається ***атомом*** або ***елементарною формулою*** логіки предикатів. Наприклад: ДІЛИТЬСЯ(x, 13), ДІЛИТЬСЯ(x, y), БІЛЬШЕ(плюс(x, 1), x), ДОРІВНЮВАТИ(x, 1), СКЛАДАТИ(студенти, сесії).

Визначення

Правильно побудованими формулами логіки предикатів називаються формули, які можна рекурсивно визначити таким чином:

1. Атом є формулою.
2. Якщо F і G — формули, то $(\neg F)$, $(F \vee G)$, $(F \wedge G)$, $(F \to G)$, $(F \sim G)$ також є формулами.
3. Якщо F — формула, а x — вільна змінна, то $(\forall x)F$ і $(\exists x)F$ теж формули.
4. Ніяких формул, крім породжених вказаними вище правилами, не існує.

Визначення

Частина формули, на яку поширюється дія квантора, називається ***областю дії квантора***.

Оскільки дія квантора може поширюватися не на всю формулу, а тільки на її частину, то змінна може бути зв'язаною в одній частині формули і вільною в другій. У цьому випадку вважають, що змінна є і зв'язаною, і вільною одночасно.

Приклад. Визначити, які змінні є зв'язаними, а які — вільними у таких формулах:

1. $A(x, y)$.
2. $\exists y\ (B(x) \to \forall x\ A(x, y))$.
3. $\exists x\ (B(x) \to \forall x\ A(x, y))$.

Розв'язок. Обидві змінні у формулі 1 є вільними. У формулі 2 змінна y є зв'язаною, а змінна x — і зв'язаною, і вільною (змінна x вільна у предикаті $B(x)$ і зв'язана у предикаті $A(x, y)$). В формулі 3 змінна x є зв'язаною, а змінна y — вільною.

В логіці висловлень інтерпретація формули полягає у приписуванні атомам істиннісних значень. У логіці предикатів поняття інтерпретації формули декілька поширюється: необхідно вказати предметну область (область значень предметних змінних) і значення констант, а також функціональних символів і предикатів, що зустрічаються у формулі.

Визначення

Інтерпретація формули F логіки предикатів складається з елементів непорожньої предметної області D, значень всіх констант, функціональних символів і предикатів, що зустрічаються в F. Вказані значення задаються таким чином:

1. Кожній константі ставиться у відповідність деякий елемент з D.
2. Кожному n-місному функціональному символу ставиться у відповідність відображення з D^n у D. Тут $D^n = (x_1, x_2, ..., x_n)$, де $x_1, ..., x_n \in D$.
3. Кожному n-місному предикату ставиться у відповідність відображення з D^n в $\{I, X\}$.

Для кожної інтерпретації на області D формула може одержати істиннісне значення I або X згідно з такими правилами:

1. Якщо задані значення формул F і G, то істиннісні значення формул $(\neg F)$, $(F \vee G)$, $(F \wedge G)$, $(F \to G)$, $(F \sim G)$ одержуються за допомогою таблиць істинності відповідних логічних операцій.
2. Формула $(\forall x)F$ одержує значення I, якщо F одержує значення I для кожного x з D, у протилежному випадку вона одержує значення X.
3. Формула $(\exists x)F$ одержує значення I, якщо F одержує значення I хоча б для одного x з D, у протилежному випадку вона одержує значення X.
4. Формула, що містить вільні змінні, не може одержати істиннісне значення.

Після уточнення поняття інтерпретації в логіці предикатів такі поняття, як ***загальнозначущість***, ***суперечливість***, ***здійсненність***, ***нейтральність*** (***незагальнозначущість***) ***формули*** і ***логічний наслідок*** можуть бути визначені точно як для формул логіки висловлень (див. п. 5.2, 5.3).

Запитання

1. Дайте визначення атома у логіці предикатів.
2. Що називається формулою в логіці предикатів?
3. Сформулюйте поняття інтерпретації формули логіки предикатів.
4. Що розуміють під областю дії квантора?
5. Порівняйте поняття інтерпретації у логіці висловлень і логіці предикатів.
6. Назвіть види формул логіки предикатів залежно від прийнятих ними істиннісних значень.
7. Запишіть правила, згідно з якими формула логіки предикатів може одержати істиннісне значення.

Завдання

1. Нехай предикат $M(x, y)$ означає «x менше y», предикат $D(x, y)$ — «x дорівнює y», а предикат $S(x, y, z)$ — «$x = y + z$». Наведені висловлення сформулюйте природною мовою:
   а) $(\forall x)\ M(0, x)$;
   б) $(\forall x)\ S(x, x, x)$;
   в) $(\exists x)\ M(x, x - 1)$;
   г) $\neg((\exists x)\ S(x, x, x))$;
   д) $(\exists y)(\exists x)\ S(x, y, z)$;
   е) $(\exists y)(\forall x)\ (D(x, y) \vee M(x, y))$;
   ж) $(\forall x)(\forall y)\ (S(x, y, y) \to M(y, x))$.
2. Нехай X — множина співробітників відділу, а $P(x)$, $Q(x)$ і $R(x)$ — предикати, що означають відповідно: x займається спортом, x вивчає іноземну мову, x має винаходи, $x \in X$. Розшифруйте:
   а) $(\exists x)\ P(x)Q(x)$; б) $(\forall x)\ P(x)(Q(x) \to R(x))$.
3. Вказати вільні і зв'язані входження змінних у такі формули, що містять предикати A і B:
   а) $(\forall x_2)\ A(x_1, x_2)$;
   б) $(\forall x_3)((\forall x_1)A(x_1, x_2) \leftarrow A(x_3, x_1))$;
   в) $(\forall x_2)\ A(x_3, x_2) \to \forall x_3\ B(x_3, x_2)$;
   г) $(\forall x_2)\ A(x_1, x_2) \to \exists x_1\ A(x_1, x_2)$;
   д) $(\exists x_2)(\forall x_1)\ A(x_1, x_2) \to B(x_1)$.
4. Задано предметну область $D = \{1, 2\}$, значення констант a і b, функціональних символів f, а також предиката P:

a	b
1	2

$f(1)$	$f(2)$
2	1

$P(1,1)$	$P(1,2)$	$P(2,1)$	$P(2,2)$
1	1	1	0

Знайти істиннісні значення таких предикатних виразів:

а) $P(a, f(a)) \wedge P(b, f(b))$;

б) $(\forall x)(\forall y)(P(x, y) \to P(f(x), f(y)))$.

5. Оцінити формулу $(\forall x)(P(x)) \to Q(f(x), a)$ на інтерпретації $D = \{1, 2\}$, $a = 1$, $f(1) = 2$, $f(2) = 1$.

$P(1)$	$P(2)$	$Q(1, 1)$	$Q(1, 2)$	$Q(2, 1)$	$Q(2, 2)$
0	1	1	1	0	1

5.8. Закони і тотожності у логіці предикатів

Колізія змінних, заміна зв'язаної змінної, комутативні і дистрибутивні властивості кванторів, закон де Моргана для кванторів

Всі закони і тотожності, які справедливі у логіці висловлень, залишаються справедливими і у логіці предикатів. Крім того, у логіці предикатів існують додаткові закони, що призначені для еквівалентного перетворення формул, що містять квантори та змінні.

Слід зауважити, що перенесення квантора на початок формули може змінити зміст предикатного висловлення. Так, наприклад, $\forall x\ F(x) \wedge \forall x\ G(x)$ не еквівалентне $\forall x\ (F(x) \wedge G(x))$. У загальному випадку слід перейменувати зв'язані змінні, щоб запобігти *колізії* — ситуації, коли у формулі одна й та ж змінна знаходиться в області дії протилежних кванторів. Наприклад, у формулі $\forall x\ (F(x)) \to \exists x\ (Q(x))$ змінна x одночасно знаходиться в області дії кванторів $\forall$ і $\exists$.

Розглянемо висловлення, що використовує предикат рівності ДОРІВНЮЄ двох чисел: $\forall x\ \exists y$ ДОРІВНЮЄ$(x + 1, y)$. Дане висловлення означає, що для будь-якого числа x існує число y, яке більше його на одиницю. Наведене висловлення є істинним. Однак, якщо поміняти порядок розташування кванторів на протилежний, то одержимо таке висловлення: $\exists y\ \forall x$ ДОРІВНЮЄ$(x + 1, y)$. Одержане висловлення означає, що існує таке число y (одне!), яке на одиницю більше будь-якого числа x. Це висловлення не відповідає попередньому і є хибним.

Для еквівалентних перетворень предикатних висловлень з кванторами необхідно використовувати наведені нижче закони. Перш ніж безпосередньо перейти до розглядання законів дій з кванторами, введемо такі позначення: $F(x)$ і $H(x)$ — одномісні предикати, $P(x, y)$ — двомісний предикат.

1. ***Заміна зв'язаної змінної:***

$$(\exists x)\ F(x) = (\exists y)\ F(y);$$

$$(\forall x)\ \mathrm{F}(x) = (\forall y)\ F(y).$$

Введення нового позначення зв'язаної змінної (тобто перейменування зв'язаної змінної) не змінює зміст формули логіки предикатів, якщо виконується така умова: ніяка вільна змінна у будь-якій частині формули не повинна після перейменування бути зв'язаною. Іншими словами, для нового позначення зв'язаної змінної слід обирати букву, яка відсутня у формулі. Наприклад:

$$(\forall x)\ (\exists y)\ P(x, y) = (\forall x)\ (\exists z)\ P(x, z).$$

У даному прикладі здійснено операцію перейменування зв'язаної змінної y.

2. ***Комутативні властивості кванторів:***

$$(\forall x)\ (\forall y)\ P(x, y) = (\forall y)\ (\forall x)\ P(x, y);$$

$$(\exists x)\ (\exists y)\ P(x, y) = (\exists y)\ (\exists x)\ P(x, y).$$

Змінювати місцями можна тільки однойменні квантори.

$$(\forall x)\ (\exists y)\ P(x, y) \neq (\exists y)\ (\forall x)\ P(x, y).$$

3. ***Дистрибутивні властивості кванторів:***

$$(\forall x)F(x) \vee G = (\forall x)(F(x) \vee G);$$

$$(\exists x)F(x) \vee G = (\exists x)(F(x) \vee G);$$

$$(\forall x)F(x) \wedge G = (\forall x)(F(x) \wedge G);$$

$$(\exists x)F(x) \wedge G = (\exists x)(F(x) \wedge G),$$

де G — формула логіки предикатів, яка не містить x;

$$(\forall x)F(x) \wedge (\forall x)H(x) = (\forall x)(F(x) \wedge H(x));$$

$$(\exists x)F(x) \vee (\exists x)H(x) = (\exists x)(F(x) \vee H(x)).$$

Сформульований дистрибутивний закон справедливий тільки для квантора загальності $\forall$ при кон'юнкції $\wedge$ і квантора

існування $\exists$ при диз'юнкції $\vee$, оскільки інші комбінації призводять до нерівностей:

$$(\forall x)F(x) \vee (\forall x)H(x) \neq (\forall x)(F(x) \vee H(x));$$
$$(\exists x)F(x) \wedge (\exists x)H(x) \neq (\exists x)(F(x) \wedge H(x)).$$

Для подолання цього обмеження дистрибутивного закону, слід використовувати заміну зв'язаної змінної:

$$(\forall x)F(x) \vee (\forall x)H(x) = (\forall x)F(x) \vee (\forall y)H(y) =$$
$$= (\forall x)(\forall y)\,(F(x) \vee H(y));$$

$$(\exists x)\mathrm{F}(x) \wedge (\exists x)H(x) = (\exists x)\mathrm{F}(x) \wedge (\exists y)H(y) =$$
$$= (\exists x)(\exists y)\,(F(x) \wedge H(y)).$$

Таким чином, у загальному випадку дистрибутивні властивості кванторів можна записати такою схемою:

$$(Q_1x)F(x) \vee (Q_2y)\,H(y) = (Q_1x)(Q_2y)\,(F(x) \vee H(y));$$

$$(Q_1x)F(x) \wedge (Q_2y)\,H(y) = (Q_1x)(Q_2y)\,(F(x) \wedge H(y)),$$

де Q_1, Q_2 — будь-який з кванторів $\exists$ або $\forall$.

4. ***Закон де Моргана для кванторів:***

$$\neg((\forall x)F(x)) = (\exists x)\neg F(x);$$
$$\neg((\exists x)F(x)) = (\forall x)\neg F(x).$$

Проілюструємо цей закон таким прикладом. Нехай предикат $F(x)$ означає, що «x є простим числом». Коли x послідовно приймає значення ряду натуральних чисел — $x = \{1, 2, 3, 4, 5, 6, 7, \ldots\}$, предикат відповідно змінює істиннісне значення:

$$F(1) = X,\ F(2) = I,\ F(3) = I;$$
$$F(4) = X,\ F(5) = I,\ F(6) = X,\ F(7) = I,\ \ldots.$$

Переконаємося у справедливості першої формули для заперечення квантора загальності:

$\neg((\forall x)F(x))$ = «не всі x є простими числами» = «існують такі x, які є непростими числами» = $(\exists x)\neg F(x) = I$.

Обидва наведені висловлення істинні. Тепер переконаємося у справедливості другої формули для заперечення квантора існування:

$\neg((\exists x)F(x))$ = «немає жодного x, яке було б простим» = = «всі x є непростими числами» = $(\forall x)\neg F(x) = X$.

Ці висловлення є хибними.

Запитання

1. Що розуміють під колізією змінних?
2. До яких наслідків може призвести перенесення квантора на початок формули? Наведіть приклади коректного і некоректного перенесення кванторів на початок формули.
3. Поясніть суть заміни зв'язаної змінної.
4. Сформулюйте комутативні властивості кванторів.
5. За дотримання якої умови правомірно використовування дистрибутивних властивостей кванторів?
6. Яким чином можна обійти обмеження дистрибутивних властивостей кванторів?
7. Запишіть формули закону де Моргана для кванторів.

Завдання

1. Опустіть знаки заперечення безпосередньо на предикати або булеві змінні:
   а) $\neg(\exists x)\ ((\neg(\forall y)\ (B(y) \vee (\exists z)C(z)) \wedge \neg A(x)))$;
   б) $\neg(\forall y)\ (\neg(\exists x)\ (A(y, z) \vee B(x)) \vee C(z))$;
   в) $\neg(\exists x)\ ((\neg(\forall y)\ (A(x) \to B)) \vee C(x, y))$;
   г) $\neg(\forall u)\ \neg(\forall v)(P(u) \to Q(v) \to A(z))$.
2. Встановіть, чи еквівалентні задані предикати:
   а) $(\exists x)(A(x) \wedge \neg B(y))$ и $\neg(\forall z)(A(z) \to B(y))$;
   б) $(\forall x)((A(x) \to B(x)) \wedge (A(x) \to \neg B(x)))$ и $\neg(\exists y)\ A(y)$.
3. Винести за дужки квантори:
   а) $(\exists v)\ C(v, y) \wedge ((\exists x)\ A(x) \vee B)$;
   б) $(\exists x)(\exists y)\ A(x, y) \wedge (\exists x)(\exists y)\ B(x, y)$;
   в) $(\exists x)\ A(x, y) \vee ((\forall x)\ B(x) \vee (\forall y)\ C(y))$;
   г) $(\exists x)(\forall y)A(x, y) \wedge (\exists x)(\exists z)\ (B(x, z) \wedge (\exists y)\ A(x, y))$.
4. Довести загальнозначущість таких формул:
   а) $(\forall x)P(x) \to (\exists y)P(y)$;
   б) $(\forall x)P(x) \vee ((\exists y)\neg P(y))$.
5. Довести, що формула $(\forall x)P(x) \wedge (\exists y)\neg P(y)$ суперечлива.
6. Довести, що формула $P(a) \to \neg((\exists x)P(x))$ несуперечлива.

5.9. Випереджені нормальні форми і логічний висновок у логіці предикатів

Випереджена нормальна форма, алгоритм зведення до випередженої нормальної форми, правила видалення/введення квантора загальності/існування

В логіці висловлень було введено дві нормальні форми: кон'юнктивна і диз'юнктивна. В логіці предикатів вводиться третя нормальна форма, що називається випередженою нормальною формою.

Визначення

Формула F в логіці предикатів знаходиться у ***випередженій нормальній формі (ВНФ)*** тоді і тільки тоді, коли вона може бути зображена у вигляді $(Q_1x_1) \dots (Q_nx_n)(M)$, де кожне (Q_ix_i), $i = 1, \dots, n$, є або $(\forall x)$, або $(\exists x)$, а M — формула, що не містить кванторів. Причому $(Q_1x_1) \dots (Q_nx_n)$ називається ***префіксом***, а M — ***матрицею формули*** F.

Для перетворення виразів довільної форми у ВНФ необхідно по черзі виконати такі етапи перетворення.

1. Виключити логічні зв'язки еквіваленції (~) та імплікації (→), виразивши їх через операції диз'юнкції, кон'юнкції і заперечення за допомогою таких законів:

$$F \to G = \neg F \vee G;$$

$$F \sim G = (\neg F \vee G) \wedge (\neg G \vee F) = \neg F \wedge \neg G \vee F \wedge G.$$

2. Опустити знаки операцій заперечення безпосередньо на предикати, використовуючи закон подвійного заперечення $\neg(\neg F) = F$ і закони де Моргана $\neg(F \vee G) = \neg F \wedge \neg G$, $\neg(F \wedge G) = \neg F \vee \neg G$, у тому числі для кванторів:

$$\neg((\forall x)\, F(x)) = (\exists x)\, (\neg F(x)); \quad \neg((\exists x)\, F(x)) = (\forall x)\, (\neg F(x)).$$

3. Якщо необхідно, перейменувати зв'язані змінні.

4. Винести квантори на початок формули, використовуючи відповідні закони, для одержання випередженої нормальної форми.

Приклад. Звести формулу $(\forall x)F(x) \to (\exists x)Q(x)$ до ВНФ.

Розв'язок. Спочатку виключимо імплікацію, потім опустимо знак операції заперечення безпосередньо на предикат і винесемо квантор на початок:

$$(\forall x)F(x) \to (\exists x)H(x) = \neg((\forall x)F(x)) \vee (\exists x)H(x) =$$
$$= (\exists x)(\neg F(x)) \vee (\exists x)H(x) = (\exists x)(\neg F(x) \vee H(x)).$$

Приклад. Одержати ВНФ для формули

$$G \equiv (\forall x)(\forall y)((\forall z)(P(x, y) \wedge P(y, z)) \to (\exists z)R(x, y, z)).$$

Розв'язок. Скористаємося наведеним вище алгоритмом.

$$G = (\forall x)(\forall y)(\neg((\forall z)P(x, z) \wedge P(y, z)) \vee (\exists z)\, R(x, y, z)) =$$
$$= (\forall x)(\forall y)((\exists z)(\neg P(x, y) \vee \neg P(y, z)) \vee (\exists u)\, R(x, y, u)) =$$
$$= (\forall x)(\forall y)(\exists z)(\exists u)(\neg P(x, y) \vee \neg P(y, z) \vee R(x, y, u)).$$

Розглянемо правила висновку, які можна використовувати для проведення дедуктивних умовиводів з висловленнями логіки предикатів, що містять квантори. Ці правила часто використовуються у ході математичних доведень без додаткового пояснення.

Правило видалення квантора загальності

$$\frac{\forall x\ F(x)}{F(c) \text{ для довільного } c \in D}$$

використовується для доведення істинності $F(c)$, де c — довільно обраний елемент предметної області D, у якій справедливе $\forall x\, F(x)$. Наприклад, із засновку «*Всі студенти бажають одержувати добрі оцінки*» робимо висновок: «*Студент Петров бажає одержувати добрі оцінки*».

Правило введення квантора загальності

$$\frac{F(c) \text{ для довільного } c \in D}{\forall x\ F(x)}$$

стверджує істинність $\forall x\ F(x)$, якщо доведена істинність $F(c)$ для будь-якого c, тобто для всіх елементів c з розглянутої предметної області D.

Правило видалення квантора існування в істинній формулі $\exists x\ F(x)$ полягає в позначенні імені елемента c (конкретного або гіпотетичного), для якого $F(c)$ істинне:

$$\frac{\exists x\ F(x)}{F(c) \text{ для деякого } c \in D}.$$

Правило введення квантора існування

$$\frac{F(c) \text{ для деякого } c \in D}{\exists x\ F(x)}$$

дозволяє вирішити, що $\exists x\ F(x)$ є істинним, коли відомий деякий елемент c, для якого істинне $F(c)$.

Крім наведених правил, у логіці предикатів у ході дедуктивного висновку можна використовувати всі правила, які застосовуються для дедуктивних висновків у логіці висловлень.

Приклад. Показати, що з тверджень «*Всі у першій групі вивчають математику*» і «*Маша — студентка першої групи*» маємо висновок: «*Маша вивчає математику*».

Розв'язок. Позначимо через $F(x)$ предикат «x є студент першої групи», а через $M(x)$ — «x вивчає математику». Тоді засновки можна записати у вигляді: $\forall x\ (F(x) \to M(x))$ і F(Маша) відповідно, а потрібний висновок — M(Маша). Для одержання цього висновку необхідно здійснити таку послідовність дій:

1. $\forall x\ (F(x) \to M(x))$ перший засновок.
2. F(Маша) $\to M$(Маша) крок 1, правило видалення квантора $\forall$.
3. F(Маша) другий засновок.
4. M(Маша) кроки 2 і 3, правило відділення.

Таким чином, одержано шуканий результат.

Запитання

1. Дайте визначення випередженої нормальної форми.
2. За допомогою яких законів можна опустити знаки операцій заперечення безпосередньо на предикати?
3. Сформулюйте алгоритм перетворення виразів довільної форми у ВНФ.
4. Назвіть правила висновку, які можна використовувати для проведення дедуктивних умовиводів з висловленнями логіки предикатів.
5. В чому полягає правило видалення квантора загальності?
6. Як і навіщо використовується правило введення квантора загальності?
7. Поясніть відмінність у трактуванні елемента предметної області у правилі видалення квантора існування від прийнятої у правилі введення квантора загальності.
8. Яке правило введення квантора існування?

Завдання

1. Звести до ВНФ такі вирази:
   а) $(\forall y)\ (F_1(y) \vee \neg(\exists x)\ P(x, y))$;
   б) $\neg(\exists x)\ ((\forall y)\ A(x, y) \wedge (\exists y)\ (\forall z)\ (C(z) \to B(x, y))$;
   в) $(\exists x)\ (\forall y)\ B(x, y) \sim (\exists x)\ A(x)$;
   г) $\neg(\forall y)\ (\exists x)\ (A(x) \to B(y))$;
   д) $(\forall x)\ F_1(x) \to \neg(\forall x)\ (F_2(y) \vee (\forall y)\ P(x, y))$.
2. Визначте, чи є формула $(\exists x)\ (P(x) \wedge Q(x))$ логічним наслідком формул $(\exists x)\ P(x)$ і $(\exists x)\ Q(x)$.
3. Застосовуючи дедуктивні правила логіки предикатів, наведіть висновки з таких засновків:

3.1. Якщо хтось з тих людей — автор цих пліток, то він глупий і безпринципний. Але ніхто з тих людей не глупий і не позбавлений принципів.

3.2. Якщо всі ці люди не хоробрі або на них не можна покластися, то вони не належать до нашої компанії. Але вони належать до нашої компанії.

3.3. Якщо хтось з підозрілих здійснив всі ці нерозкриті крадіжки, то він був ретельно підготовлений і мав співучасника. Якщо б всі крадіжки були підготовлені ретельно, то, якщо б був співучасник, вкрадено було б набагато більше. Але останнє не має місця.

3.4. Якщо один з нас піде завтра на перше заняття, то він повинен буде підвестися рано, а якщо ми підемо сьогодні ввечері у кіно, то він ляже пізно спати. Якщо будь-який з нас ляже пізно спати, а підведеться рано, то буде задовольнятися п'ятьма годинами сну. Але ми не можемо задовольнятися п'ятьма годинами сну.

3.5. В бюджеті виникне дефіцит, якщо і тільки якщо не підвищать деякі мита. Державні витрати на всі соціальні нестатки скоротяться, якщо і тільки якщо у бюджеті буде дефіцит. Деякі мита підвищать.

3.6. Якщо всі ціни одночасно підвищуються, то підвищується і заробітна плата. Всі ціни високі або застосовується регулювання цін. Якщо застосовується регулювання цін, то немає інфляції. Спостерігається інфляція.

5.10. Обчислення предикатів

Структура обчислення предикатів, правила відділення та узагальнення, правила $\forall$ і $\exists$-введення, перейменування вільних і зв'язаних змінних

Аналогічно обчисленню висловлень в логіці предикатів існує формальна система — ***обчислення предикатів***, яка займається конструюванням формул і доведенням їх загальнозначущості. Обчислення предикатів має ідентичну обчисленню висловлень структуру, а саме: мову, систему аксіом і правила висновку.

Аксіоми обчислення предикатів можна поділити на дві групи:

1) Аксіоми обчислення висловлень (можна обрати будь-яку з формальних систем S_1, S_2).
2) Предикатні аксіоми, де змінна x у формулі $F(x)$ є вільною і жодного разу не піддається дії квантора за y:

P1) $\forall x\ F(x) \to F(y)$;

P2) $F(y) \to \exists x\ F(x)$.

Формула $F(y)$ одержана з $F(x)$ заміною x на y.

Для з'ясування сенсу вимоги до входжень x у $F(x)$ розглянемо як $F(x)$ формулу $\exists y\ P(y, x)$, в якій вільне входження x знаходиться в області дії квантора $\exists y$, тобто зазначена вимога не задовольняється. Підстановка даної формули до аксіоми P1 дає таку формулу: $\forall x\ \exists y\ P(y, x) \to \to \exists y\ P(y, y)$.

Якщо одержану формулу проінтерпретувати на множині натуральних чисел N з предикатом P «бути більше», то одержимо висловлення: «якщо для всякого x знайдеться y, який більше нього, то знайдеться і y, більший за самого себе». Засновок цієї імплікації істинний на N, а його висновок хибний, тому всі висловлення є хибними.

В обчисленні предикатів використовуються такі правила висновку:

1) Правило відділення (Modus Ponens), сформульоване у п. 5.3 (див. таблицю 5.5), повністю переноситься з обчислення висловлень.

2) Правило узагальнення ($\forall$-введення):

$$\frac{F \to G(x)}{F \to \forall x G(x)},$$

де $G(x)$ містить вільні входження x, а F їх не містить.

3) Правило $\exists$-введення:

$$\frac{G(x) \to F}{\exists x G(x) \to F}$$

при тих же вимогах до F і G, що й у попередньому правилі.

Правило перейменування вільних змінних

В обчисленні предикатів з вивідності формули $F(x)$, що містить вільні входження x, жодне з яких не знаходиться в області дії квантора за y, виходить вивідність $F(y)$.

Приклад. Довести справедливість правила перейменування вільних змінних.

Розв'язок.

1. $F(x)$ (за умови).

2. $F(x) \to (G \to F(x))$ (аксіома 1 формальної системи S_2; тут як G можна обрати будь-яку довідну формулу, що не містить вільних входжень змінних x; довідність цієї формули знадобиться на кроці 5, а обмеження на x — на кроці 4).

3. $G \to F(x)$ (за правилом відділення, кроки 1, 2).

4. $G \to \forall x\ F(x)$ (за правилом узагальнення, крок 3).

5. $\forall x\ F(x)$ (наслідок з кроку 4, оскільки G — істинна формула).

6. $F(y)$ (крок 5, аксіома P1)

Правило доведено.

Правило перейменування зв'язаних змінних

В обчисленні предикатів з вивідності $\forall x\ F(x)$ виходить вивідність $\forall y\ F(y)$, а з вивідності $\exists x\ F(x)$ — вивідність $\exists y\ F(y)$ за умови, що $F(x)$ не містить вільних входжень y і містить вільні входження x, жодне з яких не входить до області дії квантора за y.

Приклад. Довести правило перейменування зв'язаних змінних для квантора загальності.

Розв'язок.

1. $\forall x\ F(x)$ (за умови).

2. $\forall x\ F(x) \to F(y)$ (аксіома P1).

3. $\forall x\ \mathrm{F}(x) \to \forall y\ F(y)$ (правило узагальнення, крок 2).

4. $\forall y\ F(y)$ (кроки 1, 3).

Необхідно зауважити, що доведення для квантора $\exists$ здійснюється аналогічно, але використовує аксіому P2 і правило $\exists$-введення.

Теорема 1

Будь-яка довідна формула обчислення предикатів тотожно істинна.

Ця теорема аналогічна відповідній теоремі обчислення висловлень і показує, що правила висновку в обчисленні предикатів зберігають загальнозначущість, тобто їх застосування до загальнозначущих формул знов дає загальнозначущі формули.

Запитання

1. Сформулюйте призначення обчислення предикатів.
2. Запишіть формули аксіом обчислення предикатів.
3. Поясніть обмеження аксіом обчислення предикатів.
4. Назвіть правила висновку обчислення предикатів.
5. Побудуйте схему правила узагальнення.
6. Визначте правило $\exists$-введення за допомогою схеми.
7. Які обмеження необхідні для правила узагальнення і $\exists$-введення? До яких наслідків може призвести недотримання вказаних обмежень?
8. Яку особливість здобуває правило підстановки в обчисленні предикатів?
9. Сформулюйте правило перейменування вільних змінних.
10. В чому полягає сутність правила перейменування зв'язаних змінних?
11. Сформулюйте твердження теореми про загальнозначимості правил висновку обчислення предикатів.

Завдання

1. Доведіть нелогічність таких міркувань:

   1.1. Всі студенти нашої групи — члени клуба «Динамо». А деякі члени клуба «Динамо» займаються спортом. Отже, деякі студенти нашої групи займаються спортом.

   1.2. Деякі студенти нашої групи — вболівальники «Динамо». А деякі вболівальники «Динамо» займаються спортом. Отже, деякі студенти нашої групи займаються спортом.

   1.3. Кожний першокурсник знайомий кимось з студентів другого курсу. А деякі другокурсники — спортсмени. Отже, кожний першокурсник знайомий з кимось із спортсменів.

2. Показати, що формула G не є логічним наслідком множини формул K:

   а) $G = (\forall x)\neg R(x)$, $K = \{(\exists x)R(x) \to (\exists x)Q(x),\ \neg Q(a)\}$;

   б) $G = (\forall x)R(x, x)$, $K = \{(\forall x)(\forall y)(R(x, y) \to R(y, x)),$ $(\forall x)(\forall y)(\forall z)(R(x, y) \wedge R(y, z) \to R(x, z))\}$;

   в) $G = (\exists x)(P(x) \wedge \neg R(x))$, $K = \{(\forall x)[P(x) \to (\exists y)(Q(y) \wedge S(x, y))],$ $(\exists x)[R(x) \wedge (\forall y)(Q(y) \to \neg S(x, y))],\ (\exists x)P(x)\}$.

5.11. Багатозначна логіка

Виникнення багатозначних логік, значення істинності висловлення, алфавіт багатозначної логіки, унарні і бінарні функції, повна система функцій багатозначної логіки

Вперше багатозначна логіка з'явилася через заперечення аристотелева закону виключеного третього. Відповідно до цього закону диз'юнктивне висловлення $p \vee \neg p$ є тавтологія, а атомар-

не висловлення p у аристотелевій логіці завжди або істинне, або хибне. Оскільки в аристотелевій логіці будь-яке висловлення може приймати тільки одне з двох значень істинності (істину або хибність), вона одержала назву ***двозначної логіки***. В 1921 році Я. Лукашевич у маленькій статті на двох сторінках розглядає трьохзначну логіку, тобто таку логіку, в якій будь-яке висловлення p може приймати одне з трьох можливих значень істинності. Незалежно від Лукашевича Е. Пост аналізує ***m-значну*** логіку, в якій висловлення p може приймати одне з m можливих значень істинності, де m — будь-яке ціле число, більше за 1. У випадку, коли m більше за 2, логіку називають ***багатозначною***. В 1930 році Лукашевич і Тарський приступають до подальшого вивчення m-значної логіки. В 1932 році поняття ***m-значної*** логіки узагальнюється Г. Рейхенбахом, що розглядає нескінченнозначну логіку, в якій для висловлення p існує нескінченна множина значень істинності.

Видатний вчений А. Гейтінг приблизно у то й же час побудував двозначну символічну логіку, виходячи з потреб інтуїціоніської математичної школи. Ця логіка, на відміну від аристотелевої, не приймає беззаперечно законів виключеного третього і подвійного заперечення. Внаслідок цього закони створеної зі спеціальними цілями логіки Гейтінга, як і закони багатозначних логік, відрізняються від законів Аристотеля. Тому такі логіки називають неаристотелевими. Символічна двозначна логіка, побудована Гейтінгом у роботі «Принципи математики», належить до неаристотелевих логік, відрізняючись від аристотелевої іншою інтерпретацією імплікації.

Подібно неевклідовим геометріям неаристотелеві логіки також знайшли собі застосування. Нескінченнозначна логіка була задумана Г. Рейхенбахом як фундамент математичної теорії ймовірності. А у 1933 році Т. Швицький помітив, що багатозначні логіки можуть бути використовувані у сучасній квантовій фізиці. Багато аспектів такого використання були досліджені Г. Біркгофом і Г. Рейхенбахом. Використання інтуїціоністами логіки Гейтінга також свідчить про математичні цінності нових логік.

Для спрощення розгляду основних положень теорії багатозначних логік обмежимося трьохзначною логікою і скористаємося методом таблиць істинності. В першу чергу,

відтворимо таблицю істинності для операції кон'юнкції (таблиця 5.11).

Таблиця 5.11. Таблиця істинності кон'юнкції

∧		q	
		I	*X*
p	*I*	*I*	*X*
	X	*X*	*X*

Таблиця 5.11 побудована таким чином: у лівому стовпці знаходяться можливі значення істинності для висловлення *p*, а у верхньому рядку — можливі значення істинності для висловлення *q*. Знаючи значення істинності вказаних висловлень, можна знайти значення їх кон'юнкції у комірці, що стоїть на перетині рядку, що відповідає значенню істинності *p*, і стовпця, відповідного значенню істинності *q*. Оскільки, за визначенням, кон'юнкція істинна в тому і тільки в тому випадку, коли обидва висловлення *p* і *q* істинні, значення *I* стоїть у лівій верхній комірці таблиці, а врешті — *X*. Зазначимо, що таблиця заповнюється на підставі одного лише визначення операції кон'юнкції.

Тепер перейдемо до трьохзначної логіки і позначимо три можливі значення істинності висловлення через *I*, «?» і *X*. Знов складемо таблицю істинності операції кон'юнкції (таблиця 5.12). Оскільки кон'юнкція істинна, коли обидва висловлення *p* і *q* істинні, ліва верхня комірка таблиці повинна містити значення *I*, крім того, *I* не може знаходитися ні в якій іншій комірці таблиці. Таким чином, залишається вісім комірок, в кожній з яких може бути записано або значення *X*, або значення «?». Всього одержується $2^8 = 256$ способів заповнення таблиці. Звідси маємо, що у трьохзначній логіці існує 256 різних визначень кон'юнкції.

Таблиця 5.12. Шаблон таблиці істинності кон'юнкції у трьохзначній логіці

∧		q		
		I	?	*X*
p	*I*	*I*		
	?			
	X			

В таблицях 5.13 і 5.14 наведено дві з 256 можливих таблиць істинності операції кон'юнкції у трьохзначній логіці.

Таблиця 5.13. Таблиця істинності кон'юнкції у трьохзначній логіці (Лукашевич та ін.)

∧		*q*		
		I	?	*X*
p	*I*	*I*	?	*X*
	?	?	?	*X*
	X	*X*	*X*	*X*

Таблиця 5.14. Таблиця істинності кон'юнкції у трьохзначній логіці (Бочар)

∧		*q*		
		I	?	*X*
p	*I*	*I*	?	*X*
	?	?	?	?
	X	*X*	?	*X*

Таблиця істинності 5.13 була запропонована Лукашевичем, Постом і Россером. Вона будується відовідно до угоди, за якою значення «?» більш хибне, ніж *I*, але менш хибне, ніж *X*, а значення кон'юнкції співпадає із значенням істинності більш хибного із складових висловлень.

Відмінне від описаного визначення кон'юнкції дає Бочар (таблиця 5.14). За Бочаром символ «?» означає нерозв'язність, а кон'юнкція висловлень *p* і *q* вважається нерозв'язною у випадку нерозв'язності хоча б одного із складових висловлень.

Розглянемо таблицю істинності операції заперечення. У випадку заперечення єдине обмеження полягає у тому, що $\neg p$ не може бути істинним при істинності висловлення *p*, і $\neg p$ не може бути хибним у випадку хибності *p*. Вказане обмеження повністю визначає таблицю істинності для заперечення у двозначній логіці і припускає 12 можливих способів визначення заперечення у трьохзначній. Нижче наведено дві з 12 можливих для заперечення таблиць істинності операції заперечення. Таблиця істинності 5.15, що запропонована Постом, заснована на угоді, згідно з якою операції заперечення

привласнюється наступне за порядком значення аргументу. Таблиця істинності 5.16 була запропонована Бочаром, Лукашевичем і Россером, які виходили з того, що $\neg(\neg p)$ повинне бути еквівалентне p.

Таблиці істинності інших логічних операцій можуть бути побудовані на підставі їх визначення за допомогою операцій кон'юнкції та заперечення. Оскільки кон'юнкція та заперечення незалежні, а решта операцій може бути через них виражена, то існує взагалі $256 \times 12 = 3072$ різних трьохзначних логік. Таким чином, кількість різних можливих структур багатозначної логіки надзвичайно велика.

Таблиця 5.15. Таблиця істинності заперечення у трьохзначній логіці (Пост)

p	$\neg p$
I	?
?	X
X	I

Таблиця 5.16. Таблиця істинності заперечення у трьохзначній логіці (Бочар та ін.)

p	$\neg p$
I	X
?	?
X	I

В деяких випадках при побудові багатозначних логік використовується таке поняття значення істинності.

Визначення

Дійсне число $T(p)$ з інтервалу $[0, 1]$, яке ставиться у відповідність висловленню p, називають ***значенням істинності*** висловлення p.

Значення істинності $T(p)$ можна розуміти як ймовірність того, що висловлення p істинне, причому два висловлення, що мають одне й те ж значення істинності, можна вважати логічно еквівалентними. Значення істинності $T(p) = 1$ означає істинність, а значення $T(p) = 0$ — хибність висловлення p. Заперечення $\neg p$ визначається за допомогою значення істинності

таким чином: $T(\neg p) = 1 - T(p)$. В трьохзначній логіці, наприклад, як значення істинності можна прийняти числа 0, 1/2 і 1. Якщо значення істинності $T(p) = 1/2$, то значення істинності $\neg p$ також дорівнює 1/2, і p виявляється логічно еквівалентним своєму запереченню. Таке висловлення може бути назване сумнівним, причому заперечення цього висловлення також виявляється сумнівним. Зазначений підхід свідчить про наявність тісного зв'язку між багатозначними логіками і теорією ймовірностей.

Багатозначна логіка розглядає однорідні логічні функції, що визначені на множині (0, 1, ..., k – 1), яка складається з k елементів. В силу однорідності сама функція *k-значної* логіки від n змінних приймає значення з тієї ж скінченної множини.

Визначення

Функції k-значної логіки визначені і приймають значення, що входять до деякої множини $B_k = \{0, 1, ..., k - 1\}$, що називається ***алфавітом*** цієї ***логіки***.

Функцію k-значної логіки однозначно визначає її таблиця значень (істинності). Множину всіх функцій *k-значної* логіки позначають P_k. Кількість функцій P_k, що залежать від n змінних, дорівнює k^{k^n}. Як і у двозначній логіці, висловлення зображуються у вигляді формул k-значної логіки. Елементарні функції зображують узагальнення аналогічних функцій двозначної логіки. Розглянемо основні функції k-значної логіки.

Унарні функції

1. Циклічне заперечення:

$$\neg x = x + 1 (\mathrm{mod}\ k),$$

де y mod k — залишок від ділення y на k.

Таким чином, ця елементарна функція зображує узагальнення заперечення у розумінні «циклічного» зміщення значень:

$$\neg x = \begin{cases} x+1, & \text{при } x \neq k-1 \\ 0, & \text{при } x = k-1 \end{cases}.$$

2. Заперечення Лукашевича:

$$N_x = k - 1 - x.$$

Наведена елементарна функція N_x є іншим узагальненням операції заперечення у розумінні «дзеркального» відображення значень.

3.Узагальнене заперечення:

$$I^{\sigma}(x)=\begin{cases}k-1, \text{ при } x=\sigma\\ 0, \text{ при } x\neq\sigma, \quad \text{де } x,\sigma\in\{0,1,\ldots,k-1\}\end{cases}$$

Елементарна функція $I^{\sigma}(x)$ при $\sigma \neq k-1$ є узагальненням деяких властивостей заперечення.

4. Характеристична функція:

$$J^{\sigma}(x)=\begin{cases}1, \text{ при } x=\sigma\\ 0, \text{ при } x\neq\sigma, \quad \text{де } x,\sigma\in\{0,1,\ldots,k-1\}\end{cases}$$

Функція $J^{\sigma}(x)$ — характеристична функція значення σ при $\sigma' \neq k-1$ зображує узагальнення операції заперечення.

Бінарні функції

1. Узагальнення кон'юнкції:
   $\min(x_i, x_j)$.
2. Інше узагальнення кон'юнкції:
   $x_i\, x_j \pmod k$.
3. Узагальнення диз'юнкції:
   $\max(x_i, x_j)$.

Визначення

Система функцій $f_1 \ldots f_n$ називається ***повною***, якщо будь-яка функція з P_k може бути зображена у вигляді формули, що складається з цих функцій.

В k-значній логіці залишаються справедливими деякі закони двозначної, а саме: асоціативності, комутативності, дистрибутивності і т. д. Подібно до функцій двозначної логіки k-значні логічні функції можуть бути задані у вигляді таблиці. Кількість стовпців у таблиці дорівнює k^n, де n — кількість змінних, що входять до функції, а кількість функцій визначається числом k^{k^n}, яке швидко зростає із збільшенням k.

В k-значній логіці існує k констант $f_0 = 0$, $f_1 = 1$, ..., $f_{k-1} = k-1$. Серед функцій однієї змінної найбільш часто використовуваними є такі:

1) Характеристичні функції i-го порядку — $f_0(x)$, $f_1(x)$, $f_2(x)$, що визначені в таблиці 5.17.

Таблиця 5.17. Функції однієї змінної у трьохзначній логіці

	X			
$F(x)$	x	0	1	2
	$f_0(x)$	2	0	0
	$f_1(x)$	0	2	0
	$f_2(x)$	0	0	2
	N_x	2	1	0
	$\neg x$	1	2	0

2) Заперечення Лукашевича $N_x = k - 1 - x$.

3) Функція циклічного заперечення $\neg x = x + 1 \pmod{}$.

Таблиці істинності вказаних функцій у трьохзначній логіці будуть мати вигляд, що зображений у таблиці 5.17. Серед функцій двох змінних найбільш важливе значення мають такі:

1) *значна* диз'юнкція — $x_1 \vee x_2 = \max(x_1, x_2)$.

2) *значна* кон'юнкція — $x_1 \wedge x_2 = \min(x_1, x_2)$.

3) Функція Шеффера — Веббах — $x_1|x_2 = x_1 \vee x_2 + 1 (\text{mod } k)$.

4) Додавання за модулем k — $x_1 + x_2 \pmod{k}$.

5) Множення за модулем k — $x_1 * x_2 \pmod{k}$.

Значення даних функцій при $k = 4$ зображено в таблиці 5.18.

Таблиця 5.18. Функції двох змінних у чотиризначній логіці

x_1	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
x_2	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$x_1 \vee x_2$	0	1	2	3	1	1	2	3	2	2	2	3	3	3	3	3
$x_1 \wedge x_2$	0	0	0	0	0	1	1	1	0	1	2	2	0	1	2	3
$x_1 \vee x_2 + 1 \pmod{k}$	1	2	3	0	2	2	3	0	3	3	3	0	0	0	0	0
$x_1 + x_2 \pmod{k}$	0	1	2	3	1	2	3	0	2	3	0	1	3	0	1	2
$x_1 * x_2 \pmod{k}$	0	0	0	0	0	1	2	3	0	2	0	2	0	3	2	1

Скориставшись поняттям характеристичних функцій для двозначного випадку, нескладно записати ДДНФ і ДКНФ у багатозначній логіці. Нагадаємо, що основну роль у ДДНФ відіграють елементарні кон'юнкції $x_1^{\sigma_1} \wedge \ldots \wedge x_n^{\sigma_n}$, які відмінні від нуля лише на одному наборі $(\delta_1, \ldots, \delta_n)$. При цьому всі вони одержані з кон'юнкції, що відповідає одиничному набору, підстановкою функції від однієї змінної x^{δ}.

Запитання

1. Що розуміють під багатозначною логікою?
2. Які існують різновиди багатозначних логік?
3. Скільки різних визначень кон'юнкції існує у чотиризначній логіці?
4. Складіть таблицю істинності кон'юнкції у трьохзначній логіці.
5. В чому полягає єдине обмеження операції заперечення у багатозначній логіці?
6. Дайте визначення поняттю значення істинності висловлення.
7. Сформулюйте визначення алфавіту *k-значної* логіки.
8. Скільки існує різних функцій k-значної логіки від n змінних?
9. Запишіть формули основних унарних функцій k-значної логіки.
10. Назвіть найважливіші бінарні функції *k-значної* логіки.
11. Яка система функцій k-значної логіки називається повною?
12. Назвіть найбільш часто використовувані функції однієї змінної.
13. Складіть таблицю основних функцій двох змінних у чотиризначній логіці.

Завдання

1. Складіть таблицю істинності функції узагальненого заперечення у трьохзначній логіці.
2. Побудуйте таблицю істинності імплікації у трьохзначній логіці, виходячи з припущення, що істина не може імплікувати хибність. Чи можливо у даній логіці єдиним чином виразити імплікацію через заперечення та кон'юнкцію?
3. За аналогією з двохзначною логікою доведіть справедливість дистрибутивного закону у k-значній логіці.
4. Побудуйте таблиці істинності функцій однієї змінної у чотиризначній логіці.
5. Запишіть формули ДДНФ і ДКНФ у k-значній логіці.

розділ 6

Теорія графів

6.1. Історичні зауваженння. Типові задачі

Прийнято пов'язувати зародження теорії графів як математичної дисципліні з роботою Леонарда Ейлера 1736 р., у якій знайдено умову існування у зв'язному графі циклу, що містить всі ребра графа (без повторень). Такий цикл тепер називається ***ейлеровим***. Як показує простий приклад (рис. 6.1), є графи, що не є ейлеровими циклами.

Рис. 6.1. Не ейлерів граф

Першоджерелом задачі про ейлерів цикл сам Ейлер називає відому головоломку про кенігсберські мости. В місті Кенігсберг (у 1736 р.) два річкових острова A, B з'єднувалися між собою і з берегами C, $Д$ річки Прегель сьома мостами (рис. 6.2).

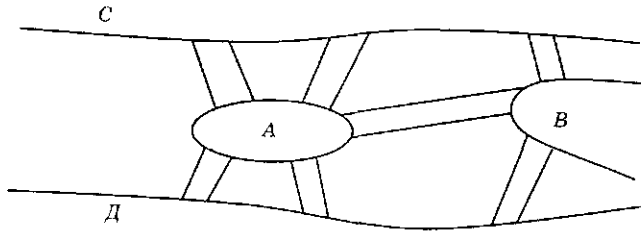


Рис. 6.2. Кенігсберські мости

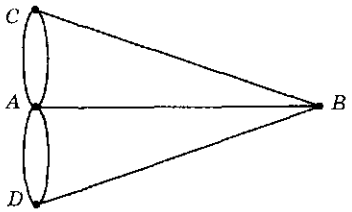


Рис. 6.3. Граф переходів по мостам

Чи можна здійснити прогулянку по місту, пройшовши по кожному мосту точно один раз, і повернутися до вихідної точки? Зобразивши ділянки суші вершинами, мости — ребрами (рис. 6.3), одержимо граф, який не є ейлеровим циклом (див. п. 6.3). Тому і задача про кенігсберські мости не має позитивного розв'язку.

Наступний крок зробив у 1847 р. Кірхгоф. Вивчаючи електричні ланцюги та абстрагуючись від фізичної природи пристроїв, він виділив окремо комбінаторно-топологічну структуру, що називається зараз ***графом ланцюга***, розробив алгоритм знаходження максимального підграфу без циклів (що називається деревом) і з його допомогою записав найменшу незалежну систему рівнянь ланцюга. Дослідження з електротехніки, електроніки, релейно-контактних схем до сьогодні індукують різні дослідження з теорії ***графів***, і навпаки.

В 1857 р. у зв'язку з проблемами органічної хімії Келі розглядав задачу перелічення всіх дерев зі степенями вершин 1 і 4. Вона пов'язана з описом ізомерів граничних (насичених) вуглеводнів C_nH_{2n+2} з даним числом (n) атомів вуглецю. Задача виявилася непростою, її сенс зрозумілий з рис. 6.4, на якому зображено 2 ізомери при $n = 4$ (бутан та ізобутан).

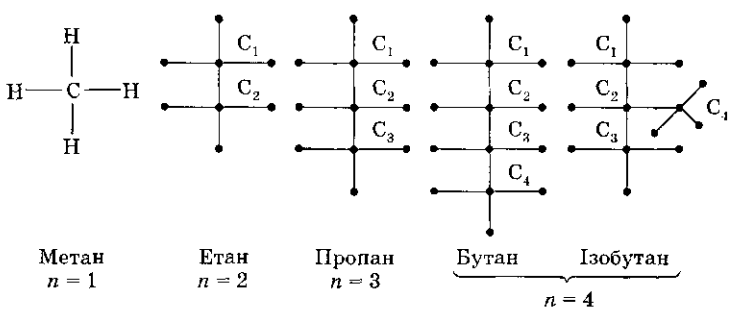


Рис. 6.4. Ізомери C_nH_{2n+2}

Добре відома задача комівояжера, що має численні застосування (в тому числі в економіці, проектуванні): для заданої

системи пунктів (міст) і доріг між ними, схематично зображених плоским[1] зв'язним графом, обрати найкоротший маршрут комівояжера, який виходить і повертається до вихідного міста так, щоб відвідати решту пунктів і тільки один раз. Задача не завжди має розв'язок, а якщо має, то відомі алгоритми працюють ефективно лише для порівняно невеликих графів — і це, незважаючи на неабияку бібліографію із задачі комівояжера.

Частковий випадок цієї задачі, що не враховує довжину переходів між пунктами, але зберігає всі її принципові труднощі, сформулював у 1859 р. В. Гамільтон на прикладі графа додекаедру: обійти замкненим маршрутом по ребрах багатогранника всі його вершини, пройшовши через кожну вершину рівно один раз (крім початку і кінця шляху). Саме цю задачу Гамільтон продав за 25 гіней одному майстрові іграшок у вигляді гри «Навкруги світу». Для довільного графа задача Гамільтона не обов'язково має розв'язок і формулюється так: Знайти ***простий цикл***, що містить всі вершини графа (такий цикл називається ***гамільтоновим***). Для графа додекаедра гамільтонів цикл існує. За певних умов сучасна задача комівояжера, що припускає неодноразове відвідання вершин, зводиться до відшукання у графі із заданими «довжинами» ребер гамільтонового циклу найменшої довжини. Згадані умови такі: кожне ребро (a, b) графа має довжину, що дорівнює довжині найкоротшого ланцюга між вершинами a і b.

Одна з найвідоміших задач математики — проблема чотирьох фарб — також належить до графів. Ще у минулому столітті було відзначено, що кожну конкретну географічну карту можна розфарбувати чотирма фарбами так, щоб будь-які дві сусідні[2] країни були пофарбовані у різні кольори. Однак доведення для довільної карти (тобто для двозв'язного плоского графа) вдалося дати з використанням ЕОМ тільки в останні роки. Починаючи з XIX століття, було надано багато помилкових доведень проблеми чотирьох фарб як професіоналами, так і не математиками — всіх приваблювала простота, естетичність і гучна популярність задачі. Слід підкреслити результат Хівуда, що довів у 1890 р. достатність п'яти фарб для розфарбування карти.

---

[1] Тобто реалізованим як геометричний граф у R^2.

[2] Що мають загальну лінійну (а не точкову) ділянку границі.

Для неплоских графів задача про розфарбування ставиться інакше — для вершин; тут є багато результатів і нерозв'язаних задач, цікавих для теорії і практики (див. р. 6.7).

Граф називається ***плоским*** (або ***планарним***), якщо його можна укласти на площині без схрещення ребер (шляхом їх неперервної деформації). Топологічний критерій планарності графа виявив Л. С. Понтрягін (1927 р.) без опублікування цього результату і незалежно — К. Куратовський (1930 р.). Через вагомість планарної реалізації електронних схем (друковані плати і ін.) останнім часом одержано інші критерії планарності та алгоритми укладення графів (Вагнер, Уїтні, Мак-Лейн, Фарі, Штейн тощо).

Починаючи з 30-х років XIX століття, популярність графів і кількість праць з чистої теорії графів та її застосувань неухильно зростає. За допомогою графа моделюються будь-які схеми, в яких виділяються більш прості частини (вершини) і зв'язки між ними (ребра). Не дивно, що графи зустрічаються у дослідженнях з соціології, психології, економіки, теорії ігор, логіки, програмування, теорії ймовірностей (ланцюги Маркова), квантової механіки (діаграми Фейнмана), хімії (структура молекул), статистичної механіки, кристалофізики, медицини (нервові, судинні та інші мережі), електро- і радіотехніки, лінгвістики, теорії розкладів, з транспортних мереж і течій, вентиляційних мереж та ін.

6.2. Неорієнтовані графи і термінологія

У подальшому буде з'ясовано, що будь-який скінченний граф (при загальному його визначенні) можна для наочності уявити як сукупність ліній, з'єднуючих задані точки у трьохвимірному просторі. Тому корисно мати визначення так званого ***геометричного графа*** у просторі R^n: це сукупність точок X і простих[1] кривих Y, кінці яких є точки з X, а самі криві попарно не мають спільних внутрішніх точок. Точки з X називаються ***вершинами***, криві з Y —

[1] Неперервних, самонеперетинних.

ребрами графа. На рис. 6.5 зображено граф у просторі R^2 (або, якщо бажано, — у R^3), де

$$X = \{x_1, x_2, x_3, x_4, x_5\};$$

$$Y = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7\}.$$

Позначення для графа: $G = (X, Y)$, $n = n_G$ — число вершин, $m = m_G$ — число ребер графа.

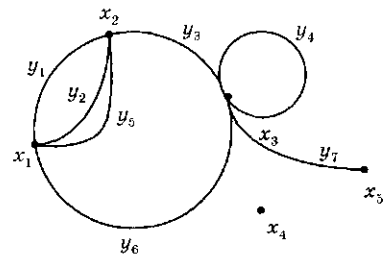


Рис. 6.5. Приклад графа у R^2

Основні терміни[1]

Суміжні вершини (x_2, x_3) — це вершини, з'єднані ребром, ***суміжні*** ребра (y_2, y_3) — це ребра, що мають спільну вершину.

Вершина x_2 і ребро y_3 ***інцидентні одна одному***, якщо точка x_2 є кінець кривої y_3. ***Петля*** (y_4) — замкнене ребро. Ізольована вершина (x_4) неінцидентна жодному ребру.

Паралельні (кратні) ребра (y_1, y_2, y_5) — це ребра, що інциденті одній парі вершин (x_2, x_1).

Маршрут (з'єднуючий вершини x_a, x_b) — скінченна послідовність ребер та інцидентних їм вершин, що складають неперервну криву з кінцями x_a, x_b. Число ребер у маршруті називається його ***довжиною*** (включаючи внесок повторюваних ребер). Так, на рис. 4. послідовність ребер y_1, y_5, y_2, y_3, y_4, y_7 (у більш докладних позначеннях — переміжна послідовність вершин і ребер x_1, y_1, x_2, y_5, x_1, y_2, x_2, y_3, x_3, y_4, x_3, y_7, x_5) є маршрут довжини 6 з кінцями x_1, x_5.

Маршрут ***замкнений***, якщо кінці його співпадають ($x_a = x_b$).

Маршрут називається ***ланцюгом***, якщо всі його ребра різні, і ***простим ланцюгом***, якщо всі його вершини (крім, може, кінців ланцюга) різні.

Цикл — це замкнений ланцюг (***простий цикл***, якщо ланцюг простий).

Граф називається ***зв'язним***, якщо будь-яка пара його вершин з'єднується ланцюгом.

Підграф — будь-яка частина графа, що сама є графом.

Компонента (зв'язності) — максимальний зв'язний підграф у графі (наприклад, граф рис. 6.5 має 2 компоненти:

[1] Для орієнтованих графів див. п. 6.4.

$\{x_4\}$ і залишкова частина). ***Степінь*** $\deg x_i = \delta(x_i)$ ***вершини*** x_i — число інцидентних їй ребер ($\delta(x_1) = 4$, $\delta(x_5) = 1$, $\delta(x_4) = 0$), причому петля враховується як два ребра ($\delta(x_3) = 5$).

Граф називається ***n-зв'язним***, якщо між будь-якими його двома вершинами знайдеться n ланцюгів, що попарно не мають спільних некінцевих вершин.

Граф $G = (X, Y)$ називається:

— ***порожнім***, якщо множина його ребер порожня;

— ***простим***, якщо він не містить петель і паралельних ребер;

— ***повним***, якщо він простий і кожна пара вершин суміжна;

— ***регулярним*** або ***однорідним*** (степені r), якщо степені всіх його вершин однакові (дорівнюють $r = \deg x_i$, $\forall x_i \in X$);

— ***деревом***, якщо він не містить циклів і зв'язний;

— ***мультиграфом***, якщо він містить паралельні ребра;

— ***псевдографом***, якщо він містить петлі та кратні ребра.

Граф $G' = (X, Y')$ називається ***доповненням*** простого графа $G = (X, Y)$ з тією ж множиною вершин X, якщо $Y \cap Y' = \varnothing$ і граф $G_0 = (X, Y \cup Y')$ є ***повним***. Інакше кажучи, у доповнювальному графі G' вершини (x_i, x_j) суміжні (з'єднані ребром $y'_k \in Y'$), якщо вони несуміжні у вихідному графі G.

Точка зчленування графа — вершина, видалення якої разом з інцидентними їй ребрами призводить до збільшення числа компонент графа.

Відстань $d(\upsilon, \vartheta)$ між вершинами υ, ϑ графа G — довжина найкоротшого ланцюга між υ, ϑ. Якщо від вершини υ до вершини ϑ не веде жодний ланцюг, то $d(\upsilon, \vartheta) = \infty$.

Діаметром $d(G)$ графа G називається максимальна відстань $d(\upsilon, \vartheta) \neq \infty$ у графі G.

Гранню (коміркою) геометричного графа у R^2 (тобто ***плоского графа***, в якому ребра перетинаються тільки у вершинах) називається така непорожня замкнена підобласть площини, що будь-які дві точки області можна з'єднати простою (Жордановою) кривою, внутрішні (не кінцеві) точки якої лежать всередині області, не перетинаючись з ребрами графа.

Границею грані вважається множина ребер і вершин графа, що належать грані. Будь-який ***скінченний плоский граф*** має в точності одну ***необмежену*** грань, яка називається ***зовнішньою гранню***. Решта граней (обмежених) називаються ***внутрішніми***.

Ексцентриситет $e(\upsilon)$ вершини $\upsilon \in X$ у зв'язному графі G є відстань від υ до найбільш віддаленої від неї вершини, а ***радіус*** $r(G)$ графа G — найменший з ексцентриситетів вершин:

$$e(\upsilon) = \max_{\forall x \in X} d(x, \upsilon), \quad r(\upsilon) = \min_{\forall \upsilon \in X} e(\upsilon).$$

Зрозуміло, що найбільший ексцентриситет дорівнює діаметру графа:

$$d(G) = \max_{\forall \upsilon \in X} e(\upsilon) = \max_{\upsilon} \max_{x} d(x, \upsilon).$$

Вершина υ називається ***центральною вершиною графа*** G, якщо на ній досягається мінімум ексцентриситетів, тобто $e(\upsilon) = r(G)$.

Центром графа G називається множина всіх його центральних вершин; центр може складатися з єдиної вершини, а може — з двох і більше вершин. Наприклад, центр простого циклу G_n містить всі n вершин. На рис. 6.6 наведено дерево G з числом вершин $n = 21$, числом ребер $m = 20$, де для кожної вершини вказано її ексцентриситет. Дерево має радіус $r(G) = 4$, діаметр $d(G) = 7$, центр дерева складається з пари вершин $\{u, \upsilon\}$, кожна з яких має мінімальний ексцентриситет 4.

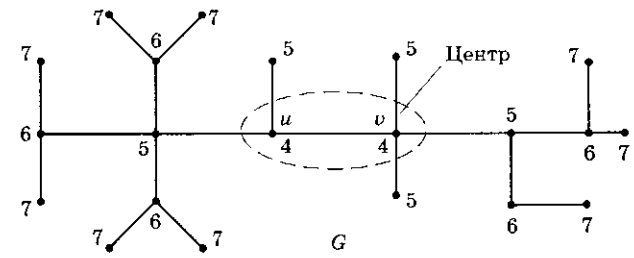


Рис. 6.6. Ексцентриситети вершин і центр графа G

Завдання

1. Чи правильно, що замкнений маршрут непарної довжини містить простий цикл? А парної довжини?
2. Довести або спростувати:
   а) об'єднання будь-яких двох різних ланцюгів, з'єднуючих дві вершини, містить простий цикл;
   б) об'єднання будь-яких двох простих ланцюгів, з'єднуючих дві вершини, містить простий цикл;
   в) граф зв'язний тоді і тільки тоді, коли множину X його вершин не можна розбити на дві підмножини X_1, X_2, що не

перетинаються, так, щоб кожне ребро було інцидентне або двом вершинам з X_1, або двом вершинам з X_2;

г) у зв'язному графі будь-які два найдовші прості ланцюги мають спільну вершину;

д) неправильно, що у кожному зв'язному графі всі найдовші прості ланцюги мають спільну вершину.

3. Якщо граф зв'язний і n — число його вершин, то яка нижня оцінка для числа його ребер?
4. Довести, що зв'язний граф залишається зв'язним при видаленні ребра тоді і тільки тоді, коли це ребро міститься у деякому циклі.
5. Довести, що множина всіх ребер скінченного зв'язного графа утворює простий цикл тоді і тільки тоді, коли степені всіх вершин кратні 2.
6. Довести, що всі ребра скінченного зв'язного графа можна включити до деякого маршруту.
7. Довести, що ланцюг z не може бути простий, якщо з множини всіх його ребер можна утворити інший ланцюг $z_1 (\neq z)$.
8. Нехай граф G має n_k вершин степені k, а решта вершин мають степінь $k + 1$. Тоді $n_k = (k + 1)n - 2m$, де n — число вершин, m — число ребер графа.

6.3. Ейлерові цикли

Заперечна відповідь у головоломці про кенігсберські мости (див. р. 6.1) випливає з такої теореми Ейлера.

Теорема 6.1

Граф містить ейлерів цикл тоді і тільки тоді, коли він зв'язний і степені всіх його вершин — парні.

□ Якщо граф ейлерів, то він зв'язний і при обході ейлерова циклу захід і вихід в чергову вершину вносить дві одиниці в її степінь (тому що ребра не повторюються). Оскільки проходяться всі ребра графа, то степені всіх вершин — парні.

Нехай тепер зв'язний граф G має парні степені вершин. Оберемо вершину x_0 і перейдемо від неї до вершини x_1 за деяким ребром y_0, пофарбувавши його подумки у будь-який колір. З вершини x_1 рушимо далі в x_2 по непофарбленому ребру (це можливо через $\deg x_1 = 2k$, $k \geq 1$), пофарбувавши його після проходження. Просуваючись таким чином по непофарбованим ребрам, ми повинні повернутися до вихідної вершини x_0 (зустрічаючи, можливо, інші вершини кілька разів),

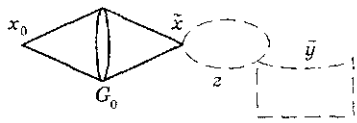


Рис. 6.7. Нарощування циклу

бо у протилежному випадку вершина $x_i \neq x_0$, з якої ми не можемо рухатися далі, має непарну степінь. Повернувшись перший раз у вершину x_0, ми пофарбуємо граф G_0, що є за побудовою ейлеровим циклом. Якщо $G_0 \neq G$, то в силу зв'язності G існує хоча б одне непофарбоване ребро $\bar{y}$, що з'єднується з підграфом G_0 непофарбованим ланцюгом z, можливо, нульової довжини (рис. 6.7). Видалимо з графа G пофарбовані ребра і одержимо граф з парними степенями вершин. Знайдемо вершину $\tilde{x}$ дотику ланцюга $z \cup \bar{y}$ до підграфу G_0. Починаючи з вершини $\tilde{x}$, можна виділити з непофарбованих ребер новий цикл $G_1(\supset z \cup \bar{y})$, який у поєднанні з G_0 також утворює ейлеровий цикл у графі $G_1 \cup G_0$. Нарощуючи таким чином цикли, ми вичерпаємо весь скінченний граф G ейлеровим циклом. ■

Завдання

1. Як нарисувати весь ейлерів цикл одним розчерком, не прибігаючи у процесі побудови до послідовного нарощування циклів?
2. Довести, що граф ейлерів тоді і тільки тоді, коли множину його ребер можна розбити на прості цикли (що попарно не перетинаються по ребрах).
3. Ейлеровим ланцюгом у графі G називається ланцюг, що містить всі ребра з G. Довести таке узагальнення теореми Ейлера: граф G має ейлеровий ланцюг тоді і тільки тоді, коли він зв'язний і число вершин непарного степеня дорівнює 0 або 2.

Доповнення

1. Задача китайського листоноші тісно пов'язана з ейлеровими циклами: у графі G, ребрам якого приписані додатні ваги C_j, знайти замкнений маршрут Q, що містить всі ребра $\{y_j\}_1^m$, з мінімальною сумарною вагою $\sum_{j=1}^{m} n_j c_j$, де n_j — число проходжень ребра y_j у маршруті Q.
   Для будь-якого зв'язного графа G задача китайського листоноші розв'язна, оскільки число проходжень ребра y_j не обмежується одиницею. Якщо G містить ейлерів цикл, то будь-який такий цикл дає розв'язок задачі листоноші: кожне ребро має мінімум проходжень і вага циклу $\sum c_j$ — мінімальна. Алгоритми розв'язку задачі китайського листоноші пропонували Беллман і Кук — метод динамічного програмування в окремому випадку,

коли $c_j = 1(\forall_j)$; у загальному випадку — Едмонде, Джонсон, Басакер, Сааті, Кристофідес.

Застосування задачі китайського листоноші досить численні. Наприклад, це мінімізовані за довжиною пробігу транспортні задачі обслуговування споживачів у кожному ребрі на заданій мережі доріг (доставка пошти, продуктів та інших вантажів, полив, патрулювання вулиць, прибирання сміття) або інспектування кожного відрізку у мережі розподілених систем (електричних, телефонних, залізничних, вентиляційних) і навіть визначення та рекомендація економного маршруту огляду музею.

6.4. Абстрактні графи та геометричні реалізації. Орієнтовані графи

Геометричний граф у R^n, що визначений у п. 6.2, зазвичай називається неорієнтованим. Якщо на кожному його ребрі обрати певний напрямок, то такий граф називається ***орієнтованим геометричним графом*** у R^n, а його ребра з напрямками — ***дугами***. Наприклад, забезпечивши ребра графа на рис. 6.5 напрямками, одержимо орієнтований граф (рис. 6.8).

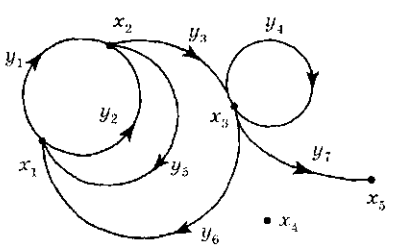


Рис. 6.8. Орієнтований граф

Навпаки, ігноруючи напрямки дуг в орієнтованому графі і розглядаючи їх як ребра, одержуємо відповідний ***неорієнтований граф***. Через це для орієнтованого графа зберігаються всі поняття та характеристики, що відносяться до відповідного неорієнтованого: суміжність двох вершин (наприклад, x_2 і x_3), інцидентність дуги і вершини (наприклад, x_2 і y_3), маршрут, ланцюг, цикл, зв'язність, степені вершини, n-зв'язність і т. д. Природно, виникають аналогічні поняття і характеристики, що враховують орієнтацію дуг, частіше всього з додаванням епітету «орієнтований».

Орієнтований маршрут (довжини n) — послідовність з n дуг (не обов'язково різних), що проходиться неперервним розчерком вздовж напрямку дуг від початкової вершини до кінцевої (наприклад, орієнтований маршрут y_1, y_5, y_2, y_3 має початком

вершину x_1, кінцем — x_3). Будь-який орієнтований маршрут є маршрутом (неорієнтованим), зворотне може не виконуватися (наприклад, маршрут y_6, y_7 між вершинами x_1, x_5 не є орієнтованим). Орієнтований маршрут без дуг, що повторюються, називається ***шляхом*** (y_1, y_3, y_7); замкнений шлях — ***контуром***. Шлях є ланцюгом, контур — циклом; зворотне, взагалі кажучи, неправильно. ***Шлях*** (контур) без внутрішніх вершин, що повторюються, зветься ***простим***. Орграф називається ***сильно зв'язним***, якщо для будь-якої пари різних вершин v, w існує шлях з v до w і шлях із w до v.

Хоча геометричні графи (в R^n) мають в якості вершин X і ребер (дуг) Y реальні геометричні об'єкти у R^n (точки і криві), метричні характеристики цих об'єктів (форми кривих, їх довжини і відстані між точками) не відіграють ролі. Приймаються до уваги лише властивості інциденції вершин x_i і ребер (дуг) y_k. У зв'язку з цим визначення графа як геометричного об'єкту в метричному просторі R^n є надмірним з логічної точки зору, хоча і наглядним, що спирається на геометричну інтуїцію.

Визначення абстрактного графа. Графом G (орієнтованим) називається трійка $G = (X, Y, f)$, де X, Y — довільні множини, $f: Y \to X \times X$ — відображення множини Y у декартовий добуток множини X на себе. Елементи множини X називаються вершинами, множини Y — дугами, відображення f — інцидентором. Елементами $X \times X$ є упорядковані пари (x_i, x_j) і рівність $f(y_k) = (x_i, x_j)$ можна інтерпретувати таким чином: «дуга» y_k виходить з «вершини» x_i і заходить до «вершини» x_j; така дуга y_k називається інцидентною вершинам x_i, x_j. Наприклад, для геометричного графа на рис. 6.8 можна побудувати відповідний йому абстрактний граф так: ввести абстрактні множини $X = \{x_i\}_{i=1}^{5}$, $Y = \{y_k\}_{k=1}^{7}$ та інцидентор f:

$$f(y_1) = f(y_2) = (x_1, x_2); \quad f(y_3) = (x_2, x_3); \quad f(y_4) = (x_3, x_3);$$

$$f(y_5) = (x_2, x_1); \quad f(y_6) = (x_3, x_1); \quad f(y_7) = (x_3, x_5).$$

Неорієнтований абстрактний граф визначається за допомогою трійки $G = (X, Y, \varphi)$, де множина X називається множиною вершин, Y — множина ребер, а $\varphi: Y \to \langle X \times X \rangle$ — відображення ребер у симетризований декартів добуток вершин на себе, в якому пари (x_i, x_j) і (x_j, x_i) ототожнені. Якщо через $\Theta: X \times X \to \langle X \times X \rangle$ позначити відповідне відображення симетризації, то для орієнтованого графа $G = (X, Y, f)$, $f: Y \to X \times X$

відповідний йому неорієнтований граф можна визначити як $G_0 = (X, Y, \Theta f)$.

Часто графи позначаються у вигляді перших двох множин трійки: $G = (X, Y)$, інцидентор f мається на увазі неявно.

Два графа G і G' називаються ***ізоморфними*** ($G \sim G'$), якщо існують взаємно однозначні відображення вершин і ребер $\varphi: X \to X'$, $\Psi: Y \to Y'$ відповідно, що зберігають відношення інциденції: $f'\psi = (\varphi \times \varphi)f$. Інакше кажучи, ребро (дуга) y інцидентне (заходить, виходить) вершині x графа G тоді і тільки тоді, коли в графі G' відповідне ребро (дуга) $y' = \Psi(y)$ інцидентне (заходить, виходить) вершині $x' = \varphi(x)$. Отже, ***ізоморфізм*** $\Phi: G \to G'$ є перетворення $\Phi(G) = \Phi(X, Y, f) = (\varphi(X), \Psi(Y); (\varphi \times \varphi)f\Psi^{-1})$, але для спрощення ми будемо писати іноді $X' = \Phi(X)$, $Y' = \Phi(Y)$, $f' = \Phi(f)$. Ізоморфізми графа у себе називаються ***автоморфізмами***; очевидно, вони утворюють групу. Навпаки, можна показати, що будь-яка абстрактна скінченна група ізоморфна групі автоморфізмів деякого графа G (зі скінченним числом вершин та дуг). *Якщо граф G ізоморфний геометричному графу G' у R^n, то G' називається* ***геометричною реалізацією графа G у просторі R^n***. Наприклад, геометричний граф у R^3 на рис. 6.9 ізоморфний геометричному графу в R^2 на рис. 6.10.

Граф (абстрактний або геометричний), що має геометричну реалізацію у R^2 (інакше — плоску реалізацію), називається ***планарним***. Геометричний граф у R^3 на рис. 6.9 — планарний, п'ятивершинний граф на рис. 6.11 — непланарний (докладніше див. завдання і доповнення 9, 10).

Рис. 6.9. Граф K_4

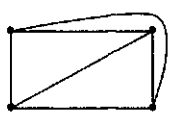


Рис. 6.10. Плоске зображення K_4

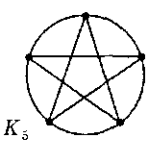


Рис. 6.11. Граф K_5

Ізоморфізм графів є відношенням еквівалентності, тому вся множина графів розбивається на класи ізоморфних графів. Часто ми не будемо розрізняти ізоморфні графи (тобто вивчати класи) і мати справу з геометричними представниками (реалізаціями).

Слід підкреслити алгоритмічну складність перевірки ізоморфізма двох графів — це комбінаторна задача, яку можна

звести до з'ясування подібності двох матриць суміжності (див. нижче п. 6.5). Геометричні зображення графів на кресленнях в цьому випадку мало ефективні. Наприклад, не відразу видно, що невеликі шостивершинні графи на рис. 6.12 ізоморфні один одному.

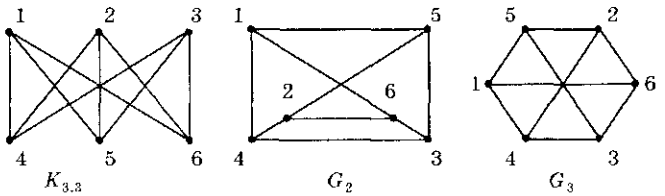


Рис. 6.12. Ізоморфні графи

Граф G називається ***скінченним***, якщо множини вершин X і ребер Y скінченні, і ***локально скінченним***, якщо степені всіх вершин скінченні. ***Нескінченні дискретні графи*** (*із зчисленними множинами* X, Y) зустрічаються у застосуваннях.

Зв'язок з відношеннями. Найпростіші властивості

Якщо R — бінарне відношення на множині X, то факт x_1Rx_2 можна зобразити наявністю дуги з елемента x_1 в елемент x_2. Таким чином, *між бінарними відношеннями та орієнтованими графами без паралельних дуг встановлюється взаємно однозначна відповідність.* Оскільки нас будуть цікавити графи, що припускають паралельні дуги (наприклад y_1, y_2 на рис. 6.8), то ми маємо справу з об'єктом більш загальним, ніж бінарні відношення.

Наприкінці торкнемося найпростіших комбінаторних властивостей скінченних графів. Оскільки поява кожного нового ребра додає по одиниці до степенів двох інцидентних йому вершин (або двійку до степені однієї вершини у випадку петлі), сума степенів всіх вершин графа в два рази перевершує число ребер $m = |Y|$ (***«лема про рукостискання»***):

$$\sum_{\forall x \in X} \delta(x) = 2|Y|.$$

Наслідок. *Число вершин непарного степеня парне.*

Дійсно, сума парних степенів — парна, тому і сума непарних степенів повинна бути парною.

В орграфі, крім степеня вершини $\delta(x)$, вводиться ***додатний степінь*** $\delta^+(x)$ — число дуг, що починаються у вершині x, і ***від'ємний степінь*** $\delta^-(x)$ — число дуг, що закінчуються у вершині x. За іншою термінологією $\delta^+(x)$, $\delta^-(x)$ — ***півстепені «виходу і заходу»*** у вершині x. Очевидно,

$$\delta^+(x) + \delta^-(x) = \delta(x),$$

$$\sum_{\forall x \in X} \delta^+(x) = \sum_{\forall x \in X} \delta^-(x) = 1/2 \sum_{\forall x \in X} \delta(x) = |Y| = m.$$

Завдання та доповнення

1. Ейлерів контур в орграфі, за визначенням, містить всі дуги і тільки по одному разу. Скоректувати доведення теореми Ейлера 6.1 (для неорграфів) так, щоб була одержана така теорема. Орграф має ейлерів контур тоді і тільки тоді, коли він є зв'язним і псевдосиметричним: $\delta^-(x) = \delta^-(x)$, $\forall x \in X$.
2. Граф $G = (X, Y, f)$ називається ***симетричним***, якщо для кожної пари вершин (x_i, x_j) число дуг, що йдуть з x_i до x_j, дорівнює числу дуг, що йдуть з x_j до x_i: $|f^{-1}(x_i, x_j)| = |f^{-1}(x_j, x_i)|$. Перевірити, що симетричний граф є псевдосиметричним, а зворотне — неправильно.
3. Довести, що будь-який контур може бути розділений на кілька простих контурів.
4. Довести, що будь-який непростий шлях з v до w можна розділити на простий шлях і один або кілька простих контурів.
5. Якщо $\delta^-(x) \geq 1$ і $\delta^-(x) \geq 1$ $\forall x \in X$, то чи правильно, що будь-яка вершина орграфа належить, щонайменше, одному контурові?
6. Довести, що якщо $\delta^+(x) > 0$ або $\delta(x) > 0$ для $\forall x \in X$, то в орграфі існує, щонайменше, один контур.
7. Довести, що орієнтований граф сильно зв'язаний тоді і тільки тоді, коли існує замкнений орієнтований маршрут, до якого кожна дуга входить, щонайменше, один раз.
8. Нехай вершини відповідають цілим числам від 0 до 12. Побудувати графи, що характеризують бінарні відношення R, де vRw означає: а) $V = W^2$; б) v і w порівнянні за модулем 4 (тобто $v - w$ кратне 4); в) v і w взаємно прості.
9. Відома головоломка — «Задача про три криниці» — полягає у з'єднанні дорогами трьох ворогуючих домів з трьома криницями (джерелами) так, щоб кожний дім з'єднувався з кожною криницею, але дороги не перетиналися (за виключенням початкових і кінцевих точок). Переконайтеся, що задача не має розв'язку: її розв'язок означав би, що граф $K_{3,3}$ (рис. 6.12) має плоску реалізацію, що неправильно.

10. Графи K_5 (рис. 6.11) і $K_{3,3}$ (рис. 6.12) надають приклади неплоских графів. Виявляється, що ці дві конфігурації є характерними для неплоских графів. Цей непростий результат було встановлено у 1927 р. Л. С. Понтрягіним (без опублікування), і у 1930 р. — незалежно К. Куратовським.

Теорема Понтрягіна — Куратовського

Граф G є планарним тоді і тільки тоді, коли він не містить підграфів виду G_1, G_2 (рис. 6.13). G_1 і G_2 називаються графами Понтрягіна — Куратовського; мають прості ланцюги між позначеними вершинами і можуть бути одержані заміною ребер у графах $K_{3,3}$ і K_5 довільними простими ланцюгами.

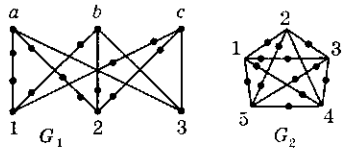


Рис. 6.13. Графи Понтрягіна — Куратовського

11. Довести строго, що графи Понтрягіна — Куратовського не планарні.

Вказівка. Достатньо розглянути графи K_5 (рис. 6.11) і $K_{3,3}$ (рис. 6.12). Припустити, що вони планарні, за теоремою Ейлера обчислити число граней f; враховуючи оцінки знизу 4 (у $K_{3,3}$) і 3 (у K_5) для числа ребер у грані, підсумувати числа ребер за всіма гранями і порівняти з фактичним числом ребер.

6.5. Матриця суміжності, ізоморфізми та операції над графами

1. Квадратна матриця A розміру $n = |X|$, елемент якої $a_{i,j}$ дорівнює числу дуг, що йдуть від вершини x_i до вершини x_j, називається ***матрицею суміжності орграфа*** $G = (X, Y)$. Для неорієнтованого графа елемент $a_{i,j}$ матриці суміжності A_0 визначається як число ребер, з'єднуючих вершини x_i і x_j (очевидно, A_0 завжди симетрична відносно головної діагоналі: $A_0 = A_0^T$). Ненульове значення елемента $a_{i,j}$ характеризує суміжність вершин x_i і x_j у графі, звідси і назва — матриця суміжності. Ясно, що граф з точністю до ізоморфізму відновлюється за матрицею $A(A_0)$. Для орграфа на рис. 6.8 і відповідного неорграфа на рис. 6.5 матриці суміжності мають вигляд:

$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}; \quad A_0 = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{bmatrix} 0 & 3 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}.$$

Очевидно, для довільного орграфа G і відповідного неорієнтованого графа G_0 («що втратив» орієнтацію дуг) матриці суміжності A і A_0 зв'язані співвідношенням $A + A^T = A_0$.

2. Вже відзначалося, що практично (і візуально для геометричних графів) важко встановити ізоморфізм двох графів (див. рис. 6.12). Матриці суміжності дозволяють сформулювати простий алгебраїчний критерій ізоморфізму.

Теорема 6.2

Графи $G = (X, Y)$, $G' = (X', Y')$ ізоморфні тоді і тільки тоді, коли вони мають однакове число вершин, і матриця суміжності $A(G')$ одержується з матриці $A(G)$ послідовними переставленнями рядків з одночасним переставлянням однойменних стовпців.

В неорієнтованих графах те ж правильне для матриць A_0. Інакше кажучи, існує така підстановка δ у групі підстановок множини $\{1, \ldots, n\}$, $n = |X| = |X'|$, що для всіх елементів матриць правильно $a_{ij}(G') = a_{\delta(i)\,\delta(j)}(G)$.

□ З визначення матриці суміжності виходить, що взаємна заміна номерів вершин x_1, x_2 у графі G призводить до переставлень місцями перших двох рядків і одночасно перших двох стовпців у матриці $A(G)$, і навпаки. Таке перетворення є найпростішим ізоморфізмом над графом G. У той же час ізоморфізм Φ над G однозначно визначається (відновлюється) за своєю дією на вершини: $\Phi(X) = X'$, отже — за деякою підстановкою δ на множині номерів вершин. Але підстановка δ розкладається у суперпозицію найпростіших підстановок $(i \leftrightarrow j)$, тому ізоморфізм $\Phi\colon G \to G'$ еквівалентний суперпозиції найпростіших ізоморфізмів і вказаному в умові теореми перетворенню над матрицею $A(G)$. ■

Кількість всіх підстановок дорівнює $n!$, тому безпосереднє застосування теореми 6.2 для розпізнавання ізоморфізму графів, що полягає у перебиранні $n!$ варіантів, неефективне при великому числі вершин $n = |X|$. Існують різні засоби та алгоритми, які для графів того чи іншого специфічного виду істотно зменшують об'єм обчислень при з'ясуванні ізоморфізму.

3. Для двох орграфів $G = (X; Y; f)$, $G_1 = (X; Y_1; f_1)$ з однаковими множинами вершин визначимо *операції додавання і множення.*

Сума $G + G_1 = (X; Y \cup Y_1; f + f_1)$ одержується об'єднанням дуг Y і Y_1. При множенні $GG_1 = (X; Y_0; f_0)$ з вершини x_i до вершини x_j йде стільки дуг, скільки існує шляхів довжини 2 у графі $G + G_1$ з вершини x_i до вершини x_j таких, що перша дуга належить Y, друга — Y_1.

Теорема 6.3

Для суми та добутку орграфів (з однаковими множинами вершин) матриці суміжності відповідно додаються і перемножуються, та навпаки:

$$A(G + G_1) = A(G) + A(G_1); \quad A(GG_1) = A(G) \cdot A(G_1).$$

□ Твердження для суми очевидно: у графі $G + G_1$ число дуг з x_i до x_j дорівнює $a_{ij}(G) + a_{ij}(G_1)$. В графі $G + G_1$ число різних шляхів з послідовністю вершин виду $\{x_i, x_k, x_j\}$, таких, що дуги з x_i до x_k належать графу G і дуги з x_k до x_j належать графу G_1, дорівнює $a_{ik}(G) \cdot a_{kj}(G_1)$ (рис. 6.14). В цій послідовності вершин (x_i, x_k, x_j) проміжна вершина x_k фіксована. Загальне число шляхів довжини 2 з першою дугою з Y, другою — з Y_1 дорівнює $\sum_{k=1}^{m} a_{ik}(G) \cdot a_{kj}(G_1)$ — загальному елементу матриці $A(G) \cdot A(G_1)$. ■

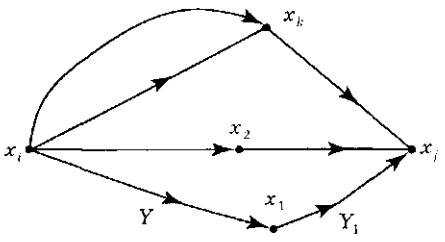


Рис. 6.14. Граф $G + G_1$

Наслідок 6.1. *Якщо A — матриця суміжності орграфа G, то елемент a_{ij}^n матриці A^n (натурального степеня n) дорівнює числу різних орієнтованих маршрутів довжини n, що йдуть з вершини x_i до x_j у графі G.*

□ Твердження правильне при $n = 1$ за визначенням матриці суміжності. Нехай воно правильне для $k = n - 1$: елемент a_{ij}^{n-1} матриці A^{n-1} дорівнює числу орієнтованих маршрутів довжини $n - 1$ з x_i до x_j у графі G і одночасно числу дуг з x_i до x_j у графі G^{n-1}, визначеному послідовним перемноженням двох чинників. Оскільки $A^n = A^{n-1}A$, то, застосовуючи теорему 6.3 до добутку графів $G^{n-1}G$, одержуємо твердження для матриці A^n (n-го кроку індукції). ■

Наслідок 6.2. *В орграфі G існує ормаршрут довжини n тоді і тільки тоді, коли $A^n \neq 0$; не існує контурів тоді і тільки тоді, коли матриця суміжності нільпотентна: $A^r = 0$ при деякому r.*

Зауваження. В зв'язку з другою частиною наслідку 2 корисно відзначити, що у випадку контуру довжини r з вершинами α, β, ..., γ всі степені A^{nr} ($n = 1, 2, \ldots$) матриці суміжності A мають відмінні від нуля діагональні елементи з номерами (α, α), (β, β), ..., (γ, γ). Якщо контурів у графі немає (зокрема, якщо більш того, немає циклів), то довжини всіх шляхів обмежені числом дуг m і $A^{m+1} = 0$, елемент a_{ij}^n матриці A^n дорівнює числу різних простих шляхів довжини n з вершини i у вершину j.

4. Питання ***досяжності*** (однієї вершини з іншої шляхом певної довжини) зв'язані, звичайно, з властивостями степенів матриці суміжності A графа. Виявляється, властивості степенів A, A^2, A^3, ... довільної матриці A з невід'ємними елементами $a_{ij} \geq 0$ (наприклад, матриці вартостей в економіці, ймовірностей переходів і т. п.) також надають цінну інформацію про зв'язки між локальними «вузлами» об'єкта у даний момент або різні моменти часу.

Практично корисні задачі про підрахунок числа контурів заданої довжини n природно зв'язані з матрицею суміжності A (достатньо проаналізувати діагональні елементи матриці A^n). Задачі такого сорту зустрічаються при кодуванні, у криптографії, лінгвістиці, при моделюванні графом дискретних фізичних процесів — у зв'язку з підсистемами, що «зациклюються».

Завдання

1. Встановити ізоморфізм графів G_1, G_2 на рис. 6.15.

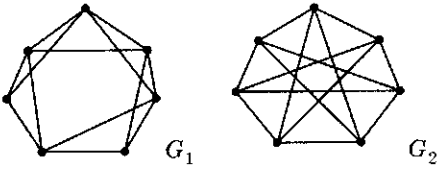


Рис. 6.15. Графи степеня 4

2. Доведіть, що:
а) граф G не є зв'язним тоді і тільки тоді, коли деяким переставленням рядків з одноіменним переставленням стовпців матриця суміжності $A(G)$ зводиться до матриці, що розпадається:

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \text{ де } A_i \text{ — квадратні матриці;}$$

б) максимальне число діагональних квадратних блоків, на які може розпадатися матриця суміжності при деякій нумерації вершин, дорівнює числу зв'язних компонент графа.

3. Дати визначення сильно зв'язного графа у термінах поведінки елементів a_{ij}^n послідовних степенів A^n ($n = 1, 2, \ldots$) матриці суміжності A.
4. Як виразити через елементи матриць A^n відстань $d(x_i, x_j)$ між вершинами x_i, x_j (довжину найкоротшого ланцюга між цими вершинами)?
5. Доведіть, що якщо G — регулярний неорієнтований граф степеня r, то всі корені λ_k «характеристичного» полінома $\det(A_0 - \lambda_I) = X_0(\lambda)$ знаходяться у колі радіуса r: $|\lambda_k| \le r$, причому один з коренів дорівнює $r(= \lambda_{k0})$.

6.6. Матриця інциденцій

1. Оскільки вершини і дуги орграфа $G = (X, Y, f)$ без петель занумеровані: $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_m\}$, дію інцидентора f можна охарактеризувати $(n \times m)$ — матрицею інциденцій $B = \{b_{ij}\}$, де за визначенням:

$$b_{ij} = \begin{cases} +1, \text{ якщо дуга } y_j \text{ виходить з вершини } x_i; \\ -1, \text{ якщо дуга } y_j \text{ заходить у вершину } x_i; \\ 0, \text{ якщо дуга } y_j \text{ не інцидентна вершині } x_i. \end{cases}$$

Для неорграфа G без петель матриця інциденцій $R = \{r_{ij}\}$ визначається так: $r_{ij} = 1$, якщо ребро y_j інцидентне вершині x_i, і $r_{ij} = 0$ — у протилежному випадку.

Для графа на рис. 6.16 і для відповідного неорграфа з втратою орієнтації дуг маємо:

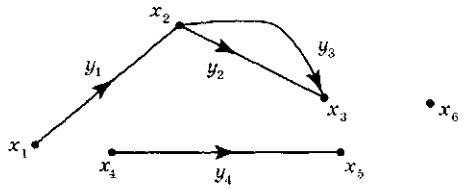


Рис. 6.16. Граф з трьома компонентами

$$B = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}; \quad R = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}.$$

Матриці орієнтованого і відповідного симетризованого неорієнтованого графів зв'язані співвідношенням $|b_{ij}| = r_{ij}$. Очевидно також, що в матриці B кожний стовпчик містить точно одну +1 і одну −1, в R — дві +1, решта елементів — нулі. З цієї ж причини сума всіх рядків у B дорівнює нульовому вектору, у R — нулю за модулем 2.

Якщо граф G має n компонент зв'язності, то, нумеруючи вершини і ребра окремими групами у кожній компоненті (або переставляючи стовпці та рядки у B і R), одержимо для матриці інциденцій $B(R)$ блочно-діагональну структуру:

$$B = \begin{bmatrix} B_1 & 0 & \dots & 0 \\ 0 & B_2 & \dots & 0 \\ & \dots & & \\ 0 & 0 & \dots & B_n \end{bmatrix}.$$

Правильне і зворотне.

2. Теорема 6.4

Ранг матриці інциденцій B зв'язного орграфа з n вершинами дорівнює $n-1$ (числу рядків без одного).

□ Внаслідок зв'язності графа кожній вершині інцидентна хоча б одна дуга, тому у матриці B немає нульових рядків. Викреслимо останній рядок $\vec{b}_n$ і припустимо, що для матриці B_0, яка залишилася, лінійна комбінація її вектор-рядків дорівнює нулю:

$$\sum_{i=1}^{n-1} \alpha_i \vec{b}_i = 0.$$

Вершина x_n зв'язана дугою y_{j_1} з деякою вершиною x_{i_1}, тому у викресленому рядку $\vec{b}_n$ елемент b_{nj_1} відмінний від нуля, а у рядку $\vec{b}_{i_1}$ — елемент $b_{i_1 j_1} \neq 0$. Оскільки у стовпці j_1 матриці B_0 тільки один елемент $b_{i_1 j_1}$ відмінний від нуля, то $\alpha_{i_1} = 0$. В графі G система вершин $\{x_n, \ldots, x_{i_1}\}$ зв'язана деякою дугою y_{j_2} з новою вершиною x_{i_2}, тому $b_{i_2 j_2} \neq 0$. Другий ненульовий елемент стовпця j_2 матриці B належить одному з рядків $\vec{b}_n$, $\vec{b}_{i_1}$, отже, $\alpha_{i_2} = 0$. Продовжуючи процес приєднання вершин (рядків): $\{x_n\}$, $\{x_n, x_{i_1}\}$, $\{x_n, x_{i_1}, x_{i_2}\}$, ..., $\{X\}$, ми послідовно одержимо нульові значення для всіх коефіцієнтів α_i у лінійній комбінації вектор-рядків матриці B_0 так, що

$$\operatorname{rang} B_0 = \operatorname{rang} B = n - 1. \quad ■$$

Матриці інциденцій і суміжності можна зв'язати цікавим співвідношенням, якщо ввести діагональну матрицю степенів вершин $\Delta = \{\delta_{ij}\}$ графа G, $\delta_{ij} = 0$ $(i \neq j)$, $\delta_{ii} = \delta(x_i)$ — степінь вершини x_i.

Теорема 6.5

Якщо G — орграф без петель, то його матриці суміжності A, інциденцій B і степенів вершин Δ зв'язані співвідношенням:

$$(A + A^T) + BB^T = \Delta. \qquad (6.1)$$

□ Елемент $\tilde{a}_{ij}$ матриці $A + A^T$ дорівнює числу Π_{ij} ребер, з'єднуючих вершини x_i, x_j; $\tilde{a}_{ii} = 2a_{ii} = 0$. Для матриці BB^T елемент з індексами i, j дорівнює $\sum_{k=1}^{m} b_{ik} b_{jk}$, що дає $(-\Pi_{ij})$ при $i \neq j$ і степінь $\delta(x_i)$ вершини x_i при $i = j$. ■

Наслідок 6.2

1. $\det(A + BB^T + A^T) = \prod_{i=1}^{n} \delta(x_i)$.

2. *Граф регулярний тоді і тільки тоді, коли матриця $(A + A^T + BB^T)$ скалярна, причому $A + A^T + BB^T = rI$, де $r = \delta(x_i)$ — степінь графа, I — одинична матриця.*

3. *Граф без петель є повним, якщо*

$$\det(A + A^T + BB^T) = (n-1)^n,$$

де $n = |X|$, *і навпаки.*

Завдання

1. Покажіть, що матриця інциденцій $B(R)$ визначає граф G з точністю до ізоморфізму.
2. Перевірити формулу (6.1) для конкретних значень матриць графа рис. 6.2:
   2.1. Для повного графа без петель і паралельних ребер
   $$(A + A^T)^2 = (n-1)(A + A^T) + BB^T.$$
   2.2. Для регулярного графа степеня r
   $$\det(A + A^T + BB^T) = r^n.$$
   Чи правильне зворотне (порівняй з п. 3 наслідку 6.2)?
   2.3. Ранг матриці інциденцій p-компонентного графа з n вершинами дорівнює $n - p$.

6.7. Розфарбування

1. Про проблему чотирьох фарб для плоских карт вже говорилося у п. 6.1. Настільки багатьох вона вводила у спокусу, що Френк Харарі писав у 1969 р.: «Гіпотезу чотирьох фарб можна на повній підставі назвати ще «хворобою чотирьох фарб», оскільки вона дуже схожа на захворювання. Вона дуже заразна. Іноді вона протікає порівняно легко, але у деяких випадках набуває затяжного або навіть загрозливого характеру. Ніяких щеплень проти неї не існує; правда, люди з достатньо здоровим організмом після короткого спалаху здобувають довічний імунітет. Цією хворобою людина може хворіти кілька разів, і вона супроводжується гострим болем, але жодного летального наслідку зареєстровано не було. Відомий, щонайменше, один випадок наслідування хвороби від батька до сина, тому, може бути, вона спадкоємна».

Порівняно недавно, у 1976 р., «хворобу чотирьох фарб» навчилися лікувати — гіпотеза підтвердилася, нарешті. Однак багатоступінчасте доведення цього факту є «річчю у собі», його настільки важко перевірити, що деякі спеціалісти сумніваються у тому, що гіпотеза вирішена. Етапи цього доведення такі. Спочатку, використовуючи ідею Г. Біркгофа про опис властивостей спеціальних «незвідних» (що не розфарбовуються чотирма фарбами) графів, математики звели число вершин

4-розфарбовних графів до 96. Потім у 1969 р. Х. Хеєш звів проблему чотирьох фарб до питання 4-розфарбування великого, але скінченного числа графів. Пізніше число таких графів було зведене до 1482. Нарешті, у 1976 р. К. Аппель і В. Хейкен (з колективом математиків і програмістів) правильно розфарбували всі 1482 графа за допомогою потужного комп'ютера, затративши на розфарбування близько 2000 годин машинного часу.

І все ж таки ця проблема, що так довго простояла, — лише окремий випадок цілого напрямку розфарбування графів. Перефразуємо задачу розфарбування країн зв'язного графа G у задачу *розфарбування вершин іншого плоского графа* $\bar{G}$ (не вимагаючи для загальності, щоб G був двозв'язним). Всередині кожної області Q_i (грані, країни) графа G, включаючи зовнішню, позначимо одну точку $\bar{x}_i$ — вершину шуканого графа $\bar{G}$.

Якщо у G країни Q_i і Q_j мають спільне ребро y, то з'єднаємо у $\bar{G}$ вершини $\bar{x}_i$ і $\bar{x}_j$ ребром $\bar{y}$, що перетинає тільки y. Граф $\bar{G}$ називається ***геометрично двоїстим*** до плоского графу G.

Наслідок. *Задача розфарбування країн географічної карти (областей плоского графа G) еквівалентна задачі розфарбування вершин геометрично двоїстого графа $\bar{G}$, також плоского.*

При цьому умова правильного розфарбування вершин полягає в тому, що суміжні (у $\bar{G}$) вершини $\bar{x}_i$, $\bar{x}_j$ набувають різних кольорів. Оскільки операція $G \to \bar{G}$ відображає весь клас зв'язних плоских графів на себе взаємно однозначно (це виходить з рівності $\bar{\bar{G}} = G$), то проблема чотирьох фарб переформулюється так:

будь-який плоский граф припускає правильне розфарбування вершин не більш, ніж чотирма фарбами.

Зауваження. Еквівалентність проблеми розфарбування країн і проблеми розфарбування вершин у класі всіх плоских графів можна встановити, не користуючись точною рівністю $\bar{\bar{G}} = G$. Важливо, що при переході до двоїстого графа $G \to \bar{G}$ розфарбування країн G переходить у розфарбування вершин $\bar{G}$, розфарбування вершин G — у розфарбування країн $\bar{G}$.

Довільний (не обов'язково плоский і зв'язний) граф $\bar{G}$ називається ***r-розфарбовним*** (*або* ***r-хроматичним***), якщо він припускає правильне розфарбування вершин r фарбами.

Не зменшуючи загальності, можна вважати, що при розфарбуванні вершин граф не містить паралельних ребер і петель, тобто є простим.

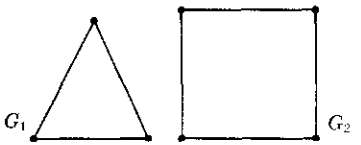


Рис. 6.17. 3-розфарбовний і 2-розфарбовний графи

Граф G_1 на рис. 6.17 3-розфарбовний, G_2 — 4, 3, 2-розфарбовний.

Мінімальне число кольорів, необхідне для правильного розфарбування вершин графа $G(\min r)$ називається ***хроматичним числом*** $\gamma(G)$.

Деякі практичні задачі пов'язані з розфарбуванням ребер графа G, яке вважається правильним, якщо кожні два суміжних ребра (інцидентних одній вершині) розфарбовуються у різні кольори. Найменше число кольорів, що припускає правильне розфарбування ребер, називається ***хроматичним класом*** $\gamma'(G)$ графа або ***індексом***. В принципі, розфарбування ребер графа G зводиться до розфарбування вершин деякого графа G^1, що називається ***дуальним***[1] за відношенням до G, тому $\gamma'(G) = \gamma(G^1)$ — хроматичний клас G дорівнює хроматичному числу дуального графа. Ця властивість дуального графа G^1 відповідним чином індукує його визначення: вершини дуального графа G^1 знаходяться у взаємно однозначній відповідності з ребрами G, і дві вершини у G^1 поєднані ребром, якщо (і тільки якщо) відповідні два ребра у G суміжні.

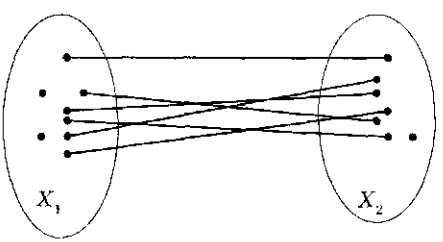


Рис. 6.18. Біхроматичність

2. Якщо $\gamma(G) = 1$, то граф G не має ребер (складається з ізольованих вершин; петлі у питаннях розфарбування вершин не враховуються). Будь-яке дерево має хроматичне число 2; граф з хроматичним числом $\gamma(G) = 2$ називається ***біхроматичним***. Будь-яке розфарбування G двома фарбами (рис. 6.18) розбиває вершини на дві однофарбні множини X_1, X_2, всередині кожної з яких вершини *попарно несуміжні* (*незалежні*, як іноді говорять). Ця обставина знаходить численні застосування на практиці і при різних доведеннях.

Нам знадобиться більш-менш очевидна лема.

---

[1] За іншою термінологією, граф G^1 називається реберним графом для G і позначається $L(G)$.

Лема 6.1

Будь-який замкнений маршрут

$$Q = \{x_0, y_1, x_1, y_2, x_2, \ldots, y_n, x_0\}$$

можна вичерпати (одержати порожній ланцюг) шляхом багатократного застосування таких операцій.

Ф. Видалення неперервно прохідного підмаршруту, що є простим циклом.

Ψ. Видалення тривіального замкненого підмаршруту довжини 2 виду $\{x_k, y_k, x_{k+1}, y_k, x_k\}$.

□ Дійсно, застосуємо до Q послідовно операцію Ф доти, поки це можливо. Оскільки підмаршрути Q_i ($\subset Q$), що залишилися, не входять до жодного циклу, то кожне ребро $q \in Q_i$ зустрічається в об'єднанні підмаршрутів $\bigcup_i Q_i$ однакове число разів у прямому і зворотному напрямках. Застосовуючи тепер послідовно операцію Ψ, одержимо порожній ланцюг. ■

Теорема 6.6 ***(Кеніг)***

Граф G є біхроматичним тоді і тільки тоді, коли всі його цикли мають парну довжину.

Зауваження. Наступні три властивості графа G еквівалентні.

1. Всі замкнені маршрути мають парну довжину.
2. Всі цикли мають парну довжину.
3. Всі прості цикли мають парну довжину.

Дійсно, імплікація $1 \Rightarrow 2 \Rightarrow 3$ очевидна, а $3 \Rightarrow 1$ випливає з леми, оскільки видалені у доведенні леми підмаршрути мають парну довжину.

□ Нехай граф G біхроматичний; реалізуємо будь-яке розфарбування двома кольорами (у кожній компоненті зв'язності). Кінці будь-якого ланцюга парної довжини мають однаковий колір, непарної — різні кольори, тому замкнені маршрути (цикли) непарної довжини відсутні. Зворотно, нехай всі замкнені маршрути графа G мають парну довжину. Пофарбуємо у колір α одну вершину x_0, потім у колір β — її оточення (множину суміжних вершин), у колір α — оточення вершин кольору β і так далі. За скінченне число кроків всі вершини графа пофарбуються у кольори α і β, причому розфарбування вершини x_i двічі означає, що між першим і другим пофарбуванням алгоритм розфарбування захопив замкнений маршрут з початком і кінцем x_i. Через парність його довжини обидва рази x_i пофарбована в один колір (α або β). ■

3. Охарактеризувати структуру *n-хроматичних* графів при $n \geq 3$ подібно до того, як це робить теорема Кеніга для біхроматичного графа, поки не вдається. Є багато часткових результатів, наприклад, для деяких класів регулярних графів, плоских графів з трикутними комірками (плоских тріангуляцій). Відзначимо, що хоча плоский (інакше — планарний) граф має хроматичне число $\gamma \leq 4$, планарність не є характеристичною ознакою 4-хроматичності графа: $K_{3,3}$ (рис. 6.12) біхроматичний ($\gamma = 2$), але не планарний.

Для одержання загальних оцінок хроматичного числа зручно ввести кілька визначень. ***Кліка*** — будь-який повний підграф графа G; ***клікове число*** (або ***щільність***) $\rho(G)$ — число вершин найпотужнішої кліки. ***Незалежна множина*** — будь-яка сукупність попарно несуміжних вершин; ***число незалежності*** $\alpha(G)$ — число вершин найпотужнішої множини незалежності. ***Додатковим до графа*** $G = (X, Y)$ називається граф $\tilde{G} = (X, \tilde{Y})$ з тією ж множиною вершин, що доповнює G до повного графа

$$G_0 = G + \tilde{G} = (X, Y \cup \tilde{Y}).$$

Приклади клік графа G (рис. 6.19) зображено на рис. 6.20, додатковий граф $\tilde{G}$ — на рис. 6.21.

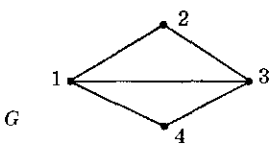


Рис. 6.19. Зв'язний граф G

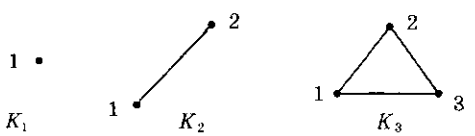


Рис. 6.20. Кліки графа G

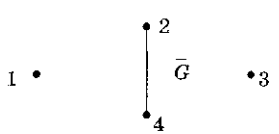


Рис. 6.21. Додатковий граф $\bar{G}$

Оскільки будь-яка множина попарно суміжних вершин при розфарбуванні вимагає кольорів за числом вершин, клікове число є нижньою оцінкою для хроматичного числа (6.2,*а*). Друга оцінка (6.2,*б*) виходить з того, що α співпадає з потужністю p_{i_0} найбільшої множини вершин, що припускає пофарбування одним кольором:

$$а)\ \gamma(G) \geq \rho(G); \qquad б)\ \gamma(G) \geq \left\lceil \frac{|X|}{\alpha(G)} \right\rceil. \tag{6.2}$$

Дійсно, нехай фіксоване деяке оптимальне розфарбування, V_i — однокольорові множини вершин, $|V_i| = p_i$. За побудовою $\bigcup_{i=1}^{\gamma} V_i = X$, $\sum_{i=1}^{\gamma} p_i = |X|$ і кожне V_i є незалежна множина. Ділення останньої рівності на α з урахуванням нерівностей $p_i \leq \alpha$ призводить до потрібної оцінки:

$$\gamma \geq \sum_{i=1}^{\gamma} \frac{p_i}{\alpha} = \frac{|X|}{\alpha} \geq \left\lceil \frac{|X|}{\alpha} \right\rceil.$$

Відзначимо без доведення більш грубу, ніж (6.2,*а*), але проте легше обчислювальну оцінку (Геллер Д. П., 1970 р.):

$$\gamma(G) \geq \frac{n^2}{n^2 - 2m}, \quad n = |X|,\ m = |Y|.$$

Цікаві оцінки суми та добутку хроматичних чисел γ і $\tilde{\gamma}$ графа G та його доповнення $\tilde{G}$, доведені Нордхаузом Е. А. і Гаддумом Ж. В. у 1956 р.:

$$2\sqrt{n} \leq \gamma + \tilde{\gamma} \leq n+1; \quad n \leq \gamma\tilde{\gamma} \leq \left(\frac{n+1}{2}\right)^2. \tag{6.3}$$

Очевидна також оцінка зверху: $\gamma \leq n + 1 - \alpha$. Нехай $\delta = \delta(G)$ найменший із степенів вершин графа G, $\Delta = \Delta(G)$ — найбільший. Справедливі такі витончені твердження.

1°. **Теорема Брукса (1941 р.).**

Якщо G — неповний зв'язний граф і $\Delta(G) \geq 3$, то $\gamma(G) \leq \Delta(G)$.

2°. *Для будь-яких натуральних чисел n, α, γ, що задовольняють нерівності $n/2 \leq \gamma \leq n + 1 - \alpha$, існує n-вершинний граф з числом незалежності α і хроматичним числом γ.*

4. Задачу розфарбування вершин графа G можна переформулювати у вигляді деякої задачі булевого програмування. Нехай $r \geq \gamma(G)$ — яка-небудь верхня оцінка хроматичного числа так, що граф G *r-розфарбовний*. Введемо матрицю булевих змінних розфарбування $T = \{t_{ij}\}$, $i = 1, \ldots, r$; $j = 1, \ldots, n$; $n = |X|$,

де для кожного розфарбування $t_{ij} = 1$, якщо вершина x_j пофарбована у i-й колір, і $t_{ij} = 0$ — у протилежному випадку. Якщо $R = \{r_{ij}\}$ — матриця інциденцій неорієнтованого графа G, то для добутку матриць TR елемент $(T \cdot R)_{ij} = \sum_{k=1}^{n} t_{ik} r_{kj}$ дорівнює: 0, якщо жоден з кінців ребра j не пофарбований у i-й колір; 1, якщо точно один кінець ребра j пофарбований у i-й колір; 2, якщо обидва кінця ребра j пофарбовані у i-й колір. Остання ситуація можлива тільки при неправильному розфарбуванні, тому умова правильності розфарбування виражається системою $r \cdot m$ нерівностей:

$$(T \cdot R)_{ij} \le 1; \quad 1 \le i \le r; \quad 1 \le j \le m = |Y|. \tag{6.4}$$

Умовою пофарбування кожної вершини x_j в один колір є

$$\sum_{i=1}^{r} t_{ij} = 1, \quad 1 \le j \le n. \tag{6.5}$$

Будь-яка булевська матриця T, що задовольняє обмеження (6.4, 6.5), забезпечує *r-розфарбування* G. Щоб одержати стандартну оптимізаційну задачу, необхідно додати функцію цілі, наприклад, мінімізувати частоту використання останнього (r-го) кольору:

$$\rho_r(T) = \sum_{j=1}^{n} t_{rj} \to \min. \tag{6.6}$$

Будь-який розв'язок T_0 задачі лінійного булевського програмування (6.4, 6.5, 6.6) дає правильне розфарбування G за допомогою $(r - 1)$ фарб, якщо $\gamma(G) < r$. В цьому випадку $\rho_r(T_0) = 0$ і можна зменшити число рядків матриці розфарбування T на одну, замінити у (6.4, 6.5, 6.6) r на $r - 1$ і розв'язати задачу заново. Продовжуючи такий процес, ми прийдемо до моменту, коли для розв'язку T чергової зменшеної задачі вперше одержиться ненульове значення цільової форми: $\rho_S(T) \ne 0$. Це буде означати, що $\gamma(G) = S$.

Існує спосіб завдання цільової форми, що одразу приводить до розфарбування з найменшим числом фарб $\gamma(G)$. Зіставимо кожному кольору i штраф p_i так, щоб $p_{i+1} > np_i$ $(n = |X|)$. Тоді цільова форма

$$Z(T) = \sum_{i=1}^{r} \sum_{j=1}^{n} p_i t_{ij} \to \min \tag{6.7}$$

з обмеженнями (6.4, 6.5) забезпечує розфарбування T_0 за допомогою $\gamma(G)$ фарб. Іншими словами, булевський розв'язок T_0 задачі (6.4, 6.5, 6.7) має нульові рядки $\gamma + 1, \ldots, r$.

Дійсно, нульовий i-й рядок у T_0 означає, що i-й колір не використовується, і якщо використовується колір $i + m$, то

переставлення двох рядків i, $i + m$ в T_0 приведе до розфарбування $\bar{T}_0$ з меншою формою (2.14): $Z(\bar{T}_0) < Z(T_0)$, що неможливо. Тому у T_0 нульові рядки стоять останніми. Далі нехай r_0 — найбільший номер ненульових рядків. Тоді $r_0 \geq \gamma = \gamma(G)$, оскільки для будь-якого правильного розфарбування треба не менше γ фарб. Припустимо, $r_0 \geq \gamma + 1$ і $\bar{T}$ — матриця правильного розфарбування з γ фарбами (хоча б одне таке розфарбування існує і задовольняє умови (6.4, 6.5)). Не зменшуючи загальності, можна вважати, що всі нульові рядки $\bar{T}$ мають номери $\gamma + 1$, ..., r. Очевидно,

$$Z(\bar{T}) = \sum_{i}^{\gamma}\sum_{j}^{n} p_i \bar{t}_{ij} \leq p_\gamma \sum_{j}^{n}\sum_{i}^{\gamma} \bar{t}_{ij} = p_\gamma \sum_{j}^{n} 1 = np_\gamma,$$

$$Z(T_0) = \sum_{i=1}^{r_0}\sum_{j=1}^{n} p_i t_{ij}^0 \geq p_{r_0} \geq p_{\gamma+1}.$$

За умови оптимальності розв'язку T_0 повинно бути $Z(T_0) \leq Z(\bar{T})$, звідки $p_{\gamma+1} \leq np\gamma$. Остання нерівність суперечить умові обрання штрафних коефіцієнтів, тому $r_0 = \gamma$.

Залежно від ситуації, замість $r \cdot m$ умов (6.4) можна використовувати $r \cdot n$ умов

$$L(1 - t_{ij}) - \sum_{k=1}^{n} t_{ik} a_{kj} \geq 0; \quad 1 \leq i \leq r; \quad 1 \leq j \leq n; \tag{6.8}$$

де L — будь-яке число, що більше n; $a_{k,j}$ — елементи матриці суміжності неорграфа G. При цьому, не зменшуючи загальності, можна вважати, що $0 \leq a_{k,j} \leq 1$, бо кратні ребра можна убрати з G, не змінюючи задачі розфарбування вершин.

Зазначимо, що при великих n і r у цільовій формі штрафні коефіцієнти мають занадто великий розкид за порядком значень; це створює незручності для обчислень і потребує особливих засобів для збереження належної точності. В сполученні з принциповими труднощами точного розв'язку задачі булевського програмування великої розмірності ми приходимо до невтішного висновку: для великих графів розфарбування і знаходження хроматичного числа методом зведення до задачі булевського програмування практично неефективні. Слід використовувати специфічні алгоритми, що враховують особливості геометрії графа. Більш того, якщо поставлена перед обчислювачем задача булевського програмування припускає інтерпретацію як задача розфарбування деякого графа G, то при великих розмірностях ефективніше шукати найоптимальніше розфарбування графа, а вже за його допомогою записувати розв'язок вихідної задачі програмування.

Для розфарбування не дуже великих графів застосовуються алгоритми перебору.

Для розфарбування великих графів, коли точні алгоритми виявляються недопустимо трудомісткими, використовуються алгоритми наближеного розфарбування, сімейство яких досить численне. Серед них є евристичні та випадково-пошукові алгоритми. При їх використанні слід проявляти відому обережність (наприклад, порівнювати результати розв'язку за кількома принципово різними алгоритмами), бо для кожного алгоритму, як правило, можна побудувати граф, для якого алгоритм дає довільно неправильну оцінку хроматичного числа.

5. Прикладні задачі, що зв'язані з розфарбуваннями, можуть бути в точності еквівалентними задачі розфарбування, але частіше містять ще додаткові обмеження.

5.1. *Задача завантаження (розміщення) n продуктів* (предметів) по ящикам (сховищам). Модельний граф G має n вершин (що відповідають продуктам), а наявність ребра (x_i, x_j) означає, що продукт x_i несумісний з x_j. Якщо місткості Q_i ящиків великі ($Q_i \geq n$), то задача розміщення у найменше число ящиків еквівалентна задачі оптимального розфарбування вершин графа G. При обмеженні місткості Q_i до звичайних умов розфарбування у позначеннях п. 6.4 додадуться обмеження

$$\sum_{j=1}^{n} t_{ij} \leq Q_i. \qquad (6.9)$$

5.2. В задачах *теорії розкладів* (інакше, *календарного планування*) операціям (оглядам) зіставляються тимчасові інтервали. Віднесемо їм вершини x_i модельного графа G, в якому є ребро (x_i, x_j) тоді і тільки тоді, коли операції x_i, x_j несумісні у часі. При однаковій довжині та довільній черговості операцій оптимальний розклад (сумарний час, що мінімізує виконання всіх робіт) еквівалентний оптимальному розфарбуванню модельного графа G. Хроматичне число дорівнює оптимальному часу виконання всіх робіт: $\gamma(G) = T_{\min}$. Крім того, різні кольори можуть відповідати, наприклад, різним ділянкам (приміщенням), і якщо на i-й ділянці не можна виконати більше Q_i операцій одночасно, то в модельному розфарбуванні з'являється додаткове обмеження (6.9).

Типовою задачею цього типу є задача складання розкладу занять. Нехай, наприклад, треба прочитати кілька лекцій (кожна по одній годині) за найкоротший час. Число груп слухачів дорівнює

числу лекцій, але все ж деякі лекції не можна читати одночасно (наприклад, їх читає один і той же лектор). Оптимальний розклад еквівалентний мінімальному розфарбуванню такого графа G, у якого вершини відповідають лекціям, а суміжність вершин означає, що відповідні лекції не можна читати одночасно.

Неважко скоректувати будь-який з алгоритмів оптимального розфарбування так, щоб враховувалися обмеження (6.9).

Завдання

1. Довести, що число вершин n регулярного біхроматичного графа парне, а множини однокольорності X_1, X_2 (рис. 6.18) мають однакові числа вершин:
$$|X_1| = |X_2| = \alpha(G) = n/2.$$
2. Довести, що:
а) для правильного розфарбування карти, що одержується при перетині кіл на площині, достатньо двох кольорів;
б) будь-яка карта, що не має вершин степеня 2, має грань, границя якої містить не більше п'яти ребер.

6.8. Дерева

Визначення

Зв'язний граф T без циклів називається ***деревом***. Орієнтація може не враховуватися, і тоді говорять про ***ребра дерева*** y_j. Якщо орієнтація враховується, то говорять про ***дуги дерева***, а саме дерево називається ***орієнтованим***, наприклад, дерево логічних можливостей, генеалогічне дерево.

Крім наведеного визначення, можна дати ще кілька еквівалентних визначень дерева.

Теорема 6.7

Для графа G з n вершинами і m ребрами рівносильні такі властивості:

1. *G зв'язний і не має циклів.*
2. *G зв'язний, і число його вершин на одиницю перевершує число ребер: $n = m + 1$.*
3. *G не містить циклів і $n = m + 1$.*
4. *G не містить циклів, але додавання ребра між будь-якими двома несуміжними вершинами призводить до появи циклу.*

5. *G зв'язний, але втрачає цю властивість при видаленні будь-якого ребра.*
6. *Будь-які дві вершини $x_i \neq x_k$ графа G з'єднує єдиний ланцюг.*

Якщо одну з властивостей 2–6 виконано, то граф G є дерево.

Доведення теореми 6.7 або її частин міститься у багатьох книгах з теорії графів. Оскільки воно не потребує спеціальної техніки і досить елементарне, надамо його читачеві як вправу.

Наслідок 6.3. *Будь-яке дерево з числом вершин $n \geq 2$ має, щонайменше, дві висячі (кінцеві) вершини.*

Дійсно, за лемою про рукостискання (п. 6.4) сума степенів вершин є $\sum_{i=1}^{n} \delta(x_i) = 2(n-1)$. Оскільки $\delta(x_i) \geq 1$ (ізольовані вершини відсутні), то хоча б дві вершини x_1, x_2 мають степінь $\delta(x_1) = \delta(x_2) = 1$, у протилежному випадку $\delta(x_i) \geq 2$ для всіх i, так що

$$\sum_{i=1}^{n} \delta(x_i) \geq 2n. \qquad \blacksquare$$

В розділі 6.2 на рис. 6.6 було наведено приклад дерева G з числом вершин $n = 21$, у якому центр складався з двох вершин $\{u, v\}$. Для простого ланцюга P_n центр складається з однієї вершини, якщо число вершин n непарне, і відповідно — з двох вершин, якщо n парне (рис. 6.22).

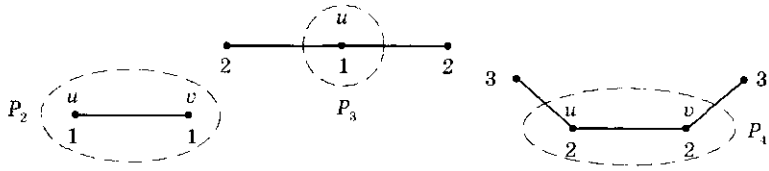


Рис. 6.22. Ексцентриситети вершин і центри ланцюгів P_2, P_3, P_4

Виявляється, для будь-якого дерева справедлива.

Теорема 6.8

Центр будь-якого дерева складається з однієї або двох суміжних вершин.

□ Для дерев P_2, P_3 з числом вершин $n = 2$, $n = 3$ твердження справедливе (див. рис. 6.22). Нехай T — дерево з n вершинами, $n > 3$. Якщо видалити з T всі висячі ребра з кінцевими вершинами, то одержане дерево T_0 ($\subset T$) має той же центр, що і дерево T. Тепер теорема виводиться за індукцією. ■

Перелічення графів і дерев: неізоморфних, кореневих і позначених

Задачі перелічення всіх «різних» (у тому чи іншому розумінні) графів із заданим числом вершин n є важкими. Вони виникали і розв'язувалися у чистому вигляді або з додатковими обмеженнями у математиці, хімії, фізиці, проектуванні та різних інженерних проблемах. Зрозуміло, що перелічення дерев з n вершинами утворює, з одного боку, більш простий, з другого боку, — базовий клас задач.

Граф $G = (X, Y)$ називається ***позначеним***, якщо його вершинам привласнені фіксовані *позначки*, наприклад, номери $1, 2, ..., n$. Два позначених графа ***однакові (не різняться)***, якщо їхні вершини позначені однією системою позначок та існує ізоморфізм (див. р. 10.4) одного графа на інший, при якому зберігаються позначки всіх вершин. На рис. 6.23 зображено всі позначені графи з числом вершин $n = 3$; їх кількість дорівнює 8.

Кількість G_{nm} (неорієнтованих) позначених (n, m) — графів (простих з n вершинами і m ребрами) дорівнює, очевидно, числу сполучень з множини різних неорієнтованих пар вершин $\{(\overline{x_i, x_j})\}$ за числом ребер m. Оскільки число вказаних пар вершин дорівнює числу сполучень C_n^2 (з n по 2), то (див. п. 10.3)

$$G_{nm} = C_{C_n^2}^m = C_s^m, \quad s = C_n^2 = \frac{n(n-1)}{2}. \tag{6.11}$$

Сумуючи числа G_{nm} за всіма можливими кількостями ребер $m = 1, 2, ..., \frac{n(n-1)}{2}$ від випадку безреберного графа до випадку повного графа з n вершинами, одержуємо *число* G_n *всіх позначених графів з n вершинами*:

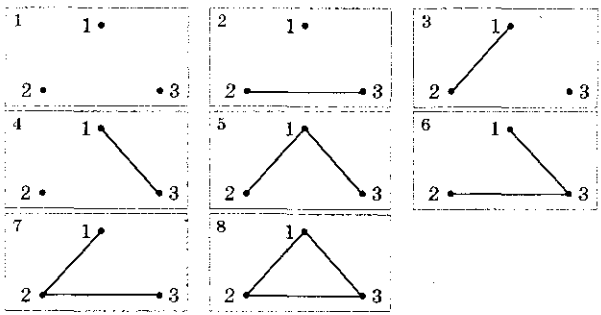


Рис. 6.23. Позначені графи з трьома вершинами

$$G_n = \sum_{m=0}^{s} C_s^m = 2^s, \quad s = C_n^2 = \frac{n(n-1)}{2}. \qquad (6.12)$$

Зображення суми в (6.12) у вигляді $2^{C_n^2}$ маємо з формули (10.14). Зокрема, $G_3 = 2^{C_3^2} = 2^3 = 8$ (див. рис. 6.23), $G_4 = 2^{C_4^2} = 2^{\frac{4\cdot 3}{1\cdot 2}} = 2^6 = 64$.

Число g_n неізоморфних графів без позначок (простих, неорієнтованих) знайти значно трудніше. Серед восьми графів рис. 6.23 без урахування позначок можна вказати тільки чотири попарно неізоморфних один одному, наприклад, у квадратах 1, 2, 7, 8. Отже, $g_3 = 4$, що вдвічі менше числа G_3 позначених графів. Число G_4 позначених чотиривершинних графів дорівнює 64, в той час, як різних неізоморфних чотиривершинних графів без позначок існує всього 11 = g_4 (див. рис. 6.24).

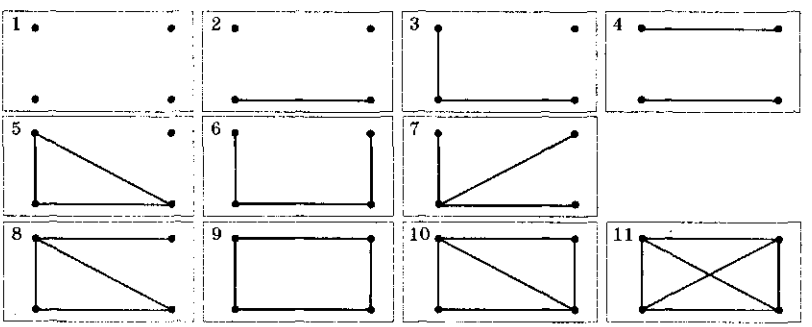


Рис. 6.24. Чотиривершинні неізоморфні графи

Якщо через g_{nm} позначити *число неізоморфних n-вершинних графів з m ребрами*, то

$$g_n = \sum_{m=0}^{s} g_{nm}, \quad s = C_n^2 = \frac{n(n-1)}{2}. \qquad (6.13)$$

Згідно з рис. 6.24 для $n = 4$ маємо: $s = 6$; $g_{4,0} = 1$ (квадрат 1), $g_{4,1} = 1$ (квадрат 2), $g_{4,2} = 2$ (квадрати 3, 4), $g_{4,3} = 3$ (квадрати 5, 6, 7), $g_{4,4} = 4$ (квадрати 8, 9), $g_{4,5} = 1$ (квадрат 10), $g_{4,6} = 1$ (квадрат 11). За формулою (6.13) $g_4 = 1 + 1 + 2 + 3 + 2 + 1 + 1 = 11$.

В загальному випадку кількості неізоморфних графів g_{nm}, g_n знаходяться за допомогою теорії перелічення конфігурацій, що створена Д. Пойа. З її допомогою підраховано, наприклад, що $g_{9,5} = 25$, $g_{9,7} = 148$, $g_{9,9} = 771$, $g_{9,15} = 21\,933$, $g_{9,18} = 34\,040$; і нарешті, число всіх графів з 9-ма вершинами дорівнює $G_9 = 308\,168$.

Перейдемо до *переліченння дерев*, позначивши через τ_n *число позначених дерев* і через t_n *число звичайних*[1] *(неізоморфних) дерев* з n вершинами. На рис. 6.23 (квадрати 5, 6, 7) видно, що $\tau_3 = 3$, $t_3 = 1$. Далі $t_4 = 2$ (рис. 6.24, квадрати 6, 7) і як видно з рис. 6.25, число позначених чотиривершинних дерев дорівнює $16 = \tau_4$.

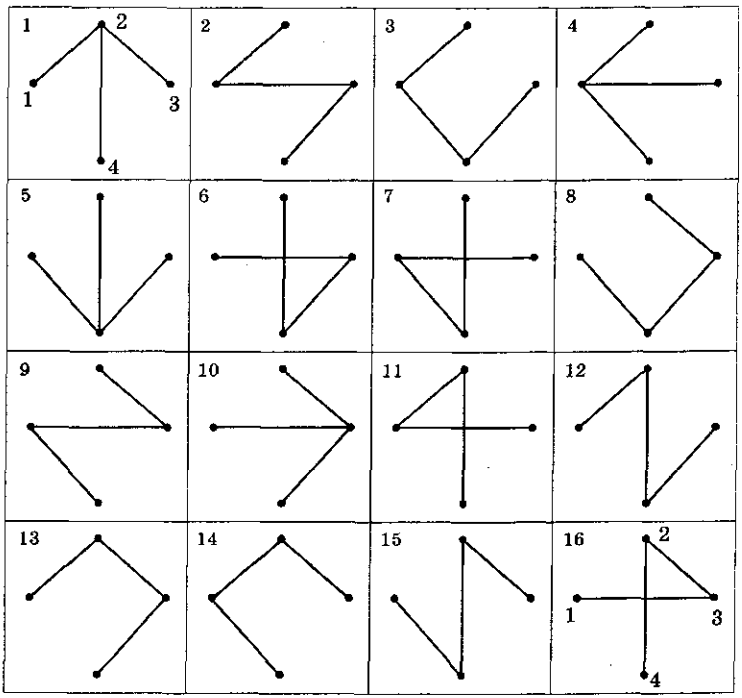


Рис. 6.25. Позначені чотиривершинні графи

Формула А. Келі (1897 г.). *Число* τ_n *позначених дерев з* n *вершинами дорівнює* n^{n-2}

$$\tau_n = n^{n-2}. \qquad (6.14)$$

Доведення наводиться нижче (див. наслідок 6.4 з теореми 6.9).

Наприклад, $\tau_2 = 2^{2-2} = 1$, $\tau_3 = 3^{3-2} = 3$, $\tau_4 = 4^{4-2} = 16$ (рис. 6.25), $\tau_5 = 5^{5-2} = 125$, $\tau_6 = 6^4 = 1296$, $\tau_7 = 7^5 = 16807$.

[1] Дерева без будь-яких позначок іноді називаються *вільними деревами*.

Числа t_n звичайних (неізоморфних) дерев є значно меншими, однак обчислити їх істотно важче. Сучасні алгоритми знаходження значень t_n і одержання конфігурацій неізоморфних дерев з n вершинами засновані на теорії перелічення Д. Пойа.

Важливий клас графів утворюють дерева з однією позначеною вершиною, яка називається ***коренем***. Само дерево з однією позначеною («виділеною») вершиною називається ***кореневим деревом***. Число кореневих дерев з n вершинами позначається через T_n; зрозуміло, що

$$t_n \le T_n \le \tau_n. \qquad (6.15)$$

Всі кореневі дерева з числом вершин $n = 3$ і $n = 4$ зображено на рис. 6.26.

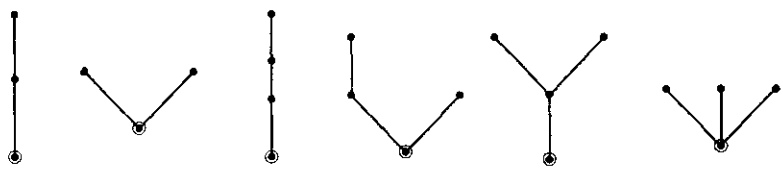


Рис. 6.26. Кореневі дерева з трьома і чотирма вершинами

В таблиці 6.1 наведено значення чисел t_n неізоморфних дерев, T_n кореневих дерев і τ_n позначених дерев з числами вершин n від 1 до 10.

Таблиця 6.1. Значення t_n, T_n, τ_n

Число вершин n	1	2	3	4	5	6	7	8	9	10
Число дерев t_n	1	1	1	2	3	6	11	23	47	106
Число кореневих дерев T_n	1	1	2	4	9	20	48	115	286	719
Число позначених дерев τ_n	1	1	3	16	125	1296	16 897	262 144	4 782 969	10^8

На рис. 6.27 зображено всі неізоморфні дерева з п'ятьма вершинами ($t_5 = 3$).

Рис. 6.27. Неізоморфні дерева з п'ятьма вершинами

Остови (каркаси) графа

Остовним підграфом (коротко — ***остовом***) або ***каркасом графа*** $G = (X, Y)$ називається підграф $H = (X, Y_H)$ *без циклів*, що містить *всі вершини* X і *максимально велике число ребер*. Якщо граф G — зв'язний, то остов H є *дерево*, якщо G містить p компонент зв'язності, то остов (каркас) H складається з p *дерев*. Два остови у G вважаються різними, якщо вони відрізняються хоча б одним ребром, тобто як графи. Для того щоб G *мав* більше одного остова, необхідно і достатньо існування хоча б одного циклу в G.

Задача визначення *числа всіх остовів графа* G і їх фактичного конструювання є важливою для застосувань. Зрозуміло, її достатньо розв'язати для зв'язного G. Для *повного (простого) графа* G *перелічення остовів є перелічення всіх позначених дерев з* $n = |X|$ вершинами, так що число остовів дорівнює n^{n-2} (див. (6.14), теорема Келі). У загальному випадку число остовів одержав Кірхгоф у 1847 р. ***Матрицею Кірхгофа*** $K(G)$ простого графа G називається $n \times n$ — матриця з елементами

$$k_{i,j} = \begin{cases} -1, \text{ якщо вершини } x_i, x_j \text{ суміжні;} \\ 0, \text{ якщо вершини } x_i, x_j \text{ не суміжні;} \\ \delta_i = \deg x_i, \text{ якщо } i = j. \end{cases}$$

Сума елементів у кожному рядку (і кожному стовпці) матриці $K(G)$ *дорівнює нулю. У будь-якої квадратної матриці з такою властивістю рядків і стовпців, а не тільки у матриці Кірхгофа* $K(G)$, *алгебраїчні доповнення* Δ_{ij} *всіх елементів* k_{ij} *рівні між собою:* $\Delta_{ij} = \Delta$.

У матриці ж $K(G)$ величина Δ дорівнює числу остовних дерев.

Теорема 6.9 ***(Кірхгоф)***

Число остовних дерев у простому зв'язному графі G *з* n *вершинами* ($n \geq 2$) *дорівнює алгебраїчному доповненню будь-якого елемента матриці Кірхгофа* $K(G)$.

Доведення спирається на вираження визначника за формулою Біне — Коші.

Наслідок 6.4. *Справедлива формула Келі (6.14) для числа позначених дерев з* n *вершинами* $\tau_n = n^{n-2}$.

□ Всі позначені дерева з n вершинами утворюють в точності множину остовних дерев *повного* графа K_n. За теоремою 6.9

їх число дорівнює алгебраїчному доповненню Δ_{11} елемента k_{11} матриці Кірхгофа

$$K(K_n) = \begin{pmatrix} n-1 & -1 & -1 & \dots & -1 \\ -1 & n-1 & -1 & \dots & -1 \\ . & . & . & \dots & . \\ -1 & -1 & -1 & \dots & n-1 \end{pmatrix}$$

розміру $n \times n$. Визначник порядку $(n-1)$

$$\Delta_{11} = \begin{vmatrix} n-1 & -1 & -1 & \dots & -1 \\ -1 & n-1 & -1 & \dots & -1 \\ \dots & \dots & \dots & \dots & \dots \\ -1 & -1 & -1 & \dots & n-1 \end{vmatrix}$$

зручно перетворити, замінивши спочатку перший рядок сумою всіх рядків, а потім, додаючи одержаний рядок з одиниць до рядків з номерами 2, 3, ..., $n-1$:

$$\Delta_{11} = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ . & . & . & \dots & . \\ -1 & n-1 & -1 & \dots & -1 \\ . & . & . & \dots & . \\ -1 & -1 & -1 & \dots & n-1 \end{vmatrix} = \begin{vmatrix} 1 & . & 1 & 1 & \dots & 1 \\ . & . & . & . & \dots & . \\ 0 & . & n & 0 & \dots & 0 \\ . & . & . & . & \dots & . \\ 0 & . & 0 & 0 & \dots & n \end{vmatrix} = n^{n-2}. \blacksquare$$

Алгоритм побудови остовного дерева. Пошук у глибину

Задача побудови якого-небудь (одного!) остовного дерева у графі $G = (X, Y)$ належить до числа найважливіших для практики і, на щастя, є найпростішою для алгоритмізації і комп'ютерної реалізації. Зрозуміло, що основна ідея побудови повинна полягати у послідовному відборі ребер, що не утворюють цикли з попередніми. Припустимо, побудову остовного дерева $T = (X, Y_T)$ закінчено. Тоді у його процесі було переглянуто всі n вершин і відібрано $(n-1)$ ребро графа G, $n-1 = |Y_T|$. Незрозуміло, скільки всього ребер буде потрібно переглянути, відбираючи ребра Y_T без циклів; можливо, у гіршому випадку всі ребра множини Y.

Враховуючи останнє зауваження, ми спочатку наведемо так званий алгоритм «пошуку у глибину», що переглядає по

одному разу всі ребра графа G і звичайно всі його вершини. Зауважимо, що пошук у глибину (скорочено ПГ) — не єдиний метод перегляду всіх вершин і ребер графа G. Наприклад, часто використовується перегляд графа «пошуком у ширину», при якому у кожній черговій вершині переглядаються всі інцидентні їй ребра без виключення і всі їх кінцеві вершини (тобто «оточення» вершини X). Метод пошуку в ширину фактично використовувався нами у розділі 6.7 при розфарбуванні біхроматичного графа у доведенні теореми Кеніга.

Якісний опис алгоритму пошуку в глибину, позначення

В процесі пошуку в глибину вершинам графа G послідовно привласнюються нові номери (ПГ-номери) від 1 до n, а ребра одержують позначки двох класів: «пряме ребро» і «зворотне ребро». Пошук починається з довільної вершини $v_0 \in X$, якій привласнюється ПГ-номер1: ПГ $(v_0) = 1$. Далі обирається будь-яке інцидентне до v_0 ребро (v_0, v), позначається позначкою «пряме ребро» і вершині v привласнюється ПГ-номер 2. Наступний (третій) крок пошуку починається у вершині v, в якій ПГ $(v) = 2$, і підкоряється рекурентній процедурі, яку ми опишемо для довільного кроку $(k + 1)$. Припустимо, k-й крок виконаний і деяка вершина x одержала ПГ-номер ПГ $(x) = k$.

Можливі дві ситуації:

1. У вершині x існує інцидентне непозначене ребро (x, w). Якщо вершина w вже має ПГ-номер, то ребро (x, w) одержує позначку «зворотне», і продовжується пошук непозначеного ребра, що інцидентне вершині x. Якщо ж вершина w не має раніше привласненого ПГ-номеру, то їй привласнюється черговий ПГ-номер ПГ $(w) = k + 1$, ребро (x, w) одержує позначку «пряме», і пошук переміщується у вершину w.
2. Всі ребра, що інцидентні вершині x, позначені на попередніх кроках. Тоді пошук повертається у вершину $\bar{x}$, що має попередній ПГ-номер ПГ $(\bar{x}) = k - 1$.

Пошук у глибину закінчується, коли всі ребра графа G виявляться позначеними.

Позначимо Y_+ — множина прямих ребер, Y_- — множина зворотних ребер. Із процедури пошуку в глибину виходить твердження.

Твердження. Якщо $G = (X, Y)$ — зв'язний граф, то підграф $T = (X, Y_+)$ є остовне дерево в G, а підграф (X, Y_-)

є додатковим до дерева T. Дерево T є орієнтованим кореневим з коренем v_0.

Перейдемо до формалізованого опису алгоритму пошуку в глибину. Попередньо інформацію про граф G зручно зобразити у вигляді списків суміжності $\{N_x\}_{x \in X}$, де N_x — список вершин w, інцидентних вершині x. В процесі пошуку формується динамічний список S вершин, в який кожна вершина x включається в точності один раз у момент привласнення їй ПГ-номера, і, можливо, виключається з S у момент повернення з x у вершину з попереднім ПГ-номером. Список S організується як стек, у якому включення і виключення вершин відбувається справа. Використовуються такі змінні: k — останній привласнений ПГ-номер; p — покажчик кінця стека S, так що $S(p)$ — ім'я останньої вершини стека S. Алгоритм формує три списка, що розширюються: список вершин SПГ, що одержали на цей момент ПГ-номера; список «прямих» ребер SY_+; список SY_- «зворотних» ребер.

Алгоритм пошуку в глибину у неорієнтованому зв'язному графі $G = (X, Y)$

1. Обрати довільну вершину $v_0 \in X$. Покласти ПГ(v_0):= 1, $S(1) := v_0$, $S\text{ПГ} := \{v_0\}$, $SY_+ := \{\varnothing\}$, $SY_- := \{\varnothing\}$.
2. Обрати досліджувану вершину $x := S(p)$.
3. Перегляд списку суміжності N_x. Знайти таку вершину $w \in N_x$, що ребро (x, w) не позначено, і перейти до кроку 4. Якщо такої вершини у списку N_x немає, то перейти до кроку 5.
4. 4.1. Якщо вершина w вже має ПГ-номер, то позначити ребро (x, w) позначкою «зворотне» і занести у кінець списку SY_-. Перейти до кроку 3 і продовжити перегляд списку суміжності N_x.

   4.2. Якщо вершина w ще не має ПГ-номера, то привласнити їй черговий ПГ-номер: $k := k + 1$, ПГ $(w) = k + 1$. Ребро (x, w) забезпечити позначкою «пряме» і занести у кінець списка SY_+; покласти $p := p + 1$, $S(p) := w$. Перейти до кроку 2.
5. Якщо для списку суміжності N_x, що переглядається на кроці 3, кожна вершина $w \in N_x$ з'єднується з вершиною x вже позначеним ребром (x, w), то повернутися від вершини x до вершини з попереднім ПГ-номером і покласти:

   $p := p - 1$ (видалення вершини x із стека S).

   Якщо $p = 0$, пошук в глибину завершується. Якщо $p \neq 0$, перейти до кроку 2.

6. Пошук у глибину завершується, коли всі ребра Y одержали позначки, тобто коли $SY_+ \cup SY_- = Y$. Тоді

$$Y_+ = SY_+,\ Y_- = SY_-,\ S\Pi\Gamma = X,\ T = (X, Y_+) —$$

остовне дерево графа G.

Зауваження. Остовне дерево T могло бути сформовано, взагалі кажучи, до закінчення перегляду всіх ребер Y, саме, у той момент, коли число прямих ребер досягло значення $n - 1$: $|SY_+| = n - 1$.

Яка *трудомісткість алгоритму пошуку в глибину*? Оскільки кожна з n вершин графа G не більш ніж два рази обробляється у стеку S (один раз включається і, можливо, один раз виключається), то робота зі стеком займає $\theta(n)$ часу[1].

Сумарне виконання кроку 4 алгоритму потребує $\theta(m)$ часу, де m — число ребер графа G (див. розділ 8). Сумарне виконання кроку 3 також можна обмежити часом $\theta(m)$: достатньо організувати додатковий масив із змінною $q = q(x)$, що вказує номер першої непереглянутої вершини у списку суміжності N_x, і починати черговий перегляд на кроці 3 з вершини $q(x)$. Отже, трудомісткість всього алгоритму становить $\theta(n + m)$. Це найкращий можливий результат, оскільки всі $n + m$ вершин і ребер повинні бути переглянуті. Про такі алгоритми кажуть, що вони мають *лінійну складність*: в оцінці часу для найгіршого варіанту обчислень число всіх «вхідних даних» алгоритму (тут це $n + m$) помножується на постійну, що не залежить від кількості даних.

Проілюструємо роботу алгоритму на графі (рис. 6.28). Для зручності первинні імена вершин не наводяться, а вказуються вже їхні ПГ-номера, які одержуються послідовно у процесі роботи алгоритму. Прямі ребра позначаються індексом «п» і однією стрілкою на суцільний лінії, зворотні ребра — індексом «0» і стрілками на пунктирній лінії. Номера ребер привласнюються у порядку їх проходження. В таблиці 6.2 наведено 15 кроків пошуку

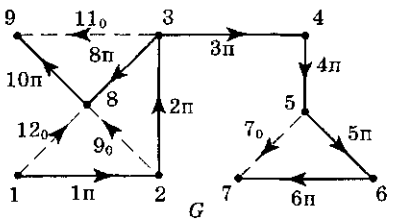


Рис. 6.28. Алгоритм пошуку в глибину

[1] Тобто при великих n витрати часу на стек S не більш ніж cn, де постійна c не залежить від n.

в глибину, на кожному кроці вказується повністю весь стек вершин S, весь масив даних SY_+, весь масив SY_- і позначка нового ребра — якщо таке є. Зупинка — після кроку 15. В рядку 15 масив SY_+ дає всю множину Y_+ ребер остовного дерева T, масив SY_- — множину Y_- ребер кодерева, хоча фактично Y_+ було одержане ще на кроці 13. Граф заданий списками суміжності N_k — множинами вершин, суміжними k-й вершині: (k = 1, 2, ..., 9): N_1 = {2, 8}, N_2 = {1, 3, 8}, N_3 = {2, 4, 8, 9}, N_4 = {3, 5}, N_5 = {4, 6, 7}, N_6 = {5, 7}, N_7 = {5, 6}, N_8 = {1, 2, 3, 9}, N_9 = {3, 8}.

Таблиця 6.2. Пошук у глибину для графа G рис. 6.28

№	S — стек вершин	Позначка ребра	Масив SY_+	Масив SY_-
1	1	1п	1п	—
2	1, 2	2п	1, 2 – п	—
3	1, 2, 3	3п	1, 2, 3 – п	—
4	1, 2, 3, 4	4п	1, 2, 3, 4 – п	—
5	1, 2, 3, 4, 5	5п	1, 2, 3, 4, 5 – п	—
6	1, 2, 3, 4, 5, 6	6п	1, 2, 3, 4, 5, 6 – п	—
7	1, 2, 3, 4, 5, 6, 7	7_0	1, 2, 3, 4, 5, 6 – п	7_0
8	1, 2, 3, 4, 5, 6	—	1, 2, 3, 4, 5, 6 – п	7_0
9	1, 2, 3, 4, 5	—	1, 2, 3, 4, 5, 6 – п	7_0
10	1, 2, 3, 4	—	1, 2, 3, 4, 5, 6 – п	7_0
11	1, 2, 3	8п	1, 2, 3, 4, 5, 6, 8 – п	7_0
12	1, 2, 3, 8	9_0	1, 2, 3, 4, 5, 6, 8 – п	7_0, 9_0
13	1, 2, 3, 8	10п	1, 2, 3, 4, 5, 6, 8, 10 – п	7_0, 9_0
14	1, 2, 3, 8, 9	11_0	1, 2, 3, 4, 5, 6, 8, 10 – п	7_0, 9_0, 11_0
15	1, 2, 3, 8	12_0	1, 2, 3, 4, 5, 6, 8, 10 – п	7_0, 9_0, 11_0, 12_0
	Зупинка			

Зауваження. Якщо вихідний граф G не є зв'язним, то алгоритм пошуку в глибину перегляне тільки одну з компонент графа і побудує остовне дерево цієї компоненти. Зрозуміло, як модифікувати алгоритм, щоб пошук перейшов до нової компоненти зв'язності і далі переглянув послідовно всі компоненти, побудувавши, зокрема, остовні дерева у всіх компонентах, тобто *остовний ліс графа G*.

Алгоритм побудови остовного дерева шляхом довільного перегляду ребер

Розглянемо ще один алгоритм побудови остовного дерева, в якому ребра графа G проглядаються у довільному порядку. В кожному кроці алгоритму множина ребер Y розбивається на три підмножини: Y_α — пофарбовані α-кольором, Y_β — пофарбовані β-кольором, Y_0 — непофарбовані. До початку роботи алгоритму $Y_\alpha = \varnothing$, $Y_\beta = \varnothing$, $Y_0 = Y$; після зупинки — $|Y_\alpha| = n - 1$, і досягнення цієї рівності є одною з двох можливих умов зупинки. Шукане остовне дерево T будується з α-ребер поступово, шляхом поповнення зв'язних піддерев, які іноді називаються «*букетами*». Поповнення α-букетів робиться доти, доки це можливо без появи α-циклів або досягається умова $|Y_\alpha| = n - 1$. Якщо $|Y_\alpha| < n - 1$, а фарбування α-кольором будь-якого ребра з Y_0 призводить до появи α-циклу, то алгоритм зупиняється, *граф G не зв'язний, остовне дерево не існує*, і можна скоректувати алгоритм для побудови остовного лісу.

Формальний опис алгоритму

Крок 1. Обрати довільне ребро y_1, позначивши його α-кольором і сформувати перший α-букет, приєднавши до ребра y_1 пару інцидентних йому вершин.

Крок 2. Обрати будь-яке непофарбоване ребро y_2 і зупинитися, якщо такого немає (в останній ситуації не існує остовного дерева в графі G з числом вершин $n > 2$). Для обраного ребра y_2 можливі чотири ситуації.

2.1. Обидві кінцеві вершини ребра y_2 належать одному α-букету. Тоді пофарбувати ребра y_2 кольором β і повернутися на початок кроку 2.

2.2. Одна з кінцевих вершин ребра y_2 належить деякому α-букетові B, інша — не належить жодному з α-букетів. Тоді y_2 пофарбувати α-кольором і поповнити букет B ребром y_2 разом з інцидентною кінцевою вершиною.

2.3. Обидві кінцеві вершини ребра y_2 не належать жодному із сформованих букетів. Тоді пофарбувати y_2 кольором α і сформувати новий букет з y_2 і двох його кінцевих вершин.

2.4. Обидві кінцеві вершини ребра y_2 належать двом різним букетам. У цьому випадку пофарбувати y_2 кольором α і злити два згаданих букета до одного.

Крок 3. Якщо всі вершини графа G увійшли до одного букету (це еквівалентне умові $|Y_\alpha| = n - 1$), закінчити процедуру, в цьому випадку підграф $T = (X, Y_\alpha)$, що складається з усіх вершин і всіх α-ребер, є остовним деревом графа G. В протилежному випадку — повернутися на початок кроку 2.

Остів мінімальної ваги

Зваженим графом називається граф $G = (X, Y)$, кожному ребру якого приписана *вага* — додатне число $c_i = c(y_i) > 0$ (іноді $c_i \geq 0$). ***Вагою підграфа*** з G називається сума ваг ребер підграфа. Задача відшукання *остова найменшої ваги* у графі G часто зустрічається у застосуваннях: при проектуванні комп'ютерних і кабельних мереж, ліній електропостачання, трубопроводів, доріг тощо. При математичному моделюванні цих реальних ситуацій роль вершин графа G відіграють відповідно комп'ютери, комп'ютерні центри, абоненти і передавачі (у кабельних мережах), джерела і споживачі електроенергії, води, газу і т. д. Роль ребер y_i і ваг c_i зваженого графа G відіграють припустимі шляхи прокладки кабелів, труб, доріг та їх вартості. Якщо зв'язний граф G «припустимих трасувань» близький до повного графа K_n, то число всіх остовних дерев дорівнює n^{n-2}, і прямий перебір всіх дерев і обчислень їх ваг може мати обтяжуючий об'єм обчислень. Наприклад, число остовів τ_n для K_n при $n = 22$ має нижню оцінку $\tau_{22} = 22^{20} > 20^{20} = 2^{20} \cdot 10^{20} > 10^{25}$. Разом з тим, для знаходження остова найменшої ваги є ціла низка ефективних алгоритмів. Опишемо два з них, що стали вже класичними.

Алгоритм Дж. Краскала знаходження остова мінімальної ваги (1956 р.)

Ідея алгоритму полягає в тому, що *на кожному кроці алгоритму обирається найкоротше ребро, що не утворює цикл з обраними раніше ребрами.*

1 *крок*. В вихідному графі $G = (X, Y)$ будується підграф $T_1 = (X, Y_1)$ з одним ребром $u_1 = y_1$ мінімальної ваги $c(u_1) = \min\limits_{\forall y_i \in Y} \{c(y_i)\}$.

Крок k. Якщо граф $T_{k-1} = (X, Y_{k-1})$ без циклів вже побудований, обираємо ребро $u_k \in Y \backslash Y_{k-1}$ так, щоб воно не утворювало циклу з ребрами Y_{k-1} і мало найменшу вагу серед всіх таких ребер. Будуємо граф T_k, приєднуючи до графа T_{k-1} ребро u_k; $T_k = (X, Y_k)$, $Y_k = Y_{k-1} \cup \{u_k\}$ $k = 2, 3, \ldots, n - 1$.

Алгоритм Р. Прима побудови остова мінімальної ваги (1957 р.)

На кожному кроці алгоритму Прима будується дерево — зв'язний підграф на підмножині X_k вершин у G, в той час, як в алгоритмі Краскала послідовні підграфи T_k без циклів містили всю множину вершин X графа G, і тому при $k < n - 1$ графи T_k були незв'язними. В іншому алгоритми схожі один на одного.

Крок 1. Обираємо у $G = (X, Y)$ ребро e_1 з мінімальною вагою $c(e_1) = \min\limits_{\forall y \in Y}\{c(y)\}$ і відзначаємо вершини $v_1, v_2 \in X$, які інцидентні ребру e_1. Будуємо дерево $T_1 = (X_1, Y_1) \subset G$, де $X_1 = \{v_1, v_2\}$, $Y_1 = \{e_1\}$.

Крок $k + 1$. Нехай дерево $T_k = (X_k, Y_k) \subset G$ вже побудоване, де $Y_k = \{e_i\}_{i=1}^{k}$, $X_k = \{v_j\}_{j=1}^{k+1}$. Серед множини ребер $Y(X_k, X'_k)$, що з'єднують вершини із X_k з вершинами з множини $X'_k = X \backslash X_k$, обираємо ребро e_{k+1} з найменшою вагою: $c(e_{k+1}) = \min\{c(y), \forall y \in Y(X_k, X'_k)\}$. Дерево T_{k+1} будується приєднанням до T_k одного ребра e_{k+1} і одної вершини $v_{k+2} \in X'_k$, інцидентній ребру e_{k+1}: $T_{k+1} = (X_{k+1}, Y_{k+1})$, $X_{k+1} = X_k \cup v_{k+2}$, $Y_{k+1} = Y_k \cup e_{k+1}$.

Зауваження. Іноді треба побудувати остів максимальної ваги у графі G. Алгоритми Краскала і Прима повністю розв'язують цю задачу, якщо у них всюди мінімальну вагу замінити на максимальну.

Орієнтовані і бінарні дерева

Дерева з орієнтованими *ребрами (дугами)* використовуються давно. Достатньо згадати генеалогічні дерева, розгалужені фізичні та інформаційні процеси. На сьогодні вони істотно використовуються у питаннях кодування, теорії мов і граматик, теорії автоматів, у логічному і синтаксичному аналізі, в теорії алгоритмів.

Визначення

Орієнтованим деревом *(інакше —* ***кореневим орієнтованим деревом****)* називається орієнтований граф без циклів з властивостями:

а) існує точно одна вершина r, що називається *коренем*, у яку не заходить жодна дуга;

б) в кожну вершину, крім кореня, заходить точно одна дуга;

в) існує шлях з кореня до будь-якої іншої вершини.

Зауваження. Обравши орієнтацію на всіх ребрах неорієнтованого графа, ми одержуємо орієнтований граф. Однак вибір довільної орієнтації на ребрах неорієнтованого дерева не обов'язково призводить до орієнтованого дерева у розумінні наведеного вище визначення. Наприклад, граф з чотирма дугами на рис. 6.29 не є орієнтованим деревом, граф на рис. 6.30 — орієнтоване дерево з коренем $r = 1$.

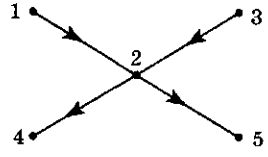


Рис. 6.29.
Неорієнтоване дерево

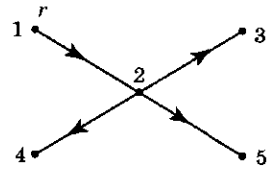


Рис. 6.30.
Орієнтоване дерево з коренем r

Відносно елементів орієнтованих дерев встановилася відповідна термінологія. ***Нащадок*** v вершини u — це вершина v, в яку веде *шлях* з вершини u; при цьому u називається ***предком*** для v. Якщо довжина цього шляху дорівнює 1, тобто з u до v веде дуга, то u — ***«батько»*** для v, а v — ***«син»*** для u. Вершина, що не має нащадків, називається ***листом***. ***Глибина вершини*** v у дереві — це довжина шляху з кореня у v. ***Висотою вершини*** v у дереві називається довжина найдовшого шляху з v до будь-якого листа. ***Висота дерева*** — це число дуг найдовшого шляху, тобто висота кореня. ***Рівень вершини*** v — це різниця між висотою всього дерева і глибиною вершини v. Висота дерева на рис. 6.31 дорівнює 3, вершина 8 має глибину 2, висоту 1, рівень 1.

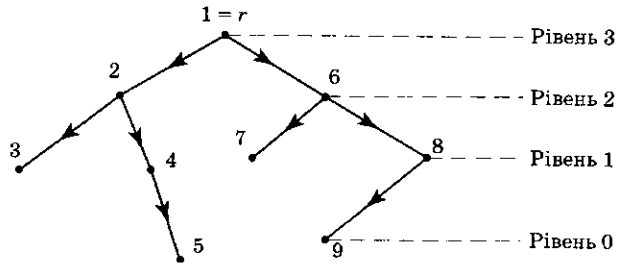


Рис. 6.31. Кореневе дерево, упорядковане за рівнями

Листями є вершини 3, 5, 7, 9; їх рівні і глибина у принципі різні: листя 3, 7 мають рівень 1, глибину 2, листя 5, 9 мають рівень 0, глибину 3. Висота ж будь-якого листа дорівнює 0.

Упорядкованим деревом (орієнтованим, розуміється) називається дерево, в якого множина синів кожної вершини упорядкована звичайно зліва направо, як на рис. 6.31: $2 < 6$, $3 < 4$, $7 < 8$.

Визначення

Бінарним (двійковим) деревом T *називається упорядковане дерево, з кожної вершини якого може виходити не більше двох дуг.*

Нагадаємо, що упорядковане дерево T є орієнтованим, з коренем, і тому у кожну вершину, що відмінна від кореня, заходить рівно одна дуга. Кожна вершина бінарного дерева може мати або двох синів — *лівого* і *правого* (так вершина 2 рис. 6.31 має лівого сина 3 і правого сина 4), або мати тільки лівого сина (лівий син 9 вершини 8), або тільки правого сина (вершина 5 — єдиний правий син вершини 4), або не мати жодного сина (листя 3, 5, 7, 9).

Лівим піддеревом вершини u називається максимальне піддерево $T_l(u_l) \subset T$, коренем якого є лівий син u_l вершини u. Відповідно ***праве піддерево*** $T_r(u_r)$ вершини u *визначається* як максимальне піддерево з коренем u_r — правим сином вершини u. На рис. 6.31 ліве піддерево вершини 6 складається з однієї вершини 7, праве піддерево — з вершин 8, 9 і дуги (8, 9).

Комп'ютерне завдання двійкового дерева T звичайно зображує два масиви: ЛІВИЙ СИН і ПРАВИЙ СИН. Якщо вершини T занумеровані числами від 1 до n, то в масиві ЛІВИЙ СИН $[i] = k$ тоді і тільки тоді, коли вершина «k» є лівим сином вершини 'i', та $[i] = 0$ тоді і тільки тоді, коли вершина 'i' не має лівого сина. Для дерева рис. 6.31 зображення вказаними двома масивами має вигляд таблиці 6.3.

Таблиця 6.3. Зображення дерева рис. 6.31

[i]	1	2	3	4	5	6	7	8	9	
ЛІВИЙ СИН	2	3	0	0	0	7	0	9	0	
ПРАВИЙ СИН	6	4	0	5	0	8	0	0	0	

Завдання

1. Перевірити, що шлях з кореня в будь-яку іншу вершину є єдиним.
2. Назвіть всі попарно не ізоморфні одне одному дерева четвертого порядку (тобто з чотирма вершинами); аналогічно — п'ятого порядку.
3. Якщо два дерева ізоморфні, то послідовності степенів їх вершин $\delta_1 \le \delta_2 \le \ldots \le \delta_v$ однакові. Чи правильне зворотне?
4. Чи правильно, що будь-який маршрут у дереві є простим ланцюгом? Довести або спростувати.
5. Назвіть неізоморфні дерева з шістьма вершинами ($t_6 = 6$).
6. Назвіть кореневі дерева з п'ятьма вершинами ($T_5 = 9$).
7. Скоректувати алгоритм побудови остовного дерева пошуком у глибину для випадку незв'язного графа та одержати відповідний алгоритм побудови остовного лісу.
8. Модифікувати алгоритм побудови остовного дерева довільним перебором ребер у алгоритм побудови остовного лісу незв'язного графа.

6.9. Теорема Пуанкаре. Фундаментальні матриці перерізів і циклів

1. В орієнтованому графі $G = (X, Y, f)$ кожному *циклу* q (тобто замкненому ланцюгу, п. 6.2) ставиться у відповідність вектор-цикл $\vec{q}$ за таким правилом: компонента q_i рядку $\vec{q}$ дорівнює нулю, якщо дуга y_i не входить до циклу q; якщо ж $y_i \in q$, то $q_i = 1$ або $q_i = -1$ залежно від того, співпадає чи ні напрямок дуги y_i з обраним додатним напрямком обходу циклу. Циклу $q = \{y_1, y_2, y_4, y_6\}$ у графі на рис. 6.32 відповідає вектор-цикл

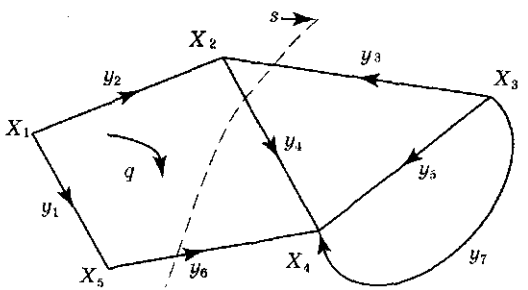


Рис. 6.32. Цикл у графі

$$\begin{array}{c} \quad y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \vec{q} = (-1 \quad 1 \quad 0 \quad 1 \quad 0 \quad -1 \quad 0) \end{array}.$$

Будь-яка *максимальна лінійно незалежна множина вектор-циклів, об'єднаних за рядками у матрицю Q, називається* ***матрицею циклів графа***[1].

Перерізом (***розрізом***) s графа G називається певна підмножина дуг, видалення якої збільшує число компонент зв'язності, тому підграф $G\backslash s$ складається з двох диз'юнктних частин G_1 і G_2, зв'язок між якими у графі G здійснюється тільки дугами перерізу s (рис. 6.33).

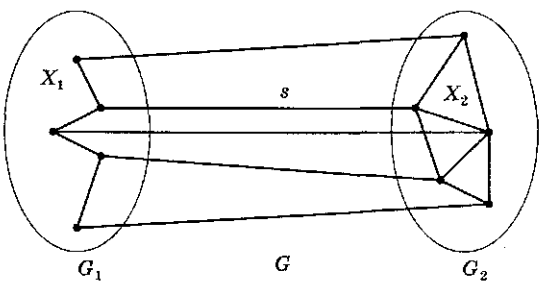


Рис. 6.33. Переріз графа

Вершини X_1 із G_1, які інцидентні дугам s, називаються першим боком перерізу; вершини X_2 із G_2, інцидентні s — другим боком перерізу s. Таким чином, підмножини вершин X_1, X_2, що не перетинаються, утворюють ***розбиття*** множини всіх вершин $X = X_1 \cup X_2$ графа G. Для перерізу (як і для циклу) довільно вводиться орієнтація — напрямок від одного боку перерізу до іншого. Вектор-переріз $\vec{s}$ — це рядок $\vec{s}$, у якого компонента $s_i = 0$, коли дуга y_i не входить у переріз s. Якщо дуга y_i належить перерізу узгоджено з обраною орієнтацією перерізу, то $s_i = 1$; якщо не узгоджено, то $s_i = -1$. Для перерізу $s = \{y_3, y_4, y_6\}$ у графі (рис. 6.32) вектор-переріз дорівнює:

$$\begin{array}{c} \quad y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \vec{s} = (0 \quad 0 \quad -1 \quad 1 \quad 0 \quad 1 \quad 0) \end{array}.$$

[1] Іноді Q називається «базисною матрицею циклів», а матрицею циклів — сукупність всіх векторів-циклів, взагалі кажучи, лінійно залежних.

Матриця S, рядки якої утворюють максимальний лінійно-незалежний набір вектор-перерізів, називається ***матрицею перерізів***[1] графа G. Стовпці матриць перерізів і циклів нумеруються однаково — дугами графа, а число їх рядків у загальному випадку різне.

Завдання

1. Побудувати матриці перерізів S і циклів Q графа на рис. 6.34 і переконатися, що їх об'єднання виду $\begin{bmatrix} S \\ Q \end{bmatrix}$ є квадратна невироджена матриця.
2. Кожний рядок $\vec{b}_i$ матриці інциденцій B (п. 10.6) є вектор-перерізом, що відповідає перерізу s_i навколо вершини x_i (рис. 6.34).

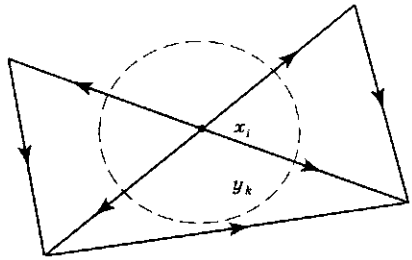


Рис. 6.34. Переріз навколо вершини

2. В графі G з m дугами кожний вектор-цикл $\vec{q}$ ортогональний будь-якому вектору-перерізу $\vec{s}$ (теорема Пуанкаре):

$$\vec{q} \cdot \vec{s}^T = (\vec{q}, \vec{s})_{R^m} = 0. \qquad (6.16)$$

Більш того, якщо ввести простір вектор-циклів H_q як лінійну оболонку вектор-циклів[2] і простір вектор-перерізів H_s як лінійну оболонку вектор-перерізів, то

$$H_q \oplus H_s = R^m. \qquad (6.17)$$

Таким чином, крім властивості ортогональності (6.16) вектор-цикли і вектор-перерізи графа з m дугами утворюють ***тотальну систему*** елементів у просторі R^m, тобто систему векторів, лінійна оболонка яких є весь простір R^m.

Доведемо спочатку співвідношення (6.16).

---

[1] Або «базисною матрицею перерізів», на відміну від матриці всіх перерізів графа.

[2] Де лінійні комбінації утворюються з будь-якими дійсними коефіцієнтами, а не тільки з цілочисловими.

Теорема 6.10 ***(А. Пуанкаре)***

Довільний вектор-переріз $\vec{s}$ *ортогональний будь-якому вектор-циклу* $\vec{q}$ *графа* G.

В сумі

$$\vec{s}\vec{q}^T = \sum_{j=1}^{m} s_j q_j = \vec{q}\vec{s}^T \qquad (6.18)$$

відмінні від нуля тільки ті доданки (s_j q_j), для яких дуга y_j входить одночасно у розглянуті переріз s і цикл q. Число таких ребер y_j парне. Дійсно, при обході циклу q проходження дуги $y_j \in s$ у напрямку орієнтації перерізу повинне компенсуватися проходженням деякої дуги $y_k \in s$ у напрямку, протилежному орієнтації s. Припустимо, дуги y_j, y_k спрямовані з додатним напрямком обходу циклу і $q_j = q_k = 1$. Тоді $sgn\ s_j = -sgn\ s_k$ і $s_j q_j + s_k q_k = 0$. Зміна напрямку дуги y_j призводить до одночасної зміни знаків чисел s_j і q_j, так що значення $s_j q_j$ зберігається. В силу відзначеної парності сума (6.18) дорівнює нулю:

$$\sum_{\forall j} s_j q_j = 0. \qquad \blacksquare$$

3. Доведення розкладання (6.17) зручно провести за допомогою так званої ***фундаментальної системи циклів і перерізів***, що відповідає фіксованому каркасу T графа G (***каркасом*** називається максимальний підграф без циклів).

Зрозуміло, що достатньо розглядати ці питання для зв'язного графа, оскільки у незв'язному графі вектори $\vec{s}$, $\vec{q}$ є прямими сумами відповідних векторів у компонентах зв'язності.

Каркас T зв'язного графа $G = (X, Y, f)$ є, очевидно, деревом $T = (X, Y^T, f^T)$, що містить всі вершини графа. ***Кодеревом*** G_c називається доповнення

$$G_c = G \backslash T = (X,\ Y \backslash Y^T = Y_c,\ f_c = f|Y_c).$$

Наприклад, один з каркасів графа G рис. 6.35 зображений жирними лініями (каркас «великої ведмедиці») і відповідно дуги кокаркаса $Y_c = \{y_7, y_8, y_9, y_{10}\}$ — тонкими лініями.

Переріз s називається ***фундаментальним*** (за деревом T), якщо він містить точно одну дугу y_k дерева T, а орієнтація перерізу s узгоджена з орієнтацією дуги y_k. Зрозуміло, що завдання дуги дерева повністю визначає відповідний фундаментальний

переріз. На рис. 6.35 пунктирами зображено фундаментальні перерізи $s_1, s_2, \ldots, s_6$ за деревом T «велика ведмедиця».

Фундаментальною матрицею перерізів S_T (за деревом T) називається матриця, рядками якої є вектор-перерізи, фундаментальні за деревом T. Виділеному дереву T на рис. 6.35 відповідає фундаментальна матриця перерізів, стовпці якої позначені номерами дуг всього графа, рядки — номерами дуг дерева, точніше, номерами відповідних фундаментальних перерізів:

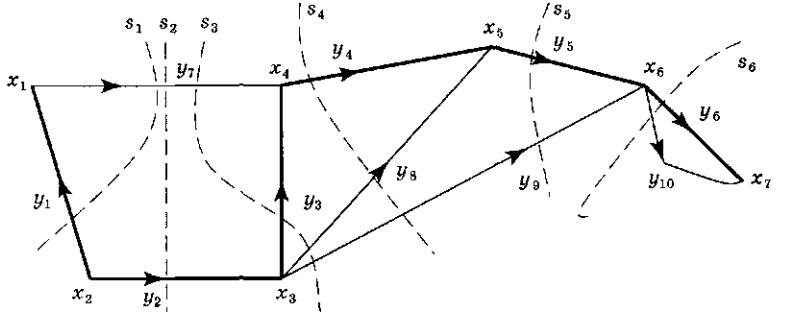


Рис. 6.35. Дерево «великої ведмедиці»

$$S_T = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc|cccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \left[\begin{array}{cccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}. \qquad (6.19)$$

Фундаментальним циклом графа G ***(за деревом*** T***)*** називається будь-який цикл, що містить точно одну дугу y_i кодерева $G\backslash T$ і орієнтований узгоджено з напрямком дуги y_i. Зрозуміло, що кожна дуга y кодерева однозначно задає фундаментальний цикл q_y, який дуга y замикає на дереві T. ***Фундаментальною матрицею циклів*** Q_T (за деревом T) називається матриця, рядки якої є всі фундаментальні вектор-цикли за деревом T. Якщо для графа на рис. 6.19 фундаментальні вектор-цикли позначити номерами відповідних дуг, кодерева y_7, y_8, y_9, y_{10}, то фундаментальна матриця циклів така:

$$Q_T = \begin{array}{c} \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \begin{array}{c} \begin{array}{cccccc|cccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \left[\begin{array}{cccccc|cccc} 1 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}. \qquad (6.20)$$

Зауваження. Відносно розбивання множини всіх дуг $Y = Y_T \cup Y_c$ на дуги дерева і кодерева фундаментальні матриці перерізів і циклів розбиваються на блоки виду:

$$S_T = \overset{Y_T \quad Y_C}{Y_T[E;\ S_0]}; \qquad Q_T = \overset{Y_T \quad Y_C}{Y_C[Q_0;\ E]}. \qquad (6.21)$$

З теореми Пуанкаре одержуємо:

Наслідок. *Для довільних матриці перерізів S і циклів Q даного графа G справедливе співвідношення ортогональності*

$$SQ^T = 0; \qquad QS^T = 0. \qquad (6.22)$$

Інформаційні блоки S_0, Q_0 (6.21) фундаментальних матриць S_T, Q_T (за одним деревом) виражаються один через другий за формулою:

$$S_0 = -Q_0^T; \qquad Q_0 = -S_0^T. \qquad (6.23)$$

Тому достатньо обчислити одну з фундаментальних матриць S_T, Q_T.

Тепер можна обґрунтувати розкладання (6.17) простору R^m у ортогональну суму простору вектор-циклів H_q і вектор-перерізів H_s. Простір H_q (H_s) містить лінійну оболонку Z_q (Z_s) рядків матриці Q_T (S_T). В силу (6.22) маємо $Z_q \perp Z_s$. У той же час через одиничні блоки у (6.21) $\dim Z_s + \dim Z_q = m$, звідки $Z_q \oplus Z_s = R^m$.

Наслідок. *Квадратна* $(m \times m)$ *матриця* $\begin{bmatrix} S \\ Q \end{bmatrix}$, *зокрема* $\begin{bmatrix} S_T \\ Q_T \end{bmatrix}$ *має максимальний ранг* m.

Наслідок (із формули (6.21)). *Для зв'язного графа з* m *дугами і* n *вершинами*

$$\operatorname{rang} S = \operatorname{rang} S_T = |Y_T| = n - 1; \quad \operatorname{rang} Q = m - n + 1 = |Y_c|.$$

Величина rang Q співпадає з цикломатичним числом ν — числом комірок плоского графа. У випадку неплоского зв'язного графа число $\nu = m - n + 1$ також називається цикломатичним,

воно співпадає з розмірністю простору вектор-циклів $\dim H_q$ (або максимальним числом лінійно незалежних вектор-циклів графа).

Завдання

1. Чи еквівалентні одна одній (і в якому сенсі) матриця інцидencій і перерізів?
2. Повна лінійно незалежна система рівнянь Кірхгофа електричного ланцюга на графі G може бути записана за допомогою матриць перерізів і циклів у векторній формі так:
   $\vec{S}\vec{I} = 0$; $\vec{Q}\vec{U} = 0$. Доведіть це.
   Тут $\vec{I}(\vec{U})$ — вектор всіх струмів (напружень) гілок.
3. Доведіть, що:
   а) будь-яка невироджена квадратна матриця, що складена із стовпців матриці перерізів, відповідає деякому каркасу графа;
   б) будь-яка невироджена квадратна матриця, що складена із стовпців матриці циклів, відповідає деякому кокаркасу у графі.
4. Переконайтеся в тому, що для плоского графа задача перелічення перерізів графа G еквівалентна задачі перелічення циклів двоїстого графа $\bar{G}$.
5. Побудувати фундаментальну матрицю перерізів і циклів двох графів Понтрягіна — Куратовського, що зображені на рис. 6.36. Переконайтеся в ортогональності матриць S і Q^T для кожного графа.

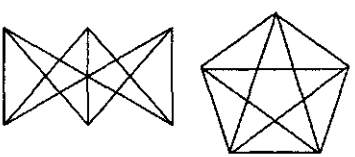


Рис. 6.36. Графи Понтрягіна — Куратовського

6. Підмножина ребер Y_0 у планарному графі $G = (X, Y, f)$ утворює простий цикл тоді і тільки тоді, коли відповідна йому підмножина ребер $\overline{Y_0}$ у геометрично двоїстому графі $\bar{G}$ утворює переріз. Доведіть це.

6.10. Найкоротші відстані та шляхи у мережах

Задачі відшукання найкоротших відстаней між вершинами графа (мережі) із заданими вагами ребер часто зустрічаються на практиці і мають очевидний економічний зміст. Це ж можна сказати і про задачі пошуку оптимальних шляхів у орієнтованих мережах. У цьому розділі наведено деякі ефективні алгоритми розв'язку таких задач: алгоритми Дейкстри, Форда, Флойда, Данцига.

Нехай $G = (X, Y)$ — зважений орієнтований зв'язний граф з додатними довжинами (вагами) дуг $c(y) \geq 0$, треба знайти найкоротший шлях (з мінімальною сумою ваг дуг) з вершини s до вершини t графа. Ефективний алгоритм було запропоновано Е. Дейкстрою у 1959 р. Його ідея така. Якщо вже знайдено довжини (ваги) $l(x_i)$ найкоротших шляхів з вершини s і вказано («пофарбовано») ті вершини $x_1 = s, x_2, \ldots, x_k$, до яких ведуть ці шляхи, то «черговий» за довжиною шлях із s до $(k + 1)$-ї вершини x_{k+1} у кінці цього шляху можна знайти за таким правилом. Позначимо через $X_k = \{x_i\}_{i=1}^{k}$ множину пофарбованих вершин. Для кожної непофарбованої вершини w переглянути довжини $c(x_i, w)$ дуг (x_i, w), $x_i \in X_k$, $w \in W_k = X \backslash X_k$ і знайти

$$\min \{c(x_i, w)\} = c(v_0, w_0), \; v_0 \in X_k, \; w_0 \in W_k. \qquad (6.24)$$

Коли вершини x_i, w не суміжні, то $c(x_i, w) = \infty$. Якщо $c(x_i, w) = \infty$ для всіх $x_i \in X_k$ і всіх $w \in W_k$, то алгоритм зупиняється: всі вершини w множини W_k і, зокрема, вершина t не досяжні із s. Якщо існує хоча б одна дуга з множини вершин X_k у множину W_k, тоді мінімум (6.24) скінченний і вершину w_0 буде знайдено. Шуканий («черговий» за довжиною) шлях веде з вершини s до вершини w_0, довжина цього шляху дорівнює $l(v_0) + c(v_0, w_0)$. Ми можемо пофарбувати вершину w_0, тобто привласнити їй позначення $x_{k+1} = w_0$, покласти $l(x_{k+1}) = l(v_0) + c(v_0, w_0)$ і сформувати більш широку множину пофарбованих вершин $X_{k+1} = \{x_i\}_{i=1}^{k+1}$. Процес закінчується, якщо $x_{k+1} = t$.

Формально в алгоритмі пофарбованій вершині $x \in X_k$ привласнюється постійна позначка $l(x)$ — довжина найкоротшого шляху із s до x. Якщо вершина $w \in W_k$ ще не має постійної позначки, тобто не пофарбована, їй привласнюється тимчасова позначка $l(w)$, що дорівнює довжині найкоротшого (s, w)-шляху, що проходить тільки по вершинам з постійними позначками (пофарбованим). Тимчасова позначка вершини w, взагалі кажучи, змінюється від кроку до кроку і перестає змінюватися, коли вершина w фарбується (одержує постійну позначку).

Практично на *k-му* кроці алгоритму будується кореневе дерево $T_k (\supset T_{k-1})$ від кореня s до вершин множини X_k з постійними позначками. На останньому кроці k_0 вершина t стає кінцевою вершиною дерева T_{k_0}.

Зауваження. Алгоритм можна скоректувати у пункті «зупинка», якщо ставити задачу знаходження найкоротших шляхів з кореня s до всієї решти вершин графа G за умови, що вони досяжні із s. В цьому випадку алгоритм зупиняється тільки тоді, коли буде побудовано дерево T_n, що містить всі вершини з X. *Таким чином, алгоритм Дейкстри побудує остовне орієнтоване дерево T_n графа G з коренем s.*

Зауваження. Алгоритм Дейкстри може знаходити найкоротші відстані між парами вершин у *неорієнтованому зв'язному зваженому графі G з додатними вагами ребер $c(y)$.*

Для застосування описаної вище процедури слід перейти до орієнтованого симетричного графа, замінивши кожне ребро парою протилежно спрямованих дуг однакової ваги. Остовне дерево T_n завжди може бути побудоване за допомогою алгоритму Дейкстри, і T_n буде складатися з найкоротших простих ланцюгів між вершиною s і рештою вершин графа G.

Формальний опис алгоритму Дейкстри побудови найкоротших шляхів із вершини s

Крок 1. Привласнення початкових значень.

Покласти $l(s) = 0$ і вважати цю позначку постійною (тобто пофарбувати вершину s). Покласти $l(w) = \infty$ для всіх вершин $w \neq s$ і вважати ці позначки тимчасовими. Покласти $p = s$ (p — ім'я останньої вершини, що одержала постійну позначку).

Крок 2. Оновлення позначок.

Для кожної непофарбованої вершини w, у яку веде[1] дуга (p, w) з вершини p, позначка $l(w)$ змінюється за правилом

$$l(w) \leftarrow \min\{\, l(w),\ l(p) + c(p, w)\}. \qquad (6.25)$$

Крок 3. Перетворення позначки у постійну.

На множині всіх непофарбованих вершин (з тимчасовими позначками) W_p знайти вершину w_0 з мінімальною позначкою: $l(w_0) = \min\limits_{w \in W_p}\{l(w)\}$. Зробити позначку $l(w_0)$ вершини w_0 постійною і покласти $p = w_0$.

Крок 4. (Якщо треба знайти лише найкоротший шлях від s до t).

---

[1] Умову «веде дуга (p, w)» можна не перевіряти, але тоді розглядаються всі вершини w з тимчасовими позначками, і якщо дуга (p, w) відсутня, то в (6.25) $c(p, w) = \infty$ і тимчасова позначка $l(w)$ не змінюється, див. приклад рис. 6.38.

4.1. Якщо $p = t$, то $l(p)$ — довжина найкоротшого (s, t) шляху.

Зупинка.

4.2. Якщо $p \neq t$, то перейти до кроку 2.

Крок 5. (Якщо треба знайти найкоротші шляхи від s до всіх вершин графа G).

Алгоритм виконується так довго, щоб всі вершини одержали постійні позначки і множина непофарбованих вершин вичерпалася: $W_p = 0$.

Зупинка.

Складність алгоритму Дейкстри

Зауважимо, що оцінка складності алгоритму не залежить від того, чи шукаються найкоротші шляхи від вершини s до всієї решти вершин або тільки від s до фіксованої вершини t. Дійсно, якщо короткий (s, t)-шлях виявляється найдовшим із всіх найкоротших (s, w)-шляхів, то обидві задачі розв'язуються за однакове число кроків. У цьому випадку кожний з кроків 2, 3, 5 виконується по $(n - 1)$ разів. Трудомісткість одного кроку 2 дорівнює $\theta(1)$, одного кроку 5 — також $\theta(1)$; трудомісткість кроку 3 має порядок середньої потужності множини W_k — числа $\frac{(n-1)+1}{2} = \frac{n}{2}$. Отже, трудомісткість всього алгоритму має порядок числа $\frac{n(n-1)}{2} + 2(n-1)$, тобто $0(n^2)$, $n = |X|$.

Приклад *(робота алгоритму Дейкстри)*

Знайдемо найкоротший шлях між вершинами s і t у зваженому графі G на рис. 6.37, одночасно вказавши найкоротші шляхи меншої довжини з вершини s до відповідних вершин (ваги дуг, що дорівнюють цілим числам, вказано на рисунку). Множина вершин є $X = \{s, a, b, e, d, t\}$, $|X| = n = 6$.

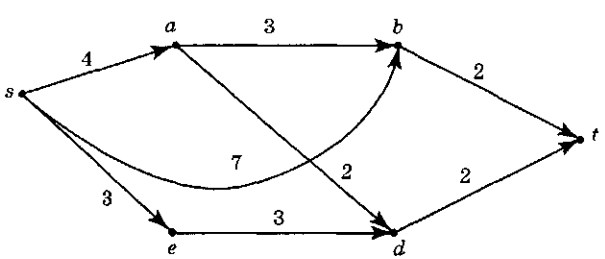


Рис. 6.37. Граф для ілюстрації алгоритму Дейкстри

Позначення. $c(u, v)$ — вага (довжина) дуги (u, v), $l(w)$ — довжина (s, w)-шляхи або позначка вершини w на даному кроці, $l_{-}(w)$ — довжина (s, w)-шляху або позначка вершини w на попередньому кроці.

Крок 1. Привласнення початкових значень.

$l(s) = 0$ — постійна позначка; s фарбується.

$l(a) = \infty$, $l(e) = \infty$, $l(b) = \infty$, $l(d) = \infty$, $l(t) = \infty$ — тимчасові позначки вершин a, e, b, d, t. Припускаємо $p = s$.

Крок 2. s. ($p = s$). Оновлення позначок.

$l(a) = \min\{l_{-}(a), l(s) + c(s, a)\} = \min\{\infty, 0 + 4\} = 4$

$l(b) = \min\{l_{-}(b), l(s) + c(s, b)\} = \min\{\infty, 0 + 7\} = 7$

$l(e) = \min\{l_{-}(e), l(s) + c(s, e)\} = \min\{\infty, 0 + 3\} = 3$

$l(d) = \min\{l_{-}(d), l(s) + c(s, d)\} = \min\{\infty, 0 + \infty\} = \infty$

$l(t) = \min\{l_{-}(t), l(s) + c(s, t)\} = \min\{\infty, 0 + \infty\}$.

Крок 3. s. ($p = s$). Перетворення позначки на постійну.

$$\min\{l(w), w \in W_p = \{a, e, b, d, t\}\} =$$

$$= \min\{4, 7, 3, \infty, \infty\} = 3 = l(e).$$

Вершина e одержує постійну позначку $l(e) = 3$, вершина e — фарбується разом з дугою (s, e). Поточне дерево T_1 найкоротших шляхів з вершини s містить вершини s, e і дугу (s, e) (див. рис. 6.38).

Припускаємо $p = e$.

Крок 2. e. ($p = e$). Оскільки $t \notin T_1$, переходимо до кроку 2 при $p = e$, оновлення позначок.

$l(a) = \min\{l_{-}(a), l(e) + c(e, a)\} = \min\{4, 3 + \infty\} = 4$

$l(b) = \min\{l_{-}(b), l(e) + c(e, b)\} = \min\{7, 3 + \infty\} = 7$

$l(d) = \min\{l_{-}(d), l(e) + c(e, d)\} = \min\{\infty, 3 + 3\} = 6$

$l(t) = \min\{l_{-}(t), l(e) + c(e, t)\} = \min\{\infty, 3 + \infty\} = \infty$.

Крок 3. e. ($p = e$). Перетворення позначки на постійну.

$$\min\{l(w), w \in W_p = \{a, b, d, t\}\} = \min\{4, 7, 6, \infty\} = 4 = l(a).$$

Вершина a одержує постійну позначку $l(a) = 4$. Вершина a і дуга (s, a) фарбуються. Поточне дерево найкоротших шляхів T_2 складається з вершин s, e, a і дуг (s, e), (s, a) (див. рис. 6.38).

Припускаємо $p = a$.

Крок 2. a. ($p = a$). Оновлення позначок, оскільки $t \notin T_2$.

$l(b) = \min\{l_{-}(b), l(a) + c(a, b)\} = \min\{7, 4 + 3\} = 7$

$l(d) = \min\{l_{-}(d), l(a) + c(a, d)\} = \min\{6, 4 + 2\} = 6$

$l(t) = \min\{l_{-}(t), l(a) + c(a, t)\} = \min\{\infty, 4 + \infty\} = \infty$.

Крок 3. *a*. ($p = a$). Перетворення позначки на постійну.

$\min\{l(w),\ w \in W_p = \{a, d, t\}\} = \min\{7, 6, \infty\} = 6 = l(d)$.

Вершина d одержує постійну позначку $l(d) = 6$. Вершина d і дуга (a, d) фарбуються. Поточне дерево найкоротших шляхів T_3 складається з вершин s, e, a, d і дуг (s, e), (s, a), (a, d) (див. рис. 6.38).

Припускаємо $p = d$.

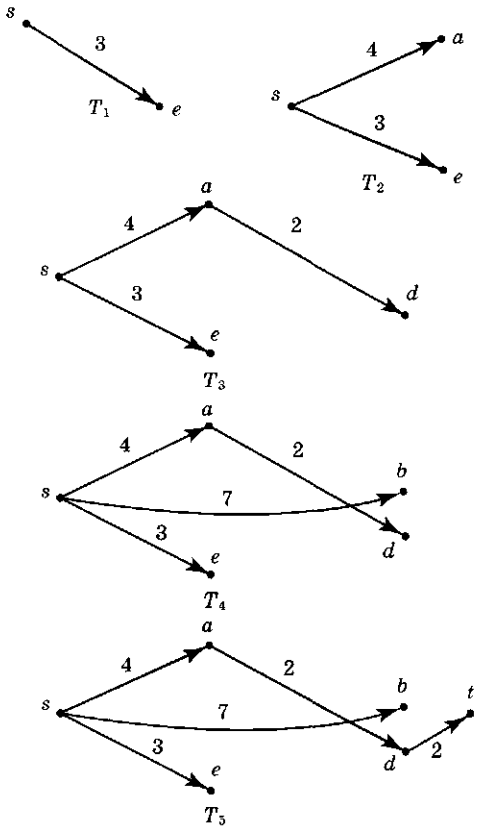


Рис. 6.38. Зростаючі орієнтовані дерева T_p найкоротших шляхів

Крок 2. *d*. ($p = d$). Оновлення позначок, оскільки $t \notin T_3$.

$l(b) = \min\{l(b),\ l(d) + c(a, b)\} = \min\{7,\ 6 + \infty\} = 7$

$l(t) = \min\{l(t),\ l(d) + c(a, t)\} = \min\{\infty,\ 6 + 2\} = 8$.

Крок 3. *d*. ($p = d$). Перетворення позначки на постійну.

$\min\{l(w),\ w \in W_p = \{b, t\}\} = \min\{7, 8\} = 7 = l(b)$.

Вершина b одержує постійну позначку $l(b) = 7$ і фарбується разом з дугою (b). Поточне дерево найкоротших шляхів T_4 складається з вершин s, e, a, d, b і дуг (s, e), (s, a), (a, d), (s, b) (див. рис. 6.38).

Припускаємо $p = b$.

Крок 2. b. $(p = b)$. Оновлення позначок, оскільки $t \notin T_4$.

$l(t) = \min\{l(t), l(b) + c(b, t)\} = \min\{8, 7 + 2\} = 8$.

Крок 3. b. $(p = b)$. Перетворення позначки на постійну.

$\min\{l(w), w \in W_p = \{t\}\} = 8 = l(t)$.

Вершина t одержує постійну позначку $l(t) = 8$ і фарбується разом з дугою (d, t). Поточне дерево найкоротших шляхів T_5 містить всі вершини графа G і дуги (s, e), (s, a), (a, d), (s, b), (d, t) (див. рис. 6.38). Дерево T_5 є остовним деревом у G.

Зупинка.

Найкоротший шлях із s до t є (s, a), (a, d), (d, t). Він не є єдиним, таку ж довжину 8 має шлях (s, e), (e, d), (d, t).

Алгоритм Форда

В алгоритмі Дейкстри передбачалося, що ваги (довжини) всіх дуг *додатні*. Можливі і невід'ємні ваги $c_j \geq 0$, оскільки послідовне стягування дуг з нульовими вагами призводить до зваженого графа з додатними вагами дуг.

Однак, якщо є дуги з *від'ємними і додатними* вагами, алгоритм Дейкстри, взагалі кажучи, не знаходить найкоротшого шляху.

Ще до появи у 1959 р. алгоритму Дейкстри було запропоновано процедури знаходження найкоротших шляхів за дугами з будь-якими знаками ваг: 1956 р. — Форд, 1957 р. — Мур, 1958 р. — Беллман. В головному ці процедури схожі; відповідна обчислювальна процедура називається зараз *алгоритмом Форда побудови найкоротших шляхів у випадку довільних ваг.*

Алгоритм Форда можна викласти як *модифікацію* алгоритму Дейкстри. Зміни алгоритму Дейкстри полягають у такому:

1. На кроці 2 «Оновлення позначок» перелік величин $l(w)$ за формулою (6.25) робиться *для всіх вершин* — як з тимчасовими, так і з постійними позначками (непофарбованими і пофарбованими). При цьому довжини $l(w)$ можуть зменшуватися як для непофарбованих вершин, так і для пофарбованих.

Поняття «постійна позначка» вершини x_p стає некоректним, і зручніше говорити, що вершина x_p *пофарбована* (на кроці 3), але, можливо, тимчасово.

2. Якщо для деякої пофарбованої вершини x_p після застосування кроку 2 відбувається зменшення «відстані» $l(x_p)$, тоді з вершини x_p і пофарбованої дуги, що веде до неї, *знімається фарбування*.

3. Алгоритм Форда закінчується тоді, коли всі вершини графа пофарбовані і ще одне виконання кроку 2 не змінює жодне з чисел $l(x)$, $\forall x \in X$.

Умова коректності алгоритму Форда, тобто завершення згідно із зауваженням 3 за скінченне число кроків, полягає у такому: *граф G не повинен містити контурів від'ємної «довжини»* (з від'ємною сумою ваг дуг).

Взагалі кажучи, виявлення контурів від'ємної довжини є окремою задачею. Однак для визначення коректності алгоритму Форда цю задачу не обов'язково розв'язувати окремо. Достатньо звичайним чином застосовувати алгоритм Форда, поки або алгоритм закінчиться відповідно до зауваження 3, або число фарбувань будь-якої одної вершини не досягне величини *n-кількості* всіх вершин. У останньому випадку граф має контур від'ємної довжини і алгоритм Форда *не коректний*.

До речі, алгоритм Форда знаходження оптимального шляху з вершини s до вершини t можна застосувати до графа з *невід'ємними вагами дуг*, однак у цьому випадку він менш ефективний за швидкістю порівняно з алгоритмом Дейкстри, оскільки складність алгоритму Форда дорівнює $\theta(n^3)$, а алгоритму Дейкстри $-\theta(n^2)$.

Алгоритми Флойда і Данцига

Задача пошуку *найкоротших шляхів між всіма парами вершин графа* може бути розв'язана шляхом послідовного привласнення позначки «початкова вершина s» *кожній вершині графа* G і наступного застосування алгоритму Дейкстри (Форда). Зрозуміло, що складність такого «багаторазового алгоритму Дейкстри» є $\theta(n^3)$, а «багаторазового алгоритму Форда» — відповідно $\theta(n^4)$. Однак існують більш ефективні методи, ніж багаторазове (саме *n-кратне*) повторення алгоритму Дейкстри або Форда — саме алгоритми Флойда (1962 р.) і Данцига (1967 р.). Обидва алгоритми застосовні як у випадку невід'ємних ваг дуг, так і у випадку ваг довільних знаків. Однак у будь-якому випадку *необхідно, щоб граф G не мав контурів від'ємної довжини*.

Попередньо пронумеруємо вершини графа G натуральними числами 1, 2, ..., n і введемо позначення $l^m_{i,j}$ для довжини найкоротшого з усіх таких шляхів від вершини i до вершини j (інакше — всіх таких (i, j) — шляхів), які як проміжні вершини можуть містити тільки перші m вершин, тобто вершини з множини $V_m = \{1, 2, ..., m\}$. При цьому $l^m_{i,j} = \infty$, якщо не існує жодного (i, j)-шляху з проміжними вершинами тільки і V_m. Для $m = 0$ вважається, що $V_m = \{\varnothing\}$ — порожня множина і l^0_{ij} — довжина найкоротшої (i, j)-дуги (тобто (i, j)-шляхи без проміжних вершин) при $i \neq j$. Для $i = j$ припускається $l^0_{i,i} = 0$.

Алгоритм Флойда ставить своєю ціллю послідовно для $m = 0, 1, 2, ..., n$ обчислювати n^2 чисел (довжин) $l^m_{i,j}$ і закінчувати обчислення одержанням $l^n_{i,j}(i, j = 1, 2, ..., n)$ шуканих найкоротших довжин шляхів між всіма парами (i, j) вершин у графі G. Якщо для фіксованого m об'єднати всі довжини $l^m_{i,j}$ у матрицю, то задача полягає у послідовному обчисленні матриць $L_0, L_1, ..., L_{m-1}, L_m, ..., L_n$.

При цьому виявляється можливим обчислити матрицю L_m, використовуючи тільки лише безпосередньо попередню матрицю L_{m-1}. Дійсно, нехай відомі всі найкоротші довжини $l^{m-1}_{i,j}(i, j = 1, ..., n)$. Розіб'ємо множину всіх (i, j) — шляхів через вершини V_m на два класи: що обов'язково проходять через вершину m і що не проходять через m (тобто які проходять тільки через вершини з V_{m-1}). Найкоротший шлях у другому класі має довжину $l^{m-1}_{i,j}$. Найкоротший шлях через вершину m у першому класі складається з двох частин: (i, j) — шляхи довжини $l^{m-1}_{i,m}$ і (m, j) — шляхи довжини $l^{m-1}_{m,j}$. Тому найкоротший (i, j) — шлях через вершину m (з проміжними вершинами тільки з V_m) має довжину $l^{m-1}_{i,m} + l^{m-1}_{m,j}$. Отже,

$$l^m_{i,j} = \min\{l_{ij}{}^{m-1},\ l^{m-1}_{i,m} + l^{m-1}_{m,j}\}. \qquad (6.26)$$

Зрозуміло, що граф G не має контурів від'ємної довжини, інакше рекурентне правило (6.26) не буде справедливим!

Тепер нам залишається привести формальний опис алгоритму Флойда.

Опис алгоритму Флойда пошуку найкоротших шляхів між всіма парами вершин

Крок 1. Перенумерувати вершини графа G числами 1, 2, ..., n. Сформувати матрицю довжин $L_0 = \{l^0_{i,j}\}$, поклавши число $l^0_{i,j}$ при $i \neq j$, що дорівнює мінімальній довжині дуги

з вершини i до вершини j або $l^0_{i,j} = \infty$, якщо таких дуг немає. Якщо $i = j$, то $l^0_{i,j} = 0$.

Крок 2. Послідовно для $m = 1, 2, \ldots, n$, виходячи з елементів $l^{m-1}_{i,j}$ матриці L_{m-1}, обчислити елементи матриці $L_m = \{l^m_{i,j}\}$ за рекурентним правилом (6.26). Фіксувати «найкоротший» (i, j)-шлях, що має довжину $l^m_{i,j}$ при її обчисленні за формулою (6.26).

При $m = n$ алгоритм закінчує роботу. Числа $l^n_{i,j}$ є довжини шуканих найкоротших (i, j)-шляхів у графі G; $i, j = 1, 2, \ldots, n$.

Зауваження. При одержані матриці L_m (по L_{m-1}) значення елементів головної діагоналі $l^m_{i,j}$, елементів mтого рядку $l^m_{m,j}$ і m-го стовпця $l^m_{i,m}$ можна не обчислювати за формулою (6.26), а покласти відразу

$$l^m_{i,i} = 0, \quad l^m_{m,j} = l^{m-1}_{m,j}, \quad l^m_{i,m} = l^{m-1}_{i,m}.$$

Дійсно, вершина m не може виступати як проміжна для найкоротшого шляху, який або починається в m, або закінчується у вершині m.

Отже, за формулою (6.26) необхідно обчислювати лише $(n-1)(n-2)$ елементів матриці L_m. Разом з тим, *порядок складності* обчислення L_m для фіксованого m дорівнює n^2. Якщо врахувати, що кількість обчислювальних матриць L_m дорівнює n, то одержується складність алгоритму Флойда порядку $\theta(n^3)$.

Алгоритм Данцига пошуку всіх найкоротших шляхів

Тут використовуються ті ж поняття і позначення, що і в алгоритмі Флойда: $1, 2, \ldots, n$ — номера вершин; $V_m = \{1, 2, \ldots, m\}$ — множина перших m вершин; $l^m_{i,j}$ — довжина «найкоротшого» (i, j)-шляху з проміжними вершинами з V_m. Матриця $L_0 = \{l^0_{i,j}\}$ $(i, j = 1, 2, \ldots, n)$ «найкоротших дуг» між всіма парами вершин графа G має розмірність $n \times n$ і формується точно так, як в алгоритмі Флойда. Однак у подальшому замість послідовних $(n \times n)$-матриць L_m будуються їх «головні» підматриці D_m розмірності $m \times m$:

$$D_m = \{l^m_{i,j}\}_{(i,j=1,2,\ldots,m)}.$$

Матриця D_1 є число $l^1_{1,1} = 0$. У подальшому всі діагональні елементи також дорівнюють нулю: $l^m_{i,i} = 0$; $1 \le i \le m$; $m = 1, 2, \ldots, n$.

В матриці D_m всі елементи, що не входять до останнього рядка і в останній стовпець, тобто елементи $l^m_{i,j}$ при $1 \le i, j \le m-1$, обчислюються точно так, як в алгоритмі Флойда за формулою (6.26). Вони дорівнюють довжинам «найкоротших»

(i, j)-шляхів між вершинами $i, j \in V_{m-1}$ з проміжними вершинами з множини V_m.

Для одержання $l^m_{i,m}$ — довжини «найкоротшого» шляху з вершини $i \in V_{m-1}$ до вершини m через вершини V_m, зауважимо, що будь-який такий шлях $S^m_{i,m}$ не може містити вершину m як проміжну і кінцеву одночасно, оскільки за умовою будь-який контур має невід'ємну довжину.

Нехай $p(\in V_{m-1})$ — вершина на шляху $S^m_{i,m}$, яка передує m. Але тоді «найкоротший» шлях $S^m_{i,m}$ можна розбити на два шляхи: $S^{m-1}_{i,p}$ і $S^m_{p,m}$, де $S^m_{p,m}$ складається з однієї дуги (p, m) і, звичайно, довжина шляху $S^m_{p,m}$ дорівнює числу $l^0_{p,m}$ — елементу матриці L_0. Зрозуміло, шлях $S^{m-1}_{i,p}$ є найкоротшим (i, p)-шляхом через вершини V_{m-1}, тому його довжина дорівнює $l^{m-1}_{i,p}$ $(i, p \in V_{m-1})$. Число $l^{m-1}_{i,p}$ є елементом попередньої матриці D_{m-1}. За побудовою

$$l^m_{i,m} = l^{m-1}_{i,p} + l^0_{p,m}. \tag{6.27}$$

Оскільки «найкоротший» шлях $S^m_{i,m}$ нам невідомий, а відоме лише його існування, то його довжину (6.27) можна знайти як такий мінімум:

$$l^m_{i,m} = \min_{1 \le k \le m-1} \{l^{m-1}_{i,k} + l^0_{k,m}\}, \quad i = 1, 2, \ldots, m-1. \tag{6.28}$$

Таким чином, елементи $l^n_{i,m}$ стовпця матриці D_m обчислюються за елементами попередньої матриці D_{m-1} і матриці L_0.

Аналогічно для елементів останнього рядка матриці D_m, тобто для довжин $l^m_{m,j}$ «найкоротших» (m, j)-шляхів через вершини V_m одержуємо формулу:

$$l^m_{m,j} = \min_{1 \le k \le m-1} \{l^0_{m,k} + l^{m-1}_{k,j}\}, \quad j = 1, 2, \ldots, m-1. \tag{6.29}$$

Залишається навести формальний опис алгоритму Данцига.

Крок 1. Перенумерувати вершини графа G числами $1, 2, \ldots, n$. Сформувати матрицю L_0 розмірності $(n \times n)$, кожний елемент якої $l^0_{i,j}$ є довжина найкоротшої дуги з вершини i до вершини j, де $i \ne j$. Якщо таких дуг взагалі немає, то $l^0_{i,j} = \infty$ $(i \ne j)$. Якщо $i = j$, то $l^0_{i,i} = 0$.

Крок 2. Послідовно за допомогою рекурентної процедури визначити послідовність квадратних матриць

$$D_1, D_2, \ldots, D_{m-1}, D_m, \ldots, D_n$$

зростаючої розмірності. Розмірність матриці D_m дорівнює $m \times m$. Елементи $l^m_{i,j} (1 \le i, j \le m)$ матриці D_m обчислюються при $i \ne j$ через елементи матриць D_{m-1} і D_0 за формулами

(6.26) для $1 \le k \le m-1$, за формулами (6.28) — для $j = m$, за формулами (6.29) — для $i = m$. Нарешті, при $i = j$ припускається[1] $l^m_{i,i} = 0$.

Крок 3. Останній крок рекурентної процедури 2 дає $(n \times n)$-матрицю D_n довжин найкоротших шляхів між всіма парами вершин графа G.

Алгоритм Данцига має таку ж складність, як алгоритм Флойда — $\theta(n^3)$.

Завдання

1. За допомогою алгоритму Дейкстри побудувати остовне дерево найкоротших шляхів із вершини x_1 до вершин графа G на рис. 6.39. Кожне неорієнтоване ребро з вагою c_j розглядається як пара протилежно спрямованих дуг з однаковими вагами c_j.

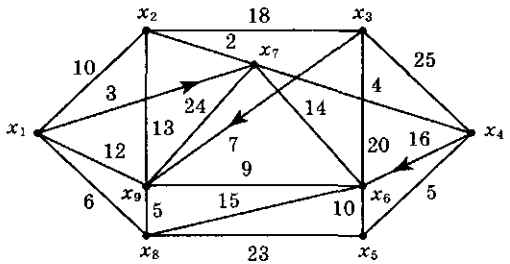


Рис. 6.39. Граф G для завдання 1
(побудова остовного дерева алгоритмом Дейкстри)

2. Дати повний опис алгоритму Форда за кроками, не спираючись на алгоритм Дейкстри.
3. Оцінити інформаційний об'єм формування «найкоротших» (i, j)-шляхів на кожній m-ій ітерації алгоритму Флойда. Розглянути приклад рис. 6.40.

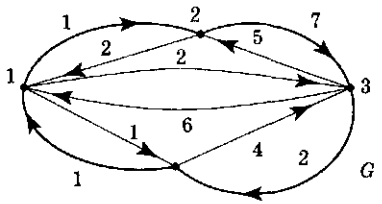


Рис. 6.40. Граф G для завдання 5
(пошук найкоротших шляхів алгоритмом Флойда)

[1] Зокрема, завжди одновимірна матриця D_1 є число нуль: $D_1 = 0$.

4. Показати, що для формування всіх найкоротших (i, j)-шляхів у графі G достатньо знати лише передостанню вершину x_{ij} найкоротшого (i, j)-шляху для всіх пар вершин (i, j) на кожному кроці алгоритму Флойда.
5. Використовуючи алгоритм Флойда, знайти найкоротші шляхи між всіма парами вершин графа G на рис. 6.40.
6. Знайти найкоротші шляхи між всіма парами вершин графа G на рис. 6.41, використовуючи алгоритм Данцига.

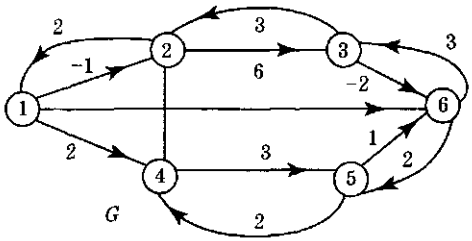


Рис. 6.41. Граф G для завдань 6 і 7

7. Порівняти роботу алгоритмів Флойда і Данцига пошуку найкоротших шляхів у графі G на рис. 6.41.
8. Чи впливає вибір нумерації вершин графа на ефективність алгоритмів Флойда і Данцига? Якщо впливає, то чому?
9. На нафтопереробному заводі є чотири місткості (A, $Б$, $В$, $Г$), і швидкість перекачки нафти з однієї місткості в іншу вказано у таблиці 6.4. Знайти два найкращих способи перекачки нафти з місткості A у місткість $Г$.

Таблиця 6.4. Швидкості перекачки нафти з 4-х місткостей

Місткість	A	$Б$	$В$	$Г$
A	0,00	0,13	0,14	0,15
$Б$	0,08	0,00	0,13	0,08
$В$	0,17	0,12	0,00	0,18
$Г$	0,10	0,06	0,13	0,00

6.11. Гамільтонові цикли і шляхи. Задача комівояжера

Джерела та визначення

Гамільтоновим циклом у графі $G = (X, Y)$ називається *простий цикл* $Q = (X, Y_0)$, *що містить всі вершини графа*, незалежно від того, чи є G орієнтованим. Вимога простоти циклу є принциповою: за гамільтоновим циклом Q можна

обійти всі вершини x графа G, відвідуючи кожну проміжну (тобто не початкову і не кінцеву) вершину тільки один раз. Сам граф G, в якому існує гамільтонів цикл, називається ***гамільтоновим графом***. Зрозуміло, що гамільтонів граф є зв'язним і, більш того, *двозв'язним*, оскільки між кожною парою вершин існує не менш ніж два різних простих ланцюга. При будь-якому $n \geq 3$ повний простий граф K_n є гамільтоновим. Повний двочастковий граф $K_{p,p}$ з рівнопотужними частками (однокольоровими множинами) також гамільтонів, див. $K_{2,2}$, $K_{3,3}$, K_5 на рис. 6.42.

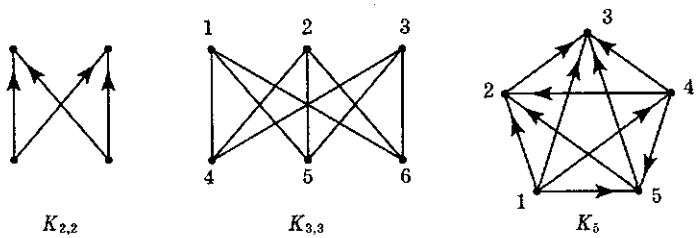


Рис. 6.42. Гамільтонові графи $K_{2,2}$, $K_{3,3}$, K_5

Безпосередньо перевіряється, що графи G_1, G_2, G_3 на рис. 6.43 не містять гамільтонових циклів і, отже, не є гамільтоновими графами. Однак всі ці графи містять гамільтонові ланцюги. ***Гамільтоновим ланцюгом*** у графі $G = (X, Y)$ називається *простий ланцюг* $Z = (X, Y_z)$, *що містить всі вершини графа*. Граф, що має гамільтоновий ланцюг, називається ***трасовним***. Гамільтоновим ланцюгом у графі G_1 є ланцюг 23514, у графі G_2 — ланцюг 623451, у графі G_3 — ланцюг 643215, а також 264315. Відзначимо, що граф G_3 не є двозв'язним, так що для трасування вимога двозв'язності графа G зайва.

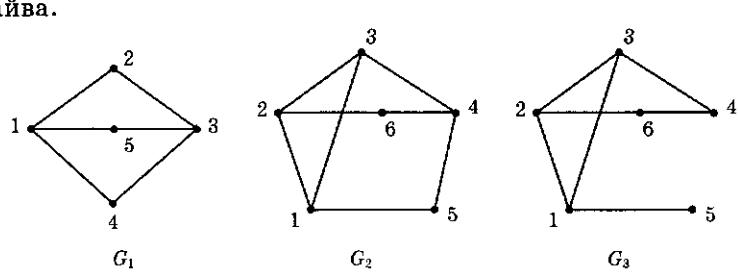


Рис. 6.43. Гамільтонові ланцюги

На рис. 6.44 зображено два негамільтонових графа $K_{2,3}$, $K_{2,4}$. Обидва вони двозв'язні. В $K_{2,3}$ існують гамільтонові ланцюги, наприклад, (4, 1, 5, 2, 3), (3, 1, 4, 2, 5), (4, 2, 5, 1, 3). В графі $K_{2,4}$ немає ані гамільтонових циклів, ані гамільтонових ланцюгів.

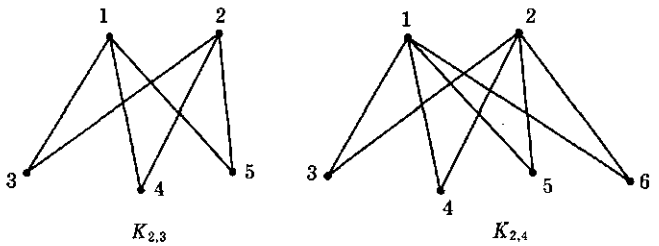


Рис. 6.44. Негамільтонові графи

Негамільтонові графи G_1, G_2, $K_{2,3}$ містять як підграфи так звані *тета-графи*, що мають хоча б дві вершини степеня 3, з'єднані трьома простими попарно різними ланцюгами довжини не менш двох. Наприклад, у графі G_1 ланцюги (1, 2, 3), (1, 5, 3), (1, 4, 3) утворюють тета-підграф з вершинами 1, 3 степеня 3. Справедлива

Теорема 6.10

Якщо двозв'язний граф не містить гамільтонового циклу, то він містить тета-підграф.

□ Нехай Q — простий цикл максимальної довжини у нашому графі $G = (X, Y)$. За умовою цикл Q не може бути гамільтоновим, тому число його вершин повинне бути менше, ніж $n = |X|$. Легко переконатися, що при $n = 3$ або $n = 4$ двозв'язний простий граф G є гамільтоновим, тому $n \geq 5$. Число вершин X_Q циклу Q (і ребер) не менше 4: $|X_Q| \geq 4$. Дійсно, якщо у двозв'язному графі G є цикл довжини 3, то до нього можна додати ланцюг з новою проміжною вершиною так, щоб виник простий цикл довжини більшої, ніж 3 (див. рис. 6.43, G_1). Існує таке ребро $(x, v) \in Y$, що $x \in Q$, $v \in Y \backslash Q$. Позначимо через a, b вершини циклу Q, суміжні з вершиною x (див. рис. 6.45). Оскільки цикл Q максимальний, то вершина v не суміжна ані з вершиною a, ані з вершиною b: у протилежному випадку можна побудувати цикл більшої довжини.

Видалимо з графа G вершину x разом з інцидентними їй ребрами, до множини яких належить і ребро (x, v). В графі, що залишився, для кожної вершини w на циклі Q (крім x)

існує простий (v, w) — ланцюг P_w, з'єднуючий вершини w і v. Серед всіх таких ланцюгів $P_w (w \in Q, w \neq x)$ оберемо найкоротший ланцюг P_c, з'єднуючий вершину v з деякою вершиною $c \in Q$. Очевидно, $c \neq a$ і $c \neq b$, інакше цикл Q не був би максимальним простим циклом у графі G. Ланцюг P_c не містить вершин циклу Q, відмінних від c, бо у протилежному випадку P_c не є найкоротший шлях із всіх ланцюгів P_w. Побудуємо у графі G підграф $G_0 = Q \cup (x, v) \cup P_c$, об'єднуючий цикл Q, ребро (x, v) і ланцюг P_c разом з інцидентними вершинами. Зрозуміло, що G_0 є тета-граф, у якому вершини третього степеня x і c з'єднуються трьома різними простими ланцюгами. ■

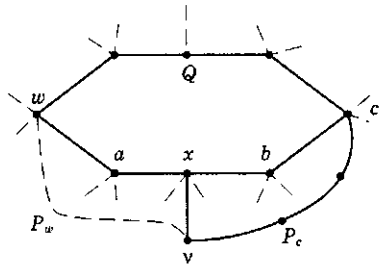


Рис. 6.45. Граф, що містить тетапідграф

Зауваження. Існування тета-підграфа є лише необхідною умовою для негамільтоновості двозв'язного графа. В двозв'язному графе $K_{3,3}$ рис. 6.42 існують тета-підграфи, однак $K_{3,3}$ — гамільтонів. Аналогічно граф G_2 на рис. 6.43 також має гамільтонів цикл (6, 2, 3, 1, 5, 4, 6) і одночасно має тета-підграф з трьома ланцюгами (2, 3, 4), (2, 6, 4), (2, 1, 5, 4) між вершинами 2, 4 третього степеня. З іншого боку, вимога двозв'язності у теоремі 6.10 істотна: негамільтонів граф G_3 на рис. 6.43 однозв'язний і не містить жодного тета-підграфа.

Постановка задачі про гамільтонів цикл з однократним проходженням всіх вершин графа може виглядати схожою або в якому-небудь сенсі двоїстою до постановки задачі про ейлерів цикл з однократним проходженням всіх ребер графа. Однак це не так. Задача про гамільтонів цикл, на жаль, не має на сьогодні ані повного теоретичного розв'язку, ані задовільного алгоритму відшукання циклу для не дуже маленьких n. Як ми бачили у розділі 6.3, і теоретичний розв'язок, і добрі алгоритми є для ейлерова циклу. Історично першим розглядав задачу обходу простим циклом всіх вершин

графа відомий ірландський математик У. Гамільтон у 1859 р. Він побудував низку таких циклів на ребрах додекаедра — правильного опуклого багатогранника з 20 вершинами і 12 п'ятикутними гранями. Плоске зображення додекаедра, яке можна трактувати як його проекцію на одну «розтягнуту» грань, зображено на рис. 6.46. Один з гамільтонових циклів зображено на рисунку жирною ламаною. Гамільтон трактував свою задачу у вигляді гри «Навколосвітня подорож», у якій пропонується обрати маршрут відвідування двадцяти міст на земній кулі, рухаючись по дорогах-ребрах додекаедра. Він навіть продав свою гру торговцеві іграшками за 25 гіней.

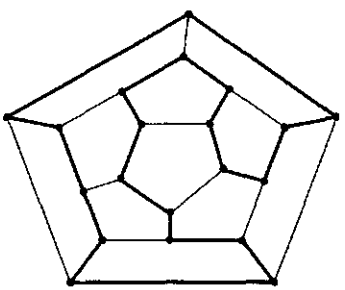


Рис. 6.46. Плоске зображення додекаедра

В орієнтованому графі поряд з гамільтоновим циклом або гамільтоновим ланцюгом часто доводиться шукати *гамільтонів контур* або *шлях*, обхід яких здійснюється тільки за *напрямками дуг*. ***Гамільтонів шлях*** — *це простий* (u, v)*-шлях у графі* G*, що містить всі вершини графа*. Якщо $u = v$, то це — ***гамільтонів контур***. Будь-який гамільтонів шлях є гамільтонів ланцюг, зворотне, взагалі кажучи, неправильно. Графи $K_{2,2}$ і K_5 на рис. 6.42 є гамільтоновими без врахування орієнтації дуг, однак вони не мають гамільтонових контурів. При цьому граф K_5 має гамільтонів шлях, у якому послідовність проходження вершин є 1, 4, 5, 2, 3. В графі $K_{2,2}$ рис. 6.42 гамільтонів шлях відсутній.

Подібна ситуація неможлива у симетричному графі: кожному гамільтоновому ланцюгу (циклу) неорієнтованого графа відповідає пара протилежно спрямованих гамільтонових шляхів (контурів) у відповідному орієнтованому симетричному графі.

Застосування гамільтонових ланцюгів і шляхів досить численні. В *теорії розкладів* окремі процедури (дії) зображуються

точками — вершинами x_1, x_2, ..., x_n, а з вершини x_i, веде дуга до x_j, якщо після виконання процедури x_i припустимий перехід до виконання процедури x_j. Гамільтонів шлях у такому модельному графі задає «розклад» послідовного виконання всіх «n» процедур.

В *дослідженні операцій* типовою є така ситуація. Є технічний пристрій (верстат, комп'ютер і т. п.) і n операцій (задач), кожну з яких пристрій здатен здійснити (вирішити) *після виконання відповідної перенастройки (перепрограмування)*. Точніше, при переході від i-ої задачі до j-ої задачі перепрограмування пристрою потребує витрати t_{ij} одиниць часу або іншого ресурсу. Необхідно знайти послідовність виконання завдань, при якій час кожного перепрограмування не перевершує заданого значення τ. Будується простий модельний граф з множиною вершин $\{1, 2, ..., n\}$, у якому дуга (i, j) присутня тоді і тільки тоді, коли $t_{ij} \le \tau$. Будь-який гамільтонів шлях у модельному графі задає шукану послідовність виконання завдань.

Розглянемо відому *задачу про книги*, в якій принцип моделювання за допомогою графа є досить повчальним і може застосовуватися для розв'язку інших задач. Для виробництва n книг друкар має дві машини: друкарську та для переплетення. На друкування k-ї книги витрачається a_k часу, на її переплетення витрачається b_k часу. Зрозуміло, що у переплетення k-а книга надходить після її друкування, тому деякий час з початку роботи верстат для переплетення може простоювати. Знайти порядок роботи з книгами, при якому час виконання всього замовлення на n книг буде мінімальним. Неважко побачити, що, маючи оптимальний порядок роботи, можна перебудувати його без зменшення повного часу виконання замовлення так, щоб книги надходили на машину для переплетення в тій же послідовності, в якій вони надходять на друкарську машину. Отже, оптимальний порядок виготовлення множини книг $(x_1, x_2, ..., x_n)$ визначається деяким переставленням $(x_{i_1}, x_{i_2}, ..., x_{i_n})$. Можна показати, що при оптимальному порядку i-а книга повинна буде надійти до друку раніше k-ї книги (не обов'язково підряд), якщо і тільки якщо $\min\{a_i, b_k\} \le \min\{a_k, b_i\}$. Тепер зрозуміло, як побудувати модельний граф. На множині вершин $\{x_1, x_2, ..., x_n\}$ проводиться дуга (x_i, x_k), якщо виконано зазначену нерівність. Гамільтонів шлях у цьому графі вказує шуканий оптимальний порядок виготовлення книг.

Ознаки існування гамільтонових циклів, шляхів і контурів

Одержання умов гамільтоновості графів є популярною задачею, до кінця не розв'язаною на сьогодні. Популярність «живиться» не тільки прикладною цінністю умов, але і рідкісною простотою та природністю самої постановки задачі про гамільтонів цикл, що не вимагає попередніх математичних знань і символьних позначень у формулюванні.

Повний граф має гамільтонів цикл, тому, висловлюючись нестрого, якісно можна припустити, що чим більше ребер у графі і чим більше «рівномірно» вони розподілені, тим вище ймовірність існування гамільтонова циклу. Наведені достатні умови гамільтоновості графа підтверджують це припущення.

Всі графи припускаються *зв'язними і простими.*

Теорема 6.11 ***(Г. Дірак, 1952 р.)***

Якщо число n вершин графа G не менше трьох і степінь $\delta_i = \deg x_i$ *будь-якої вершини* x_i *не менше* $\frac{n}{2}(\delta_i \geq \frac{n}{2})$, *то граф G є гамільтоновим.*

Сформульована ознака Дірака є очевидним наслідком більш загальної ознаки гамільтоновості, що встановлена у 1960 р. Оре.

Теорема 6.12 ***(О. Оре, 1960 р.)***

Якщо у графі G з n вершинами ($n \geq 3$) *сума степенів будь-яких двох вершин u, v є не меншою, ніж n* ($\deg u + \deg v \geq n$), *то граф G гамільтонів.*

В свою чергу, ознаку гамільтоновості Оре можна вивести з більш «пізніх» ознак Л. Поша і В. Хватала.

Теорема 6.13 ***(В. Хватал, 1972 р.)***

Нехай для упорядкованої за зростанням послідовності степенів вершин $\delta_i = \deg x_i$

$$\delta_1 \leq \delta_2 \leq \ldots \leq \delta_n \qquad (6.29)$$

графа G виконані імплікації

$$(\delta_k \leq k) \Rightarrow (\delta_{n-k} \geq n-k), \quad \forall k : 1 \leq k < \frac{n}{2},$$

тоді G — гамільтонів граф.

Існують ознаки гамільтоновості графів, що є похідними від деяких інших графів, зокрема, для степенів G^p і реберних графів $L(G)$. Нагадаємо, що вершини реберного (або дуального) графа $L(G)$ знаходяться у взаємно однозначній відповідності з ребрами графа G, а дві вершини у $L(G)$ з'єднані ребром, якщо і тільки якщо відповідні ребра у G суміжні. Степінь G^p графа $G = (X, Y)$ тут розуміється як граф з тією ж множиною вершин X, в якій вершини w, $v \in X$ з'єднані ребром (суміжні), якщо і тільки якщо у G відстань між w і v не більше p: $d(w, v) \le p$ у графі G. Наприклад, якщо C_5 — цикл з п'ятьма вершинами і п'ятьма ребрами, то $(C_5)^2 = K_5$ — повний п'ятивершинний граф. Цей приклад і саме визначення G^p підказує, що у будь-якого зв'язного графа G деякий степінь G^p повинен бути гамільтоновим графом. Із гамільтоновості степеня G^p виходить гамільтоновість G^{p+1}.

Яка нижня грань степенів p, що забезпечують гамільтоновість G^p?

Теорема 6.14 ***(Д. Караганіс, 1968 р.)***

Для зв'язного графа G з числом вершин $n \ge 3$ степінь G^3 є гамільтоновим графом.

Теорема 6.15 ***(Г. Флейшнер, 1971 р.)***

Якщо G — двозв'язний граф з числом вершин $n \ge 3$, то G^2 — гамільтонів граф.

Теорема 6.16 ***(Ф. Харарі, С. Неш-Вільямс, 1965 р.)***

Реберний граф $L(G)$ гамільтонів тоді і тільки тоді, коли у G існує цикл, що містить хоча б по одній вершині з кожного ребра графа G.

Наслідок. *Якщо граф G або ейлерів, або гамільтонів, то реберний граф $L(G)$ гамільтонів.*

Яка різниця між задачами про пошук гамільтонового циклу і пошук гамільтонового ланцюга у графі? З одного боку, гамільтонів граф завжди містить незамкнений гамільтонів ланцюг. З іншого боку, приклади графів G_1, G_3 на рис. 6.43 свідчать, що гамільтонові ланцюги можуть існувати у негамільтоновому графі, тому клас гамільтонових графів міститься у класі трасовних графів, не співпадаючи з ним. Указані дві задачі мають однакову складність, а будь-який трасовний граф G досить просто укладається у будь-який

гамільтонів граф G_v, множина вершин якого на одну вершину v перевершує множину вершин вихідного графа G. Саме, якщо (a, b)-ланцюг Z є гамільтоновим ланцюгом у трасовному графі $G = (X, Y)$, то додавання однієї нової вершини v і двох нових ребер (a, v), (b, v) призводить до графа $G_v = (X \cup v, Y \cup (b, v) \cup (v, a))$ з гамільтоновим циклом $Z \cup (b, v) \cup (v, a)$.

Наприклад, негамільтонів трасовний граф G_1 на рис. 6.43 є підграфом гамільтонового графа G_v на рис. 6.47, де додані ребра зображено пунктиром. При цьому пара доданих ребер (b, v), (v, a) «замикає» конкретний гамільтонів ланцюг Z у графі G. Щоб не зв'язувати вкладення трасовного графа у гамільтонів з визначеним гамільтоновим ланцюгом, часто нову вершину v з'єднують зі всіма вершинами вихідного графа G і одержують граф G_{dv} з *домінуючою вершиною* v. Тоді будь-який гамільтонів ланцюг у G доповнюється до відповідного гамільтонового циклу в G_{dv} (рис. 6.47).

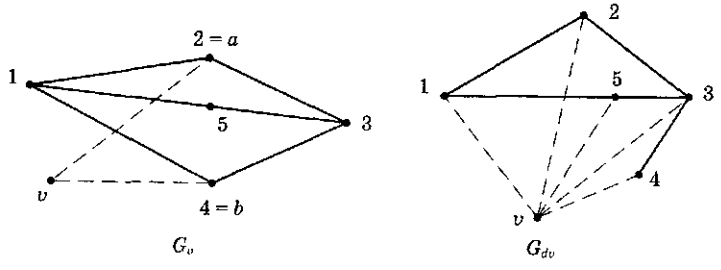


Рис. 6.47. Доповнення до гамільтонового графа G_v і до гамільтонових циклів G_{dv}

Ознаки існування *гамільтонових шляхів* і *контурів* в орієнтованому графі є сенс розглядати тільки для *несиметричних орграфів*. Однією з найпростіших є ознака Кеніга.

Теорема 6.17 ***(Кеніг)***

В повному орграфі G (будь-яка пара вершин якого з'єднується хоча б в одному напрямку) завжди існує гамільтонів шлях[1].

□ Виділимо у G простий (x_1, x_r)-шлях Z довжини $(r - 1)$ з послідовністю попарно різних вершин $[x_1, x_2, ..., x_r]$ і покажемо, що з його допомогою можна одержати простий шлях

[1] Але не гамільтонів контур (див. орграф K_3, рис. 6.42).

довжини r з попарно різними вершинами. Нехай v — вершина графа $G = (X, Y)$, що не належить шляху Z. Припустимо, що для будь-якого $k \in \{1, 2, 3, ..., r - 1\}$ не існує шляху в G довжини r з послідовністю вершин $[x_1, x_2, ..., x_k, v, x_{k+1}, ..., x_r]$. Тоді при кожному k справедливі імплікації

$$(x_k, v) \in Y \Rightarrow (v, x_{k+1}) \notin Y \Rightarrow (x_{k+1}, v) \in Y, \qquad (6.30)$$

причому остання імплікація випливає з повноти графа G. Якщо існує дуга $(v, x_1) \in Y$, то ми відразу одержуємо шуканий шлях довжини r з послідовністю вершин $[v, x_1, x_2, ..., x_r]$. Якщо дуга (v, x_1) відсутня в G, то існує дуга (x_1, v), і в силу (6.30) послідовно одержуємо, що граф G містить дуги $(x_2, v) \in Y$, $(x_3, v) \in Y$, ..., $(x_r, v) \in Y$. Але тоді існує простий шлях довжини r з послідовністю попарно різних вершин $[x_1, x_2, ..., x_r, v]$ (рис. 6.48).

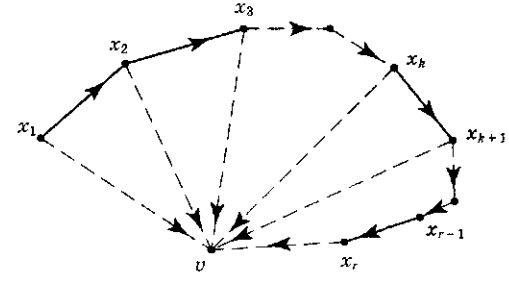


Рис. 6.48. Послідовність попарно різних вершин

Повторюючи процедуру збільшення шляху скінченне число раз, одержимо простий шлях довжини $n - 1$ ($n = |X|$), що містить всі вершини з X. Це і є шуканий гамільтонів шлях. ■

Наслідок. *Учасників турніру, в якому відбувається зустріч кожної пари і за правилами неможливі нічийні результати, завжди можна після турніру упорядкувати (ранжувати) так, щоб кожний попередній був переможцем безпосередньо наступного.*

Відзначимо деякі алгебраїчні властивості графів, що мають гамільтонів шлях або гамільтонів контур. Для довільної підмножини S вершин орграфа $G = (X, Y)$ визначимо множину S_+ всіх тих вершин v, в які ведуть дуги з вершин множини S:

$$S_+ = \{v \in X: (x, v) \in Y, \forall x \in S\}.$$

Дефіцитом простого графа $G = (X, Y)$ називається число

$$\delta_0 = \max_{S \subset X}(|S| - |S_+|). \qquad (6.31)$$

Звичайно, $|S|$ позначає число елементів множини S.

Наступна теорема містить алгебраїчні умови, виконання яких необхідно для існування гамільтонового контуру і гамільтонового шляху в орграфі.

Теорема 6.18

Якщо у графі існує гамільтонів контур, то дефіцит графа (6.31) дорівнює нулю: $\delta_0 = 0$. *Якщо існує гамільтонів шлях, то* $0 \le \delta_0 \le 1$.

Задача комівояжера

Загальна[1] задача комівояжера полягає у такому: використовуючи задану систему транспортних сполучень (доріг і т. п.) між пунктами (містами, фірмами і т. п.) у конкретній зоні обслуговування, відвідати всі пункти у такій послідовності, щоб пройдений маршрут був найкоротшим із всіх можливих.

На мові теорії графів або мереж загальна задача комівояжера має таке формулювання: у зваженому зв'язному графі G знайти найкоротший маршрут, що проходить через всі вершини графа. В постановці задачі можлива додаткова вимога *замкненості* маршруту комівояжера (повернення комівояжера у пункт вихідного перебування). Зрозуміло, що у будь-якому зв'язному графі G загальна задача комівояжера (і замкнена і незамкнена) завжди має розв'язок.

Існує ще одна постановка, в якій *додатково* до попередньої задачі треба, щоб кожний пункт обслуговування комівояжер відвідував *тільки один раз*. Очевидно, в такій постановці задача комівояжера не завжди має розв'язок, а якщо має, то маршрут комівояжера в графі G є найкоротшим гамільтоновим ланцюгом або циклом. У зв'язку із сказаним будемо називати останню постановку ***«гамільтоновою» задачею комівояжера*** (на відміну від «загальної»). Якщо шукається *замкнений* маршрут, то для розв'язності «гамільтонової» задачі комівояжера необхідно і достатньо, щоб граф G був гамільтоновим, для

[1] На відміну від «гамільтонової» задачі комівояжера.

незамкненого гамільтонового маршруту — граф G повинен бути *трасовним.*

В *орієнтованому* зваженому графі G зустрічається також задача пошуку ***загального орієнтованого маршруту комівояжера*** — найкоротшого орієнтованого маршруту, що містить всі вершини графа, і відповідно, ***орієнтованого шляху*** або ***контуру комівояжера*** (що містить кожну вершину один раз). Замкнений орієнтований (або неорієнтований) маршрут комівояжера при загальній негамільтоновій постановці задачі не обов'язково є гамільтоновим контуром (відповідно гамільтоновим циклом). Наприклад, граф G на рис. 6.49 має один гамільтонів контур і кілька гамільтонових циклів, всі їх довжини дорівнюють 24, і кожний містить три дуги. Однак найкоротшим замкненим ормаршрутом комівояжера (непростим) є замкнений маршрут з чотирма дугами (a, b), (b, a), (a, c), (c, a), який проходить через кожну вершину двічі і має довжину 8.

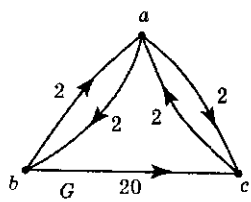


Рис. 6.49. (a, b), (b, a), (a, c), (c, a) — найкоротший замкнений ормаршрут

В яких випадках гамільтонів контур (або цикл) є розв'язком загальної задачі комівояжера?

Теорема 6.19

Якщо функція $d(x, v)$ ваг (довжин) ребер (або дуг) між парами вершин x, v у графі $G = (X, Y)$ з числом вершин $n = |X| \geq 3$ задовольняє нерівність трикутника

$$d(x, v) \leq d(x, z) + d(z, v), \quad \forall z \neq x,\ z \neq v,\ z \in X, \quad (6.32)$$

та існує розв'язок загальної задачі комівояжера, то існує також розв'язок «гамільтонової» задачі комівояжера.

Умова трикутника означає, що у графі G довжина «однокрокового» шляху[1] (ланцюга) між вершинами x і v, скінченна або нескінченна, *не перевершує* довжини будь-якого «двохкрокового» (x, v)-шляху ((x, v)-ланцюга) з однією проміжною вершиною.

□ Нехай Q — оптимальний замкнений маршрут комівояжера, для визначеності — орієнтований, і припустимо, Q не є

[1] Тобто шляху (ланцюга) $l(x, v)$ без проміжних вершин між x і v; він складається або з ребра (x, v), або відсутній.

гамільтоновим контуром у графі G. Тоді існує вершина $z \in X$, яка повторюється при обході контуру Q, щонайменше, двічі. Припустимо, при першому проходженні комівояжер потрапляє у вершину z з попередньої вершини x, а виходить з z до наступної вершини v. За умовою теореми повинен існувати «однокроковий» шлях $l(x, v)$ за дугою (x, v), який не довше двохкрокового шляху (x, z, v) (рис. 6.50). Отже, у маршруті Q два однократних проходження дуг (x, z), (z, v) можна замінити одним проходженням дуги (x, v).

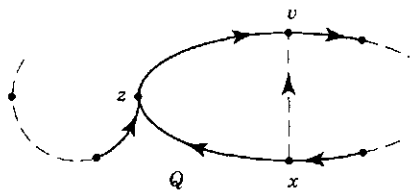


Рис. 6.50. Заміна двох однократних проходжень дуг (x, z), (z, v) на одне проходження дуги (x, v)

В результаті одержується замкнений маршрут комівояжера Q_1 довжини $d(Q_1) \le d(Q)$ з числом кроків, тобто числом проходжень дуг, меншим, ніж у маршруті Q. При цьому число проходжень вершини z зменшується на одиницю (точніше, на один захід і один вихід), а число проходжень решти вершин графа зберігається. Застосовуючи цей процес до маршруту Q_1, ми побудуємо замкнений ормаршрут комівояжера Q_2 довжини $d(Q_2) \le d(Q_1)$ з числом проходжень дуг, меншим, ніж у Q_1, і т. д. Після скінченного числа кроків ми одержимо замкнений маршрут комівояжера Q_p, у якому кожна вершина зустрічається точно один раз. Отже, Q_p є гамільтонів контур у графі G. ■

Якщо граф G не задовольняє нерівність трикутника (6.32), то формально загальну задачу комівояжера необхідно розв'язувати окремо від гамільтонової задачі комівояжера. Однак фактично *можна розв'язок загальної задачі зводити до розв'язку «гамільтонової»*. Для цього достатньо простим перетворенням перебудувати граф G і одержати граф G_t, що задовольняє нерівність (6.32) за допомогою заміни у G довжини $d(x, v)$ кожної дуги (x, v), що не задовольняє (6.32), на довжину найкоротшого шляху з x до v. Нехай Q_t — розв'язок «гамільтонової» задачі комівояжера у графі G_t. Кожну дугу[1] (v, w) з гамільто-

[1] Включаючи відсутню дугу з формальною довжиною ∞.

нового контуру Q_t, довжина якої була раніше зменшена при переході $G \to G_t$, слід замінити найкоротшим (v, w)-шляхом з графа G. Одержаний таким чином маршрут Q є оптимальним розв'язком загальної задачі комівояжера у графі G.

Наприклад, у графі G на рис. 6.49 дуга (b, c) не задовольняє умову (6.32): $20 > 2 + 2$. Тому слід замінити довжину 20 дуги (b, c) на довжину $4 = 2 + 2$ найкоротшого (b, c)-шляху (b, a, c). В одержаному зваженому графі G_t (топологічно він співпадає з G, відрізняються лише ваги дуги (b, c)) гамільтонів контур (a, b, c, a) має довжину $2 + 4 + 2 = 8$ і є оптимальним маршрутом комівояжера у графі G_t. Заміняючи у Q_t «скорочену» дугу (b, c) найкоротшим (b, c)-шляхом (b, a, c), одержуємо оптимальний у графі G ормаршрут комівояжера (a, b, a, c, a) довжини 8, що включає чотири дуги і не є гамільтоновим.

Практичні методи і алгоритми

В самому несприятливому випадку (повний граф і фатальна невдача) оптимальний гамільтонів ланцюг або цикл у графі G, що починається у заданій вершині x_1, можна побудувати шляхом різних перестановок на множині решти вершин $\{x_2, x_3, ..., x_n\}$. Отже, обчислювальна складність задач Гамільтона і комівояжера має порядок $(n - 1)!$ Внаслідок такої значної обчислювальної складності для розв'язку задачі комівояжера створено багато алгоритмів, як автономних, так і таких, що базуються на інших оптимізаційних алгоритмах, — потокових, що будують мінімальний остів тощо. При цьому точні алгоритми, що гарантують одержання маршруту комівояжера у будь-якому випадку, є трудомісткими і застосовні до мереж не дуже високої розмірності. Наближені алгоритми у більшості випадків застосовні до більш громіздких мереж, однак іноді призводять до неоптимального маршруту [10, 42, 47]. Розглянемо коротко два алгоритми.

Алгебраїчний алгоритм Йоу, Даніельсона, Дхавана

В алгоритмі послідовно будуються всі прості шляхи простого орграфа G за допомогою послідовного перемноження $(n \times n)$-матриць, що містять символи вершин $x_1, x_2, ..., x_n$. Нехай $A = \{a_{ij}\}_1^n$ — матриця суміжності орграфа G. Через $V = \{v_{ij}\}_1^n$ позначимо так звану *модифіковану матрицю суміжності графа* G, в якій $v_{ij} = x_j$, якщо існує дуга з x_i до x_j, і $v_{ij} = 0$ — якщо

такої дуги немає. Тобто, на місці одиниць у матриці суміжності A ставляться символи x_j вершин, в які заходять відповідні рядкам дуги:

$$a_{ij} = 1 \Leftrightarrow v_{ij} = x_j; \qquad a_{ij} = 0 \Leftrightarrow v_{ij} = 0.$$

«*Внутрішнім добутком вершин*» шляху $(x_1, x_2, x_3, \ldots, x_{k-1}, x_k)$ називається формальний алгебраїчний вираз (слово) $x_2 \cdot x_3 \cdot \ldots \cdot x_{k-1}$, що не містить початкової і кінцевої вершин шляху. При $k = 2$ добуток вважається рівним 1.

Запропонований алгоритм будує послідовність матриць

$$P_1, P_2, \ldots, P_{n-1}, V \cdot P_{n-1},$$

де $P_s = \{p_s(i, j)\}_{i,j=1}^{n}$ — матриця, елемент якої $p_s(i, j)$ дорівнює сумі внутрішніх добутків *всіх простих шляхів* з x_i до x_j довжини $s (1 \le s \le n-1)$, якщо $i \ne j$ і $p_s(i, i) = 0$. Зрозуміло, що $P_1 = A$ (матриця суміжності). Якщо матриця P_s вже обчислена, то для одержання P_{s+1} спочатку будується матриця

$$P^0_{s+1} = V \cdot P_s = \{p^0_{s+1}(i, j)\}, \quad p^0_{s+1}(i, j) = \sum_k V_{ik} \cdot p_s(k, j). \qquad (6.33)$$

Її елемент $p^0_{s+1}(i, j)$ дорівнює сумі внутрішніх добутків *всіх таких ланцюгів* (не тільки простих!) з вершини x_i до вершини x_j довжини $s+1$, серед яких *непростими* є в точності ті ланцюги, внутрішні добутки яких містять у чинниках $p_s(k, j)$ з (6.33) вершину x_i. Виключивши з суми у (6.33) доданки, що містять x_i, одержуємо алгебраїчний вираз $p_{s+1}(i, j)$, що дорівнює сумі внутрішніх добутків всіх *простих* (x_i, x_j) — шляхів довжини $s+1$, де $i \ne j$. Вважаючи $p_{s+1}(i, i) = 0$, одержимо шукану матрицю P_{s+1} всіх простих незамкнених шляхів довжини $s+1$.

Нарешті, при $s = n-1$ матриця P_{n-1} дає всі *гамільтонові шляхи* (що мають довжину $n-1$) у графі G між всіма парами вершин. Гамільтонові контури одержуються додаванням дуги (якщо вона існує), що з'єднує кінець гамільтонового шляху з його початком. Інакше кажучи, гамільтонові контури перелічуються членами внутрішніх добутків вершин, що містяться на діагональних елементах матриці $V \cdot P_{n-1}$.

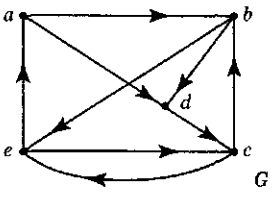


Рис. 6.51. Орграф G

Проілюструємо роботу алгоритму на прикладі орграфа G рис. 6.51 з множиною вершин $X = \{a, b, c, d, e\}$.

Матриця суміжності A і модифікована матриця суміжності V мають вигляд:

$$A=\begin{array}{c}\\ a\\ b\\ c\\ d\\ e\end{array}\!\!\begin{array}{c}\begin{array}{ccccc}a & b & c & d & e\end{array}\\ \begin{bmatrix}0&1&0&1&0\\0&0&0&1&1\\0&1&0&0&1\\0&0&1&0&0\\1&0&1&0&0\end{bmatrix}\end{array};\quad V=\begin{array}{c}\\ a\\ b\\ c\\ d\\ e\end{array}\!\!\begin{array}{c}\begin{array}{ccccc}a & b & c & d & e\end{array}\\ \begin{bmatrix}0&b&0&d&0\\0&0&0&d&e\\0&b&0&0&e\\0&0&c&0&0\\a&0&c&0&0\end{bmatrix}\end{array}.$$

Вважаємо $P_1 \equiv A$. Обчислюючи матрицю $P_2^0 = V \cdot P_1$ і замінюючи в ній підкреслені діагональні елементи нулями, одержуємо P_2. Викреслені з діагоналі вершини e, c є проміжними у 2-контурі (c, e, c) і (e, c, e) відповідно.

$$P_2^0=\begin{array}{c}\\ a\\ b\\ c\\ d\\ e\end{array}\!\!\begin{array}{c}\begin{array}{ccccc}a & b & c & d & e\end{array}\\ \begin{bmatrix}0&0&d&b&b\\e&0&d+e&0&0\\e&0&\underline{e}&b&b\\0&c&0&0&c\\0&a+c&0&a&\underline{c}\end{bmatrix}\end{array};\quad P_2=\begin{array}{c}\\ a\\ b\\ c\\ d\\ e\end{array}\!\!\begin{array}{c}\begin{array}{ccccc}a & b & c & d & e\end{array}\\ \begin{bmatrix}0&0&d&b&b\\e&0&d+e&0&0\\e&0&0&b&b\\0&c&0&0&c\\0&a+c&0&a&0\end{bmatrix}\end{array}.$$

Далі знаходимо $P_3^0 = V \cdot P_2$ і після заміни діагональних елементів нулями — матрицю простих 3-шляхів P_3

$$P_3^0=\begin{array}{c}\\ a\\ b\\ c\\ d\\ e\end{array}\!\!\begin{array}{c}\begin{array}{ccccc}a & b & c & d & e\end{array}\\ \begin{bmatrix}\underline{be}&dc&bd+be&0&dc\\0&\underline{dc+ea+ec}&0&ea&dc\\be&ea+\underline{ec}&\underline{bd+be}&ea&0\\ce&0&0&\underline{cb}&cb\\\underline{ce}&0&ad&ab+cb&\underline{ab+cb}\end{bmatrix}\end{array};$$

$$P_3=\begin{array}{c}\\ a\\ b\\ c\\ d\\ e\end{array}\!\!\begin{array}{c}\begin{array}{ccccc}a & b & c & d & e\end{array}\\ \begin{bmatrix}0&dc&bd+be&0&dc\\0&0&0&ea&dc\\be&ea&0&ea&0\\ce&0&0&0&cb\\0&0&ad&ab+cb&0\end{bmatrix}\end{array}.$$

Викреслені у матриці P_3^0 діагональні елементи відповідають 3-контурам, які з відповідної причини не можуть бути далі частинами простих 4-шляхів. Викреслені у P_3^0 елементи ec, ce на місцях (3, 2), (5, 1) відповідають непростим шляхам (c, e, c, b), (e, c, e, a) відповідно, точніше, вершини c, e є внутрішніми (проміжними) у цих шляхах. У матриці P_3 значення $(bd + be)$ елемента (1, 3) означає, що послідовність вершин $\{bd\}$ є внутрішньою у простому 4-шляху (a, b, d, c) і послідовність вершин $\{b, e\}$ є внутрішньою у простому 4-шляху (a, b, e, c).

В матриці $P_4^0 = V \cdot P_3$ підкреслені діагональні елементи відповідають 4-контурам у графі G, а підкреслені недіагональні елементи — непростим 4-шляхам, що містять контур довжини 3 або 2 (довжину контуру вказує відстань від початкової вершини шляху до місця її повторення у внутрішній послідовності). Після заміни у P_3^0 підкреслених елементів нулями одержується матриця P_4 всіх гамільтонових шляхів графа G, оскільки тут $4 = n - 1$. Загальна кількість гамільтонових 4-шляхів дорівнює 10: це число всіх ненульових доданків у матриці P_4. Якщо гамільтонові шляхи (a, b, d, c, e) і (a, d, c, b, e), що відповідають елементу (1, 5) матриці P_4, доповнити дугою (e, a), то ми одержимо два *гамільтонових контури графа* G: $Q_1 = (a, b, d, c, e, a)$, $Q_2 = (a, d, c, b, e, a)$.

$P_4^0 =$

	a	b	c	d	e
a	$\underline{dce}$	0	0	$\underline{bea}$	$bdc+dcb$
b	dce	0	ead	$\underline{eab}+\underline{ecb}$	$\underline{dcb}$
c	0	0	$\underline{ead}$	$bea+\underline{ecb}+eab$	$\underline{bdc}$
d	cbe	cea	0	$\underline{cea}$	0
e	$\underline{cbe}$	$adc+\underline{cea}$	$abd+\underline{abe}$	$\underline{cea}$	$\underline{adc}$

;

$P_{n-1} = P_4 =$

	a	b	c	d	e
a	0	0	0	0	$bdc+dcb$
b	dce	0	ead	0	0
c	0	0	0	$bea+eab$	0
d	cbe	cea	0	0	0
e	0	adc	abd	0	0

.

Якщо будь-який з решти восьми гамільтонових 4-шляхів доповнити відповідною дугою, одержується один з двох контурів Q_1 або Q_2, так що інших гамільтонових контурів у графі G

немає. Доповнення всіх десяти незамкнених гамільтонових шляхів до гамільтонового контуру Q_1 або Q_2 перелічуються за допомогою внутрішніх добутків вершин, що стоять на головній діагоналі матриці $V \cdot P_4 = \{\gamma_{ik}\}$. Наприклад, елемент $\gamma_{22} = dcea + eadc$ відповідає другій вершині «b» і гамільтоновим контурам (b, d, c, e, a, b) і (b, e, a, d, c, b), з яких перший співпадає з Q_1, другий — з Q_2. Таким чином, *кожний діагональний елемент матриці* $V \cdot P_{n-1}$ *перелічує всі гамільтонові контури графа за* допомогою внутрішніх добутків, що виникають при обході контурів з початком у тій вершині, якій відповідає діагональний елемент γ_{ii}.

Якщо не треба знаходити всі незамкнені гамільтонові шляхи, а необхідно лише перелічити *всі гамільтонові контури графа*, то *алгебраїчний алгоритм* можна декілька спростити, скоротивши об'єм обчислень орієнтовно у n разів. А саме, достатньо обчислити лише один діагональний елемент матриці $V \cdot P_{n-1}$, наприклад, γ_{11}, а для цього можна знаходити і зберігати у пам'яті не всі елементи матриць P^0_{n-1}, P_{n-1}; P^0_{n-2}, P_{n-2}; ...; P_1, *а лише їх перші стовпці.*

Зауваження. Після одержання всіх гамільтонових незамкнених шляхів або гамільтонових контурів розв'язок задачі комівояжера[1] (відповідно незамкнений або замкнений) зводиться до порівняння ваг шляхів або контурів і знаходження мінімальної ваги. Описаний алгебраїчний метод розв'язку задач Гамільтона і комівояжера завжди призводить до точного розв'язку, однак є досить трудомістким, припускає використання обчислювальних або програмних засобів високого рівня, що ефективно оперують з великими об'ємами символьних перетворень.

Завдання

1. Підрахувати кількості всіх гамільтонових циклів у графах K_4, K_5 і перелічити їх.
2. За яких умов повний дводольний граф $K_{p,q}$ є гамільтоновим? Коли $K_{p,q}$ є трасовним?
3. Знайти тета-підграфи у негамільтонових графах $K_{2,3}$, $K_{2,4}$ на рис. 6.44.
4. Нехай гратчастий граф G утворений p горизонтальними лініями і q вертикальними. Кожна точка перерізу вважається

[1] Розуміється, у гамільтоновій постановці (див. перехід до графу G_t в кінці попереднього розділу).

вершиною графа, а кожний відрізок між сусідніми точками перетину — ребром графа G. При яких p і q граф G є гамільтоновим?

5. Показати, що множина оптимальних розв'язків задачі комівояжера не залежить від того, яка з вершин обирається як початкова при обході.

6.12. Течії у мережах

В цьому розділі буде розглянуто зважений граф — граф, кожній дузі якого приписана течія деякої речовини. Такий граф є зручною моделлю при дослідженні цілого ряду задач у транспорті, зв'язку та інших областях, зв'язаних з дійсним або уявним рухом товарів, інформації або людей. Введемо визначення і позначення, типові для цього кола питань.

Мережею будемо називати орієнтований зв'язний граф без петель і паралельних ребер. Зауважимо, що течії у неорієнтованих графах можна зобразити у вигляді течій у відповідних орієнтованих; течії у незв'язних графах можуть вивчатися покомпонентно; течія у петлі не впливає на розподіл течії між вершинами.

Розглянемо мережу $G = (X, Y)$, $|X| = n$, $|Y| = m$. Нехай кожній дузі $y_j \in Y$ поставлено у відповідність невід'ємне дійсне число c_j, що інтерпретується як ***пропускна здатність (місткість) дуги*** y_j. Позначимо через $x_i \to X$ множину дуг, що виходять з вершини x_i, через $X \to x_i$ — множина дуг, що заходять до вершини x_i.

*Течією у мережі G **з вершини** x_s **до вершини** x_t **величини*** v називається невід'ємна, визначена на дугах y_j, функція φ: $Y \to R_+ \cup \{0\}$, така, що

$$\sum_{y \in x \to X} \varphi(y) - \sum_{y \in X \to x} \varphi(y) = \begin{cases} v, & x = x_s \\ 0, & x \neq x_s, x_t, \\ -v, & x = x_t \end{cases} \tag{6.34}$$

$$\varphi(y_j) \le c_j,\ j = 1, \ldots, m. \tag{6.35}$$

Вершина x_s називається *джерелом*, вершина x_t — ***стоком***, а решта вершин — ***проміжними вузлами***. Число $Q(x_i) = \sum_{y \in x_i \to X} \varphi(y) - \sum_{y \in X \to x_i} \varphi(y)$ називається ***чистою течією*** з вершини x_i відносно φ. Число $\varphi(y)$ називається ***течією за дугою*** y. Зауважимо, що якщо «реальна» течія за дугою від'ємна, то її можна

зробити додатною, обравши відповідну орієнтацію дуги y. Систему рівнянь (6.34) можна переписати у векторному вигляді:

$$B\Phi = l, \qquad (6.36)$$

де B — матриця інциденцій розмірності $n \times m$, $\Phi = (\varphi(y_1) \ldots \varphi(y_m))^T$, $l = (0..0\underset{s}{\upsilon}0..0\underset{t}{-\upsilon}0..0)^T$. Оскільки ранг матриці інциденцій дорівнює $n - 1$, то система рівнянь (6.34) надлишкова: $\sum_{i=1}^{n} Q(x_i) = 0$. Зауважимо також, що течія φ з x_s до x_t величини υ є течія величини $-\upsilon$ з x_t до x_s.

Приклад

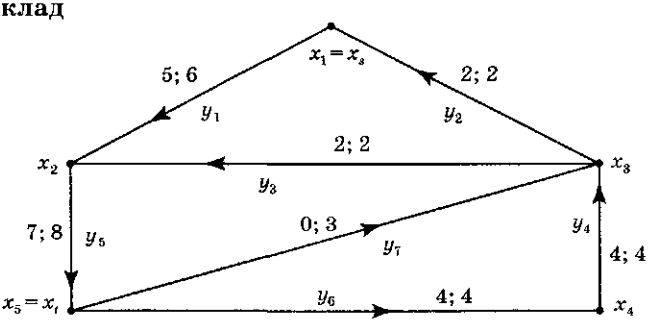


Рис. 6.52. Течія величини 3

На рис. 6.52 наведено приклад мережі, що складається з п'яти вузлів і восьми дуг, в якій розглядається течія з x_1 до x_5. Кожній дузі приписані два числа: перше — величина течії за дугою, друга — пропускна здатність дуги. Величина цієї течії дорівнює 3. Дійсно,

$$Q(x_1) = 5 - 2 = 3, \qquad Q(x_2) = 7 - (5 + 2) = 0,$$
$$Q(x_3) = -4 - 0 + 2 + 2 = 0, \quad Q(x_4) = -4 + 4 = 0, \qquad (6.37)$$
$$Q(x_5) = 4 + 0 - 7 = -3.$$

Систему рівнянь (6.37) можна записати у векторному вигляді $B\Phi = l$ (6.36):

$$B = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{pmatrix}, \quad \Phi = \begin{pmatrix} 5 \\ 2 \\ 2 \\ 4 \\ 7 \\ 4 \\ 0 \end{pmatrix}, \quad l = \begin{pmatrix} 3 \\ 0 \\ 0 \\ 0 \\ -3 \end{pmatrix}.$$

Задача про максимальну течію

Задача полягає у знаходженні такої множини течій за дугами, щоб величина $Q(x_s)$ була максимальною. Введемо необхідні визначення.

Переріз S відокремлює x_s від x_t, якщо вершини x_s, x_t належать різним бокам перерізу: $x_s \in X_s$, $x_t \in X_t$, $X = X_s \cup X_t$. ***Пропускною здатністю*** $c(S)$ ***перерізу*** S називається сума пропускних здатностей дуг перерізу, що починаються з X_s і закінчуються в X_t:

$$c(S) = \sum_{y_j \in (X_s \to X_t)} c_j .$$

Алгоритм розміщення позначок заснований на такий теоремі.

Теорема 6.20. ***Теорема про максимальну течію і мінімальний розріз***

Для будь-якої мережі максимальна величина течії з x_s до x_t дорівнює мінімальній пропускній здатності перерізу, що відокремлює x_s від x_t.

□ Покажемо спочатку, що величина υ будь-якої течії φ не перевершує пропускну здатність будь-якого перерізу (X_s, X_t), що відокремлює x_s від x_t. Оскільки функція φ є течією, то вона задовольняє рівняння (6.34) збереження течії:

$$\begin{aligned}
&\sum_{y \in x_s \to X} \varphi(y) - \sum_{y \in X \to x_s} \varphi(y) = \upsilon, \\
&\sum_{y \in x \to X} \varphi(y) - \sum_{y \in X \to x} \varphi(y) = 0, \quad x \neq x_s, x_t, \\
&\sum_{y \in x_t \to X} \varphi(y) - \sum_{y \in X \to x_t} \varphi(y) = -\upsilon.
\end{aligned} \tag{6.38}$$

Складемо ті рівняння з (6.38), які відповідають вершинам з X_s. Враховуючи, що $x_s \in X_s$, $x_t \in X_t$, одержуємо:

$$\upsilon = \sum_{y \in X_s \to X} \varphi(y) - \sum_{y \in X \to X_s} \varphi(y).$$

Вся множина вершин розпадається на два боки: $X = X_s \cup X_t$. Одержуємо

$$v = \sum_{y \in X_s \to X_s \cup X_t} \varphi(y) - \sum_{y \in X_s \cup X_t \to X_s} \varphi(y) =$$

$$= \sum_{y \in X_s \to X_s} \varphi(y) + \sum_{y \in X_s \to X_t} \varphi(y) - \sum_{y \in X_t \to X_s} \varphi(y) - \sum_{y \in X_t \to X_s} \varphi(y) =$$

$$= \sum_{y \in X_s \to X_t} \varphi(y) - \sum_{y \in X_t \to X_s} \varphi(y) \le \sum_{y \in X_s \to X_t} \varphi(y) \le \sum_{y \in X_s \to X_t} c(y) = c(X_s, X_t).$$

Тепер нам залишилося показати, що існують деякі течія φ і переріз (X_s, X_t), для яких величина течії дорівнює пропускній здатності перерізу. Як видно, всі течії від X_s до X_t обмежені і серед них можна обрати максимальну течію φ. За її допомогою визначимо переріз (X_s, X_t), для якого

$$\sum_{y \in X_s \to X_t} \varphi(y) = c(X_s, X_t), \quad \sum_{y \in X_t \to X_s} \varphi(y) = 0.$$

Визначимо множину X_s — одну з боків перерізу — рекурентно:

1) $x_s \in X_s$.

2) Якщо $x_i \in X_s$, дуга y веде з вершини x_i до будь-якої вершини x_j і $\varphi(y) < c(y)$, то $x_j \in X_s$.

3) Якщо $x_i \in X_s$, дуга y веде з деякої вершини x_k до вершини x_i і $\varphi(y) > 0$, то $x_k \in X_s$.

Крок 1) побудови множини X_s означає, що джерело x_s належить побудованому боку перерізу. Покажемо, що стік тоді лежить на іншому боці перерізу — $x_t \in X_t = X \backslash X_s$. Припустимо суперечне: нехай $x_t \in X_s$. Тоді існує «неорієнтований» ланцюг, що веде від джерела x_s до стоку x_t, такий, що для кожної дуги y_i ланцюга з напрямком, що співпадає з орієнтацією «від джерела до стоку» $\varphi(y_i) < c$, а для кожної дуги y_j ланцюга з напрямком, що не співпадає з орієнтацією «від джерела до стоку» $\varphi(y_j) > 0$. (Такий ланцюг називається ***аугментальним ланцюгом течії***).

Позначимо через $l_1 = \min\{c(y_i) - \varphi(y_i)\}$, за всіма дугами y_i ланцюга з напрямком, що співпадає з орієнтацією «від джерела до стоку», $l_2 = \min\{\varphi(y_j)\}$, за всіма дугами y_j ланцюга з напрямком, що не співпадає з орієнтацією «від джерела до стоку», $l = \min(l_1, l_2)$. Течію φ можна збільшити на l, збільшивши на l течію на дугах ланцюга, що ведуть «від джерела до стоку» і зменшивши на l течію на дугах ланцюга, що ведуть «від стоку до джерела». Це суперечить тому, що величина течії φ є максимальною величиною припустимої течії від вершини x_s до вершини x_t. ■

В розглянутому вище прикладі течія не є максимальною. Мінімальну пропускну здатність має переріз, що складається з дуг y_1, y_2: $c(X_s, X_t) = 6$. При цьому
$X_s = \{x_s\}$, $X_t = \{x_2, x_3, x_4, x_t\}$, $X_s \to X_t = \{y_1\}$, $X_t \to X_s = \{y_2\}$.

Алгоритм розміщення позначок для задачі про максимальну течію

Доведення теореми 6.20 дає алгоритм для побудови мінімального перерізу (X_s, X_t), що відокремлює x_s від x_t, і максимальну течію φ від x_s до x_t. Цей алгоритм запропонований Л. Фордом і Д. Фалкерсоном. Алгоритм починає роботу з відомої припустимої течії φ (наприклад, з нульової). Потім обчислення розвиваються у вигляді послідовності «розміщення позначок» (операція *А*), кожна з яких або призводить до течії з більшою величиною (операція *Б*), або ж закінчується висновком, що розглянута течія максимальна.

Будемо припускати, що пропускні здатності c_j дуг мережі цілочислові.

Операція А (розміщення позначок). Кожна вершина може знаходитися в одному з трьох станів: вершині приписана позначка і вершина переглянута (тобто вона має позначку і всі суміжні з нею вершини «оброблені»), вершина позначена, але не переглянута, вершина не позначена. Позначка вершини x_i має один з двох видів: $(+x_j, l)$ або $(-x_j, l)$. Частина $+x_j$ позначки першого типу означає, що течія припускає збільшення вздовж дуги (x_j, x_i) на величину l. Частина $-x_j$ позначки другого типу означає, що течія припускає зменшення вздовж дуги (x_i, x_j) на величину l. Спочатку всі вершини не мають позначок.

Крок 1. Джерелу x_s привласнюється позначка $(+, \infty)$. Вершина x_s позначена, але «не переглянута».

Крок 2. Візьмемо будь-яку позначену, але не переглянуту вершину. Нехай вона має позначку $(\pm x_k, l(x_i))$. «Переглянемо» її, тобто переглянемо всі вершини, що суміжні з нею, і позначимо ті з них, які ще непозначені.

Всім непозначеним вершинам x_j, в які йдуть дуги y_r з x_i і для яких $\varphi(y_r) < c_r$, приписуємо позначку $(+x_i, l(x_j))$, де
$$l(x_j) = \min(l(x_i), c_r - \varphi(y_r)).$$

Всім непозначеним вершинам x_j, з яких йдуть дуги y_r до x_i і для яких $\varphi(y_r) > 0$, приписуємо позначку $(-x_i, l(x_j))$, де
$$l(x_j) = \min(l(x_i), \varphi(y_r)).$$

Тепер вершина x_t і позначена, і переглянута, а вершина x_j, позначена, але не переглянута.

Крок 3. Повторювати крок 2 доти, доки або стік — вершина x_t — буде позначена, або вершина x_t буде не позначена і ніяких інших позначок не можна буде розмістити. В першому випадку переходимо до операції B, а в другому випадку алгоритм закінчує роботу з максимальною течією φ. У другому випадку множина позначених вершин і множина непозначених вершин утворюють два боки мінімального перерізу (X_s, X_t).

Операція Б (збільшення течії)

Крок 4. Покласти $x = x_t$ і перейти до кроку 5.

Крок 5. Якщо позначка у вершині x має вигляд $(+z, l(x))$, то змінити течію вздовж дуги (z, x) з $\varphi(z, x)$ на $\varphi(z, x) + l(x)$.

Якщо позначка у вершині X має вигляд $(-z, l(x))$, то змінити течію вздовж дуги (x, z) з $\varphi(x, z)$ на $\varphi(x, z) - l(x)$.

Крок 6. Якщо $z = x_s$, то стерти всі позначки і повернутися до кроку 1, щоб знов почати розміщувати позначки, але використовуючи вже поліпшену течію, що знайдена на кроці 5.

Якщо $z \neq x_s$, то взяти $x = z$ і повернутися до кроку 5.

Алгоритм розміщення позначок (операція A) зображує пошук шляху від джерела x_s до стоку x_t, вздовж якого можна збільшити течію, не перевершуючи пропускних здатностей на дугах і зберігаючи течію у проміжних вершинах. Якщо стік виявився позначеним, то такий шлях (відповідно — аугментальний ланцюг) знайдений і вздовж нього легко збільшити течію. Якщо ж операція A закінчилася, а стік виявився непозначеним, то течія була максимальною — її не можна збільшити. Множина дуг, що ведуть з позначених вершин до непозначених, утворює мінімальний переріз, оскільки позначені вершини саме складають множину X_s, що визначена у доведенні теореми 6.20.

Головна причина обчислювальної ефективності алгоритму розміщення позначок полягає в тому, що якщо яка-небудь вершина позначена і переглянута, то у подальшому її можна не враховувати. Приписування вершині x деякої позначки відповідає виділенню шляху з x_s до x, який може бути початком

аугментального ланцюга. Таких шляхів з x_s до x може бути багато, але достатньо знайти хоча б один з них.

Число, на яке збільшується течія в операції *Б*, є натуральним. Отже, якщо величина припустимої течії, з якої почалася робота алгоритму, ціла, то всі наступні течії будуть цілочисловими. Звідси виходить, що алгоритм *скінченний*. Сформулюємо теорему.

Теорема 6.21. ***Теорема про цілочисловість***

Якщо пропускні здатності дуг c_j цілі, то існує максимальна течія, що має також цілу величину.

Реалізація алгоритму Форда — Фалкерсона для мережі рис. 6.52

1. Візьмемо течію, що зображена на рисунку 6.52 як початкова припустима течія. Вона має величину 3.

2. Привласнимо джерелу, вершині x_1, позначку $(+, \infty)$. Вершина x_1 позначена, але не переглянута.

3. Переглянемо вершини, суміжні з вершиною x_1. Вершині x_2 привласнимо позначку $(+x_1, 1)$, а вершині x_3 — позначку $(-x_1, 2)$ ($\varphi(x_1, x_2) = 5 < 6 = c_1$, $\varphi(x_3, x_1) = 2 > 0$). Вершина x_1 позначена і переглянута, а вершини x_2 і x_3 позначені, але не переглянуті.

4. Переглянемо вершини, суміжні з вершиною x_3. Із вершин, суміжних з вершиною x_3, не позначені вершини x_4 і x_5. Вершині x_4 привласнимо позначку $(-x_3, 2)$, оскільки $\varphi(x_4, x_3) = 4 > 0$ і $\min(2, 4) = 2$. Вершину x_5 не позначаємо, оскільки $\varphi(x_5, x_3) = 0$.

5. Переглянемо вершини, суміжні з вершиною x_2. Вершині x_5 привласнимо позначку $(+x_2, 1)$, оскільки $\varphi(x_2, x_5) = 7 < 8 = c_5$. Стік позначений. Переходимо до операції *Б* — збільшення течії.

6. Стік має позначку $(+x_2, 1)$. Тому збільшуємо течію вздовж дуги (x_2, x_5) на 1.

7. Вершина x_2 має позначку $(+x_1, 1)$. Тому збільшуємо течію вздовж дуги (x_1, x_2) на 1. Ми одержали нову течію величини 4 (рис. 6.53).

8. Стираємо всі позначки.

9. Привласнюємо вершині x_1 позначку $(+, \infty)$.

10. Переглянемо вершини, суміжні з вершиною x_1. Вершині x_3 привласнимо позначку $(-x_1, 2)$. Вершину x_2 не позначаємо, оскільки $\varphi(x_1, x_2) = 6 = c(y_1)$.

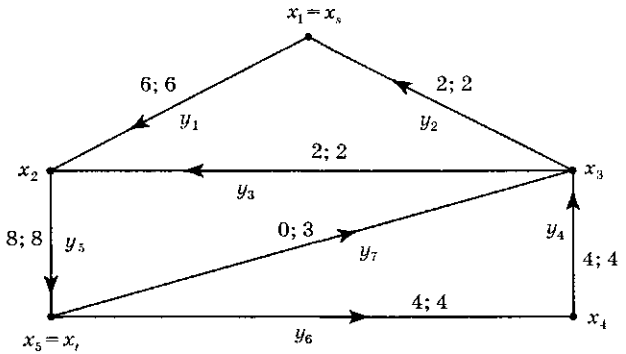


Рис. 6.53. Течія величини 4

11. Переглянемо вершини, суміжні з вершиною x_3. Вершині x_4 привласнимо позначку $(-x_3, 2)$, оскільки $\varphi(x_4, x_3) = 4 > 0$, $l(x_3) = 2$ і $\min(2, 4) = 2$.

12. Переглянемо вершини, суміжні з вершиною x_4. Вершині x_5 привласнимо позначку $(-x_4, 2)$, оскільки $\varphi(x_5, x_4) = 4 > 0$, $l(x_4) = 2$ і $\min(2, 4) = 2$. Стік позначений. Переходимо до операції *Б* — збільшення течії.

13. Стік має позначку $(-x_4, 2)$. Тому зменшуємо течію вздовж дуги (x_5, x_4) на 2.

14. Вершина x_4 має позначку $(-x_3, 2)$. Тому зменшуємо течію вздовж дуги (x_4, x_3) на 2.

15. Вершина x_3 має позначку $(-x_1, 2)$. Тому зменшуємо течію вздовж дуги (x_3, x_1) на 2. Ми одержали нову течію величини 6 (рис. 6.54). З теореми 6.20 ми знаємо, що ця течія є максимальною. Перевіримо це.

16. Стираємо всі позначки.

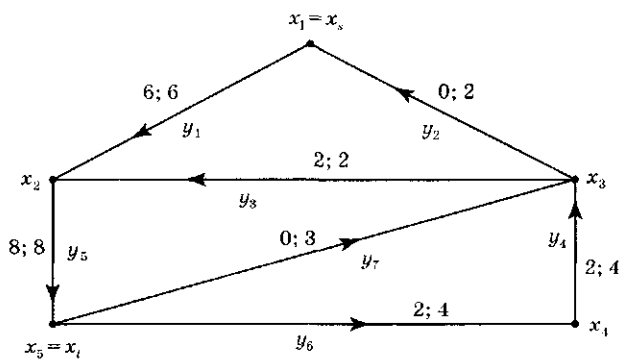


Рис. 6.54. Максимальна течія

17. Привласнимо вершині x_1 позначку $(+, \infty)$.

18. Вершини, суміжні вершині x_1, не можна позначити, оскільки дуга (x_1, x_2) насичена — $\varphi(x_1, x_2) = \varphi(y_1) = c(y_1) = 6$, а через дугу (x_3, x_1) течія не передається. Стік залишився непозначеним. Отже, одержана течія є максимальною. Дуги (x_1, x_2) і (x_3, x_1) утворюють мінімальний переріз. Множина позначених вершин утворює той його бік, який містить джерело: $X_s = \{x_1\}$. Непозначені вершині утворюють інший бік перерізу, що містить стік: $X_t = \{x_2, x_3, x_4, x_5\}$. Побудована течія має вигляд $\varphi(y_1) = 6$, $\varphi(y_5) = 8$, $\varphi(y_6) = 2$, $\varphi(y_4) = 2$, $\varphi(y_3) = 2$, $\varphi(y_2) = \varphi(y_7) = 0$.

Графи з багатьма джерелами і стоками

Алгоритм Форда — Фалкерсона застосовний і для визначення величини максимальної течії між множиною вершин-джерел і множиною вершин-стоків. Розіб'ємо множину вершин на множину джерел $X_+ = \{x \in X: Q(x) > 0\}$, що створюють течію, множину стоків $X_- = \{x \in X: Q(x) < 0\}$, що споживають течію і множину проміжних вершин $X_0 = \{x \in X: Q(x) = 0\}$, що зберігають течію φ: $X = X_+ \cup X_- \cup X_0$. Перетворимо течію φ у течію, яка має тільки одне джерело та один стік, збільшивши кількість вершин у мережі. Для цього додамо дві нові вершини — «фіктивне» джерело x_s і «фіктивний» стік x_t. З'єднаємо вершину x_s зі всіма дійсними джерелами. Цим дугам припишемо течію, що утворена відповідним джерелом. А від кожного дійсного стоку спрямуємо дугу до «фіктивного» стоку x_t. Цим дугам припишемо течію, що споживається відповідним стоком. При цьому пропускні здатності доданих дуг вважаємо нескінченними. В результаті одержуємо мережу з одним джерелом і одним стоком. Застосовуючи до неї алгоритм Форда — Фалкерсона, знаходимо максимальну течію, яка є максимальною і для вихідної мережі.

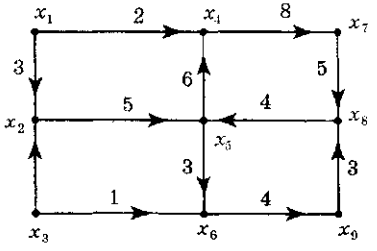


Рис. 6.55. Мережа з трьома стоками

Проілюструємо на прикладі перетворення мережі з кількома джерелами і кількома стоками до мережі з одним джерелом і одним стоком.

Приклад

На рис. 6.55 зображено мережу з двома джерелами x_1 і x_3 і трьома стоками x_7, x_8, x_9:

$Q(x_1) = 5$, $Q(x_3) = 3$, $Q(x_7) = 3$, $Q(x_8) = 4$, $Q(x_9) = 1$,
$Q(x_2) = Q(x_4) = Q(x_5) = Q(x_6) = 0$.

Перетворимо цю мережу до мережі з одним джерелом і одним стоком.

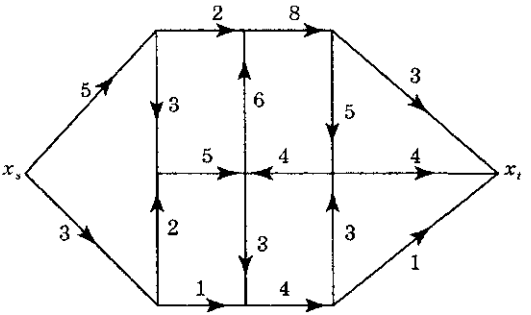


Рис. 6.56. Перетворена мережа

На рис. 6.56 зображено мережу вже з одним джерелом x_s і одним стоком x_t.

Задача про багатополюсну максимальну течію

Розглянемо задачу знаходження максимальної течії для всіх пар вузлів у неорієнтованій мережі. Цю задачу можна розглядати як узагальнення задачі з одним джерелом і одним стоком і її можна розв'язати, застосовуючи алгоритм Форда — Фалкерсона для кожної пари вершин. Більш ефективним є алгоритм, що запропонований Р. Гоморі і Т. Ху.

Алгоритм Гоморі — Ху

1. Оберемо деякі дві вершини графа. Позначимо одну з них через x_s, а іншу через x_t.

2. Застосуємо алгоритм Форда — Фалкерсона і знайдемо максимальну течію з джерела x_s у стік x_t. При цьому множина позначених вершин і множина непозначених вершин утворюють два боки мінімального перерізу X_s і X_t.

3. Оберемо дві вершини графа x_i і x_j, що лежать з одного боку від перерізу. Наприклад, нехай x_i, $x_j \in X_s$.

4. Замінимо дуги з мінімального перерізу (X_s, X_t) однією дугою, а вершини боку перерізу, в якому не лежать вершини x_i, x_j, (наприклад, X_t), — однією вершиною. Пропускну здатність

у так визначеній дузі покладемо рівній пропускній здатності перерізу (X_s, X_t).

5. Покладемо: $x_i = x_s$, $x_j = x_t$ і повернемося до другого кроку.

Можна показати, що після того, як буде обрано $n - 1$ пара вершин, ми визначимо всі $\frac{n(n-1)}{2}$ величин максимальної течії для вихідної мережі. Основна ідея алгоритму полягає в ітеративній побудові максимального остовного дерева, гілки якого відповідають перерізам, а пропускні здатності гілок дорівнюють пропускним здатностям цих перерізів. Якщо б ми застосували алгоритм Форда — Фалкерсона до кожної пари вершин, то нам би довелося його застосувати $\frac{n(n-1)}{2}$ разів. В алгоритмі ж Гоморі — Ху максимальна течія між парою вершин визначається за допомогою алгоритму розміщення позначок тільки $n - 1$ разів.

Проілюструємо на прикладі алгоритм Гоморі — Ху.

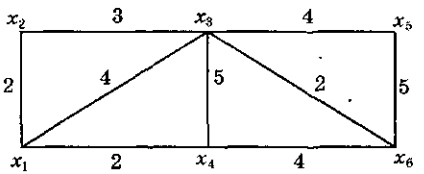


Рис. 6.57. Мережа з пропускними здатностями

Приклад

Розглянемо мережу, зображену на рис. 6.57. Числа, приписані дугам, відповідають їх пропускним здатностям. Треба для кожної пари вузлів мережі визначити величину максимальної течії між ними. Ця задача розв'язується за $n - 1 = 6 - 1 = 5$ ітерацій алгоритму Гоморі — Ху. Якщо алгоритм Форда — Фалкерсона розміщення позначок застосовувався б до кожної пари вузлів, то треба було б розв'язати 15 задач про максимальну течію.

Реалізація алгоритму Гоморі — Ху для мережі з прикладу рис. 6.57

1. Оберемо вершини $s = x_1$ і $t = x_2$. Мінімальну пропускну здатність відносно джерела $s = x_1$ і стоку $t = x_2$ має переріз з боками $X_s = \{x_1, x_3, x_4, x_5, x_6\}$ і $X_t = \{x_2\}$. За теоремою 6.20 величина максимальної течії між вершинами x_1 і x_2 дорівнює пропускній здатності перерізу (X_s, X_t): $v_{12} = v_{21} = c(X_s, X_t) = 2 + 3 = 5$. Об'єднаємо вершини з X_s в одну

вершину і з'єднаємо її з вершиною x_2 гілкою з пропускною здатністю 5 (рис. 6.58).

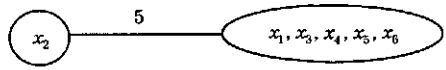


Рис. 6.58. Перший крок алгоритму

2. Оберемо вершини $s = x_1$ і $t = x_3$. Мінімальну пропускну здатність відносно джерела $s = x_1$ і стоку $t = x_3$ має переріз з боками $X_s = \{x_1\}$ і $X_t = \{x_2, x_3, x_4, x_5, x_6\}$. За теоремою 6.20 величина максимальної течії між вершинами x_1 і x_3 дорівнює пропускній здатності перерізу (X_s, X_t): $v_{13} = v_{31} = c(X_s, X_t) = 2 + 4 + 2 = 8$. Об'єднаємо вершини x_3, x_4, x_5, x_6 в одну вершину і з'єднаємо її з вершиною x_1 гілкою з пропускною здатністю 8 (рис. 6.59).

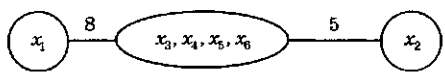


Рис. 6.59. Другий крок

3. Оберемо вершини $s = x_4$ і $t = x_3$. Мінімальну пропускну здатність відносно джерела $s = x_4$ і стоку $t = x_3$ має переріз з боками $X_s = \{x_4\}$ і $X_t = \{x_1, x_2, x_3, x_5, x_6\}$. За теоремою 6.20 величина максимальної течії між вершинами x_4 і x_3 дорівнює пропускній здатності перерізу (X_s, X_t): $v_{43} = v_{34} = c(X_s, X_t) = 2 + 5 + 4 = 11$. Об'єднаємо вершини x_3, x_5, x_6 в одну вершину і з'єднаємо її з вершиною x_4 гілкою з пропускною здатністю 11 (рис. 6.60).

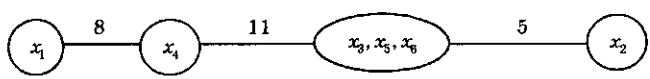


Рис. 6.60. Третій крок

4. Оберемо вершини $s = x_5$ і $t = x_3$. Мінімальну пропускну здатність відносно джерела $s = x_5$ і стоку $t = x_3$ має переріз з боками $X_s = \{x_5\}$ і $X_t = \{x_1, x_2, x_3, x_4, x_6\}$. Величина максимальної течії між вершинами x_5 і x_3 дорівнює пропускній здатності перерізу (X_s, X_t): $v_{53} = v_{35} = c(X_s, X_t) = 5 + 4 = 9$. Об'єднаємо вершини x_3, x_6 в одну вершину і з'єднаємо її з вершиною x_5 гілкою з пропускною здатністю 9 (рис. 6.61).

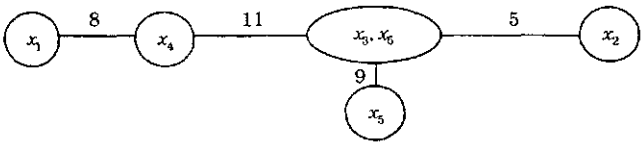


Рис. 6.61. Четвертий крок

5. Оберемо вершини $s = x_6$ и $t = x_3$. Мінімальну пропускну здатність відносно джерела $s = x_6$ і стоку $t = x_3$ має переріз з боками $X_s = \{x_6\}$ і $X_t = \{x_1, x_2, x_3, x_4, x_5\}$. Величина максимальної течії між вершинами x_6 і x_3 дорівнює пропускній здатності перерізу (X_s, X_t): $v_{63} = v_{36} = c(X_s, X_t) = 5 + 6 + 4 = 15$. З'єднаємо вершину x_3 з вершиною x_6 гілкою з пропускною здатністю 15 (рис. 6.62).

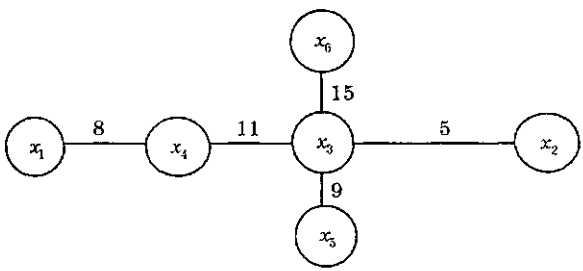


Рис. 6.62. Остаточне дерево перерізів

За деревом перерізів, зображеним на рис. 6.62, легко знайти решту величин максимальних течій. Наприклад, $v_{16} = v_{61} = 8$, оскільки у ланцюгу дерева перерізів, що з'єднує вершини x_1 і x_6, мінімальна пропускна здатність гілок дорівнює 8. Запишемо величини максимальних течій у вигляді матриці:

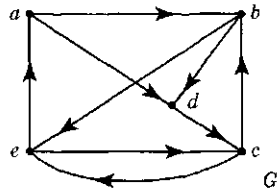
$$V = \begin{pmatrix} - & 5 & 8 & 8 & 8 & 8 \\ 5 & - & 5 & 5 & 5 & 5 \\ 8 & 5 & - & 11 & 9 & 15 \\ 8 & 5 & 11 & - & 9 & 11 \\ 8 & 5 & 9 & 9 & - & 9 \\ 8 & 5 & 15 & 11 & 9 & - \end{pmatrix}.$$

Тут на перетині i-рядка і j-стовпця стоїть величина максимальної течії між вершинами x_i і x_j.

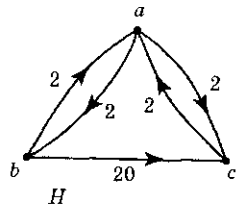
Завдання

1. Чи може у ланцюга бути два однакових ребра? Дві однакові вершини?
2. Чи може дерево містити цикл?
3. Чи будь-який цикл в орієнтованому графі є контуром?
4. Чи можуть від'ємні числа бути елементами матриці суміжності A деякого графа?
5. Чому дорівнює сума елементів за стовпцем матриці інциденцій деякого графа?
6. Чи дорівнює сума за рядком сумі за стовпцем у матриці суміжності A неорієнтованого графа? Орієнтованого?
7. Що означає рівність нулю матриці суміжності A у деякій степені k?
8. Чи є зв'язним граф, у якого число ребер $m < n - 1$, де n — число вершин?
9. Скільки ребер містить простий повний граф з n вершинами?
10. Чи правильно, що серед степенів вершин графа є хоча б два однакових?
11. Якщо граф зв'язний і n — число його вершин, то яка нижня оцінка для числа його ребер?
12. Чи є плоскими дводольні графи $K_{2,2}$ і $K_{3,3}$?
13. Побудувати матрицю суміжності дводольних графів $K_{2,2}$ і $K_{3,3}$.
14. Яку кількість компонент зв'язності може бути у графа з n вершинами?
15. Яку геометричну інформацію несуть елементи матриці A^2, де A — матриця суміжності орієнтованого графа?
16. Чи обмежена множина хроматичних чисел класу всіх плоских графів?
17. Знайти хроматичні числа графів Понтрягіна — Куратовського $K_{3,3}$ і K_5.
18. Чи будь-яке натуральне число n може бути хроматичним числом деякого графа?
19. Чому дорівнює хроматичне число дерева?
20. Чи може число ребер дерева дорівнювати числу його вершин?
21. Скільки існує різних графів з 4-ма вершинами? С 5-ма?
22. Як обчислити число лінійно незалежних циклів графа?
23. Побудувати матрицю інциденцій графа, що зображує реберний каркас куба.
24. В якому випадку матриця суміжності орієнтованого графа є симетричною?
25. Побудувати матриці суміжності та інциденцій графа, одержаного з'єднанням центра правильного п'ятикутника з його вершинами.
26. Побудувати фундаментальну матрицю циклів двочасткового графа $K_{3,3}$.

27. Побудувати фундаментальну матрицю перерізів двочасткового графа $K_{3,3}$.
28. Побудувати фундаментальні матриці циклів і перерізів графа $K_{2,4}$.
29. Чи існує гамільтонів цикл у двочасткових графах $K_{2,2}$, $K_{3,3}$, $K_{2,3}$, $K_{2,4}$?
30. Знайти гамільтонові контури у графі G.



31. В графі H вказати найкоротший контур комівояжера і найкоротший замкнений маршрут комівояжера (з можливим повторенням вершин).



32. Яке число ребер остовного дерева зв'язного графа G з n вершинами і m ребрами?
33. Як зв'язані між собою хроматичне число γ і клікове число ρ графа G?
34. Побудувати приклади клік і незалежних множин графів $K_{3,3}$, $K_{2,3}$ і графа із задачі 30.
35. Яка властивість є достатньою умовою того, щоб довільний граф містив гамільтонів цикл?
36. Чи правильно, що будь-який остів графа G має, щонайменше, 1 спільне ребро з кожним розрізом?
37. Чи будь-яка множина $\upsilon(G)$ незалежних циклів буде «фундаментальною»? ($\upsilon(G)$ — це цикломатичне число графа G)
38. Будь-який цикл графа G можна зобразити у вигляді суми за модулем 2 фундаментальних циклів. Чи правильне зворотне твердження, тобто що будь-яка сума за модулем 2 фундаментальних циклів обов'язково надасть єдиний цикл?
39. Чи однозначно обирається множина фундаментальних циклів?
40. Навести приклади сильно зв'язного, слабо зв'язного, незв'язного графів.

41. Чому дорівнює сума всіх елементів рядка матриці суміжності A? Чому дорівнює сума елементів у стовпці?
42. Чи справедливе твердження, що будь-який повний симетричний граф містить гамільтонів цикл?
43. Показати, що ранг матриці інциденцій B зв'язного графа з n вершинами дорівнює $n - 1$.
44. Чому дорівнює число позначених графів з n вершинами?
45. Як зв'язані число позначених парних графів з n вершинами і число всіх позначених графів з числом вершин $n - 1$?
46. Чому дорівнює число зв'язних позначених графів з n вершинами?
47. Чи будь-який граф є ейлеровим?

розділ 7

Мови та граматики

7.1. Задача формалізації мов та перекладу

Необхідність формального задання мов та розв'язку задачі перекладу в програмуванні

Сучасні ЕОМ являють собою складні комплекси пристроїв, що постачаються розвиненими системами математичного забезпечення, які містять мови програмування і транслятори з цих мов, операційні системи, диспетчери, монітори, різного роду обслуговуючі програми, а також системи стандартних і типових підпрограм. Усі ці засоби можуть бути реалізовані як програмним, так і схемним способом. Процес розробки систем математичного забезпечення, як і проектування самих ЕОМ, надзвичайно складний і трудомісткий. Одним з основних джерел задач є проблема задання мови та оптимального перекладу з однієї мови на іншу. Задача перекладу може бути сформульована таким чином: існують дві алгоритмічні мови і деякий алгоритм, написаний на одній з них; треба знайти оптимальну за заданими критеріями реалізацію цього алгоритму на іншій мові. В програмуванні за звичаєм першою є деяка мова програмування, орієнтована на те чи інше коло задач, а другою — внутрішня мова машини, на якій розв'язуються ці задачі. Таким чином, йдеться про переклад з мови програмування на машинну мову з одночасною оптимізацією вихідної програми. Ця задача розв'язується при проектуванні трансляторів і компіляторів. У той же час вихідним може бути алгоритм роботи деякого пристрою ЕОМ, записаний на призначеній для цієї цілі алгоритмічній мові, а мова, на яку

перекладається даний алгоритм, — це мова схем. Тоді задача полягає в одержанні оптимальної схеми, що реалізує алгоритм роботи даного пристрою або деякої його частини. Процес розв'язку таких задач на практиці поділяється на проміжні етапи, на кожному з яких робиться оптимізація і присутня своя мова. Формальне задання мови необхідне як для самого процесу перекладу, так і для визначення класу припустимих записів на цій мові, тобто для визначення коректності або припустимості запису, що перекладається на іншу мову. Мови складаються з рядків символів і, отже, весь обчислювальний процес можна розглядати як перетворення однієї множини рядків на іншу, тобто як математичний об'єкт.

Запитання

1. В яких випадках для роботи ЕОМ необхідний розв'язок задачі перекладу з однієї мови на іншу?
2. Для чого розв'язується задача формального задання мов?
3. Чим відрізняється процес перекладу в літературі від перекладів, що виконується при роботі ЕОМ?

7.2. Перетворення рядків символів

Алфавіт, рядок, символ, підалфавіт, поширення алфавіту, порожній рядок, довжина рядку, конкатенація рядків, підрядок

Формальний алфавіт зображує множину неділених символів. Скінченні послідовності букв алфавіту називаються рядками, вираженнями у цьому алфавіті.

Алфавіти ми будемо позначати великими буквами латинського алфавіту, наприклад, A, B, C.

Символи ми будемо позначати маленькими буквами латинського алфавіту з індексами або без індексів, наприклад, a_1, a_2, a_n.

Оскільки алфавіти є множинами, то до них застосовні теоретико-множинні операції: $\cap$, $\cup$, $\subset$, $\subseteq$, $\in$, $\notin$ тощо.

Визначення

Якщо A і B — алфавіти та $A \subseteq B$, то A називається підалфавітом B і B — поширенням A.

Рядки є упорядкованими сукупностями символів алфавіту і, отже, є елементами декартова добутку $A^n = \underbrace{A \times A \times \ldots \times A}_{n}$. Тут

A — алфавіт, n — довжина рядків. Рядки ми будемо записувати у вигляді послідовності символів, наприклад, $a_1a_2...a_n$. Символи також є рядками для випадку $n = 1$. В тексті значення рядку наводиться у лапках, наприклад, рядки «*abcd*», «1 + 2 =», «00001110101». Рядки за звичаєм позначаються маленькими буквами грецького алфавіту, наприклад, α, β, γ.

Визначення

Порожній ***рядок*** — це рядок, який не має символів. Позначимо його ε. Порожній рядок не є символом і не належить ніякому алфавіту. Це можна записати так: $\forall A\ \varepsilon \notin A$, де A — алфавіт.

Через A^* позначається множина всіх рядків алфавіту A, включаючи порожній рядок. Зрозуміло, що $A^* = A^0 \cup A^1 \cup A^2 \cup ...$, де $A^0 = \varepsilon$ — порожній рядок, $A^1 = A$ — різні рядки алфавіту A довжини 1, $A^2 = A \times A$ — різні рядки алфавіту A довжини 2, ... Множина A^* нескінченна.

Через A^+ позначається множина всіх рядків алфавіту A, не включаючи порожній рядок: $A^+ = A^1 \cup A^2 \cup A^3 ...$; $A^* = A^0 \cup A^+$.

Довжина рядку α дорівнює кількості символів у рядку і позначається $|\alpha|$. Якщо $\alpha = a_1a_2...a_n$, то $|\alpha| = n$. Довжина порожнього рядку дорівнює нулю: $|\varepsilon| = 0$

Визначення

Конкатенацією або злиттям рядків α і β називається рядок αβ. Наприклад, конкатенацією рядків «*sdf*», «*htuwq*» є рядок «*sdfhtuwq*». Вважається, що $\alpha\varepsilon = \varepsilon\alpha = \alpha$.

Запис α^n означає $\underbrace{\alpha\alpha...\alpha}_{n\text{ раз}}$, α^1 означає α, α^0 означає ε. Тут в якості α може бути як рядок символів, так і символ.

Наприклад, $0^31^2 = 00011$. Тут a^nb^mc означає рядок, що складається з n символів a, за якими йдуть m символів b, за якими йде один символ c.

Визначення

Рядок β є підрядком рядку вигляду αβγ, причому α і γ можуть бути будь-якими, в том числі і порожніми рядками.

Порожній рядок ε є підрядком будь-якого рядка.

Наприклад підрядками рядку «$x1 = y1 + z1$» є рядки «$x1$», «$x1$ =», «$1 = y1 + z$», «+», порожній рядок ε та інші рядки. Число підрядків у скінченного рядка — скінченне.

Запитання

1. Назвіть два різних підходи до вивчення явищ з інформаційної точки зору.
2. Чи є алфавітний спосіб достатнім для зображення будь-якої дискретної інформації?
3. Наведіть приклади алфавітного способу зображення у повсякденному житті:
   а) неперервної інформації;
   б) дискретної інформації.
4. Дайте визначення алфавітного способу задання інформації, алфавіту, рядку.
5. Дано два алфавіти A^1 і A^2, причому $A^1 \subseteq A^2$. Як називається A^1 *за відношенням до* A^2? *Як називається* A^2 *за відношенням до* A^1?
6. Що таке порожній рядок?
7. Дайте визначення операції конкатенації рядків.

Завдання

1. Визначте алфавіти для запису:
   а) арифметичних виразів;
   б) азбуки Морзе;
   в) послідовності ходів при грі у шахи.
   Наведіть приклади рядків у цих алфавітах.
2. Для алфавітів $A = \{a, b, c\}$, $B = \{0, 1\}$ складіть:
   а) A^*, B^*;
   б) A^+, B^+;
3. Нехай задано рядок α = «послідовність». Знайдіть довжину рядка $|\alpha|$.
4. Побудуйте рядки $0^5 1^2$, $a^3 b^2 c$, $0^n A 1^n$, для $n = 3, 5, 0$.
5. α = «alpha» β = «bet» Що є конкатенацією рядків $\alpha\beta$?
6. Які з наведених рядків є підрядками рядка α = «індустріалізація»:
   β_1 = «індус», β_2 = «індустрія», β_3 = «три», β_4 = «акція»?

7.3. Задання мов за допомогою граматик

Мова, що розпізнає і породжує граматики, термінальні і нетермінальні символи, продукція, початковий символ, вивід рядків

Визначення

Мова L — це множина рядків скінченної довжини у заданому скінченному алфавіті.

Перше важливе питання: як описати мову L в тому випадку, коли вона нескінченна. Якщо довжини всіх рядків обмежені, то L складається із скінченного числа рядків і можна скласти

список всіх рядків з L. Однак для багатьох мов не можна або небажано встановити верхню границю припустимої довжини рядку мови. Такі мови не можна визначити переліком всіх рядків. Необхідний опис мови повинен бути скінченним, у той час, як мова може бути нескінченною. Відомо кілька таких способів опису мов. Один з них полягає у використанні системи, що називається ***граматикою***. Залежно від способу визначення мови граматики ділять на ті, що розпізнають, і ті, що породжують.

Визначення

Граматика, що розпізнає, — це граматика, яка для будь-якого рядка може *визначити*, є цей рядок правильним або ні. Граматика, ***що породжує***, — це граматика, яка може *побудувати* будь-який правильний рядок і не будує неправильних рядків.

Ми будемо розглядати тільки граматики, що породжують, називаючи їх у подальшому просто граматиками. Для кожної граматики визначаються дві скінченні множини, що не перетинаються: множина T термінальних символів і множина N нетермінальних символів.

З ***термінальних*** символів утворюються рядки мови, яка визначається, це означає, що множина термінальних символів складає алфавіт цієї мови. Наприклад, якщо мова йде про граматику, що породжує арифметичні вирази, у цьому випадку термінальними символами є цифри, дужки і знаки операцій. ***Нетермінальні*** символи використовуються як допоміжні або проміжні у процедурах побудови рядків мови, але вони не входять до остаточного («правильного») виразу мови. Для позначення нетермінальних символів будемо використовувати великі букви латинського алфавіту.

Серцевину граматики складає скінченна множина продукцій.

Визначення

Продукція — це упорядкована пара рядків, першим елементом якої є будь-який рядок, складений із символів алфавіту $N \cup T$, причому цей рядок обов'язково містить хоча б один нетермінальний символ. Другим елементом пари є будь-який рядок, що складений із символів алфавіту $N \cup T$. Множина продукцій P описує процес породження рядків мови.

Множину перших елементів продукцій ми можемо записати як $(N \cup T)^* N\ (N \cup T)^*$. Тут $(N \cup T)^*$ — це множина рядків, складених із символів алфавіту $N \cup T$, N — множина нетермінальних символів, з яких береться один обов'язковий для першого рядку. Множину других елементів продукцій ми можемо записати як $(N \cup T)^*$. При запису перший і другий елемент продукції будемо розділяти символом «→». Таким чином, множину всіх можливих продукцій можна записати як $(N \cup T)^* N(N \cup T)^* \to (N \cup T)^*$.

Породження рядків у граматиках починається з особливого рядка, що складається з одного виділеного символу, який називається ***початковим символом***. Він позначається S.

Наведемо формальне визначення граматики.

Визначення

Граматикою називається структура $G = (N, T, P, S)$, де
N — скінченна множина нетермінальних символів;
T — неперетинна з N скінченна множина термінальних символів;
P — множина продукцій — скінченна підмножина множин $(N \cup T)^* N(N \cup T)^* \times (N \cup T)^*$;
S — початковий символ з N.

Застосування продукцій граматики призводить до породження (виводу) рядків у граматиці.

Наведемо приклад граматики.

Приклад. Розглянемо граматику $G1 = (\{A, X, S\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (,)\}, P, S)$, де P складається з набору таких продукцій:

1. $S \to A$.
2. $A \to A \cdot A$.
3. $A \to A + A$.
4. $A \to A - A$.
5. $A \to (A)$.
6. $A \to X$.
7. $X \to 1$.
8. $X \to 2$.
9. $X \to 3$.
10. $X \to 4$.
11. $X \to 5$.
12. $X \to 6$.
13. $X \to 7$.
14. $X \to 8$.
15. $X \to 9$.
16. $X \to 0$.

В цій граматиці множина нетермінальних символів — $N = \{A, X, S\}$, множина термінальних символів — $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (,)\}$, граматика містить 16 продукцій,

що складають множину P, початковий символ граматики S належить N.

Граматика визначає мову рекурсивно, за допомогою виведених рядків.

Визначення

Виведені рядки (рядки, які можна вивести) у граматиці визначаються таким чином:

Нехай $G = (N, T, P, S)$ — деяка граматика, тоді

1) S — виведений рядок;

2) Якщо $\alpha\beta\gamma$ — виведений рядок і продукція $\beta \to \delta$ міститься в P, то $\alpha\delta\gamma$ — теж виведений рядок.

Мова L(G), що визначається граматикою G, — це множина рядків, які складаються тільки з терміналів і виводяться у граматиці G. Граматики еквівалентні, якщо породжені ними мови рівні. Рівність мов визначається за правилом рівності множин.

Наприклад, продукцією у деякій граматиці може бути пара $AB \to CDE$. Застосовуються ці продукції таким чином:

1. Нехай AB є підрядком рядка, виведеного в граматиці G.
2. Замінимо у цьому рядку AB на CDE — одержимо новий рядок.
3. Цей новий рядок теж вважається виведеним в граматиці G.

Зокрема, якщо рядок $FGABH$ виведений, то $FGCDEH$ теж виведений.

Можливість виводу рядка у граматиках позначається так:

$\varphi \Rightarrow \psi$ (читається «ψ безпосередньо виведений з φ») означає, що результатом застосування однієї продукції з P до рядка φ є рядок ψ.

$\varphi \Rightarrow^* \psi$ (читається «ψ виведений з φ») означає, що існує послідовність $\varphi \Rightarrow \alpha_1 \Rightarrow ... \Rightarrow \alpha_k \Rightarrow \psi$.

Приклад. Повернемося до граматики $G1 = (\{A, X, S\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (,)\}, P, S)$ з попереднього прикладу.

Ця граматика породжує вирази із цифр, що містять знаки суми і добутку.

Виведемо рядок $(1 + 5) \cdot 7$.

Номер продукції, використаної на кожному кроці, вказано біля знака «$\Rightarrow$».

$$S \Rightarrow {}_1A \Rightarrow {}_2A \cdot A \Rightarrow {}_5(A) \cdot A \Rightarrow {}_3(A + A) \cdot A \Rightarrow {}_6(X + A) \cdot A \Rightarrow$$
$$\Rightarrow {}_6(X + X) \cdot A \Rightarrow {}_6(X + X) \cdot X \Rightarrow {}_7(1 + X) \cdot X \Rightarrow$$
$$\Rightarrow {}_{11}(1 + 5) \cdot X \Rightarrow {}_{13}(1 + 5) \cdot 7$$

Запитання

1 Дайте визначення мови.
2 Чому для того, щоб описати мову, необхідно застосувати спеціальні формальні системи (наприклад, граматики)?
3 Які граматики є тими, що розпізнають, і які тими, що породжують?
4 Дайте визначення граматики та її складових.
5 Яким чином граматики, що породжують, визначають мову? Які рядки називаються виведеними у граматиці?

Завдання

1 Для граматики $G1$, розглянутій у прикладі, зробіть вивід рядка $(4 - 5) + 2 \cdot 3$.
2. Змініть запис попереднього виводу, використовуючи символ $\Rightarrow^*$.
3. Змініть список продукцій граматики $G1$ так, щоб у виразах, що у неї припускаються, могли бути присутні:
   а) однозначні і двозначні числа;
   б) числа з будь-якою кількістю розрядів;
   в) від'ємні числа.
4. Перепишіть список продукцій граматики $G1$, не змінюючи породжену нею мову:
   а) зменшивши число використовуваних нетермінальних символів;
   б) збільшивши число використовуваних нетермінальних символів.

7.4. Форма Бекуса — Наура

Форма Бекуса — Наура запису продукцій граматики

Для зручності запису продукцій граматик використовується нотація, що введена Бекусом — ***форма Бекуса — Наура*** (***форма Бекуса, БНФ***). Вона особливо корисна, коли ми використовуємо елементи з N, які можна переплутати з елементами з T. Ця нотація полягає у такому:

1. Символ «→» заміняється на «::=».

2. В кутові дужки «< >» беруться нетермінальні символи.

3. Знак «|» використовується для скороченого запису продукцій, позначає «або» і об'єднує групу продукцій з однаковою лівою частиною, але різними правими частинами.

Приклад. Розглянемо граматику $G_{\text{число}}$:

$G_{\text{число}} = (\{\text{Число, Знак, Ціле Число, Дрібна Частина, Цифра, } S\},$
$\{+, -, 0, ..., 9,,\}, P, S)$,

де P складається з набору продукцій:

1. Число → Знак ЦілеЧисло ДрібнаЧастина.
2. Знак → +.
3. Знак → −.
4. Знак → ε.
5. ЦілеЧисло → Цифра.
6. ЦілеЧисло → Цифра ЦілеЧисло.
7. ДрібнаЧастина →, ЦілеЧисло.
8. ДрібнаЧастина → ε.
9. Цифра → 0.
10. Цифра → 1.
11. Цифра → 2.
12. Цифра → 3.
13. Цифра → 4.
14. Цифра → 5.
15. Цифра → 6.
16. Цифра → 7.
17. Цифра → 8.
18. Цифра → 9.

Запишемо продукції цієї граматики відповідно БНФ:

<Число> ::= <Знак> <Ціле Число> <Дрібна Частина>

<Знак> ::= + | − | ε

<Ціле Число> ::= <Цифра> | <Цифра> <Ціле Число>

<ДрібнаЧастина> ::=, <Ціле Число> | ε

<Цифра>:: = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

На практиці при запису граматик можуть використовуватися не всі символи БНФ, а тільки «|» або «<>», «|».

Запитання

1. Для чого застосовується форма Бекуса — Наура?
2. В чому полягає форма Бекуса — Наура?
3. Назвіть переваги запису продукцій граматик за допомогою БНФ.

Завдання

1. Запишіть продукції граматики $G1$ (п. 7.3) у БНФ.
2. Яким продукціям граматики $G = (\{A, B, S\},\{a, b\},P, S)$ відповідає запис $S \to aA \mid bA \mid ABba$?
3. Запишіть у БНФ продукції граматики

   $G = (\{A, B, S\},\{0, 1\},P, S)$:

   1) $S \to AB$.
   2) $AB \to 000$.
   3) $A \to 001$.
   4) $B \to 110$.
   5) $A \to 010$.
   6) $B \to 101$.
   7) $A \to 100$.
   8) $B \to 011$.
   9) $S \to BA$.
   10) $BA \to 111$.
   11) $S \to \varepsilon$.

4. Яка мова визначається граматикою $G_{\text{число}}$? Зробіть вивід рядків «−12,178», «100», «0,5», «0» у граматиці $G_{\text{число}}$.

7.5. Типи граматик

Ієрархія Хомського: граматики загального вигляду, контекстно-залежні, контекстно-вільні, регулярні. Проблеми належності, порожності, еквівалентності для мов

Граматики можна класифікувати за виглядом їх продукцій. Така класифікація називається ***ієрархією Хомського.***

Нехай $G = (N, T, P, S)$ — граматика.

Тип 0. Будь-яка граматика визначеного раніше вигляду називається граматикою ***загального вигляду*** (з фразовою структурою, без обмежень).

Тип 1. Граматика називається ***контекстно-залежною (КЗ)*** або такою, що **не укорочує**, якщо кожна продукція з P має вигляд $\alpha \to \beta$, де $|\alpha| \le |\beta|$ або $\alpha \to \varepsilon$, де ε — порожній рядок (ε-продукції).

Тип 2. Граматика називається ***контекстновільною (КВ)***, якщо кожна продукція з P має вигляд $A \to \alpha$, де $A \in N$, $\alpha \in (N \cup T)^*$.

Тип 3. Граматика називається ***праволінійною*** (вирівняною вправо), якщо кожна продукція з P має вигляд $A \to xB$ або $A \to x$, де $A, B \in N$, $x \in T^*$. Граматика називається ***ліволінійною*** (вирівняною вліво), якщо кожна продукція з P має вигляд $A \to Bx$ або $A \to x$, де $A, B \in N$, $x \in T^*$. Ліволінійні і праволінійні граматики називаються ***регулярними.***

Ця ієрархія є монотонною відносно включень, тобто всі граматики типу 3 є граматиками типу 2, всі граматики типу 2 — граматиками типу 1 тощо, як зображено на рис. 7.1. Ця ієрархія включення справедлива і для мов, що породжені. При цьому мові привласнюється тип і якісна характеристика граматики, що породжує, — загального виду, КЗ, КВ, регулярна мова відповідно.

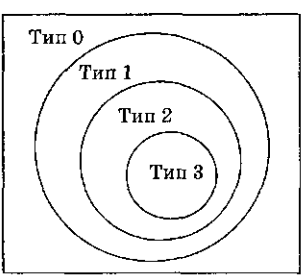


Рис. 7.1. Типи граматик

Наприклад, оскільки кожна праволінійна граматика є контекстно-вільною, то кожна праволінійна мова є КВ-мовою, та існують КВ-мови, що не є регулярними.

Включення типів є строгими в тому розумінні, що існують мови, які належать до типу k і не належать до типу $k+1$ $(0 \le k \le 2)$.

Але якщо мову можна генерувати граматикою типу k, то можливо, що її можна генерувати граматикою типу $k+1$. Наприклад, мова, що генерується за допомогою нерегулярної граматики, може виявитися регулярною.

Вже перше обмеження, що накладається на продукції граматик типу 1, визначає важливу властивість мови, що розпізнається. Для граматик, що не укорочують, без ε-продукцій доведено, що мови, які ними породжуються, легко розпізнаються, тобто за скінченне число кроків для будь-якого рядка можна визначити, чи належить цей рядок мові чи ні.

Друге обмеження на продукції у граматиці типу 2 звільняє від впливу контексту. Наприклад, продукція виду $\alpha_1\beta\alpha_2 \to \alpha_1\gamma\alpha_2$ означає дозвіл заміняти β на γ тільки у контексті α_1, α_2. Такі продукції називаються контекстно-зв'язаними. В протилежність цьому продукції, що не використовують контексту (правила КВ-граматик), називаються ***контекстно-вільними***. Граматики типу 1 називаються контекстно-залежними, оскільки можуть містити контекстно-зв'язані продукції.

Порівнюючи продукції граматики $G1$ з прикладу п. 7.3 і обмеження в ієрархії Хомського, приходимо до висновку, що ця граматика є контекстно-вільною. Наведемо приклади інших граматик і визначимо, до якого типу належить кожна з них.

Приклад. Ця граматика породжує послідовність (набір) із цифр 0, 1, 2, де кожна наступна цифра дорівнює сумі двох попередніх за модулем 3.

$$G_{\text{сумма}} = (\{A, C, S\}, \{0, 1, 2\}, P, S),$$

де P складається з набору продукцій:

1. $S \to AAC$.	8. $02C \to 022C$.
2. $A \to 0$.	9. $10C \to 101C$.
3. $A \to 1$.	10. $11C \to 112C$.
4. $A \to 2$.	11. $12C \to 120C$.
5. $C \to \varepsilon$.	12. $20C \to 202C$.
6. $00C \to 000C$.	13. $21C \to 210C$.
7. $01C \to 011C$.	14. $22C \to 221C$.

Приклад виводу: $S \Rightarrow {}_1AAC \Rightarrow {}_20AC \Rightarrow {}_402C \Rightarrow {}_8022C \Rightarrow$ $\Rightarrow {}_{14}0221C \Rightarrow {}_{13}02210C \Rightarrow {}_9022101C \Rightarrow {}_70221011C \Rightarrow {}_{10}02210112C \Rightarrow$ $\Rightarrow {}_502210112$.

За видом продукцій визначаємо, що ця граматика є контекстно-залежною.

Приклад. Побудуємо граматику G_{nn}, що породжує рядки виду 0^n1^n, $n \in N$ (запис a^n означає n символів «a», розміщених підряд).

$$G_{nn} = (\{S\}, \{0, 1\}, P, S),$$

де P складається з двох продукцій:

1. $S \to 0S1$.
2. $S \to \varepsilon$.

За видом продукцій легко визначити, що ця граматика відноситься до типу контекстно-вільних. Однак ця граматика не є регулярною, оскільки вона не є лівобічною або правобічною.

Приклад. Побудуємо граматику, що породжує рядки виду $0^n1^n2^n$, $n \in N$.

$$G_{nnn} = (\{A, B, S\}, \{0, 1, 2\}, P, S),$$

де P складається із семи продукцій:

1. $S \to 0SAB$.
2. $S \to \varepsilon$.
3. $BA \to AB$.
4. $0A \to 01$.
5. $1A \to 11$.
6. $1B \to 12$.
7. $2B \to 22$.

За видом продукцій визначаємо, що це контекстно-залежна граматика. Дійсно, в лівій частині продукцій 4–7. розташовано по два символи, перший з яких термінальний, а для контекстно-вільних граматик у лівій частині продукції повинен знаходитися тільки один символ, який є нетермінальним.

З чотирьох типів граматик ієрархії Хомського клас контекстно-вільних граматик найбільш важливий з точки зору застосувань до мов програмування і компіляції. За допомогою КВ-граматики можна визначити велику частину синтаксичної структури мови програмування. Крім того, вона є основою різних схем задання перекладів.

Чи існує у деякій фіксованій КВ-граматиці вивід вхідного рядка і якщо так, то визначити вивід цього рядка — одне з основних питань, яке розв'язують компілятор та інші програми ідентифікації і перекладу. Для спрощення цієї задачі застосовуються дерева виводу та еквівалентні перетворення граматик. Для граматик як засобів опису мови потрібно розв'язувати такі проблеми.

Проблема належності: «Дано опис мови визначеного типу і рядок ω, чи належить ω цій мові?»

Проблема порожності: «Дано опис мови визначеного типу; чи порожня ця мова?» Порожність мови означає, що не існує рядків, що належать цій мові.

Проблема еквівалентності: «Дано два описи мов однакового типу; чи співпадають ці мови?»

Доведено, що всі ці проблеми розв'язні для класу КВ-граматик.

Очевидно, що граматики в основному містять рекурсії, тобто з рядка, що містить деякий нетермінальний символ, виводиться за один або кілька кроків рядок, що містить той же самий нетермінал і тому не є «фінальним» рядком виводу, тобто власне рядком мови. Нагадаємо, що фіналом виводу повинен бути рядок мови, що складається тільки з термінальних символів. Граматики, що не містять рекурсій, породжують ***скінченні мови*** і великого інтересу не представляють. Граматики можна класифікувати за видом рекурсій, що в них містяться, як це зроблено при визначенні типу 3-граматик. Праволінійні граматики називають також ***праворекурсивними***, а ліволінійні — ***ліворекурсивними***. Граматики, що містять продукції виду $X \Rightarrow^* \alpha X \beta$, где $X \in N$, α, $\beta \in (N \cup T)^+$, називають такими, що ***самі включають.***

Від того, якого виду рекурсії містить граматика, залежить організація виводу для ефективного породження необхідних рядків мови.

Запитання

1. Визначити ієрархію Хомського для граматик.
2. Чи може:
   а) контекстно-вільна граматика бути регулярною;
   б) регулярна граматика не бути контекстно-вільною?
3. Чи завжди:
   а) контекстно-залежна граматика є регулярною;
   б) контекстно-вільна граматика є тією, що не укорочує?
4. Який тип мов породжує кожний з типів граматик?
5. Наведіть приклад впливу контексту на визначення значення слів «ключ», «лук» у реченнях російської мови. Що називають контекстом?
6. Наведіть приклад контексту у продукціях граматик.
7. Наведіть приклади продукцій, що містять ліву рекурсію, праву рекурсію, самовключення.

Завдання

1. Для граматики $G_{\text{сумма}}$ виконайте вивід іншого можливого рядка.
2. Доведіть, що граматика $G_{пп}$ дійсно виводить всі ланцюжки виду 0^n1^n. Наведіть приклад виводу.
3. Доведіть, що граматика $G_{ппп}$ дійсно виводить всі ланцюжки виду $0^n1^n2^n$. Наведіть приклад виводу.
4. Для граматики $G = (\{A, B, S\}, \{a, b\}, P, S)$ визначте тип в ієрархії Хомського. Список продукцій P такий:
   а) $S \to aAB$, $A \to Bb$, $B \to \varepsilon$;
   б) $S \to aB$, $AB \to a$;
   в) $S \to ABA$, $A \to aB$, $B \to ab$;
   г) $S \to bA$, $A \to aB$, $B \to abA$.
5. Чи існують регулярні скінченні мови?

7.6. Регулярні вирази і мови

Мови, що визначені граматиками типу 3 ієрархії Хомського, називаються ***регулярними***. Для завдання регулярних мов (множини рядків) використовуються також спеціальні вирази, що називаються регулярними виразами.

Визначення

Регулярний вираз у деякому алфавіті A визначається рекурсивно:
$\varnothing$ — регулярний вираз (порожня множина рядків);
ε — регулярний вираз (порожній рядок);
x — регулярний вираз, якщо $x \in A$ (x — символ алфавіту A);
XY, $X \cup Y$, X^* — регулярні вирази, якщо X, Y — регулярні вирази, де:
XY — конкатенація рядків, що визначені виразами X і Y;
$X \cup Y$ — об'єднання множини рядків, що визначені X і Y;
X^* — конкатенація будь-якого числа рядків, що визначені виразом X' або порожній рядок.

В регулярних виразах для визначення області дії операцій використовуються дужки. Якщо дужки відсутні, то пріоритет операцій такий:

1. X^*.
2. XY.
3. $X \cup Y$.

Приклад 1. Вираз 10^* розуміється як $1(0^*)$ і означає рядки, першим символом яких є «1», а потім іде будь-яке число нулів або жодного нуля. Наприклад, «100000000», «10», «1» та ін.

Приклад 2. Вираз $(10)^*$ означає будь-яку кількість «10», що йдуть поспіль, або порожній рядок. Наприклад, «101010101010», «10», ε тощо.

Приклад 3. Вираз $aaa \cup bbb$ означає два рядки — рядок «*aaa*» або рядок «*bbb*».

Приклад 4. Вираз $a(a \cup b \cup c)^*$ означає будь-який рядок, що складається із символів «*a*», «*b*», «*c*», який починається із символу «*a*», наприклад, «*abccbbaaabbcc*», «*aaaaaaaa*», «*accc*»,«*a*» та ін.

Зрозуміло, що регулярний вираз є рядком деякої мови. Доведено, що мову можна задати за допомогою регулярної граматики тоді і тільки тоді, коли її можна задати за допомогою регулярного виразу. Тому регулярну мову іноді називають регулярною множиною, маючи на увазі множину її рядків.

Приклад 5. Розглянемо граматику $G_{\text{число}}$ з п. 7.4.

Ця граматика є регулярною, оскільки кожна продукція з P має вид $A \to xB$ або $A \to x$, де $A, B \in N$, $x \in T^*$. Побудуємо регулярний вираз, що визначає ту ж мову, що і граматика. Число, що будується за допомогою граматики $G_{\text{число}}$, починається знаком, але знак може бути відсутнім. Вказівку знака або його відсутність реалізують продукції 2, 3, 4. Цьому відповідає регулярний вираз $(+ \cup - \cup \varepsilon)$. Далі йде ціла частина числа (продукції 1, 5, 6), яка визначається як ціле число — одна або більше цифр, причому нулі на початку числа припускаються. Регулярний вираз для цілої частини такий: $(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)$ $(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$. Таким же чином визначається і регулярний вираз для дрібної частини, з тією різницею, що перед дрібною частиною знаходиться кома (продукція 7), і дрібна частина може бути відсутня (продукція 8), тому ми додамо знак $\cup \varepsilon$.

Регулярний вираз для дрібної частини має вид:

$$(,(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)$$
$$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*) \cup \varepsilon.$$

Об'єднавши одержані вирази за допомогою конкатенації, одержимо шуканий регулярний вираз:

$$(+ \cup - \cup \varepsilon)\ (0 \cup 1 \cup ... \cup 9)(0 \cup 1 \cup ... \cup 9)^*$$

$$((,(0 \cup 1 \cup ... \cup 9)\ (0 \cup 1 \cup ... \cup 9)^*) \cup \varepsilon).$$

Цей вираз визначає ту ж мову, що і граматика $G_{\text{число}}$.

Запитання

1. Як зв'язані регулярні граматики, регулярні вирази і регулярні мови?
2. Дайте визначення регулярного виразу.
3. Як визначається пріоритет операцій у регулярних виразах?
4. Чи існує мова, задана за допомогою регулярної граматики, яку не можна задати за допомогою регулярного виразу? Чи існує мова, задана за допомогою регулярного виразу, яку неможливо задати за допомогою регулярної граматики?

Завдання

1. Для наведених регулярних виразів надайте словесний опис множини рядків і визначте, чи належить рядок «1011» цій множині:
   а) 10^*1^*; д) $0^*(10 \cup 11)^*$;
   б) $1(01)^*1^*$; е) $1^*01(0 \cup 1)$;
   в) $(10)^*(11)^*$; ж) $1(00)^*(11)^*$;
   г) $(10)^*1011$; з) $(1 \cup 00)(01 \cup 0)1^*$.
2. Опишіть наведені мови за допомогою регулярних виразів і граматик:
   а) рядки, що починаються з одного або більше нулів, за якими йде одна або більше одиниць;
   б) рядки з букв російської мови, що починаються з голосної букви;
   в) рядки з букв російської мови, що містять як підрядки слово «комп'ютер»;
   г) правильні речення російської мови, які можна скласти з набору слів: {Поздоровляю, Ви, дуже, добре, погано, відмінно, не, розв'язали, виконали, завдання, задачу, приклад}.
3. Складіть регулярні вирази, що визначають такі множини рядків:
   а) {ε, 0};
   б) {0, 11};
   в) {0, 11, 000};
   г) {00010, 01010, 00011, 01011}.
4. Складіть регулярні вирази для граматик $G = (\{A, B, S\}, \{0, 1\}, P, S)$ з такими наборами продукцій P:
   а) $S \to 0A$, $S \to 1B$, $A \to 0$, $B \to 0$;
   б) $S \to 1A$, $S \to 0$, $S \to \varepsilon$, $A \to 0B$, $B \to 1B$, $B \to 1$;
   в) $S \to 1B$, $S \to 0$, $A \to 1A$, $A \to 0B$, $A \to 1$, $A \to 0$, $B \to 1$.

7.7. Дерева виводів. Стратегії виводів

Дерево виводу. Стратегії виводу: зверху вниз, зліва направо, знизу вверх

В граматиці може бути кілька виводів, що еквівалентні у тому значенні, що в кожному виводі застосовуються одні й ті ж продукції в одних і тих же місцях, але у різному порядку. Визначити поняття еквівалентності двох виводів для граматик довільного виду складно, але у випадку КВ-граматик можна ввести зручне графічне зображення класу еквівалентних виводів, що називається деревом виводу.

Визначення

Позначене упорядковане дерево D називається ***деревом виводу*** (або ***деревом розбору***) у КВ-граматиці $G = (N, T, P, S)$, якщо виконані такі умови:

1. Корінь дерева D позначений S, а кожна вершина дерева D — символом з $N \cup T \cup \{\varepsilon\}$.

2. Для будь-якої внутрішньої вершини X_0 дерева і упорядкованого списку її синів $X_1, X_2, \ldots, X_n$ множина продукцій граматики P включає таку продукцію $A \to a_1, a_2, \ldots, a_n$, що $A, a_1, a_2, \ldots, a_n$ є позначками вершин $X_0, X_1, X_2, \ldots, X_n$ відповідно.

3. Якщо корінь дерева має єдиного нащадка, що позначений ε, то цей нащадок домінує над деревом, яке складається з єдиної вершини ε, і $S \to \varepsilon$ — продукція з множини P.

На рис. 7.2 зображено дерева виводів у граматиці G з продукціями $S \to aSbS \mid bSaS \mid \varepsilon$.

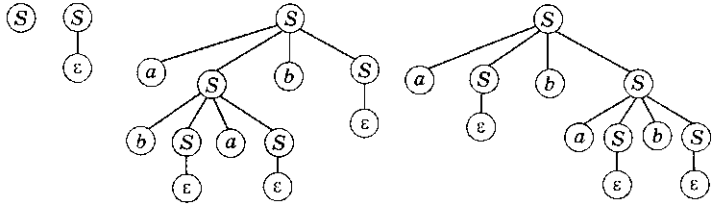


Рис. 7.2. Дерева виводів у граматиці з продукціями $S \to aSbS|bSaS|\varepsilon$

Оскільки будь-яке дерево виводу є упорядкованим, для його висячих вершин існує єдине лінійне упорядкування. Виписуючи позначки висячих вершин у цій упорядкованості, можна одержати рядок, вивід якого зображений цим деревом.

Вивід у граматиках схематично зображується у вигляді трикутника, зображеного на рис. 7.3, і заповнюється відповідно до набору продукцій граматики.

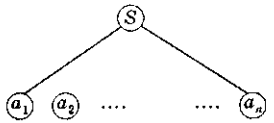
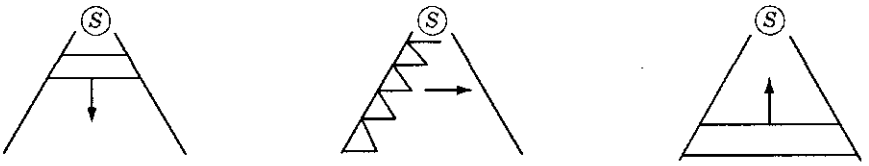


Рис. 7.3. Схема виводу в граматиках

В більшості випадків заповнення трикутника не можна розумно виконати за один крок, і за звичаєм воно одержується за допомогою послідовності піддерев. Ці послідовності піддерев можуть бути одержані кількома стратегіями, які зображені на рис. 7.4.



а) Зверху вниз

б) Зліва направо

в) Знизу вверх

Рис. 7.4. Стратегії виводу (розбору) у граматиках

При граматичному розборі порядок заповнення дерева зводиться до однієї з двох стратегій: ***зверху вниз*** (рис. 7.4,*а*) і ***знизу вверх*** (рис. 7.4,*в*). При заповненні зверху вниз — ***стратегія спадного аналізу*** — всі попередники кожної заповнюваної у дереві внутрішньої вершини до даного моменту вже заповнені. Заповнення знизу вверх — ***стратегія східного аналізу*** — характеризується тим, що всі спадкоємці кожної заповнюваної у дереві вершини вже належать побудованій частині дерева. Таким чином, при заповненні дерева зверху вниз рухання здійснюється від його кореня до висячих вершин, а при заповненні знизу вверх — навпаки. При заповненні зліва направо — ***стратегія горизонтального аналізу*** (рис. 7.4,*б*) — на кожному кроці відбувається заміна самого лівого нетермінального символу термінальним.

Запитання

1. Що таке вивід у граматиці?
2. Які виводи еквівалентні?

3. Що таке дерево виводу?
4. Який символ є вершиною дерева виводу?
5. Символи якого алфавіту є листями дерева виводу?
6. Які стратегії виводу у граматиках ви знаєте?

Завдання

1. Застосуйте різні стратегії виводу до граматик, що наведені як приклади у пп. 7.3–7.5.
2. Побудуйте дерева для виводів, що одержані при розв'язанні завдання 1.
3. G — граматика, $N = \{S\}$, $T = \{a, b, c\}$, початковий символ — S і продукції: $S \to bcS$, $S \to bbS$, $S \to a$, $S \to cb$. Побудуйте дерева виводів для:
   а) *bcbba*;
   б) *bbbcbba*;
   в) *bcabbbbbcb*.
4. G — граматика, $N = \{A, B, C, S\}$, $T = \{a, b, c\}$, початковий символ — S і продукції: $S \to AB$, $A \to Ca$, $B \to Ba$ |Xβ| b, $C \to cb$|β. Використовуючи різні стратегії виводу, визначте, чи належать ці рядки мові, породженій граматикою G:
   а) *baba*;
   б) *abab*;
   в) *cbaba*;
   г) *bbbcba*.

7.8. Побудова граматики мови програмування

Алфавіт мови, службові слова, ідентифікатори, оператори

Розглянемо деякі аспекти формального зображення мови програмування на прикладі мови Pascal.

Текст програми повинен міститися на дисковому файлі і зображувати послідовність рядків, що складаються із символів, які утворюють алфавіт мови. Рядки програми закінчуються спеціальними керуючими символами, що не входять до алфавіту.

Алфавіт мови Pascal складається з таких символів:

- Заголовні і рядкові латинські букви та символ '_':

  $A, B, C..., X, Y, Z, a, b, c, ..., x, y, z, \_$.

Букви використовуються для формування ідентифікаторів і службових слів.

- Десять арабських цифр від 0 до 9:
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Цифри використовуються для запису чисел та ідентифікаторів.

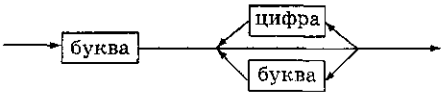
- Двадцять два спеціальних символів:
  + − * / = > < . , ; : @ ' ( ) [ ] { } # $ ^.

Спеціальні символи використовуються для конструювання знаків операцій, виразів, коментарів, а також як синтаксичні роздільники.

Символи з алфавіту мови використовуються для побудови базових елементів програм — лексем. Лексема — мінімальна одиниця мови, що має самостійне значення.

В числі лексем мови Pascal знаходяться службові (зарезервовані) слова. Це обмежена група слів, що побудовані з букв. Кожне службове слово зображує неподільне утворення, значення якого фіксоване у мові. Наприклад: begin, do, for, or, string, virtual.

Ідентифікатори (імена) — ще один вид лексем, що використовується у мові Pascal. Вони позначають імена змінних, констант, типів, позначок, процедур і функцій та формуються з букв і цифр згідно з відповідною діаграмою:



Побудуємо граматику G_{id}, яка породжує ідентифікатори:

G_{id} = ({БЦ, Буква, Цифра, Ідент.},
{*A*,..,*Z*, *a*,.., *z*,_,0,1,..,9}, *P*, Ідент.),

де *P* складається з набору продукцій:

1. <Ідент.>:: = <Буква> <БЦ>.
2. <БЦ>:: = <Буква> <БЦ> | <Цифра> <БЦ> | ε.
3. <Буква>:: = *A* | *B* | *C* | ... | *X* | *Y* | *Z* | *a* | *b* | *c* | ... | *x* | *y* | *z* | _.
4. <Цифра>:: = 0 | 1 | 2 | ... | 9.

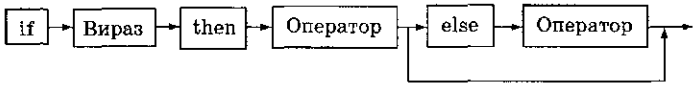
Центральним елементом мови Pascal є оператор. Набір операторів мови Pascal складає мінімальну множину конструкцій, необхідних для наглядного та однозначного зображення алгоритмів.

Мова містить такі оператори:

– оператор привласнювання;
– оператор процедури;

– оператор переходу;
– складовий оператор;
– умовний оператор;
– оператор варіанту;
– оператор циклу з передумовою;
– оператор циклу з постумовою;
– оператор циклу з параметром;
– оператор над записами;
– порожній оператор.

Розглянемо граматику G_{if}, що породжує умовні оператори, які схематично можна зобразити так:



G_{if} = ({Оператор, Вираз, Інакше, Умовний Оператор}, A, P, Умовний Оператор),

де A — алфавіт мови Pascal.

P складається з набору продукцій:

1. <Умовний Оператор>:: = if <Вираз.> then <Оператор> <Інакше>;
2. <Інакше>→ else <Оператор> | ε
3. <Оператор>::=begin<Оператор>; <Оператор>end | <Оператор>

....

Очевидно, що в наборі продукцій не вистачає продукцій породження операторів, виразів та ін., які не наведено тут через їх громіздкість. Тільки різновидів операторів мови Pascal, як було наведено раніше, існує близько 10.

Таким чином, для утворення граматики мови програмування необхідно задати такі продукції для породження всіх її одиниць, як оператор, ідентифікатор, вираз, опис, визначення, число та ін. Першою продукцією у граматиці мови програмування є продукція, що визначає конструкцію програми.

Запитання

1. Що складає алфавіт мови програмування?
2. Що таке ідентифікатор, оператор, умовний оператор, оператор циклу?

3. Який принцип побудови граматики мови програмування?
4. В яких випадках рядок, побудований за допомогою граматики G_{id}, не буде припустимим ідентифікатором у програмі?

Завдання

1. Запишіть у формі Бекуса — Наура граматику, яка буде породжувати оператор циклу із знайомої вам мови програмування.
2. Знайдіть записану у формі Бекуса — Наура граматику відомої вам мови програмування. Спробуйте побудувати невідомі вам синтаксичні конструкції мови, користуючись записаною у такій формі граматикою.
3. Побудуйте граматику, що виводить ідентифікатори використовуваної вами мови програмування. Запишіть, якщо можливо, відповідний регулярний вираз.
4. Побудуйте контекстно-вільну граматику для будь-якого відомого вам оператора циклу. Запишіть, якщо можливо, відповідний регулярний вираз.

розділ 8

Алгоритми

8.1. Поняття алгоритму

Інтуїтивне поняття алгоритму, алфавітні оператори, детерміновані та недетерміновані алгоритми

Як і поняття множини, у загальному випадку поняття алгоритм не має строгого математичного (і, якщо завгодно, логічного) визначення, і доводиться задовольнятися інтуїтивними уявленнями, заснованими на досвіді. Оскільки досвід у різних галузях свій, то і слово «алгоритм» вживається у різних значеннях: алгоритм поведінки, алгоритм побудови, алгоритм розв'язання проблеми (задачі) і т. д. Інакше кажучи, під алгоритмом у широкому значенні розуміють послідовність визначених дій або операцій, більш чи менш елементарних.

Тут ми використовуємо термін «алгоритм» виключно у значенні ***процедури обчислення***, що припускає програмування і комп'ютерну реалізацію. При цьому обчислення розглядається як у числовому, так і в алгебраїчному (символьному) алфавітах. Таке поняття алгоритму також не формалізоване, однак дозволяє висловити вимоги, які повинні задовольняти стандартні перетворення, що утворюють «обчислювальний» алгоритм. У першу чергу, ***розпізнаність*** та ***однозначна визначеність*** (***детермінованість***) для будь-якого представника з класу користувачів, яким адресується алгоритм. Далі — ***універсальність*** (***масовість***) алгоритму, тобто здатність «переробляти» будь-які вихідні дані з припустимої множини. Нарешті, ***дискретність*** і ***скінченність*** (результативність) ***алгоритму*** — одержання остаточного («вихідного») результату

за скінченне число кроків або одержання інформації про те, що результат існує.

Як приклади визначимо принциповий план побудови алгоритмів розв'язку двох відомих задач; повний опис цих алгоритмів потребує значно більших зусиль.

Приклад. Розглянемо алгоритм додавання цілих чисел у десятковій системі числення. Цей алгоритм перероблює рядки виду «$a + b$» у алфавіті, що складається з 10 цифр і знака додавання, у рядки у тому ж алфавіті, з якого виключено знак суми. Скінченна система правил, що визначають алгоритм, складається з правила порозрядного додавання, правила знаходження відповідних один одному розрядів доданків (у тому числі молодших розрядів), правил додавання цифр (таблиця додавання) і правила переносу у старший розряд у випадку «переповнення» молодшого.

Приклад. Розглянемо гру в шахи, трактуючи її як перетворення рядків у скінченному алфавіті. Як алфавіт тут природно використовувати стандартні у шаховій теорії символи. Рядками повинні бути скінченні послідовності символів алфавіту. Деякі з таких послідовностей зображують шахові позиції, інші — просто безглузді набори символів (наприклад, *aee*11*)*. Алгоритм задається лише для рядків, що зображують осмислені позиції, і полягає у перетворенні будь-якої заданої позиції у позицію, що виникає з неї після виконання чергового ходу. Як відомо, правила переміщення шахових фігур ще не визначають якийнебудь розумний вибір ходу у тій чи іншій конкретній позиції і, отже, не дозволяють побудувати осмислений алгоритм гри. Однак, як показує досвід, можна скласти скінченну систему стратегічних правил, що дозволяють у кожній конкретній позиції обирати єдиний найкращий (у деякому значенні) хід. Приєднуючи цю систему правил до правил переміщення фігур, ми одержуємо алгоритм, який природно назвати алгоритмом шахової гри. Слід відразу зазначити, що залежно від того, яким чином визначається якість позиції, змінюється і значення поняття «найкращого» ходу. Ця обставина визначає існування не одного єдиного алгоритму шахової гри, а множини таких алгоритмів.

Ми залишаємо осторонь питання про ймовірносні алгоритми, де вибір образу для того чи іншого рядка визначається не однозначно, а випадково відповідно до яких-небудь ймовірносних

критеріїв. Зрозуміло, що стосовно шахових алгоритмів розгляд випадкових переходів не є доречним. Таким чином, традиційно будемо називати алгоритмом лише строго детерміновану систему правил.

Визначення

Алгоритм, в якому на кожному кроці можна однозначно визначити наступний крок, називається ***детермінованим***.

Відповідно до прийнятої нами дискретної точки зору перетворення інформації можна зобразити як відображення множини рядків у деякому скінченному алфавіті в множину рядків у тому ж самому або будь-якому іншому скінченному алфавіті. Назвемо такі відображення ***алфавітними операторами***.

Алгоритми можна трактувати як алфавітні оператори, що задані за допомогою скінченних систем правил. У такому визначенні також немає математичної точності. Не подолавши цього недоліку, не можна зробити поняття алгоритму предметом математичної теорії. Було зроблено численні спроби уточнення поняття алгоритму, які привели до встановлення способів точного задання алгоритмів різних класів.

Запитання

1. Наведіть приклади алгоритмів, що використовуються у повсякденному житті.
2. Дайте визначення алфавітного оператора.
3. Який алгоритм називається детермінованим? Наведіть приклад.

8.2. Нормальні алгоритми Маркова

Нормальні алгоритми Маркова, принцип нормалізації, алгоритмічна розв'язність

Алгоритми Маркова, що називаються також ***нормальними алгоритмами***, перетворюють рядки, які задані у будь-якому скінченному алфавіті A, у рядки у тому ж самому алфавіті A. Нагадаємо, що рядок у алфавіті A — це упорядкована скінченна сукупність символів цього алфавіту. Нормальний алгоритм задається скінченною послідовністю підстановок.

Підстановка — це упорядкована пара рядків, необов'язково однакової довжини, що розділена символом «→». Наприклад, підстановкою є $ab \to cd$. Підстановка застосовна до вихідного рядка, якщо цей рядок містить як підрядок першу компоненту підстановки. В результуючому рядку маємо замість першої компоненти підстановки другу. Наприклад, застосувавши підстановку $ab \to cd$ до рядку «*sabfg*», одержимо рядок «*scdfg*». До рядку «*sbaff*» ця підстановка не застосовна, оскільки підрядка«*ab*» у рядку «*sbaff*» немає. Підстановки також можуть бути позначені як заключні. Застосовують нормальний алгоритм до вихідного рядка таким чином:

1) Із списка підстановок обираємо першу підстановку, яка застосовна до вихідного рядка.

2) Якщо жодна підстановка із списка підстановок не застосовна до вихідного рядка, то алгоритм припиняє роботу.

3) Застосовуємо підстановку з п. 1 до вихідного рядка, одержуємо результуючий рядок.

4) Якщо виконана підстановка є заключною, то алгоритм припиняє роботу. Інакше переходимо до п. 5.

5) Результуючий рядок вважаємо вихідним і переходимо до п. 1.

Зауваження: алгоритм може не містити заключних підстановок.

Порядок застосування підстановок, вказаний у п. 1, має велике значення для правильної роботи алгоритму. Список підстановок проглядається, починаючи з підстановки № 1. Якщо до вихідного рядку застосувати підстановку № 1 не можна, то розглядається підстановка № 2 тощо. Якщо й остання підстановка не застосовна до вихідного рядка, то алгоритм припиняє роботу, як вказано у п. 2.

Якщо нормальний алгоритм, застосований до деякого вихідного рядка, не припиняє роботу, а працює нескінченно, то вважається, що він не застосовний до цього вихідного рядка, оскільки не видає результату.

Приклад. Розглянемо приклад нормального алгоритму, який розміщує букви в алфавітному порядку. Алфавіт — $A = \{a, b, c\}$. Набір підстановок:

1. $ba \to ab$.
2. $ca \to ac$.
3. $cb \to bc$.

Застосуємо цей алгоритм до рядка «*cbabc*». В процесі роботи алгоритму послідовно одержуємо рядки:

«*cbabc*» вихідна;
«*cabbc*» підстановка 1;
«*acbbc*» підстановка 2;
«*abcbc*» підстановка 3;
«*abbcc*» підстановка 3.

Результуючий рядок — «*abbcc*».

Приклад. В алфавіті $A = \{a, b, c, d, e, ..., y\}$ до рядка «*abcd*» список підстановок

1. $a \to c$;
2. $b \to d$;
3. $cdc \to ab$

не застосовний, оскільки виникнення циклу

$$(abcd) \xrightarrow{1} (cbcd) \xrightarrow{2} (cdcd) \xrightarrow{3} (abcd)$$

призводить до нескінченної роботи алгоритму Маркова, якщо не вказано, яка з трьох підстановок у списку є заключною.

Приклад. Алфавіт $A = \{1, 0, \cap, \cup, (,)\}$. Набір підстановок:

1. $1 \cap 0 \to 0$.
2. $1 \cap 1 \to 1$.
3. $0 \cap 1 \to 0$.
4. $0 \cap 0 \to 0$.
5. $1 \cup 0 \to 1$.
6. $1 \cup 1 \to 1$.
7. $0 \cup 1 \to 1$.
8. $0 \cup 0 \to 0$.
9. $(1) \to 1$.
10. $(0) \to 0$.

Застосуємо нормальний алгоритм до рядка «$(0 \cup 1) \cap 1 \cup 0$». В процесі роботи алгоритму послідовно одержуємо рядки:

«$(0 \cup 1) \cap 1 \cup 0$» вихідна;
«$(0 \cup 1) \cap 1$» підстановка 5;
«$(1) \cap 1$» підстановка 7;
«$1 \cap 1$» підстановка 9;
«1» підстановка 2.

Результуючий рядок — «1».

Можна показати, що будь-який алгоритм еквівалентний деякому нормальному алгоритму. Інакше кажучи, для будь-якого алгоритму над довільним скінченним алфавітом A можна побудувати нормальний алгоритм, що перероблює рядки точно так, як і вихідний алгоритм.

Таким чином, можна сформулювати ***принцип нормалізації алгоритму***, який полягає в тому, що будь-який алгоритм

в алфавіті A еквівалентний деякому нормальному алгоритму в алфавіті A. Коротко формулюють так: ***будь-який алгоритм в A нормалізується.***

Принцип нормалізації, з одного боку, стверджує існування нормального алгоритму для розв'язку кожної задачі, для якої існує алгоритм в A. З іншого боку, він дозволяє стверджувати, що якщо для розв'язку якої-небудь задачі нормальний алгоритм неможливий (його не існує), то не має сенсу шукати будь-який інший алгоритм, тобто принцип нормалізації є засобом дослідження питання ***алгоритмічної розв'язності*** або ***нерозв'язності задачі.***

Запитання

1. Що являють собою алгоритми Маркова?
2. В якому порядку підстановки застосовуються до вихідного рядка?
3. В якому випадку вважається, що підстановка застосовна до вихідного рядка?
4. В якому випадку алгоритм Маркова припиняє роботу?
5. В якому випадку вважається, що алгоритм Маркова не застосовний до вихідного рядка?
6. Сформулюйте принцип нормалізації.

Завдання

1. Застосуйте алгоритм з прикладу 1 до рядків «*cabcc*», «*cccba*».
2. Для алфавіту $A = (0, 1, 2, 3)$ побудуйте нормальний алгоритм, що розміщує цифри вихідного рядка у порядку спадання.
3. Застосуйте алгоритм з прикладу 3 до рядків «$1 \cup (0 \cap 1) \cup 1$», «$(0 \cap 1 \cup 0) \cup 0 \cap 1$».
4. В алфавіті $A = \{0, 1, \neg, \cap, \cup, (,)\}$ побудуйте нормальний алгоритм, який обчислює значення виразу аналогічно прикладу 3 (у виразі може бути присутнім знак «$\neg$» — «не»).

8.3. Алгоритми та рекурсивні функції

Обчислювана функція, тезис Черча, базові рекурсивні функції, оператори побудови рекурсивних функцій

Вивчаючи поняття алгоритму, можна звернутися до вже вивченого поняття функції. Вважається, що такі об'єкти, як рекурсивні функції існують або не існують точно тоді ж, коли і алгоритми.

Теоретико-множинне визначення функції виглядає так: функцією $y = f(x)$ називається відображення множини X у множину Y, яке кожному елементу $x \in X$ ставить у відповідність

точно один елемент $y \in Y$. Алгоритм можна вважати конструктивним правилом, що ставить у відповідність кожному припустимому значенню незалежної змінної значення функції, тобто правилом, що індукує функцію. Не для будь-якої функції існує алгоритм, що обчислює її значення.

Визначення

Функція, для якої існує будь-який конструктивний спосіб одержання її значення за значенням аргументу *за скінченне число кроків*, називається ***обчислюваною.***

Очевидно, що функції, які індукуються алгоритмами, обчислювані. В деяких питаннях замість розгляду алгоритмів можна обійтися вивченням обчислюваних функцій, наприклад, якщо буде доведено, що для розв'язку деякої проблеми не існує обчислюваної функції, то тим самим буде доведене і неіснування алгоритму. Вивчаючи різні обчислювані функції, А. Черч у 1936 р. висловив твердження, відоме тепер як тезис Черча.

Тезис Черча: Будь-яка обчислювана функція від n натуральних аргументів, що приймає натуральні значення, тотожно дорівнює деякій рекурсивній функції з тим же числом незалежних змінних.

Тезис Черча зіставляє інтуїтивне поняття обчислюваної функції з точним математичним поняттям рекурсивної функції. Стверджується, що яким би не був алгоритм, що оперує цілими невід'ємними числами, існує алгоритм, супутній рекурсивній функції, який йому еквівалентний. Алгоритмом, що супутній рекурсивній функції, називається алгоритм обчислення її значення (тобто той, що є законом завдання рекурсивної функції). Розглянемо формальне визначення класу рекурсивних функцій.

Рекурсивною є функція, що використовує саму себе у своєму визначенні. Рекурсії ми зустрічаємо як у математиці, так і в повсякденному житті. Наприклад, рекурсивне визначення рядка можна сформулювати так: «Рядком називається або порожній рядок, або рядок, до якого приписана буква».

Другий приклад рекурсії — метод математичної індукції для доведень. Цей метод полягає у такому:

1) Перевіряємо істинність твердження для x_0.

2) Припускаємо, що воно істинне для x_i.

3) Якщо, використовуючи 2), вдається довести, що твердження істинне для x_{i+1}, то можемо зробити висновок про істинність цього твердження для x_i при будь-якому $i \in N$.

Клас рекурсивних функцій будується шляхом вказівки деякого числа елементарних (базових) функцій і трьох операцій над функціями, застосування яких і породжує весь клас.

Визначення

Наведені функції оголошуються елементарними або ***базовими функціями,*** та їх область визначення, і область значення задаються з поширеного натурального ряду $N_0 = N \cup \{0\}$:

1) Функції будь-якого числа змінних $O_n(x_1, ..., x_n)$, тотожно рівні нулю, де n — число змінних:
$O_n(x_1, ..., x_n) = 0, \forall x_i$.

2) Функції вибору $V_{n,i}(x_1, ..., x_n)$ або селекторні функції, де n — число змінних, i — номер змінної у списку змінних функції, значення якої приймає селекторна функція:
$V_{n,i}(x_1, ..., x_n) = x_i$.

3) Функція проходження, що позначена через $\lambda(x) = x + 1$. Значенням функції є число, що безпосередньо є наступним за x.

Нижній індекс біля знаку функції позначає ***ранг функції*** — кількість змінних цієї функції. Наприклад, функція F_n залежить від n змінних. Нижній індекс біля позначення змінної означає порядковий номер цієї змінної у списку змінних функції.

Для кожної базової функції існує простий алгоритм, що обчислює її значення. Він називається ***супутнім алгоритмом*** цієї ***функції***.

Наведемо ***оператори***, за допомогою яких з рекурсивних функцій можна одержати нові рекурсивні функції.

Оператор підстановки (суперпозиції) S породжує функцію Φ_m від m аргументів $(x_1, ..., x_m)$ за функцією F_n від n аргументів $f_1, ..., f_n$. Функція Φ_m визначається умовою $\Phi_m = F_n(f_1, ..., f_n)$. Оператор підстановки обчислює значення функцій $f_1, ..., f_n$, що залежать від m аргументів, і приймає їх за значення відповідних цим функціям аргументів функції F_n, після чого обчислює значення функції F_n. Результат вважається значенням функції Φ_m і позначається

$$\Phi_m = S(F_n, f_1, f_2, \ldots, f_n).$$

Оператор примітивної рекурсії R за двома функціями φ_{n-1} і Ψ_{n+1} будує функцію n незалежних змінних $\Phi_n = R(\varphi_{n-1}, \Psi_{n+1})$:

$$\Phi_n(x_1, \ldots, x_{n-1}, 0) = \varphi_{n-1}(x_1, \ldots, x_{n-1});$$

$$\Phi_n(x_1, \ldots, x_{n-1}, i+1) = \Psi_{n+1}(x_1, \ldots, x_{n-1}, i, \Phi_n(x_1, \ldots, x_{n-1}, i)),$$

де $i = 0, 1, \ldots, x_{n-1}$.

Оператор мінімізації μ: $\varphi_{n+1} \to \Phi_n$ за заданою функцією $\varphi_{n+1}(x_1, \ldots, x_n, x_{n+1})$ будує «мінімізовану» функцію Φ_n від меншого числа аргументів за таким правилом. Аргументу x_{n+1} функції $\varphi_{n+1}(x_1, \ldots, x_n, x_{n+1})$ надають послідовно значення 0, 1, 2 і т. д. доти, доки функція φ_{n+1} не стане перший раз дорівнювати нулю. Одержане значення x_{n+1} приймають за значення результуючої функції:

$$\Phi_n(x_1, \ldots, x_n) = x_{n+1}, \text{ якщо } \varphi_{n+1}(x_1, \ldots, x_n, x_{n+1}) = 0.$$

Якщо до закінчення процесу на якому-небудь кроці φ_{n+1} стане невизначеною, то процес безрезультатно обривається.

Рекурсивними називаються базові функції і функції, що одержуються шляхом застосування до базових функцій скінченного числа операторів підстановки, рекурсії та мінімізації. Як вже відзначалося, яким би не був алгоритм, що оперує цілими невід'ємними числами, існує еквівалентний йому алгоритм, супутній деякій рекурсивній функції. Це означає, що виконання алгоритму у певному значенні еквівалентне обчисленню значення рекурсивної функції.

Розглянемо найпростіші рекурсивні функції.

Функція $F(x) = x + 1$ співпадає з базовою функцією $\lambda(x)$, а значить є рекурсивною.

Функція $F(x, y) = x + y$ також є рекурсивною. Доведемо це.

1) Побудуємо рекурсивну функцію $f(x, y, z) = z + 1$, застосувавши оператор підстановки до базових функцій $V_{3,3}(x, y, z) = z$ и $\lambda(z) = z + 1$. Підставимо $\lambda(z)$ замість z. Одержана функція $V_{3,3}(x, y, \lambda(z)) = z + 1$ еквівалентна функції $f(x, y, z) = z + 1$.

2) Застосуємо оператор рекурсії до функцій $V_{1,1}(x) = x$ і $f(x, y, z) = z + 1$.

Поклавши у визначенні оператора рекурсії $\Phi_2(x_1, x_2) = F(x_1, x_2)$, $x_1 = x$, $x_2 = y$, $x_3 = z$, $\Psi_{2+1}(x, y, z) = f(x, y, z)$, $\Phi_2(x, 0) = F(x, 0) = V_{1,1}(x) = x$, одержуємо послідовно:

$$F(x, 0) = V_{1,1}(x) = x$$
$$F(x, 0) = f(x, 0, F(x, 0)) = f(x, 0, x) = x + 1$$
$$F(x, 2) = f(x, 1, F(x, 1)) = f(x, 1, x + 1) = x + 2$$
...
$$F(x, y) = f(x, y - 1, F(x, y - 1)) = F(x, y - 1) + 1 =$$
$$= x + (y - 1) + 1 = x + y.$$

Шукана функція $F(x, y) = x + y$ зображена за допомогою оператора рекурсії R через базові функції: $F(x, y) = R(V_{1,1}(x), f(x, y, z))$. Аналогічно доводиться, що $F(x, y) = x \cdot y$ також зображувана у вигляді рекурсивної функції.

Зауважимо, що алгоритм, супутній рекурсивній функції, не є єдиним. Так, у прикладі $F(x, y) = x + y$ на першому кроці обчислення функції $F(x, y)$ можна побудувати функцію f як $f(x, y, z) = \lambda(V_{3,3}(x, y, z)) = z + 1$.

Згідно з тезисом Черча для будь-якої обчислюваної функції можна побудувати еквівалентну їй рекурсивну функцію. Якщо для розв'язку якої-небудь задачі можна побудувати рекурсивну функцію, то існує і алгоритм, що розв'язує цю задачу. В протилежному випадку не існує і алгоритму, що розв'язує цю задачу.

Запитання

1. Дайте визначення обчислюваної функції.
2. Наведіть приклади рекурсії.
3. Чи може будь-яка обчислювана функція бути зображена у рекурсивному вигляді?
4. Для чого в теорії алгоритмів застосовуються рекурсивні функції?
5. Побудувати алгоритм обчислення функції $F(x, y) = x \cdot y$ через базові функції та оператори.

8.5. Приклади побудови алгоритмів

Правильний алгоритм, алгоритм сортування вставками, алгоритм сортування злиттям, принцип «розділяй та володарюй»

В програмуванні алгоритм — це формально описана обчислювальна процедура, що використовує вихідні дані, які називаються також ***входом алгоритму*** або його ***аргументом***, і видає результат обчислень на вихід. Формулювання задачі

містить вимоги, яким повинно задовольняти розв'язання задачі, а алгоритм, що розв'язує цю задачу, знаходить об'єкт, що задовольняє ці вимоги.

Побудову алгоритмів розглянемо на прикладі задачі сортування:

Вхід: Послідовність n чисел $(a_1, a_2, ..., a_n)$.

Вихід: Переставлення $a'_1, a'_2, ..., a'_n$ вихідної послідовності, для якої $a'_1 \le a'_2 \le ... \le a'_n$.

Наприклад, одержавши на вхід (31, 41, 59, 26, 41, 58), алгоритм сортування повинен видати на вихід (26, 31, 41, 41, 58, 59).

Багато алгоритмів використовують сортування як проміжний крок. Є багато різних алгоритмів сортування; вибір одного з них у конкретній ситуації залежить від довжини послідовності, що сортується, від того, наскільки вона вже відсортована, а також від типу пам'яті, що використовується.

Визначення

Алгоритм вважають ***правильним***, якщо при будь-якому припустимому (для даної задачі) вході він закінчує роботу і видає результат, що задовольняє вимоги задачі. В цьому випадку говорять, що алгоритм розв'язує дану обчислювальну задачу.

Неправильний алгоритм може (для деякого входу) зовсім не зупинитися або дати неправильний результат. Проте неправильний алгоритм може бути корисний, якщо помилки досить рідкі. Але це все ж таки, мабуть, виняток, ніж правило.

Алгоритм може бути записаний на російській, англійській або іншій природній мові у вигляді комп'ютерної програми або навіть у машинних кодах. Важливо тільки, щоб процедура обчислень мала точний опис, що не припускає двозначних тлумачень на кожному кроці.

В наших прикладах будемо записувати алгоритми за допомогою псевдокоду, що нагадує мови програмування. Тут іноді дозволено описувати дії алгоритму «своїми словами», якщо так виходить зрозуміліше. Крім того, опускаються технологічні подробиці (обробка помилок та ін.), які необхідні у реальній програмі.

Наведемо основні угоди, які використовуються у псевдокоді:

1. Знак «←» означає присвоєння значення.

2. Цикли *while*, *for*, *repeat* і умовні конструкції *if*, *then*, *else* мають те ж значення, що і у Pascal.

3. Індекс масиву пишеться у квадратних дужках; $A[i]$ є i-й елемент масиву A. Знак «..» виділяє частину масиву. Символ $A[1..j]$ позначає ділянку масиву A, що включає $A[1]$, $A[2]$, ..., $A[j]$.

4. Відступ від лівого поля вказує на рівень вкладеності. Наведемо приклад запису процедури:

```
PROCEDURE 1 (X, n, s, k)
1. s ← 0
2. k ← 0
3. for i ← 1 to n do
4. s ← s + X[i]
5. if s = 0 then
6. k ← i
7. x ← k + 1.
```

Тіло циклу for (рядок 3) складається з рядків 4–6, а до умови if (рядок 5) належить тільки рядок 6, але не 7. Це робить зайвим спеціальні команди типу begin та end для початку і кінця блоку. В реальних мовах програмування така угода застосовується рідко, оскільки ускладнює читання програм, що переходять із сторінки на сторінку.

5. Змінні (у нашому прикладі — i) локальні всередині процедури.

6. Символ ◊ починає коментарій, що іде до кінця рядка.

Сортування вставками

Сортування вставками зручне для сортування стислих послідовностей. Саме таким способом звичайно сортують карти: тримаючи у лівій руці вже упорядковані карти й узявши правою рукою чергову карту, ми вставляємо її у потрібне місце, порівнюючи з тими, що вже маємо і йдучи справа наліво.

Запишемо цей алгоритм у вигляді процедури INSERTIONSORT, параметром якої є масив $A[1..n]$ (послідовність довжини n, підлягаюча сортуванню). Позначимо число елементів у масиві A через length[A]. Послідовність сортується «на місці» без додаткової пам'яті: крім масиву ми використовуємо лише фіксоване число комірок пам'яті. Після виконання процедури INSERTIONSORT масив A упорядкований за зростанням.

```
INSERTIONSORT (A)
1   for j ← 2 to length [A] do
2       key ← A[j]
3       ◊ додати A[j] до відсортованої частини A[1...j − 1]
4       i ← j − 1
5       while i > 0 and A[i] > key do
6       A[i + 1] ← A[i]
7       i ← i − 1
8       A[i + 1] ← key.
```

На рис. 8.1 зображено роботу алгоритму при $A = \{5, 2, 4, 6, 1, 3,\}$, $n = 6$, індекс j вказує чергову карту (тільки що узяту зі стола). Ділянки $A[1...j-1]$ складають вже відсортовані карти (ліва рука), $A[j+1...n]$ — ще не переглянуті. В циклі for індекс j перебігає масив зліва направо. Ми беремо елемент $A[j]$ (рядок 2 алгоритму) і переміщуємо вправо елементи, що йдуть перед ним і більші за нього за величиною (починаючи з $(j-1)$-го), звільняючи місце для взятого елемента (рядки 4–7). В рядку 8 елемент $A[j]$ ставиться на звільнене місце.

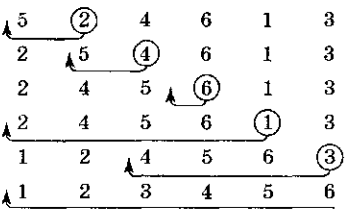


Рис. 8.1. Робота процедури INSERTIONSORT
Вихідний рядок — $A = \{5, 2, 4, 6, 1, 3, \}$. Позиція j зображена колом

Сортування злиттям. Принцип «розділяй та володарюй»

Є багато стандартних способів, що використовуються під час побудови алгоритмів. Сортування вставками є прикладом алгоритму, що діє за кроками: до відсортованої частини масиву додаються елементи один за одним. Покажемо у дії інший підхід, який називають «розділяй та володарюй», і побудуємо за його допомогою значно швидкіший алгоритм сортування.

Багато алгоритмів за своєю природою рекурсивні: розв'язуючи деяку задачу, вони викликають самих себе для розв'язку її підзадач. Ідея методу «розділяй та володарюй» полягає як

раз у цьому. Спочатку задача розбивається на кілька підзадач меншого розміру. Потім ці задачі розв'язуються (за допомогою рекурсивного виклику — або безпосередньо, якщо розмір достатньо малий). Нарешті, їх розв'язки комбінуються, й одержується розв'язок вихідної задачі.

Для задачі сортування ці три етапи виглядають так. Спочатку масив розбивається на дві половини меншого розміру. Потім кожна з половин сортується окремо. Після цього залишається з'єднати два упорядкованих масиви половинного розміру в один. Рекурсивне розбиття задачі на менші відбувається доти, доки розмір масиву не дійде до одиниці (будь-який масив довжини 1 можна вважати упорядкованим).

З'єднання двох упорядкованих масивів в один виконується за допомогою допоміжної процедури MERGE (A, p, q, r). Параметрами цієї процедури є масив A і числа p, q, r, що вказують границі ділянок, які зливаються. Процедура припускає, що $p \le q \le r$ і що ділянки $A[p..q]$ і $A[q+1..r]$ вже відсортовані, і зливає їх в одну ділянку $A[p..r]$.

Тепер складемо процедуру сортування злиттям MERGESORT (A, p, r), яка сортує ділянку $A[p..r]$ масиву A, не змінюючи решту частини масиву. При $p \ge r$ ділянка містить максимум один елемент і тим самим вже відсортована. У протилежному випадку ми відшукуємо число q_1, яке ділить $[p-r]$ з числом елементів $n = r - p + 1$ на дві приблизно рівні частини $A[p..q]$ (містить $\lceil n/2 \rceil$ елементів) і $A[q+1..r]$ (містить $\lfloor n/2 \rfloor$ елементів). Тут через $\lfloor x \rfloor$ ми позначаємо цілу частину x (найбільше ціле число, менше або таке, що дорівнює x), а через $\lceil x \rceil$ — найменше ціле число, більше або таке, до дорівнює x.

```
MERGESORT (A, p, r)
1  if p < r then
2  q ← ⌊(p + r)/2⌋
3  MERGESORT (A, p, q)
4  MERGESORT (A, q + 1, r)
5  MERGE (A, p, q, r).
```

Весь масив тепер можна відсортувати за допомогою виклику MERGESORT (A, 1, *length*[A]). Якщо довжина масиву n = *length*[A] є степінь двійки, то у процесі сортування відбудеться злиття пар елементів у відсортовані ділянки довжини 2, потім злиття пар таких ділянок у відсортовані ділянки довжини 4 і так далі

до n (на останньому кроці з'єднуються дві відсортвані ділянки довжини $n/2$). Цей процес зображено на рис. 8.2.

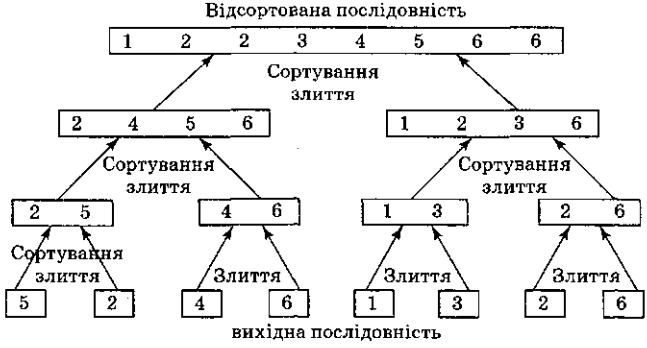


Рис. 8.2. Сортування злиттям для масиву $A = \{5, 2, 4, 6, 1, 3, 2, 6\}$

Запитання

1. Що називається алгоритмом у програмуванні?
2. Який алгоритм можна вважати правильним?
3. Запишіть алгоритми сортування вставками та злиттям, поясніть принципи їх роботи.

Завдання

1. Ідучи за зразком (рис. 8.1), зобразіть, як працює InsertionSort на вході $A = (31, 41, 59, 26, 58)$.
2. Змініть процедуру InsertionSort так, щоб вона сортувала числа у незростаючому порядку (замість неспадного).
3. Розглянемо таку задачу пошуку:
   Вхід: Послідовність n чисел $A = \{a_1, a_2, ..., a_n\}$ і число v.
   Вихід: Індекс i, для якого $v = A[i]$, або спеціальне значення NIL, якщо v не зустрічається в A. Напишіть алгоритм лінійного пошуку, який послідовно переглядає A у пошуках v.
4. Йдучи за зразком (рис. 8.2), покажіть роботу сортування злиттям для масиву $A = (3, 41, 52, 26, 38, 57, 9, 49)$.
5. Напишіть текст процедури Merge (A, p, q, r).

8.6. Складність алгоритмів

Складність опису та обчислення алгоритму, розмір задачі, часова складність, місткісна складність, порядок складності, аналіз складності алгоритмів сортування

Розглядаючи різні алгоритми розв'язку одної і тієї ж задачі, слід проаналізувати, скільки обчислювальних ресурсів вони потребують, і обрати найбільш ефективний алгоритм. Безумовно, ефективність залежить від вибору обчислювальної

моделі. В наших прикладах розглядається однопроцесорна машина з довільним доступом (randomaccess machine, RAM), що не передбачає паралельного виконання операцій.

Розрізняють складність самого алгоритму і складність його опису (реалізації).

Визначення

Складність опису алгоритму — це величина, що характеризує довжину опису алгоритму. ***Складність алгоритму*** (складність обчислення алгоритму) — це функція, що дає числову оцінку тривалості роботи алгоритму з моменту надходження на його вхід початкових даних до моменту одержання алгоритмом вихідних даних.

Для оцінки складності алгоритмів існує багато критеріїв. Найчастіше розглядається оцінка часу повного розв'язку задачі і зріст місткості пам'яті при збільшенні об'єму вхідних даних.

Визначення

Розміром задачі називається число, яке виражає міру кількості вхідних даних.

Наприклад, розміром задачі множення матриць може бути найбільший розмір матриць-чинників. Розміром задачі на графі може бути число ребер даного графа.

Визначення

Час, що витрачає алгоритм, як функція розміру задачі, називається ***часовою складністю*** цього алгоритму.
Об'єм пам'яті, що використовує алгоритм, як функція розміру задачі, називається ***місткісною складністю*** цього алгоритму.

При заданих обмеженнях пам'яті та швидкості комп'ютера складність алгоритму визначає в решті решт розмір задач, які можна розв'язати за цим алгоритмом. Розглянемо часову складність на прикладі алгоритмів сортування.

Аналіз сортування вставками

Час сортування вставками залежить від розміру масиву, що сортується: чим більше масив, тим більше треба часу. За звичаєм вивчається функціональна оцінка зростання часу роботи алгоритму від зростання розміру входу. Проте для алгоритму сортування вставками важливий не тільки розмір масиву, але

й порядок його елементів: якщо масив майже упорядкований, то часу треба менше.

Як виміряти розмір входу? Це залежить від конкретної задачі. В одних випадках розміром розумно вважати число елементів на вході (як у задачі сортування). В інших — більш природно вважати розміром загальне число бітів, що необхідне для зображення всіх вхідних даних. Іноді розмір входу вимірюється не одним числом, а кількома (наприклад, число вершин і число ребер графа).

Час роботи алгоритму виразимо через число елементарних кроків, які виконує цей алгоритм під час роботи. Питання тільки в тому, що вважати елементарним кроком. Будемо припускати, що один рядок псевдокоду вимагає досить визначеного фіксованого числа операцій (якщо тільки це не словесний опис якихось складних дій). Будемо розрізняти також виклик процедури (на який іде фіксоване число операцій) та її здійснення, яке може бути довгим.

Повернемося до процедури алгоритму InsertionSort та відзначимо біля кожного рядка її вартість (число операцій) та число разів, яке цій рядок здійснюється. Для кожного j від 2 до n (тут $n = length[A]$ — розмір масиву) підрахуємо, за скільки разів буде здійснено рядок 5, і позначимо це число через t_j. Зауважимо, що рядки всередині циклу виконуються на один раз менше, ніж перевірка, оскільки остання перевірка виходить з циклу.

	InsertionSort (A)	вартість	число разів
1.	for $j \leftarrow 2$ to $length[A]$ do	c_1	n
2.	key $\leftarrow A[j]$	c_2	$n-1$
3.	◊ додати $A[j]$ до відсортованої частини $A[1..j-1]$.	0	$n-1$
4.	$i \leftarrow j-1$	c_4	$n-1$
5.	while $i > 0$ and $A[i] >$ key do	c_5	$\sum_{j=2}^{n} t_j$
6.	$A[i+1] \leftarrow A[i]$	c_6	$\sum_{j=2}^{n} (t_j - 1)$
7.	$i \leftarrow i-1$	c_7	$\sum_{j=2}^{n} (t_j - 1)$
8.	$A[i+1] \leftarrow$ key	c_8	$n-1$.

Рядок вартістю c, повторений m разів, дає внесок cm у загальне число операцій (для кількості використаної пам'яті цього сказати, взагалі кажучи, не можна). Склавши внески всіх рядків, одержимо вираз, що визначає часову складність процедури InsertionSort — $T(n)$:

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) +$$
$$+ c_5 \sum_{j=2}^{n} t_i + c_6 \sum_{j=2}^{n} (t_i - 1) + c_7 \sum_{j=2}^{n} (t_j - 1) + c_8(n-1).$$

Час роботи алгоритму залежить не тільки від n, але і від того, який саме масив розміру n поданий на вхід. Найбільш сприятливий випадок, коли масив вже відсортований. Тоді цикл у рядку 5 завершується одразу після першої перевірки (оскільки $A[i] \le \text{key}$ при $i = j - 1$), так що всі t_i рівні 1, і загальний час є

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5(n-1) + c_8(n-1) =$$
$$= (c_1 + c_2 + c_4 + c_5 + c_8)n - (c_2 + c_4 + c_5 + c_8).$$

Таким чином у найбільш сприятливому випадку час $T(n)$, необхідний для сортування масиву розміру n, є лінійною функцією від n, тобто має вигляд $T(n) = an + b$ для деяких констант a і b (ці константи визначаються обраними значеннями $c_1, \ldots, c_8$).

Якщо ж масив розміщений у зворотному (спадному) порядку, то час роботи процедури буде максимальним, кожний елемент $A[j]$ доведеться порівнювати зі всіма елементами $A[1], \ldots, A[j-1]$. При цьому $t_j = j$. Обчислимо суми

$$\sum_{j=2}^{n} j = \frac{2+n}{2}(n-1) = \frac{n(n+1)}{2} - 1;$$
$$\sum_{j=2}^{n} (j-1) = \sum_{j=2}^{n-1} j = \frac{1+n-1}{2} \cdot (n-1) = \frac{n(n-1)}{2}.$$

Одержуємо, що у гіршому випадку час роботи алгоритму дорівнює

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5\left(\frac{n \cdot (n+1)}{2} - 1\right) +$$
$$+ c_6 \frac{n(n-1)}{2} + c_7 \frac{n(n-1)}{2} + c_8(n-1) =$$
$$= \left(\frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2}\right)n^2 + \left(c_1 + c_2 + c_4 + \frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2} + c_8\right)n -$$
$$- (c_2 + c_4 + c_5 + c_8).$$

Тепер функція $T(n)$ — квадратична, тобто має вигляд $T(n) = an^2 + bn + c$, де константи a, b і c визначаються значеннями $c_1, \ldots, c_8$.

Цей приклад показує, що часи роботи алгоритму у гіршому та кращому випадках можуть сильно відрізнятися. Для одержання гарантованих рамок оцінюється максимальний час роботи для входів заданого розміру.

В деяких випадках нас буде цікавити також середній час роботи алгоритму на входах даної довжини. Безумовно, ця величина залежить від обраного розподілу ймовірностей, і на практиці реальний розподіл входів може відрізнятися від передбачуваного, яке звичайно вважають рівномірним. Іноді рівномірний розподіл моделюють, використовуючи датчик випадкових чисел.

Наш аналіз часу роботи процедури INSERTIONSORT був заснований на кількох припущеннях, що спрощують. Спочатку ми припустили, що час виконання i-го рядка постійний і дорівнює c_1. Потім ми огрубіли оцінку до $an^2 + bn + c$. Далі будемо враховувати тільки порядок величин. Час роботи у гіршому випадку має порядок зростання n^2 (ми відкинули члени менших порядків і коефіцієнт при n^2). Це записують так: $T(n) = \Theta(n^2)$. В цьому випадку говорять, що часова складність алгоритму — порядку n^2.

Алгоритм з меншим порядком часової складності більш переважний: якщо, скажімо, один алгоритм має час роботи $\Theta(n^2)$, а інший — $\Theta(n^3)$, то перший більш ефективний, щонайменше, для більших n.

Аналіз сортування злиттям

Час роботи процедури злиття MERGE (A, p, q, r) є $\Theta(n)$, де n — загальна довжина ділянок, що зливаються $(n = r - p + 1)$. Це легко пояснити на картах. Нехай ми маємо дві купки карт (сорочкою вниз), і у кожній купці карти йдуть зверху вниз у зростаючому порядку. Як зробити з них одну купку? На кожному кроці ми беремо меншу з двох верхніх карт і кладемо її (сорочкою уверх) у результуючу купку. Коли одна з вихідних купок стає порожньою, ми додаємо всю решту карт другої купки до результуючої купки. Зрозуміло, що кожний крок вимагає обмеженого числа дій і загальне число дій є $\Theta(n)$.

Для спрощення будемо припускати, що розмір n вхідного масиву є степінь двійки (це обмеження не дуже істотне). Тоді на кожному кроці ділянка, що сортується, ділиться на дві рівні половини (див. рис. 8.2). Розбиття на частини (обчислення границі) вимагає часу $\Theta(1)$, а злиття — часу $\Theta(n)$. Одержуємо співвідношення:

$$T(n) = \begin{cases} \Theta(1), & \text{якщо } n = 1, \\ 2T(n/2) + \Theta(n), & \text{якщо } n > 1. \end{cases}$$

Це — рекурентне співвідношення. Можна довести, що з нього виходить оцінка

$$T(n) = \Theta(n \log n),$$

де через log ми позначаємо двійковий логарифм (основа логарифмів не відіграє ролі, оскільки призводить лише до зміни константи). Тому для більших n сортування злиттям більш ефективне, ніж сортування вставками з часовою складністю $\Theta(n^2)$.

Запитання

1. Чим відрізняється складність алгоритму від складності опису алгоритму?
2. Що називається розміром задачі?
3. Що називається часовою і місткісною складністю алгоритму?
4. Від чого залежить час роботи алгоритму сортування вставками? В якому випадку він буде максимальним, а в якому — мінімальним?
5. Який порядок складності процедури сортування вставками? Проведіть обчислення.
6. Від чого залежить час роботи алгоритму сортування злиттям?
7. Який порядок складності процедури сортування вставками? Проведіть обчислення.

Завдання

1. Будемо сортувати масив з n елементів так: переглянемо його і знайдемо мінімальний елемент, який скопіюємо у першу комірку іншого масиву. Потім переглянемо масив знову і знайдемо наступний елемент і т. д. Такий спосіб сортування можна назвати сортуванням вибором. Запишіть цей алгоритм на псевдокоді. Вкажіть час його роботи у кращому і гіршому випадках, використовуючи Θ-позначення.

2*. Дана послідовність чисел x_1, x_2, ..., x_n. Покажіть, що за час $\Theta(n \log n)$ можна визначити, чи є в цій послідовності два однакових числа.

3. Як записати вираз $n^3/1000 - 100n^2 - 100n + 3$ за допомогою Θ-позначень?

4. Майже будь-який алгоритм можна трохи змінити, радикально зменшивши час його роботи у кращому випадку. Як?

5*. Доведіть за індукцією, що якщо

$$T(n) = \begin{cases} \Theta(1), & \text{при } n = 1, \\ 2T(n/2) + \Theta(n), & \text{при } n = 2^k \text{ и } k > 1, \end{cases}$$

то $T(n) = \Theta(n \log n)$.

6. Сортування вставками можна оформити як рекурсивну процедуру: бажаючи відсортувати $A[1..n]$, ми (рекурсивно) сортуємо $A[1..n-1]$, а потім ставимо $A[n]$ на правильне місце у відсортованому масиві $A[1..n-1]$. Напишіть рекурентне співвідношення для часу роботи такої процедури.

7. При пошуку у відсортованому масиві ми можемо спочатку порівняти шуканий елемент із середнім елементом масиву і довідатися, в якій половині його слід шукати, а потім застосувати ту ж ідею рекурсивно. Такий спосіб називається двійковим пошуком. Напишіть відповідну програму, використовуючи цикл або рекурсію. Поясніть, чому час її роботи є $\Theta(\log n)$?

8. Зауважимо, що цикл while у рядках 5–7 процедури Insertion-Sort переглядає елементи відсортованої ділянки $A[1..j-1]$ поспіль. Замість цього можна було б використовувати двійковий пошук (завдання 7), щоб знайти місце вставки за час $\Theta(\log n)$. Чи вдасться таким чином зробити загальний час роботи рівним $\Theta(n \log n)$?

9. Сортування вставками для малих кусків.
Сортування злиттям більш швидке, ніж сортування вставками, але для маленьких n співвідношення зворотне. Тому має сенс достатньо короткі куски не розбивати далі, а застосувати до них сортування вставками. Питання в тому, де слід провести границю.

9.1. Нехай масив розміру n розбитий на k частин розмір n/k. Покажіть, що можна відсортувати всі частини окремо за допомогою сортування вставками за час $\Theta(nk)$.

9.2. Покажіть, що після цього можна злити всі частини в один упорядкований масив за час $\Theta(n \log(n/k))$.

9.3. Загальний час роботи такого змішаного алгоритму є $\Theta(nk + n \log(n/k))$. Яка максимальна швидкість зростання k як функції від n, при якій цей час так і є $\Theta(n \log n)$?

9.4. Як би ви стали обирати оптимальне значення k на практиці?

10. Число інверсій.

Нехай $A[1..n]$ — масив з n різних чисел. Нас буде цікавити число інверсій у цьому масиві, тобто число пар $i < j$, для яких $A[i] > A[j]$.

10.1. Вкажіть п'ять інверсій у масиві (2, 3, 8, 6, 1).

10.2. Яке максимально можливе число інверсій у масиві довжини n?

10.3. Як зв'язаний час роботи алгоритму сортування вставками і число інверсій? Поясніть свою відповідь.

10.4. Побудуйте алгоритм, який рахує число інверсій у масиві довжини n за час $\Theta(n \log n)$. Вказівка: модифікуйте алгоритм сортування злиттям.

8.7. Використання швидких алгоритмів

Дослідження залежності можливого розміру задачі від складності алгоритму і швидкості роботи комп'ютера

Часто різниця між поганим і добрим алгоритмом більш істотна, ніж між швидким і повільним комп'ютером. Нехай ми хочемо відсортувати масив з мільйона чисел. Що швидше — сортувати його вставками на швидкому комп'ютері, який виконує 100 мільйонів операцій за секунду, або злиттям на повільному комп'ютері, який виконує 1 мільйон операцій за секунду? Нехай до того ж сортування вставками написане надзвичайно економно і для сортування n чисел треба лише $2n^2$ операцій. У той же час алгоритм злиття написаний без особливої турботи про ефективність і вимагає $50n \cdot \log n$ операцій. Для сортування вставками мільйона чисел одержуємо

$$\frac{2(10^6)^2 \text{ операцій}}{10^8 \text{ операцій за секунду}} = 20000\,\text{с} \approx 5{,}56\,\text{г}$$

для суперкомп'ютера і всього

$$\frac{50(10^6)\log(10^6) \text{ операцій}}{10^6 \text{ операцій за секунду}} \approx 1000\,\text{с} \approx 17\,\text{хв}$$

для іншого (повільного) комп'ютера. Перевага більш ефективного алгоритму очевидна.

Розглянемо ще один приклад. Припустимо, у нас є п'ять алгоритмів A_1, ..., A_5 з такими часовими складностями:

Алгоритм	Часова складність — $T(n)$
A_1	n
A_2	$n \log n$
A_2	n^2
A_4	n^3
A_5	2^n

В таблиці 8.1 наведено розміри задач, які можна розв'язати за одну секунду, одну хвилину, одну годину кожним з цих пяти алгоритмів.

Таблиця 8.1. Залежність можливих розмірів задачі від складності алгоритму

Алгоритм	Часова складність	Максимальний розмір задачі (кількість умовних одиниць інформації), яку можна розв'язати за		
		1 с	1 хв	1 г
A_1	n	1000	6×10^4	$3{,}6 \times 10^6$
A_2	$n \log_2 n$	140	4893	2×10^5
A_3	n^2	31	244	1897
A_4	n^3	10	39	153
A_5	2^n	9	15	21

Нехай одиниця інформації перероблюється за одну мілісекунду. Тоді алгоритм A_1 може обробити за одну секунду вхід розміру 1000 одиниць інформації, в той час, як A_5 — вхід розміру не більше 9 одиниць інформації.

Розглянемо, як вплине збільшення швидкості комп'ютера в 10 разів на можливості використання алгоритмів A_1 ... A_5 для розв'язку задач з часовою складністю порядку n^2. Нехай s_1 — розмір деякої задачі 1. Нехай час розв'язку її на першому комп'ютері — T_1. На іншому комп'ютері ця задача розв'язується у 10 разів швидше, тобто для другого комп'ютера правильно $T_1/10 = \Theta(s_1^2)$. Визначимо, якого розміру задачу може розв'язати другий комп'ютер за час T_1. Це означає, що необхідно знайти s_2 таке, що правильно $T_1 = \Theta(s_1^2)$. Виходячи з рівності $T_1/10 = \Theta(s_1^2)$, виконаємо перетворення:

$$T_1 = 10\ \Theta(s_1^2) = (3{,}16)^2\ \Theta(s_1^2) = \Theta((3{,}16 s_1)^2).$$

Одержуємо, що $s_2 = 3{,}16s_1$. Це означає, що для алгоритмів з часовою складністю порядку n^2 десятиразове збільшення швидкості обчислень збільшує розмір задачі, яку можна розв'язати, тільки в 3 рази.

В таблиці 8.2 вказано збільшення максимального розміру задачі, яку можливо розв'язати при збільшенні швидкості машини у 10 разів.

Таблиця 8.2. Залежність можливого розміру задачі від складності алгоритму

Алгоритм	Часова складність	Максимальний розмір задачі	
		до прискорення у 10 разів	після прискорення у 10 разів
A_1	n	s_1	$10s_1$
A_2	$n \log_2 n$	s_2	приблизно $10s_2$
A_3	n^2	s_3	$3{,}16s_3$
A_4	n^3	s_4	$2{,}15s_4$
A_5	2^n	s_5	$s_5 + 3{,}3$

Порівнюючи наведені таблиці, приходимо до висновку, що застосування більш діючого алгоритму для розв'язку задачі у багатьох випадках дає більший виграш у часі, ніж застосування більш швидкодіючої машини. Оскільки обчислювальні машини працюють все швидше і ми можемо розв'язувати всі великі задачі, саме складність алгоритму визначає те збільшення розміру задачі, якого можна досягнути із збільшенням швидкості машини. Ми бачимо, що розробка ефективних алгоритмів не менш важлива, ніж розробка швидкої електроніки.

Запитання

1. Що є більш важливим: ефективність алгоритму або швидкість роботи комп'ютера? Обґрунтуйте відповідь.
2. Побудуйте таблицю залежності максимально припустимого розміру задачі для розв'язання за 1 с і 1 хвилину для алгоритмів з часовими складностями порядку n, n^2, 2^n.
3. Наведіть розрахунок, як впливає збільшення швидкості роботи комп'ютера у 10 разів на припустимий розмір задачі для алгоритмів з часовими складностями порядку n, n^2, 2^n.

Завдання

1. Нехай сортування вставками і злиттям здійснюються на одній і тій же машині і вимагають $8n^2$ і $64n \log n$ операцій відповідно.

Для яких значень n сортування вставками є більш ефективним? Як можна покращити алгоритм сортування злиттям?

2. При якому найменшому значенні n алгоритм, що вимагає $100n^2$ операцій, ефективніше за алгоритм, що вимагає 2^n операцій?
3. Порівняння часу роботи. Нехай є алгоритм, що розв'язує задачу розміру n за $f(n)$ мікросекунд. Який максимальний розмір задачі, яку він зможе розв'язати за час t? Знайти його для функцій і часів, наведених у таблиці.

$f(n)$	1 с	1 хв	1 г	1 день	1 рік	1 ст
$\log n$						
$n \log n$						
n						
n^2						
n^3						
2^n						
$n!$						

розділ 9

Автомати

9.1. Загальна характеристика автоматів

Автомат, схема автомата, односторонній автомат, тільки читаючий автомат, читаючий та пишучий автомат, пам'ять автомата, такт роботи автомата, керуючий пристрій, початкова та заключна конфігурація, детермінований та недетермінований автомати

Обчислення за алгоритмом можна розглядати як деякий процес, який описується своєю множиною станів, початковим станом і правилами переходу із стану до стану. Ці переходи можуть виконуватися залежно від зовнішніх впливів (вхідних даних), під впливом деякого випадкового механізму або якось інакше, наприклад, за вибором деякої «живої істоти». Процес, у якому переходи виконуються під впливом зовнішніх дій, моделюється за допомогою автомата. Автомат — це схематизований алгоритм. Використовуючи автомати, можна моделювати багато машин, включаючи компоненти комп'ютера.

Коли визначено задачу, яку необхідно розв'язати, виникають три питання. Перший — чи можливе розв'язання цієї задачі. Другий — яким чином можна розв'язати цю задачу. Третій — яка складність розв'язку цієї задачі. Використовуючи такі обчислювальні моделі, як автомати, можна дослідити питання розв'язності і складності різних задач. Як і граматики, автомати також використовуються під час дослідження граматичної структури мов.

Автомат можна зобразити у вигляді схеми (рис. 9.1).

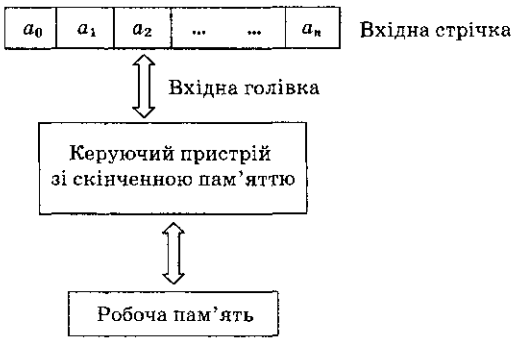


Рис. 9.1. Схема автомата

ОСНОВНІ ПОНЯТТЯ. Автомат складається з трьох частин — ***вхідної стрічки***, ***керуючого пристрою*** зі скінченною пам'яттю і ***допоміжної***, або ***робочої, пам'яті***.

Вхідну стрічку можна розглядати як лінійну послідовність кліток, або комірок, причому кожна комірка містить точно один вхідний символ з деякого скінченного ***вхідного алфавіту***. Ліву і праву комірки можуть займати особливі ***кінцеві маркери***, позначимо їх λ. Порожні комірки позначимо ε. За допомогою вхідної стрічки зображується інформація, що поступає на вхід автомата.

Вхідна голівка у кожний момент читає одну вхідну комірку. За один крок роботи автомата вхідна голівка може зміститися на одну комірку вліво, залишитися нерухомою або зміститися на одну комірку вправо. Автомат, який ніколи не переміщує свою вхідну голівку вліво, називається ***одностороннім***. Схему вхідної стрічки та вхідної голівки зображено на рис. 9.2.

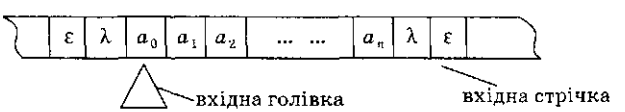


Рис. 9.2. Вхідна стрічка та вхідна голівка

Вхідна голівка може бути ***тільки читаючою***, якщо під час роботи автомат не утворює рядок — результат. Існують

такі автомати, вхідна голівка яких ***читаюча*** та ***пишуча***. Такі автомати можуть записувати результуючий рядок або змінювати символи на вхідній стрічці.

Пам'ять автомата — це структура, в якій записуються, зберігаються і зчитуються додаткові дані, що використовуються автоматом при роботі. Для кожного виду автоматів строго визначено тип пам'яті, функції доступу до пам'яті та перетворення пам'яті. Автомат може не мати пам'яті. Тип пам'яті часто визначає назву автомата.

Робота автомата складається з послідовності ***тактів***. Кожний такт складається з таких дій:

1. Читається поточний вхідний символ.
2. За допомогою функції доступу досліджується пам'ять, і до неї заноситься деяка інформація.
3. Змінюється стан керуючого пристрою.
4. Записується вихідна інформація у комірки вхідної стрічки.
5. Вхідна голівка зміщується на одну комірку вліво, вправо або зберігається у початковому стані.

Поточний вхідний символ та інформація, що витягнута з пам'яті, разом з поточним станом керуючого пристрою визначають, яким повинен бути цей такт.

Таким чином, на кожному такті визначається поточна ***конфігурація*** автомата, до якої входять:

— поточний стан керуючого пристрою;
— поточний зміст вхідної стрічки;
— поточний зміст робочої пам'яті.

Конфігурація автомата змінюється на кожному такті під впливом керуючого пристрою.

Керуючий пристрій складається зі скінченної множини станів $S = \{s_0 \dots s_n\}$ і відображень, які залежно від попередньої конфігурації дозволяють визначити нову конфігурацію автомата, напрямок переміщення вхідної голівки (якщо вона не одностороння), інформацію на друку (якщо в автоматі передбачено функцію друку), інформацію, що заносять у пам'ять і витягують з пам'яті (якщо автомат має робочу пам'ять).

Автомат починає роботу з ***початкової*** конфігурації. При початковій конфігурації керуючий пристрій знаходиться у заданому початковому стані, вхідна голівка оглядає лівий символ на вхідній стрічці, і пам'ять має заздалегідь встановлений

початковий зміст. Конфігурація, при якій автомат припиняє роботу, називається ***заключною***. Автомат припиняє роботу, перейшовши в один із станів заздалегідь виділеної множини заключних станів, або тоді, коли перегляд вхідних даних завершений. Умови припинення роботи чітко визначаються для кожного конкретного автомата.

Керуючий пристрій є ***недетермінованим***, якщо для кожної конфігурації існує скінченна множина можливих конфігурацій (більше однієї), так що в будь-яку з них автомат може перейти на наступному кроці. Керуючий пристрій називається ***детермінованим***, якщо для кожної конфігурації існує не більше однієї можливої наступної конфігурації. Відповідно автомат називається детермінованим і недетермінованим залежно від виду керуючого пристрою.

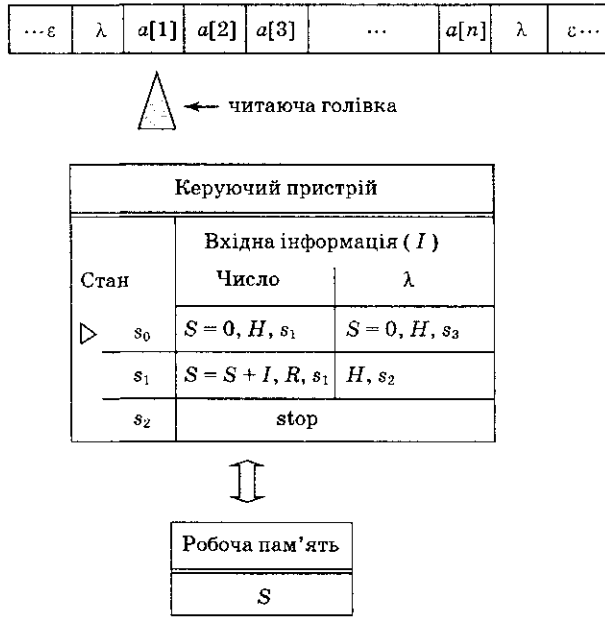
Недетермінований автомат розглядається як множина паралельно працюючих детермінованих його екземплярів. Наприклад, якщо для даної конфігурації існують три можливих наступних конфігурації, то розглядається три паралельно працюючих детермінованих автомата. Недетермінізм скінченного автомата не слід змішувати з «випадковістю», при якій автомат може випадково обрати один із станів з фіксованими ймовірностями, але сам автомат завжди є тільки в одному екземплярі.

Запитання

1. Яким чином зв'язані поняття автомату і алгоритму?
2. Для дослідження яких питань використовуються автомати?
3. Зобразіть схему автомата і назвіть складові частини автомата.
4. Поясніть принцип дії і побудову кожної частини автомата.
5. Що включає такт роботи автомата?
6. Визначте поняття конфігурації автомата. Що таке початкова та кінцева конфігурації автомата?
7. Чим відрізняються детермінований і недетермінований скінченні автомати?

Завдання

1. Визначте такти роботи автомата, що розв'язує задачу підрахунку суми елементів масиву, схема автомата зображена на рисунку (припускаємо, що додавання є елементарною операцією для цього автомата).



В таблиці керуючого пристрою вказано дії, здійсненні автоматом, і стан, в який автомат переходить залежно від поточного стану і значення вхідної інформації.

Позначення:

▷ —покажчик поточного стану;

H — читаюча голівка залишається на місці;

R — читаюча голівка зміщується вправо;

I — інформація, що прочитана на даному кроці.

2. Складіть схеми автоматів за прикладом схеми у завданні 1, що виконують такі дії:
   а) визначення довжини вхідного рядка;
   б) підрахунок кількості нулів у вхідному рядку.

9.2. Розпізнавачі

Розпізнавач, припустимий рядок, визначена розпізнавачем мова

Розглядаючи граматики, ми визначили два способи задання мов: розпізнаванням рядків та породженням рядків. Для граматик в основному застосовується другий спосіб. Автомати, що використовуються для визначення мови, частіше є тими,

що розпізнають, їх називають ***розпізнавачами***. За допомогою розпізнавачів, також як і за допомогою граматик, можна однозначно визначити мову.

Звичайно припускається, що вхідна голівка розпізнавача тільки читає.

Розглянемо деякий рядок ω. Розпізнавач ***допускає*** вхідний рядок ω, якщо, починаючи з початкової конфігурації, в якій рядок ω записаний на вхідній стрічці, розпізнавач може зробити послідовність кроків, що закінчується завершальною конфігурацією. Слід вказати, що, починаючи з даної початкової конфігурації недетермінований розпізнавач може виконати багато різних послідовностей кроків. Якщо хоча б одна з цих послідовностей закінчується завершальною конфігурацією, то початковий вхідний ланцюжок буде допущений.

Мова, що визначена розпізнавачем, зображує множину вхідних ланцюжків, які вона допускає.

Для кожного класу граматик з ієрархії Хомського існує клас розпізнавачів, що визначає той же клас мов. Цими розпізнавачами є скінченні автомати, автомати з магазинною пам'яттю, лінійно обмежені автомати та машини Тьюринга. Точніше мови з ієрархії Хомського можна охарактеризувати так:

1. Мова регулярна (праволінійна або ліволінійна) тоді і тільки тоді, коли вона визначається скінченним автоматом.
2. Мова контекстно-вільна тоді і тільки тоді, коли вона визначається автоматом з магазинною пам'яттю.
3. Мова контекстно-залежна тоді і тільки тоді, коли вона визначається лінійно обмеженим автоматом.
4. Мова визначається граматикою загального виду тоді і тільки тоді, коли вона визначається машиною Тьюринга.

Всі наведені чотири типи автоматів описуються у наступних розділах.

Запитання

1. Яку задачу розв'язують автомати, що називаються розпізнавачами?
2. Яким чином визначається, що автомат допускає вхідний рядок?
3. Що таке мова, визначена розпізнавачем?
4. Як зв'язані різні види розпізнавачів та граматик?

9.3. Скінченні автомати

Скінченний автомат, автомат з виходом і без виходу, завдання автомата таблицею і діаграмою станів, автомати Мілі і Мура, відповідність скінченного розпізнавача регулярній граматиці і регулярному виразу

Скінченний автомат складається тільки з вхідної стрічки та керуючого пристрою із скінченною пам'яттю (рис. 9.3). Число його станів скінченне, що і визначає його назву. Керуючий пристрій може бути детермінованим або недетермінованим, але вхідна голівка — одностороння — що зміщується вправо.

Рис. 9.3. Схема скінченного автомата

Ми визначимо скінченний автомат, задавши скінченну множину його станів, допустимі вхідні символи, допустимі вихідні символи, початковий стан і множину завершальних станів. Задаються також функція переходів станів, яка за даним поточним станом і поточним вхідним символом вказує всі можливі наступні стани, і функція друку.

Визначення

Скінченний автомат — це структура виду

$$M = (S, I, O, f, g, s_0, F),$$

де S — скінченна множина ***станів***;
I — скінченна множина допустимих ***вхідних символів***;
O — скінченна множина допустимих ***вихідних символів***;

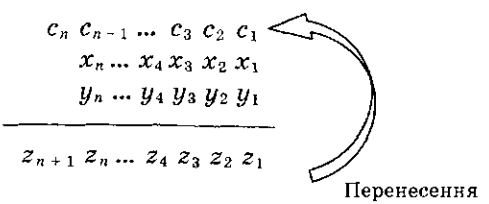
f — відображення множини $S \times I$ у множину S, що визначає вибір наступного стану (функцію f називають ***функцією переходів***);
g — відображення множини $S \times I$ у множину O, що визначає вихідний символ (функцію g називають ***функцією виходів***);
$s_0 \in S$ — ***початковий стан*** керуючого пристрою;
$F \subseteq S$ — множина ***завершальних станів***.

Автомат може не мати функції g і множини O, якщо не передбачено видачі вихідної інформації. В цьому випадку вважається, що автомат ***без виходу***. Якщо автомат містить функцію g, тоді вважається, що він з ***виходом***. Множина F може бути не визначеною.

Скінченний автомат можна задати за допомогою ***таблиці***. В цій таблиці для кожної комбінації станів автомата і вхідного символу вказується наступний стан і вихідний символ. Крім того, скінченний автомат можна задати за допомогою орієнтованого графа, що називається ***діаграмою станів***. У цьому графі вершини зображують стани автомата, дуги — переходи з одного стану в інший. Кожна дуга позначена вхідним і вихідним символом, що відповідає цьому переходу.

Приклад. Побудувати скінченний автомат, здійснюючий додавання двох натуральних чисел, використовуючи їх бінарне зображення.

Пояснимо процедуру порозрядного бінарного додавання. Нехай є два числа, записані за допомогою n-розрядного двійкового коду: $x_n \ldots x_2 x_1$ і $y_n \ldots y_2 y_1$. Спочатку додаються x_1 і y_1, у результаті чого виходить результат z_1 і перенесення c_1. Після цього робиться додавання x_2, y_2 і c_1, у результаті виходить z_2 і перенесення c_2. Ця процедура продовжується до кроку n, на якому при додаванні x_n, y_n і c_{n-1} виходить z_n і c_n. Остання сума z_{n+1} дорівнює перенесенню c_n.



Скінченний автомат, що виконує це додавання, можна побудувати, використовуючи тільки два стани.

Розглянутий автомат $M = (S, I, O, f, g, s_0)$ має множину станів $S = \{s_0, s_1\}$, множину вхідних символів $I = \{00, 01, 10, 11\}$, множину вихідних символів $O = \{0, 1\}$, початковий стан — $s_0 \in S$. Будемо вважати, що автомат припиняє роботу, коли закінчилося надходження вхідних даних, це означає, що додавання завершено.

Вхідними даними на одному такті є два біти — відповідні біти доданків, тому можливі варіанти вхідної інформації для одного такту — 00, 01, 10, 11. Переходи побудовані відповідно до одержуваної суми, яка зображена вихідним бітом, і перенесенням, яке зображене станом (s_0 — перенесення дорівнює 0, s_1 — перенесення дорівнює 1).

Відображення f: $S \times I \to S$ (функція переходів) і g: $S \times I \to O$ (функція виходів) зображені у таблиці 9.1. В таблиці зображено, що якщо біти, які додаються, — 00 і поточний стан s_0 (перенесення немає), то автомат залишається у стані s_0 (перенесення немає) і видає результат 0. Якщо біти, що додаються, — 00 і поточний стан s_1 (перенесення — 1), то автомат залишається у стані s_0 (перенесення немає) і видає результат 1. Якщо, наприклад, біти, що додаються, — 10 і поточний стан s_1 (перенесення — 1), то автомат залишається у стані s_1 (перенесення — 1) і видає результат 0.

Таблиця 9.1. Функції переходів f і виходів g скінченного автомата для додавання

Стани S	f					g			
	I:	00	01	10	11	00	01	10	11
s_0		s_0	s_0	s_0	s_1	0	1	1	0
s_1		s_0	s_1	s_1	s_1	1	0	0	1

Діаграму станів, що відповідає цьому автомату, представлено на рис. 9.4.

Біля кожної стрілки переходу вказано вхідну інформацію і вихідний сигнал. Наприклад, запис 01.1 біля петлі у стані s_0 означає, що якщо на вхід автомата із станом s_0 поступає пара бітів 01, то на вихід необхідно видати 1 і як наступний стан зберегти s_0. Розглянемо роботу автомата. Якщо, наприклад, поточний стан — s_1 і на вхід поступає 01, наступний стан

буде s_1, а вихід — 0. Відбувається додавання $0 + 1 + 1 = (10)_2$. Тому вихідний сигнал — 0, а перенесення 1 визначає наступний стан s_1.

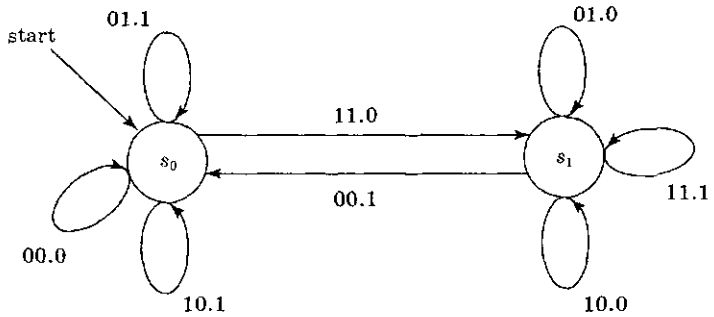


Рис. 9.4. Скінченний автомат для додавання

В розглянутому нами автоматі вихідна інформація відповідає переходам, однак існують автомати, в яких вихідна інформація залежить тільки від стану, в який переходить автомат. Залежно від цього розрізняють два види скінченних автоматів.

Визначення

Скінченний автомат, у якому вихідна інформація поставлена у відповідність переходам, називається ***автоматом Мілі***. Скінченний автомат, у якому вихідна інформація поставлена у відповідність станам, називається ***автоматом Мура***.

Визначення означає, що в автоматах Мілі функція виходів — це залежність вихідної інформації від поточного стану і поточної вхідної інформації, а в автоматах Мура функція виходів — це залежність вихідної інформації тільки від поточного стану.

Скінченний автомат для додавання з нашого прикладу є автоматом Мілі. Розглянемо приклад автомата Мура.

Приклад. Побудуємо автомат Мура, який на виході друкує «1», якщо кількість одиниць, що прочитані автоматом у вхідному рядку, ділиться на 3, і друкує «0» у протилежному випадку (автомат також друкує 0, якщо не прочитано жодної

одиниці у початковому стані s_0). Вхідний алфавіт автомата $I = \{0, 1\}$. Назвемо цей автомат A_3.

Визначимо стани автомата: s_0 — початковий стан, s_1 — прочитано число одиниць, що дає при діленні на 3 у залишку 1, s_2 — прочитано число одиниць, що дає при діленні на 3 у залишку 2, s_3 — прочитано число одиниць, кратне 3. Звичайно, що вихідну інформацію в такому автоматі можна поставити у відповідність станам. Побудуємо таблицю переходів і граф автомата (таблиця 9.2, рис. 9.5).

Таблиця 9.2. Функції переходів f і виходів g автомата Мура A_3

S \ I	f		g
	0	1	
s_0	s_0	s_1	0
s_1	s_1	s_2	0
s_2	s_2	s_3	0
s_3	s_3	s_1	1

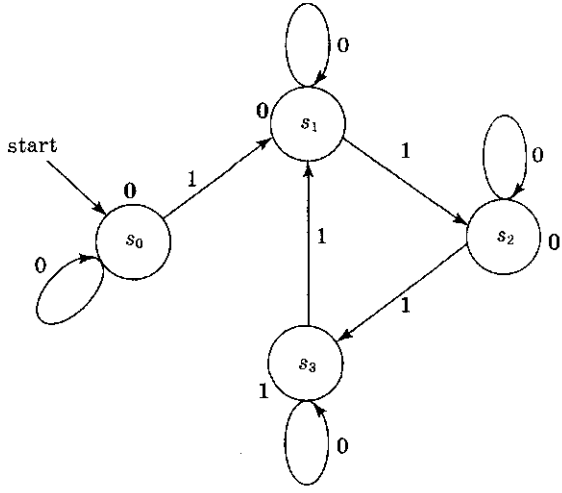


Рис. 9.5. Автомат Мура A_3

Припустимо, на вхід автомата поступив рядок «0001010100». Розглянемо, яку вихідну інформацію видасть автомат. Побудуємо відповідну таблицю (таблиця 9.3).

Таблиця 9.3. Обробка вхідної інформації «0001010100» автоматом A_3

Прочитаний рядок	f	g
0	s_0	0
00	s_0	0
000	s_0	0
0001	s_1	0
00010	s_1	0
000101	s_2	0
0001010	s_2	0
00010101	s_3	1
000101010	s_3	1
0001010100	s_3	1

Після прочитання рядка «0001010100» автомат друкує «1», тобто кількість одиниць, що прочитав автомат у вхідному рядку, ділиться на 3.

За допомогою скінченних автоматів можна розпізнавати рядки мов. Автомат A_3, що побудований у попередньому прикладі, можна розглядати як розпізнавач мови, що складається з рядків бітів, які містять число одиниць, кратне 3. Назвемо цю мову L_3. Мові L_3 належать, наприклад, рядки «111», «01000110001010100», «01010101010101010» тощо. Для розпізнання мови звичайно не потрібна наявність у автомата вихідних сигналів, але визначається множина кінцевих станів автомата. Якщо прочитання вхідного рядка автомат завершив, знаходячись у кінцевому стані, то рядок вважається розпізнаним. У нашому прикладі множина кінцевих станів складається з єдиного стану s_3.

Твердження. Мова є ***регулярною*** тоді і тільки тоді, коли вона розпізнається за допомогою скінченного автомата.

Оскільки існує скінченний автомат, що розпізнає мову L_3, то отже, L_3 — регулярна мова, та її можна задати за допомогою регулярного виразу і регулярної граматики. Складемо відповідний регулярний вираз. У стані s_0 автомат знаходиться, пропускаючи будь-яку кількість нулів, на початку вхідного

рядка, це відповідає регулярному виразу 0^*. Стан s_1 допускає одну одиницю (при переході із s_0 до s_1), потім може йти будь-яка кількість нулів, автомат залишається у стані s_1. Таким чином, стан s_1 відповідає приєднанню виразу 10^*. Аналогічно переходу від s_1 до стану s_2, від s_2 — до s_3 відповідає дворазове приєднання рядка виду 10^*. Стан s_3 є кінцевим. Складемо вираз за допомогою операції конкатенації: $0^*10^*10^*10^*$. Оскільки із стану s_3 автомат може знову перейти до стану s_1, прочитавши одиницю, і потім також послідовно пройти наступні стани скільки завгодно разів від s_1 до s_3, то у формі регулярного виразу це виглядає так: $(10^*10^*10^*)^*$. Таким чином, регулярний вираз для мови L_3 має вигляд:

$$L_3: 0^*10^*10^*10^*(10^*10^*10^*)^*.$$

Можна побудувати граматику для мови L_3. Це є граматика

$$G_3 = (\{A, B, C, S\}, \{0, 1\}, P, S),$$

де P складається з набору продукцій:

1. $S \to AB$.
2. $A \to C1C1C1C$.
3. $B \to 1C1C1CB \mid \varepsilon$.
4. $C \to 0C \mid \varepsilon$.

Дійсно, продукція 4 відповідає виразу 0^*, тоді продукція 3 відповідає виразу $(10^*10^*10^*)^*$, а продукція 2 — виразу $0^*10^*10^*10^*$.

Запитання

1. Зобразіть схему скінченного автомата та опишіть його пристрій.
2. Дайте визначення скінченного автомата, його складових.
3. Опишіть два способи завдання скінченного автомата.
4. Чим відрізняються скінченні автомати з виходом і без виходу.
5. Дайте визначення автоматів Мілі і Мура.
6. Яким чином функціонують скінченні розпізнавачі?
7. Який вид мов визначається скінченними автоматами? За допомогою яких формальних систем, крім скінченних автоматів, можна задати цей вид мов?

Завдання

1. Побудуйте діаграми для скінченних автоматів, що зображені такими таблицями:

а)

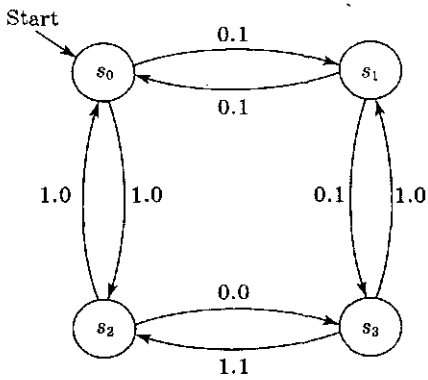
S	f	g
	0 1	0 1
s_0	s_1 s_0	0 1
s_1	s_0 s_2	0 1
s_2	s_1 s_1	0 0

б)

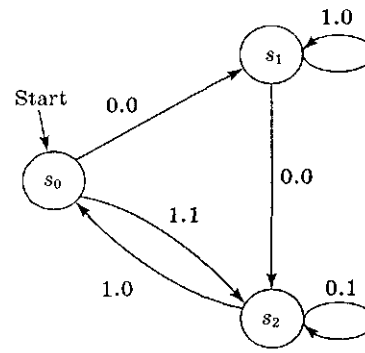
S	f	g
	0 1	0 1
s_0	s_1 s_0	0 0
s_1	s_0 s_2	1 1
s_2	s_1 s_1	0 1
s_3	s_1 s_2	1 0

2. Побудуйте таблиці станів для автоматів, що задані діаграмами:

а)



б)



3. Побудуйте скінченний автомат, який змінює кожний парний біт у вихідному рядку, починаючи з другого, а непарні біти, починаючи з першого, залишає незмінними.
4. Побудуйте скінченний автомат, що моделює роботу замка для сейфа, що містить числа від 1 до 40. Відчиняється комбінацією: поворот праворуч на 10, ліворуч на 8, праворуч на 37. Вхідні дані складаються з пар: напрямок повороту (L, R), число від 1 до 40. Якщо вважати цей автомат розпізнавачем, то яку мову він визначає?
5. Побудуйте скінченний автомат, який визначає, чи є останніми символами вхідних даних слово «автомат». Вхідні дані — рядок з букв російського алфавіту. Побудуйте відповідний регулярний вираз.
6. Дослідіть, яким чином можна здійснити побудову скінченного автомата, що відповідає заданому регулярному виразу, і навпаки.

9.4. Автомати з магазинною пам'яттю

Автомат з магазинною пам'яттю, розпізнавання рядків автоматом, мова, що визначається автоматом

Автомат з магазинною пам'яттю (МП-автомат) — це скінченний автомат, що має пристрій пам'яті типу стек. У стеці елементи інформації заносяться до самої верхньої комірки пам'яті, зміщуючи тим самим весь її зміст на одну комірку вниз, і дістаються тільки із самої верхньої комірки пам'яті, зміщуючи тим самим весь її зміст на одну комірку вверх. При цьому у кожний момент часу доступний тільки самий верхній елемент пам'яті. Теоретично, об'єм пам'яті МП-автомата не обмежений. Схему МП-автомата зображено на рисунку 9.6.

Рис. 9.6. Автомат з магазинною пам'яттю

Визначення

Автомат з магазинною пам'яттю — це структура

$$M = (S, I, O, Z, f, g, s_0, z_0, F),$$

де S — скінченна множина ***станів***;

I — скінченна множина допустимих ***вхідних символів***;

O — скінченна множина допустимих ***вихідних символів***;
Z — скінченна множина допустимих ***символів пам'яті***;
f — ***функція переходів*** — відображення множини $S \times I \times Z$ у множину $S \times Z$, що визначає вибір наступного стану і символ, який заноситься до пам'яті;
g — ***функція виходів*** — відображення множини $S \times I \times Z$ у множину O, що визначає вихідний символ;
$s_0 \in S$ — ***початковий стан*** керуючого пристрою;
$z_0 \in Z$ — ***початковий символ*** — символ, що знаходиться у пам'яті (стек) у початковий момент;
$F \subseteq S$ — множина ***кінцевих станів***.

Приклад. Роботу автомата з магазинною пам'яттю розглянемо на прикладі МП-розпізнавача, що допускає множину $\{0^n1^n \mid n = 0, 1, 2, \ldots\}$. Запис 0^n1^n означає множину рядків, що складаються з рівної кількості нулів та одиниць, наприклад, «000111», «01», «0000011111». В п. 7.5 цю мову було позначено через $L_{пп}$. Вона є контекстно-вільною, але не є регулярною.

Розглянутий автомат $M = (S, I, Z, f, s_0, z_0, F)$ має множину станів $S = \{s_0, s_1, s_2\}$, множину вхідних символів $I = \{0, 1\}$, множину символів пам'яті $Z = \{0, 1\}$, початковий стан s_0, кінцевий стан s_0, спочатку у пам'ять записаний символ $x = z_0$. Будемо вважати, що автомат допускає вхідний рядок $a_1, a_2, \ldots, a_n \in I^*$, якщо після прочитання цього рядка (тобто послідовного огляду голівкою всіх символів a_i рядка), автомат переходить у кінцевий стан.

Функція переходів f задається такими значеннями:

$$f(s_0, 0, x) = \{(s_1, 0x)\};$$
$$f(s_1, 0, 0) = \{(s_1, 00)\};$$
$$f(s_1, 1, 0) = \{(s_2, \varepsilon)\};$$
$$f(s_2, 1, 0) = \{(s_2, \varepsilon)\};$$
$$f(s_2, \varepsilon, x) = \{(s_0, \varepsilon)\}.$$

Рядок ε — порожній рядок. Перехід $f(s_0, 0, x) = \{(s_1, 0x)\}$ означає, що якщо автомат знаходиться у стані s_0, одержує на вхід 0 і дістає з пам'яті поточний символ x, то він повинен перейти у стан s_1, повернути у пам'ять символ x і помістити у стек символ 0. Перехід $f(s_2, 1, 0) = \{(s_2, \varepsilon)\}$ означає, що якщо

автомат знаходиться у стані s_2, одержує на вхід символ 1 і дістає з пам'яті поточний символ 0, то він повинен залишитися у стані s_2 і в пам'ять нічого не додавати. Таким чином, виходить, що на цьому такті із стека пам'яті «виштовхнутий» і загублений символ 0. Кінцева конфігурація автомата — (s_0, ε). При роботі автомат зчитує у пам'ять першу половину ланцюжка, що складається з нулів, а потім видаляє з пам'яті по одному нулю на кожну одиницю, що поступає на вхід. Крім того, переходи станів гарантують, що всі нулі передують одиницям. Наприклад, для вхідного ланцюжка 0011 автомат здійснить таку послідовність тактів:

$$(s_0, 0011, x) \to (s_1, 011, 0x) \to (s_1, 11, 00x) \to$$
$$\to (s_2, 1, 0x) \to (s_2, \varepsilon, x) \to (s_0, \varepsilon).$$

Символ «→« означає перехід від такту до такту. Для кожного такту записана конфігурація, яка включає поточний стан, частину вхідного ланцюжка, що залишилася непрочитаною, зміст пам'яті. Наприклад, для конфігурації $(s_1, 011, 0x)$ поточний стан — s_1, непрочитана частина вхідного ланцюжка — 011, поточний вхідний символ (розміщений ліворуч) — 0. Зміст пам'яті — $0x$, символ, що дістається (розміщений ліворуч) — 0.

За допомогою автоматів з магазинною пам'яттю можна розпізнавати рядки мови двома способами. Перший — рядок вважається розпізнаним, якщо після його прочитання автомат переходить у завершальний стан. Другий — якщо після прочитання рядка стек пам'яті виявляється порожнім. У розглянутому нами прикладі вимагалося виконання обох умов.

Твердження. Мова є ***контекстно-вільною*** тоді і тільки тоді, коли вона розпізнається автоматом з магазинною пам'яттю.

Запитання

1. Зобразіть схему автомата з магазинною пам'яттю. Поясніть пристрій блоку пам'яті МП-автомата.
2. Дайте визначення автомата з магазинною пам'яттю.
3. Якими способами можна розпізнавати рядки за допомогою МП-автомата?
4. Який вид мов можна розпізнавати за допомогою МП-автомата?

Завдання

1. Переконайтеся, що для мови $L_{пл}$: 0^n1^n , $n \geq 0$ неможливо побудувати скінченний розпізнавач. Побудуйте скінченний розпізнавач для цієї мови при обмеженні $0 \leq n \leq 10$.

2. Зробіть розпізнавання рядків «000011» «111000» за допомогою МП-розпізнавача для мови L_{nn} (цей розпізнавач розглядався як приклад у цій главі). Переконайтеся, що ці рядки не можуть бути розпізнані.
3. Побудуйте автомат з магазинною пам'яттю, який розпізнає мову, що складається з рядків $\omega\omega^r$, де ω визначається регулярним виразом $(a \cup b)(a \cup b)^*$.
4. Визначте, яку мову задають граматики. Побудуйте автомат з магазинною пам'яттю, який розпізнає цю мову.

   а)

   $G = (\{A, S\}, \{a, b, d\}, P, S)$,

   P:

   1. $S \to Ad$
   2. $A \to aAbb \mid abb$.

   б)

   $G = (\{A, S\}, \{a, b, d\}, P, S)$,

   P:

   1. $S \to SS \mid A$
   2. $A \to 0A1 \mid 01$.

5. Напишіть алгоритм, що моделює роботу МП-автомата.

9.5. Машина Тьюринга. Лінійно-обмежені автомати

Машина Тьюринга, тезис Черча — Тьюринга, розпізнавання рядків, мови, що розпізнаються, лінійно-обмежені автомати

Машина Тьюринга, названа на честь англійського математика Алана Тьюринга, є найбільш потужним автоматом із всіх розглянутих нами. Вона містить всі елементи скінченного автомата, крім того, її вхідна стрічка нескінченна в обох напрямках, голівка, що зчитує, є також і тією, що друкує, вона може переміщатися вздовж стрічки в обох напрямках.

Твердження. За допомогою машини Тьюринга можна моделювати будь-який обчислювальний процес, який можна назвати алгоритмом. Це твердження відоме як тезис Черча — Тьюринга. За допомогою машини Тьюринга можна розпізнавати будь-які мови, визначені граматиками загального виду.

Схему машини Тьюринга зображено на рисунку 9.7.

Рис. 9.7. Схема машини Тьюринга

Визначення

Машина Тьюринга — це структура

$$T = (S, I, f, s_0),$$

де S — скінченна множина ***станів***;
I — скінченна множина ***допустимих символів*** стрічки, включає порожній символ λ;
f — відображення множини $S \times I$ у множину $S \times I \times \times \{L, R\}$, що визначає для комбінації стану і вхідного символу вибір наступного стану, друкований символ, напрямок переміщення голівки — L (ліворуч), R (праворуч);
$s_0 \in S$ — ***початковий стан*** керуючого пристрою.

Оскільки стрічка нескінченна, то інформація, яка повинна бути прочитана, займає лише частину стрічки. Порожні комірки займає символ λ. В початковому стані голівка, що читає і пише, встановлюється на ліву з непорожніх комірок, і керуючий пристрій знаходиться у стані s_0. Машина працює так, що її дії визначаються функцією f. Машина зупиняється, якщо для поточної комбінації стану і вхідного символу функція f не визначена.

Може бути задана множина кінцевих станів. Кінцеві стани визначаються, в основному, коли машина моделює розпізнавання рядків.

Приклад. Побудуємо машину Тьюринга, що розпізнає рядки виду 0^n1^n, $n \geq 1$. Множина цих рядків утворює контекстно-вільну мову $L_{пп}$.

Покладемо $T = (S, I, f, s_0)$,

де

$S = \{s_0, ..., s_6\}$,

$I = \{0, 1, M\}$,

s_0 — початковий стан;

s_6 — кінцевий стан.

Функцію переходів f зображено у таблиці 9.4. В кожній комірці міститься номер наступного стану, друкований символ і напрямок переміщення голівки (L — ліворуч, R — праворуч).

Таблиця 9.4. Функція переходів f

	Вхідна інформація			
	0	1	M	λ
s_0	s_1 M R	-	-	-
s_1	s_1 0 R	s_1 1 R	s_2 M L	s_2 λ L
s_2	-	s_3 M L	-	-
s_3	s_4 0 L	s_3 1 L	s_5 M R	-
s_4	s_4 0 L	-	s_0 M R	-
s_5	-	-	s_6 M R	-
s_6	-	-	-	-

Вхідний рядок вважається розпізнаним, якщо після його прочитання машина переходить до кінцевого стану s_6. При розпізнаванні використовується допоміжний символ M, який записується замість вже прочитаних символів рядка.

Розглянемо послідовність кроків даної машини Тьюринга при розпізнаванні рядка «000111» = $0^3 1^3$.

На першому кроці машина T зчитує перший нуль, замінює його на M — «M00111», переходить у стан s_1, переміщується праворуч, поточним стає другий 0. На другому кроці машина зчитує поточний нуль, не змінює його, залишається у стані s_1, переміщується праворуч, поточним стає третій 0, і т. д. Зобразимо ці дії схематично таким чином:

(s_0) ... $\lambda\underline{0}00111\lambda$ ... $\rightarrow (s_1)$ ... $\lambda M\underline{0}0111\lambda$ ... $\rightarrow$

$\rightarrow (s_1)$ ... $\lambda M0\underline{0}111\lambda$ ... $\rightarrow (s_1)$ ... $\lambda M00\underline{1}11\lambda$ ...$\rightarrow$

$\rightarrow (s_1)$ ... $\lambda M001\underline{1}1\lambda$ ... $\rightarrow (s_1)$ ... $\lambda M0011\underline{1}\lambda$ ... $\rightarrow$

$\rightarrow (s_1)$ ... $\lambda M00111\underline{\lambda}$ ... $\rightarrow (s_1)$ ... $\lambda M00111\underline{\lambda}$ ... $\rightarrow$

$\rightarrow (s_2)$ ... $\lambda M0011\underline{1}\lambda$ ... $\rightarrow (s_3)$ ... $\lambda M001\underline{1}M\lambda$ ... $\rightarrow$

$\rightarrow (s_3)$ ... $\lambda M00\underline{1}1M\lambda$ ... $\rightarrow (s_3)$...$\lambda M0\underline{0}11M\lambda$ ... $\rightarrow$

$\rightarrow (s_4)$ ... $\lambda M\underline{0}011M\lambda$ ... $\rightarrow (s_4)$ ... $\lambda\underline{M}0011M\lambda$ ... $\rightarrow$

$\rightarrow (s_0)$ ... $\lambda M\underline{0}011M\lambda$ ... $\rightarrow (s_1)$ ... $\lambda MM\underline{0}11M\lambda$ ... $\rightarrow$

$\rightarrow (s_1)$ ... $\lambda MM0\underline{1}1M\lambda$ ... $\rightarrow (s_1)$ ... $\lambda MM01\underline{1}M\lambda$ ... $\rightarrow$

$\rightarrow (s_1)$ ... $\lambda MM011\underline{M}\lambda$ ... $\rightarrow (s_2)$ ... $\lambda MM01\underline{1}M\lambda$ ... $\rightarrow$

$\rightarrow (s_3)$ ... $\lambda MM0\underline{1}MM\lambda$ ... $\rightarrow (s_3)$ ... $\lambda MM\underline{0}1MM\lambda$ ... $\rightarrow$

$\rightarrow (s_4)$ ... $\lambda M\underline{M}01MM\lambda$ ... $\rightarrow (s_0)$ ... $\lambda MM\underline{0}1MM\lambda$ ... $\rightarrow$

$\rightarrow (s_1)$ ... $\lambda MMM\underline{1}MM\lambda$ ... $\rightarrow (s_1)$ ... $\lambda MMM1\underline{M}M\lambda$ ... $\rightarrow$

$\rightarrow (s_2)$ ... $\lambda MMM\underline{1}MM\lambda$ ... $\rightarrow (s_3)$ ... $\lambda MM\underline{M}MMM\lambda$ ... $\rightarrow$

$\rightarrow (s_5)$ ... $\lambda MMM\underline{M}MM\lambda$ ... $\rightarrow (s_6)$ ... $\lambda MMMM\underline{M}M\lambda$ ... $\rightarrow$

$\rightarrow$ зупинка (перехід не визначений).

Оскільки машина завершила роботу у стані s_6, то це означає, що рядок $0^3 1^3$ розпізнаний. Отже, контекстно-вільна мова L_{nn} може бути задана за допомогою автомата з магазинною пам'яттю. Розглянемо мову $0^n 1^n 2^n$ (назвемо її L_{nnn}). Мова L_{nnn} є контекстно-залежною і вимагає більш складний автомат для розпізнавання. Побудуємо машину Тьюринга, що розпізнає цю мову.

Приклад. Машина Тьюринга, що розпізнає мову L_{nnn}. Покладемо

$$T = (S, I, f, s_0),$$

де

$S = \{s_0, \ldots, s_9\}$,
$I = \{0, 1, 2, M, E\}$,
s_0 — початковий стан;
s_9 — кінцевий стан.

Функцію f зображено у таблиці 9.5.

Таблиця 9.5. Функція f

	Вхідна інформація					
	λ	0	1	2	*E*	*M*
s_0	s_9 λ *L*	s_1 0 *L*	-	-	-	-
s_1	s_2 *E R*	-	-	-	-	-
s_2	-	s_3 *M R*	-	-	-	s_2 *M R*
s_3	-	s_3 0 *R*	s_4 *M R*	-	-	s_3 *M R*
s_4	-	-	s_4 1 *R*	s_5 *M R*	-	s_4 *M R*
s_5	s_6 λ *L*	-	-	s_5 2 *R*	-	-
s_6	-	-	-	s_7 2 *L*	-	s_8 *M L*
s_7	-	s_7 0 *L*	s_7 1 *L*	s_7 2 *L*	s_2 *E R*	s_7 *M L*
s_8	-	-	-	-	s_9 *E L*	s_8 *M L*
s_9	-	-	-	-	-	-

Машини Тьюринга можуть моделювати не тільки розпізнавання рядків мови, але і будь-які інші обчислювальні алгоритми. Очевидно, що для виконання таких найпростіших обчислень, як додавання або множення, потрібні досить громіздкі машини Тьюринга. Декілька спрощує структуру цих машин додавання додаткових паралельних вхідних і вихідних стрічок. Але зрозуміло, що машина Тьюринга не є оптимальним пристроєм для практичного виконання обчислень або розпізнавання мов. Основна ціль їх створення полягає у дослідженні з їх допомогою питань складності алгоритму та алгоритмічної розв'язності. Якщо вдається довести, що для розв'язку деякої задачі неможливо побудувати машину Тьюринга, то це означає, що не існує й алгоритму для її розв'язку.

Визначення

Недетермінований розпізнавач, пам'яттю якого є первинно порожня стрічка машини Тьюринга довжиною не більше вхідного ланцюжка, називається ***лінійно-обмеженим автоматом***.

Мова визначається лінійно-обмеженим автоматом тоді і тільки тоді, коли вона контекстно-залежна.

Запитання

1. Сформулюйте тезис Черча — Тьюринга.
2. Які мови можна розпізнавати за допомогою машини Тьюринга?
3. Зобразіть схему машини Тьюринга.
4. Дайте визначення машини Тьюринга.
5. Яка основна ціль створення машини Тьюринга?
6. Що зображує лінійно-обмежений автомат, і які мови він визначає?

Завдання

1. Дослідіть роботу машини Тьюринга для розпізнавання мови L_{nnn}: $0^n1^n2^n$ (конструкцію машини наведено як приклад у цьому розділі). Використовуйте як вхідну інформацію рядки, що належать і не належать даній мові.
2. Нехай машина Тьюринга задана таблицею.

	Вхідна інформація		
	0	1	λ
s_0	s_0 0 R	s_1 1 R	s_3 λ R
s_1	s_0 0 R	s_2 0 L	s_3 λ R
s_2	s_3 0 R	-	-
s_3	-	-	-

Визначте, який вид буде мати інформація на стрічці після зупинки машини, якщо на початку роботи на вхід подано рядок «... λλ0̲10110λλ ...» (голівка встановлена на підкреслений символ).

3. Побудуйте машину Тьюринга (алфавіт стрічки {0, 1}), яка:
   а) змінює перший 0 на стрічці на 1 і решту символів залишає без зміни;
   б) змінює всі символи 1 на стрічці на 0, крім лівої одиниці;
   в) розпізнає рядки, що закінчуються нулем;
   г) розпізнає рядки, що містять парне число одиниць.
4. Складіть алгоритм, що моделює роботу машини Тьюринга. На вхід алгоритму надходить таблиця машини та вхідний рядок.

розділ 10

Комбінаторика

10.1. Передмова

Комбінаторика або комбінаторний аналіз — розділ дискретної математики, яка вивчає комбінації і перестановку предметів, взаємне розташування частин скінченних множин предметів довільного походження, а також нескінченних множин, які задовольняють деякі умови підпорядкованості. Виникла комбінаторика у XVII ст. Але у самостійну наукову дисципліну комбінаторний аналіз сформувався лише у XX ст. Комбінаторні методи застосовуються в теорії ймовірностей, випадкових процесах, статистиці, математичному програмуванні, плануванні експериментів. Розглядаються задачі, в яких доводиться обирати ті чи інші предмети, розташовувати їх у певному порядку і знаходити серед різноманітних комбінацій найкращі. Комбінаторика тісно зв'язана з теоріями чисел, графів, скінченних автоматів. Її досягнення використовуються під час планування та аналізу наукових експериментів, у лінійному та динамічному програмуванні, у математичній економіці, у системах проектування та керування, у комп'ютерних науках та інших галузях науки і техніки.

Виділяють такі проблеми комбінаторного аналізу:

1) *Задачі на перелічення*, в яких необхідно визначити кількість розміщень елементів скінченної множини, що задовольняють певні умови. Для розв'язку задач перелічення розроблено різноманітні методи, серед яких слід відзначити метод продуктивних функцій і метод перелічення Пойа.

2) *Задачі про існування та побудову.* В задачах такого класу розглядаються питання: чи має місце визначена конфігурація частин скінченної множини з деякими властивостями; якщо така конфігурація існує, то як її побудувати. При цьому важливу роль відіграють чисельні та алгебраїчні методи.

3) *Задачі про вибір.* Задачі такого типу досліджують умови, за яких можна здійснити вибір підмножини або деякої сукупності частин множини так, щоб задовольнити певні вимоги. Для розв'язку задач про вибір, крім комбінаторних методів, треба застосовувати алгебраїчний апарат.

Історія розвитку комбінаторики

Китайські рукописи XII–XIII ст. до н. е. описували навколишню дійсність як об'єднання двох початків — чоловічого і жіночого. Вісім рисунків з трьох рядів символів зображували землю, гори, воду та інші стихії — сума перших восьми натуральних чисел (36) утілювала Всесвіт.

Легенда про китайського імператора Ію, який жив 4000 років тому, розповідає: існувала священна черепаха, на панцирі якої був зображений рисунок, що містив дев'ять чисел (рис. 10.1).

4	9	2
3	5	7
8	1	6

Рис. 10.1. Магічний квадрат

Додавши числа у рядках, стовпцях або діагоналях магічного квадрату, зображеного на рис. 10.1, одержимо одне й те ж число — 15.

Конкретні комбінаторні задачі, які стосувалися переліку предметів або невеликих їх груп, стародавні греки розв'язували без помилок.

Існують суміжні задачі між комбінаторикою та теорією чисел.

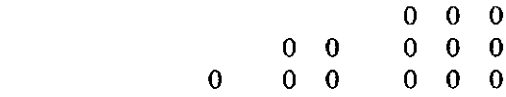


Рис. 10.2. Зображення квадратів чисел

Старогрецькі філософи запропонували поняття «квадрату числа» — квадрати натуральних чисел зображувалися камінцями, як зображено на рис. 10.2. За допомогою комбінацій предметів та обчислення кількості таких комбінацій було одержано числа, які дорівнюють сумі своїх дільників, наприклад, $6 = 1 + 2 + 3$; «дружні числа», кожне з яких дорівнює сумі дільників деякого іншого числа.

Великий розвиток у середні віки в Європі та Азії одержали різноманітні числові забобони і тлумачення, зв'язані із заміною букв відповідними числами (греки позначали числа за допомогою букв — перші 9 літер алфавіту позначали цифри від 1 до 9, наступні за ними — від 10 до 90, а останні 9 літер — числа від 100 до 900). В середні віки вчені, які називалися кабалістами, піддавали такому «аналізу» слова Біблії та інших священних книг і робили на підставі своїх досліджень пророцтва про майбутнє світу. Особливо виділялися чортова дюжина — число 13, комбінації цього числа з днями тижня та астрономічними явищами — затьмареннями або появою комет; число диявола — 666.

Поряд з кабалістами і містиками комбінаторикою у середньовіччі займалися астрологи.

Астрологія — наука, яка вивчає об'єднання планет і їх взаєморозташування між собою, — також дала поштовх для формулювання і розв'язку класичних комбінаторних задач. Астрологів цікавило питання про рух планет і їх вплив на долю людини. Особливе значення приділяли вони об'єднанням планет — зустрічам різних планет в одному знаку Зодіаку.

Астролог Бен Езра (1140 г. н. е.) підрахував кількість сполучень семи планет по дві, по три і т. д. Виявилося

$$C_7^2 = C_7^5, \quad C_7^3 = C_7^4,$$

де C_n^k — число сполучень з n різних предметів по k.

Остаточно формулу числа сполучень одержав Леві Бен Герман у XIV ст.:

$$C_n^k = \frac{n(n-1)\ldots(n-k-1)}{1 \cdot 2 \cdot \ldots \cdot k}.$$

На початку XVII ст. цю формулу заново вивів французький математик П. Ерігон.

Велику увагу класифікації видів суджень приділяла схоластична наука. У схоластиці дивно переплітalися дослідження

богословів з дослідженнями проблем, які примикають до комбінаторики, математичної логіки, теорії множин та інших сучасних розділів математики. Великими знавцями схоластичних досліджень були засновники теорії множин — *Бернард Больцано* та *Георг Кантор*. Сперечаючись про взаємовідношення членів пресвятої трійці, супідрядності ангелів, архангелів, херувимів та серафимів, схоласти були змушені розглядати різноманітні відношення порядків та ієрархії. Наприклад, загробний світ, що описаний Данте у «Божественній комедії», з його колами пекла і різними частинами чистилища і раю.

Схоласт *Раймонд Лулій* створив у XIII ст. машину, складену з кількох кіл, на яких були нанесені основні предикати, суб'єкти, атрибути та інші поняття схоластичної логіки. Повертаючи ці кола, він одержував різноманітні об'єднання понять і сподівався знайти з їх допомогою істину.

Розв'язуючи питання про добуток коренів будь-якого степеня, арабські алгебраїсти дійшли до формули для степеня суми двох чисел, відомою під назвою «біном Ньютона». Напевне, цю формулу знав поет и математик *Омар Хайям*, який жив у XI–XII ст. н. е. В XIII–XV ст. н. е. таку формулу наводять у своїх роботах деякі арабські вчені.

Італієць *Леонардо Пізанський*, на прізвисько *Фібоначчі*, у своїй книзі «Liber Abaci», що видана у 1202 р., систематизував всю арифметику арабів, деякі відомості з геометрії *Евкліда* і додав до них результати своїх досліджень. Робота Фібоначчі містила і нові комбінаторні задачі, наприклад, про знаходження найменшої кількості гир, за допомогою яких можна одержати будь-яку цілу вагу від 1 до 40 фунтів. Розглядав Леонардо і знаходження цілих розв'язків рівнянь. Далі аналогічні задачі призвели до розв'язку задачі про кількість натуральних розв'язків систем рівнянь та нерівностей, яка може розглядатися як одна з фундаментальних частин комбінаторики.

Але головною заслугою Леонардо перед комбінаторикою було те, що він сформулював та розв'язав «задачу про кроликів». Ще за часів грецьких математиків були відомі дві послідовності, кожний член яких одержувався за визначеними правилами з попередніх членів — арифметична та геометрична прогресії. В задачі Леонардо з'явилася нова послідовність, кожний член якої (починаючи з третього) дорівнює сумі двох попередніх членів: $u_n = u_{n-1} + u_{n-2}$. Подібні формули одержали назву рекурентних (від латинського recurrere — повертатися). Метод

рекурентних формул виявився з часом одним з найпотужніших для розв'язку комбінаторних задач.

Нарди, шашки, шахи, японські облавні шашки «го» — в цих іграх треба розглядати різноманітні з'єднання пересувних фігур, і перемагає той, хто їх краще вивчить, прорахує виграшні комбінації та уникне програшних.

Чималий поштовх до розвитку комбінаторики дали азартні ігри, які існували дуже давно, але одержали особливе поширення після хрестових походів. Найбільше поширення одержала гра у кісті.

Роботи *Блеза Паскаля* і *П'єра Ферма* (Франція, XVII ст.) поклали початок комбінаторної (дискретної) теорії ймовірностей.

В 1666 р. *Готфрид Вільгельм Лейбниць* запропонував фундаментальні поняття комбінаторики. До області комбінаторики Лейбниць відносив і «універсальну характеристику» — математику суджень, тобто прообраз математичної логіки.

Після робіт Паскаля та Ферма, Лейбниця та Л. Ейлера можна було вже говорити про комбінаторику як про самостійну частину математики, тісно пов'язану з іншими розділами науки, як теорія ймовірностей, вчення про ряди і т. п. Наприкінці XVIII ст. німецький вчений Гінденбург і його учні зробили навіть спробу вибудувати загальну теорію комбінаторного аналізу.

В XIX ст. під час дослідження з комбінаторики почали простежуватися зв'язки цієї теорії з визначниками, скінченними геометріями, групами, математичною логікою і т. д.

Однією з найбільш складних загадок у біології XX ст. була будова «нитей життя» — молекул білка і нуклеїнових кислот. Виявилося, що молекули білка — це об'єднання кількох довгих ланцюгів, що складені з 20 амінокислот. Після відкриття будови ДНК виникло питання: яким чином молекули ДНК передають організму інструкції про побудову ланцюгів амінокислот, з яких складаються білки. Дослідження показали, що мова йде про 20 амінокислот. Американський фізик *Г. Гамов* сформулював задачу так: як за допомогою 4 видів нуклеотидів можна зашифрувати 20 видів амінокислот? Ця задача, як і багато інших задач генетики, має комбінаторний зміст і, зокрема, зв'язана з таким розділом комбінаторики, як *кодування*.

Криптографія — наука про шифрування — була відома з давніх часів; одна з областей криптографії — розшифровка письмен (текстів) древніх цивілізацій. Ключ до прочитання

ієрогліфів, загублений кілька тисячоліть тому, було знову знайдено саме завдяки комбінаторним методам у читанні забутих письменностей, заснованих на спостереженнях за текстом, на зіставленні повторюваності комбінацій слів і граматичних форм. Також потрібний аналіз призначення надпису, часу та умов його складення і т. д. Типовий приклад з цієї області надає історія розшифровки клинописних надписів.

10.2. Первинні поняття комбінаторного аналізу

Деякі класичні задачі комбінаторики

А. **Магічний квадрат.** «Розмістити числа 1, 2, 3, 4, 5, 6, 7, 8, 9 у вигляді квадрату так, щоб сума чисел будь-якого із стовпців, рядків і діагоналей була однією й тією ж».

Магічний квадрат 3-го порядку було зображено на рис. 10.1. Більш складна задача — побудова магічних квадратів 4-го порядку; доведена можливість побудови всього 880 типів таких квадратів. Існують алгоритми побудови квадратів вищих порядків і магічних кубів.

Б. **Шахові задачі.** Добре відома задача про ферзі: «Поставити на шахову дошку найбільшу кількість ферзів таким чином, щоб жоден з них не зміг взяти іншого».

Розв'язок тут досить очевидний — більше 8 ферзів на дошку поставить не вдається.

Оскільки ферзь б'є по горизонталі, вертикалі і діагоналі шахової дошки розміром 8 × 8 кліток, то більше 8 ферзів поставити неможливо. Задачу можна розв'язати прямим перебором варіантів, і виявиться, що вісім ферзів можна розмістити таким чином, причому всього є 92 варіанти такого розміщення. Один з варіантів як приклад наведено на рис. 10.3.

В загальноприйнятих позначеннях шахових вертикалей і горизонталей вказаний варіант розміщення ферзів записується так:

$$(a, 6), (b, 3), (c, 5), (d, 8), (e, 1), (f, 4), (g, 2), (h, 7).$$

Наведемо ще один припустимий варіант розміщення:

$$(a, 6), (b, 1), (c, 5), (d, 2), (e, 8), (f, 3), (g, 7), (h, 4).$$

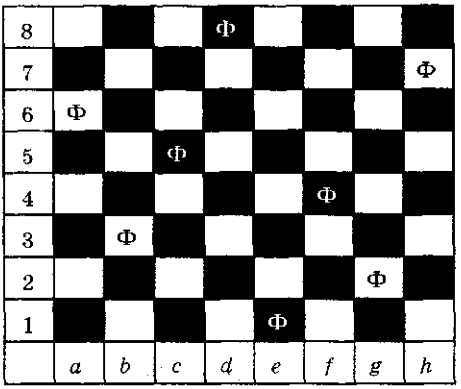


Рис. 10.3. Розміщення ферзів таким чином, що жоден з них не може взяти іншого

На дошці розміром $n \times n$, де число n не повинне ділитися ані на 2, ані на 3, можна поставить по n ферзів n різноманітних кольорів таким чином, щоб ферзі одного кольору не били один одного.

Якщо розмістити шахових коней, то виявиться, що будь-який з них на дошці розміром 2×4 може бити тільки одну клітку і на цій дошці можна розмістити тільки чотирьох коней, які не б'ють один одного. На дошці 8×8 можна таким чином поставити тільки $8 \cdot 4 = 32$ коня, оскільки дошка розміром 8×8 кліток розбивається на вісім полів розміром 2×4. На дошці розміром $n \times n$ можна поставити $(n \cdot n)/2$ коней, що не б'ють один одного, якщо n парне, і $(n \cdot n + 1)/2$ коней, якщо n непарне.

В. **Латинські квадрати і блок-схеми.** Квадрат розміром n рядків і n стовпців, складений з n літер таким чином, щоб кожна буква входила лише один раз у кожний стовпець і кожний рядок, називають ***латинським***. На рис. 10.4 зображено латинський квадрат розміром 4×4.

$$\begin{matrix} A & B & C & D \\ B & A & D & C \\ C & D & A & B \\ D & C & B & A \end{matrix}$$

Рис. 10.4. Латинський квадрат 4×4

Якщо два латинських квадрати «накласти» один на один таким чином, щоб кожна пара букв (велика і маленька) A, B, C, D і a, b, c, d зустрілися тільки один раз, то такі квадрати називають ***ортогональними.***

Ортогональний латинський квадрат розміром 4×4 зображено на рис. 10.5.

Ab	*Db*	*Ba*	*Cc*
Bc	*Ca*	*Ad*	*Db*
Cd	*Bb*	*Dc*	*Aa*
Da	*Ac*	*Cb*	*Bd*

Рис. 10.5. Пара ортогональних латинських квадратів

Першим задачу про ортогональні латинські квадрати розглянув Л. Ейлер (задача про «офіцерське каре»). Ця задача не має розв'язку при $n = 2$ і $n = 6$.

Г. Теорема про представників

Нехай у деякій множині X виділені підмножини $X_1, X_2, ..., X_n$. *Для того, щоб у* X *можна було обрати* n *різних елементів-представників* $a_1, a_2, ..., a_n$, *таких, що* $a_1 \in X_1$, $a_2 \in X_2$, ..., $a_n \in X_n$, *необхідно і достатньо виконання такої умови: для кожного значення* $k = 1, 2, ..., n - 1$ *об'єднання будь-яких* k *обраних підмножин* X *повинне містити, щонайменше,* k *неоднакових елементів.* Сформульована теорема про представників була доведена Ф. Холлом. Сама проблема пошуку представників має інші назви або тлумачення. Наприклад, назва «*задача про селянські весілля*» виникло тому, що Герман Вейль вперше сформулював подібну задачу у декілька жартівливому тоні: «В селі відносно кожного парубка і дівчини відомо, дружать вони чи ні. Якщо для будь-яких k парубків об'єднання множин їх подруг містить, щонайменше, k дівчат, то кожний парубок може обрати собі дружину з числа своїх подруг». Тут X — множина всіх дівчат; $X_1, X_2, ..., X_n$ — підмножини, які складаються з подруг першого, другого, ..., n-го парубка. Доводять цю теорему методом математичної індукції за кількістю парубків.

Зауважимо, що Ф. Холл довів справедливість теореми про різних представників для будь-якої системи підмножин $X_1, X_2, ..., X_n$ множини X, яка може містити і нескінченне число елементів. Остаточній варіант теореми Ф. Холла еквівалентний *теоремі Д. Кеніга про матриці* (будь-яка з теорем легко

виводиться одна з одної). Теорема Кеніга стосується булевих матриць, які зустрічаються у логіці, цілочисловому програмуванні, в теорії графів і мереж. Будь-який стовпець або рядок матриці називаються її «*лінією*».

Теорема Д. Кеніга ***(про матриці)***

Нехай елементи прямокутної матриці складаються із нулів та одиниць. Мінімальне число ліній, які містять всі одиниці матриці, дорівнює максимальному числу q таких одиниць у матриці, серед яких не існує двох розміщених на одній лінії.

Кожна підмножина з q одиниць матриці із вказаною властивістю назвемо ***базисною множиною одиниць.*** Розглянемо приклади булевих матриць:

$$A = \begin{array}{c} 4 \\ 3 \\ 2 \\ 1 \end{array}\!\!\begin{bmatrix} 0 & 1^* & 0 & 0 & 0 \\ 0 & 0 & 1^* & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\!\!, \qquad B = \begin{array}{c} 4 \\ 3 \\ 2 \\ 1 \end{array}\!\!\begin{bmatrix} 1^* & 1 & 0 & 0 \\ 1 & 1^* & 0 & 0 \\ 0 & 0 & 1^* & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}\!\!.$$
$$\begin{array}{ccccc} a & b & c & d & e \end{array} \qquad \begin{array}{cccc} a & b & c & d \end{array}$$

У матриці A базисна множина одиниць позначена зірочками, вона єдина. Число q *дорівнює* двом, а мінімальна множина ліній, які містять всі одиниці A, не одна. Власне, існує 4 мінімальних множини ліній:

{рядок 3, рядок 4}, {стовпець b, стовпець c},
{рядок 4, стовпець c}, {рядок 3, стовпець b}.

У матриці B число q дорівнює трьом. Мінімальна множина ліній, які містять всі одиниці, одна: {стовпець a, стовпець b, рядок 2}.

Базисна множина одиниць позначена зірочками, але вона не одна.

Завдання

Перерахувати всі базисні множини одиниць у матриці B.

Д. Правила суми і добутку

Правило суми. Якщо множина M — це об'єднання множин M_1, M_2, M_3, ..., M_k, які не перетинаються попарно, тоді кількість елементів множини зв'язані співвідношенням

$$|M| = |M_1| + \ldots + |M_k|.$$

Тут $|M|$ — кількість елементів множини M.

Нехай предмет a_1 можна обрати m_1 способами, предмет a_2 — m_2 способами, ..., предмет a_k — m_k способами. Тоді вибір предмета a_1, або a_2, ..., або a_k може бути зроблений

$$m_1 + m_2 + \ldots + m_k \qquad (10.1)$$

способами.

Підкреслимо, що тут сполучник «або» вживається у розділовому вмісті, тобто вибір a_i виключає вибір будь-якого іншого предмета.

Приклад. З міста A у місто B вирушають 10 потягів, 5 літаків і 3 автобуси. Скількома способами одній людині можна дістатися з A до B?

За правилом суми всього існує $10 + 5 + 3 = 18$ способів.

Правило добутку. Якщо M_1, M_2, M_3, ..., M_k — скінченні множини і $M = M_1 \times M_2 \times \ldots \times M_k$ — їх декартів добуток, то

$$|M| = |M_1 \times M_2 \times \ldots \times M_k| = |M_1| \cdot |M_2| \cdot \ldots \cdot |M_k|.$$

Нехай предмет a_1 можна обрати m_1 способами, предмет a_2 — m_2 способами, ..., предмет a_k — m_k способами і нехай вибір предмета a_1 не впливає на кількість способів вибору предметів a_2, ..., a_k; вибір предмета a_2 не впливає на кількість способів вибору предметів a_1, a_3, ..., a_k і т. д. Тоді вибір *упорядкованої* множини предметів $(a_1, a_2, \ldots, a_k)$ у вказаному порядку можна здійснити

$$m_1 \cdot m_2 \cdot \ldots \cdot m_k \qquad (10.2)$$

способами.

Приклад. Є 17 парубків і 21 дівчина. Скільки танцювальних пар вони можуть утворити?

Спочатку оберемо парубка — це можна зробити 17 способами, після цього кожний парубок обере собі партнершу (21 спосіб). За правилом добутку вибір упорядкованої множини танцювальних пар (парубок, дівчина) складає $17 \cdot 21 = 357$ пар.

Е. Складний вибір предметів

Часто у комбінаторних задачах вибір предметів здійснюється у кілька етапів, причому в деяких з них виконуються умови, за яких діють правила суми, в інших — правило добутку. Взагалі «складовий», «складний» вибір предметів, який здійснюється шляхом послідовного вибору «простих» предметів, у практичних задачах зустрічається досить часто. Зручним

засобом аналізу можливостей і перебору випадків вибору є побудова кореневого дерева — кожній послідовності можливих виборів відповідає шлях, який спрямований від кореня X_0 до кінцевих вершин дерева. Кінець будь-якої з дуг першого рівня відповідає можливому результату вибору першого рівня і т. д. Таким чином, дерево будується рівень за рівнем, поки не будуть вичерпані всі можливості вибору в послідовності, яка розглядається. Кінець кожної дуги k-рівня дерева відповідає єдиній послідовності результатів перших k виборів.

Приклад. Код Морзе. Скільки різних символів може бути записано кодом Морзе (•, –), якщо для їх запису використовується не більше п'яти знаків?

Існує п'ять способів вибору довжини символу: 1, 2, 3, 4 або 5 знаків. Із кореня X_0 вийдуть 5 дуг:

$$X_0X_1, \quad X_0X_2, \quad X_0X_3, \quad X_0X_4, \quad X_0X_5.$$

При довжині коду 1 існують 2 символи • і –, тому з вершини X_1 відповідно вибору довжини символу, який дорівнює 1, проводимо дві дуги: $X_1\,X_{11}$ і $X_1\,X_{12}$. При довжині коду, яка дорівнює 2, існують 4 символи: ••, •–, –•, ––; з вершини X_2 проводимо 4 дуги:

$$X_2X_{21}, \quad X_2X_{22}, \quad X_2X_{23}, \quad X_2X_{24}.$$

При довжині коду, яка дорівнює 3, маємо 8 символів. Отже, з вершини X_3 проводимо 8 дуг. Аналогічно 16 символів довжини 4 і 32 символи довжини 5. Загальна кількість кінцевих вершин дорівнює $2 + 4 + 8 + 16 + 32 = 62$. Це і є шукане число символів.

10.3. Перестановки, розміщення, сполучення

Розміщення і перестановки без повторень

Нехай $M = \{a_1, a_2, \ldots, a_n\}$ — фіксована множина.

Визначення

Упорядковані підмножини з k елементів $(b_1, b_2, \ldots, b_k)$ множини M називаються ***перестановками*** k елементів. Можна сказати, що k-перестановки $(b_1, b_2, \ldots, b_k)$ — це розміщення у визначеному порядку k елементів з множини M.

Визначення
k-перестановки множини з *n* елементів називаються ***розміщеннями з n по k*** елементів.

Якщо у перестановці беруть участь всі елементи множини (*n*-перестановка), то використовують термін ***перестановка***, а кількість всіх *n-перестановок* позначають через P_n.

Два розміщення з *n* по *k* різні, якщо вони складаються з різних елементів або з однакових елементів, але розміщених у різному порядку.

Позначення. A_n^k — ***кількість k-розміщень n-множини M*** (читається *A* з *n* по *k*). При утворенні розміщень перший елемент може бути обраний *n* способами, оскільки існує можливість незалежного вибору з *n* елементів, другий — $(n - 1)$ різними способами, оскільки незалежний вибір здійснюється для решти з $(n - 1)$ елемента, *k*-й елемент — відповідно $n - k + 1$ способами. Застосуємо правило добутку:

$$A_n^k = n(n-1)\ldots(n-k+1) = \frac{n!}{(n-k)!}. \qquad (10.3)$$

Якщо $n = k$, тоді A_n^n — кількість всіх способів упорядкування цих елементів:

$$P_n = A_n^n = n!. \qquad (10.4)$$

Приклад. $M = \{1, 2, 3\}$. Побудувати різні перестановки множини *M* по 2 і по 3 елементи.

Перестановки (розміщення) по 2 елементи:

(1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 2).

Їх кількість $A_3^2 = 3 \cdot 2 = 6$.

Перестановки по 3 елементи, тобто просто перестановки елементів множини *M*:

(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1).

Їх кількість $P_3 = 3! = 1 \cdot 2 \cdot 3 = 6$.

Розміщення і перестановки обов'язково враховують порядок елементів.

Розміщення і перестановки з повтореннями

Перестановки з *n* елементів у кількості (10.4) та розміщення з *n* елементів по *k* у кількості (10.3) зображують ***упорядковані вибірки*** попарно різних елементів початкової множини *M*, $|M| = n$. Часто доводиться робити упорядковані вибірки *з повтореннями* деяких елементів. Наприклад, такими

є 3-вибірки (1, 1, 2), (1, 2, 1), (2, 1, 1), (2, 3, 3), (3, 2, 3) із множини $M = \{1, 2, 3\}$. Такі загальні упорядковані k-вибірки (з повтореннями або без них) іноді називають ***кортежами*** (довжини k). Два кортежі (тобто дві загальні вибірки) вважаються *однаковими*, якщо вони мають однакову довжину і на місцях з однаковими номерами стоять однакові елементи.

Визначення

Розміщенням з повтореннями з n елементів по k (***k-розміщенням*** з повторенням з n) називається кортеж (вибірка з повторенням) довжини k з n елементів.

Кількість k-розміщень з повтореннями з n елементів обчислюється за формулою:

$$\overline{A_n^k} = n^k. \qquad (10.5)$$

Дійсно, після заповнення першого місця кортежу довжиною k одним з n елементів (що можливо зробити n варіантами) заповнити друге місце кортежу можна знову будь-яким елементом з усієї множини (повторюючи в одному з варіантів елемент, який знаходиться на першому місці), і так далі k разів. За правилом добутку одержимо, що $\overline{A_n^k} = n \cdot n \ldots n = n^k$.

Інше доведення цього співвідношення проводиться методом математичної індукції за *k-кількістю* елементів у розміщенні при фіксованому значенні n.

Приклад. Якщо у вигляді початкової множини взяти будь-яке слово, наприклад, БРАТ, тоді одержимо 24 слова з 4 різних букв (24 перестановки без повторень відповідно до формули (10.4) для $n = 4$). Якщо можливі повторення букв, наприклад, БРРР, БАБА та ін., тоді ми одержимо кількість слів $\overline{A_4^4} = 4^4 = 256$. Множина 3-слів, що використовують чотири різні букви (таких, як БАР, РАБ, АРА та ін.), складається з $4^3 = 64$ слів.

Перш ніж розглядати перестановки з повтореннями, з'ясуємо, скільки неоднакових кортежів довжини 4 можна утворити, переставляючи між собою букви у слові ГАГА. Таких перестановок маємо тільки 6 (замість числа перестановок $P_4 = 4! = 24$ з чотирьох різних букв без повторень):

ГАГА, ГААГ, ГГАА, АГАГ, ААГГ, АГГА.

Правило утворення цих 6 кортежів можна переформулювати інакше, більш сприятливо для узагальнення.

Чотириелементна множина M складається з двох елементів множини $M_1 = \{А, А\}$ типу «$А$» і двох елементів множини $M_2 = \{Г, Г\}$ типу «Г». Скільки існує 4-вибірок (перестановок) з елементів M таких, які містять два елементи типу А і два елементи типу Г?

Узагальнення цього прикладу забезпечує вдале формулювання та швидкий розв'язок задачі про *кількість упорядкованих перестановок із позначеними індексами повторень.*

Формулювання. Нехай множина M з n предметів ($|M| = n$) складається з множини M_1 предметів першого типу кількістю $n_1(|M_1| = n_1)$, множини M_2 предметів другого типу кількістю n_2 і т. д. до множини M_k предметів k-го типу кількістю $n_k(|M_k| = n_k)$, причому

$$M = \overset{k}{\underset{i=1}{U}} M_i, \quad n_1 + n_2 + \ldots + n_k = n, \quad M_i \cap M_j = \varnothing, \; i \neq j.$$

Скільки існує таких n-перестановок з повтореннями, які містять n_1 елементів першого типу, n_2 елементів другого типу, ..., n_k елементів k-го типу?

Елементи першого типу можна переставляти між собою $n_1!$ способами, але оскільки всі елементи однакові, такі перестановки нічого не змінюють.

Перестановки елементів одного типу можна робити незалежно від перестановок елементів іншого типу, тому за правилом добутку елементи перестановок можна переставляти між собою $n_1!n_2!\ \ldots\ n_k!$ способами так, що загальна n-перестановка не зміниться. Отже, множина всіх $n!$ перестановок із n елементів початкової множини розпадається на частини, які складаються з $n_1!n_2!\ \ldots\ n_k!$ однакових перестановок. Кількість різноманітних перестановок із вказаними повтореннями $n_1, n_2, \ldots\ n_k$ дорівнюють

$$P(n_1, n_2, \ldots, n_k) = \frac{n!}{n_1!n_2!\ldots n_k!}, \tag{10.6}$$

де $n = n_1 + n_2 + \ldots + n_k$. Список $(n_1, n_2, \ldots\ n_k)$ називають ***специфікацією перестановки.***

Сполучення

В тих випадках, коли не цікавий порядок елементів у комбінації, а цікавий лише склад комбінації, говорять про *сполучення* (іноді використовується термін «*комбінація*»).

Сполучення без повторень

Визначення

Сполученням з n елементів по k (***k-сполученням*** з ***n***) називається будь-яке k-розміщення (або ***k-вибірка***) з цих елементів, в якому враховується лише склад елементів і не враховується їх порядок.

Кількість k-сполучень з n елементів позначається C_n^k, або $\binom{n}{k}$, читається «кількість сполучень з n по k».

Із неупорядкованої множини елементів $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ можна утворити $k!$ упорядкованих k-розміщень позначених елементів. Тому кількість C_n^k всіх k-сполучень з n елементів у $k!$ разів менше, ніж кількість A_n^k всіх упорядкованих розміщень з n елементів по k:

$$C_n^k = \frac{A_n^k}{k!} = \frac{n!}{(n-k)!k!}. \qquad (10.7)$$

Ця формула співпадає з формулою (10.6), якщо $n_1 = k$, $n_2 = n - k$:

$$P(k, n-k) = \frac{n!}{(n-k)!k!}, \qquad (10.8)$$

іншими словами,

$$C_n^k = P(k, n-k). \qquad (10.9)$$

Приклад. У футбольному чемпіонаті беруть участь 17 команд. За умовою, що 3 останні команди залишають вищу лігу, скільки варіантів такого завершення чемпіонату?

Оскільки нас не цікавить взаємний порядок команд, які посіли останні місця у лізі, застосуємо формулу підрахунку сполучень — кількості варіантів:

$$C_{17}^3 = \frac{17!}{3!(14)!} = \frac{17 \cdot 16 \cdot 15}{1 \cdot 2 \cdot 3} = 17 \cdot 8 \cdot 5 = 680.$$

Сполучення з повтореннями

Маємо предмети n типів у такій кількості, що предметів кожного типу не менше, ніж k екземплярів. Скільки існує неупорядкованих k-вибірок (комбінацій) предметів з можливими повтореннями?

Ця проблема має дотепну «кондитерську» ілюстрацію. В магазині продаються тістечка чотирьох сортів (типів): бізе,

наполеони, пісочні та еклери. Скільки існує комбінацій придбання 7 тістечок? Тут $n = 4$, $k = 7$.

Якщо ідентифікувати типи тістечок номерами 1, 2, 3, 4 у тій послідовності, як вони записані, то приклади придбання тістечок мають вигляд кортежів довжини 4:

$$(1, 1, 1, 4), (2, 2, 2, 1), (2, 0, 0, 5), (0, 2, 3, 2), \ldots \quad (10.10)$$

Другий кортеж означає, що ми придбали 2 бізе, 2 наполеони, 2 пісочника, 1 еклер.

Наведемо еквівалентне формулювання цієї задачі. Знайти кількість C_n^k ***сполучень по k елементів з повтореннями із n елементів*** множини $A = \{a_i\}_1^n$.

Позначимо через k_i кількість повторень елемента a_i у визначеному сполученні (комбінації)

$k_i \in \{0, 1, 2, \ldots, k\}$, $i = 1, 2, \ldots, n$. Тоді $k_1 + k_2 + \ldots + k_n = k$.

Задачу розв'язують за допомогою кодування — зашифруємо кожну комбінацію нулями та одиницями: для кожного типу пишуть стільки одиниць, скільки предметів цього типу входить у комбінацію і, якщо предмети деякого типу не входять у комбінацію, тоді на відповідному місці пишеться 0. Сусідні типи відокремлюють один від одного нулем, якщо обидва сусідніх типа містять хоча б по одній одиниці. Так, чотири комбінації (10.10) з «кондитерського прикладу» кодуються відповідно таким чином:

(1, 0, 1, 0, 1, 0, 1, 1, 1, 1), (1, 1, 0, 1, 1, 0, 1, 1, 0, 1);
(1, 1, 0, 0, 0, 1, 1, 1, 1, 1), (0, 1, 1, 0, 1, 1, 1, 0, 1, 1).

При кодуванні одержимо стільки одиниць, скільки предметів входить до комбінації, тобто k одиниць і $(n - 1)$ нулів. Різним комбінаціям (або неупорядкованим вибіркам) відповідають різні перестановки з повтореннями, а кожній перестановці відповідає комбінація нулів та одиниць. Таким чином, кількість ***k-сполучень з повтореннями з елементів n типів*** $\overline{C_n^k}$ дорівнює кількості $P(k, n - 1)$ перестановок з повтореннями з k одиниць і $(n - 1)$ нулів:

$$\overline{C_n^k} = P(k, n-1) = \frac{(k+n-1)!}{k!(n-1)!} = C_{n+k-1}^k. \quad (10.11)$$

Додамо умову — до комбінації повинні входити елементи кількох фіксованих типів (r типів) $r \le n$. Щоб виконати цю умову, візьмемо одразу ж по одному елементу з даних r типів.

Таким чином, у k-сполученні виявляться зайнятими r місць, інші $(k-r)$ місць можна заповнити будь-якими елементами, тому комбінацій шуканого виду буде

$$\overline{C_n^{k-r}} = C_{n+k-r-1}^{k-r} = \frac{(k+n-r-1)!}{(k-r)!(n-1)!}. \qquad (10.12)$$

Завдання

1. В задачі з тістечками підрахувати кількість варіантів купівлі 7 тістечок.
2. За тих же самих умов задачі підрахувати: скільки існує таких варіантів купівлі 7 тістечок, які містять:
   а) хоча б одне бізе;
   б) не менш ніж 3 еклери.

Властивості сполучень

Арифметичний трикутник (трикутник Паскаля)

Послідовні значення C_n^k можна підрахувати за допомогою формул (10.13), які перевіряються безпосередньо за допомогою (10.7):

$$C_n^k = C_n^{n-k}, \quad C_n^k = C_{n-1}^{k-1} + C_{n-1}^k. \qquad (10.13)$$

Побудуємо так званий арифметичний трикутник.

1. Покладемо $C_0^0 = 1$, запишемо це значення у перший рядок.

2. В другому рядку запишемо значення $C_1^0 = 1$, $C_1^1 = 1$ таким чином, щоб значення C_0^0 опинилося над проміжком між числами другого рядка (рис. 10.6).

3. Наступний рядок: перше і останнє (третє) числа є $C_2^0 = C_2^2 = 1$. Між цими числами запишемо значення $C_2^1 = 2$. За формулою (10.13) $C_2^1 = C_1^0 + C_1^1$, тобто C_2^1 дорівнює сумі чисел попереднього рядка, що розміщені ліворуч та праворуч від нього: $C_2^1 = C_1^0 + C_1^1 = 1 + 1 = 2$.

4. За таким же правилом заповнюємо наступні рядки — по краям пишемо числа $C_n^0 = C_n^n = 1$, а всі проміжні значення одержуємо як суми двох чисел попереднього рядка, що розміщені ліворуч та праворуч від шуканого значення. В результаті одержимо ***арифметичний трикутник***, перші рядки якого наведено на рис. 10.6.

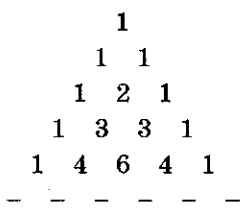


Рис. 10.6. Арифметичний трикутник

При побудові арифметичного трикутника кожне число n-рядка бере участь у формуванні двох чисел $(n+1)$-го рядка. Якщо додати числа $(n+1)$-го рядка через одно, то одержана сума дорівнює сумі всіх чисел n-рядка.

Через одно числа $(n+1)$-го рядка можна додавати двома способами — підсумувати числа, що розміщені на непарних позиціях у рядку, або ті, що розміщені на парних позиціях. В обох випадках одержимо однакове число, яке дорівнює сумі чисел у рядку з попереднім номером n. Цю властивість запишемо у вигляді

$$C_n^0 + C_n^2 + C_n^4 + \ldots = C_n^1 + C_n^3 + C_n^5 + \ldots = C_{n-1}^0 + C_{n-1}^1 + C_{n-1}^2 + \ldots + C_{n-1}^{n-1}.$$

Обрані тотожності

З властивостей арифметичного трикутника випливає, що сума чисел $(n+1)$-го рядка вдвічі більше суми чисел n-го рядка. В першому рядку стоїть число 1, і сума чисел першого рядка 1. Тому у $(n+1)$-му рядку сума чисел дорівнює 2^n, тобто

$$C_n^0 + C_n^1 + \ldots + C_n^n = 2^n. \qquad (10.14)$$

Ще кілька властивостей наведемо без доведення:

$$C_{n-1}^0 + C_n^1 + C_{n+1}^2 + \ldots + C_{n+m-1}^m = C_{n+m}^m; \qquad (10.15)$$

$$C_n^n + C_{n+1}^n + C_{n+2}^n + \ldots + C_{n+m-1}^n = C_{n+m}^{n+1}. \qquad (10.16)$$

Звідси

$$1 + 2 + 3 + \ldots + m = \frac{m(m+1)}{2} = C_{m+1}^2; \qquad (10.17)$$

$$1^2 + 2^2 + 3^2 + \ldots + m^2 = \frac{m(m+1)(2m+1)}{6};$$

$$1^3 + 2^3 + 3^3 + \ldots + m^3 = \frac{m^2(m+1)^2}{4};$$

$$(C_p^0)^2+(C_p^1)^2+\ldots+(C_p^p)^2=C_{2p}^p;$$

$$C_n^0-C_n^1+C_n^2-\ldots+(-1)^nC_n^n=0;$$

$$C_n^k\cdot C_{n-k}^{m-k}=C_m^k\cdot C_n^m.$$

10.4. Формула включень та виключень. Застосування

Загальна формула. Поставимо задачу: «Нехай є N предметів, деякі з яких мають властивості α_1, α_2, ..., α_n. При цьому окремий предмет може не мати жодної з цих властивостей або мати одну з властивостей, або мати властивості α_i, α_j, ..., α_h і, можливо, деякі з інших властивостей. Позначимо через $N(\alpha_i, \alpha_j, \ldots, \alpha_h)$ кількість предметів, які мають властивості α_i, α_j, ..., α_h. Якщо предмети не мають властивості α_k, позначимо це штрихом α_k'. Наприклад, $N(\alpha_1, \alpha_2, \alpha_3')$ — кількість предметів із властивостями α_1, α_2, які не мають властивості α_3. Про інші властивості нам невідомо, чи мають їх ці предмети чи ні.

Загальна формула включень та виключень підраховує кількість предметів, позбавлених всіх без виключення властивостей і має вигляд:

$$N(\alpha_1', \alpha_2', \ldots, \alpha_n') = N - N(\alpha_1) - N(\alpha_2) - \ldots - N(\alpha_n) +$$
$$+N(\alpha_1,\alpha_2)+N(\alpha_1,\alpha_3)+\ldots+N(\alpha_1,\alpha_n)+\ldots+N(\alpha_{n-1},\alpha_n)-N(\alpha_1,\alpha_2,\alpha_3)-$$
$$-\ldots-N(\alpha_{n-2},\alpha_{n-1},\alpha_n)+\ldots+(-1)^nN(\alpha_1,\alpha_2,\ldots,\alpha_n). \qquad (10.18)$$

Якщо $(\alpha_i\alpha_j \ldots \alpha_k\alpha_n)$ — комбінація властивостей без урахування їх порядку, то знак «+» у формулі (10.18) ставиться за умови, що кількість властивостей парна і «−» — якщо непарна. Наприклад, член $N(\alpha_1, \alpha_3, \alpha_8, \alpha_{11})$ входить із знаком «+», а член $N(\alpha_2, \alpha_4, \alpha_6)$ — із знаком «−».

Таким чином, у формулі включень та виключень спочатку виключаються всі предмети, які мають хоча б одну з властивостей α_1, α_2, ..., α_n, потім включаються предмети, які мають хоча б дві з цих властивостей, далі виключаються ті, які мають, щонайменше, три властивості і т. д.

Доведення проводиться методом індукції за кількістю властивостей.

Накладемо такі обмеження на кількість елементів, які мають деякі з властивостей α_1, α_2, ..., α_n:

$$N(\alpha_1) = \ldots = N(\alpha_n) = N^{(1)},$$
$$N(\alpha_1, \alpha_2) = N(\alpha_1, \alpha_3) = \ldots = N(\alpha_{n-1}, \alpha_n) = N^{(2)},$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$N(\alpha_1, \ldots, \alpha_k) = \ldots = N(\alpha_{n-k+1}, \ldots, \alpha_n) = N^{(k)},$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$N(\alpha_1, \alpha_2, \ldots, \alpha_n = N^{(n)}).$$

Таким чином, додатково припускається, що *кількість елементів* з будь-якими властивостями залежить тільки від кількості цих властивостей, а не від типу (або номерів) властивостей, що характерні для предметів. Звідси випливає, що сумарна кількість предметів, які мають рівно k властивостей, дорівнює

$$N(\alpha_1, \ldots, \alpha_k) + \ldots + N(\alpha_{n-k+1}, \ldots, \alpha_n) = C_n^k N^{(k)}, k = 1, 2, \ldots, n.$$

За вказаними обмеженнями формула включень та виключень має вигляд (10.19):

$$N(\alpha'_1, \ldots, \alpha'_n) = N - C_n^1 N^{(1)} + C_n^2 N^{(2)} - \ldots + (-1)^n C_n^n N^{(n)}. \quad (10.19)$$

Застосування формули включень і виключень до перестановок з обмеженнями

Кількість зміщень D_n

Знайти кількість зміщень D_n — перестановок з n елементів, із яких жоден не залишається у первинному місці.

Задача про зміщення відома давно: наприклад, нею займався ще Л. Ейлер. Значення D_n швидко одержується за допомогою формули (10.19) включень та виключень з обмеженнями. Нехай $N = P_n$ — кількість всіх перестановок із n елементів, $N^{(1)} = P_{n-1}$ — кількість таких перестановок, в яких один фіксований елемент α_i залишається на своєму місці. Взагалі, $N^{(k)} = P_{n-k}$ — кількість перестановок, в яких k фіксованих елементів (із цих n елементів) залишаються на своєму місці, $k = 1, \ldots, n$.

Тоді кількість зміщень D_n дорівнює $N(\alpha'_1, \ldots, \alpha'_n)$ — числу перестановок, у яких кожний елемент α_i не залишається на своєму місці. Згідно з (10.19) маємо

$$D_n = P_n - C_n^1 P_{n-1} + C_n^2 P_{n-2} - \ldots + (-1)^n C_n^n P_0 = n![1 - \frac{1}{1!} + \frac{1}{2!} + \ldots + \frac{(-1)^n}{n!}].$$

Безсумнівно, $D_1 = 0$.

Якщо $n = 0$, домовимося, що $D_0 = 1$.

Зміщення неповної кількості елементів

Кількість перестановок $D_{n,r}$ із n елементів, при яких рівно r елементів залишаються на первинних місцях, а інші $n-r$ обов'язково змінюють своє місце, дорівнює

$$D_{n,r} = C_n^r D_{n-r}.$$

Спочатку слід вибрати, які саме r елементів залишуться на місці. Це можна зробити C_n^r способами. Інші $(n-r)$ елементів можна переставляти будь-якими способами, лише б жоден з них не опинився на своєму початковому місці. Це можна зробити D_{n-r} способами. За правилом добутку одержимо $C_n^r D_{n-r}$ шуканих перестановок.

Розіб'ємо всі перестановки на класи залежно від того, скільки елементів залишаються незміщеними при цій перестановці

$$\sum_{r=0}^{n} D_{n,r} = \sum_{r=0}^{n} C_n^r D_{n-r} = n!,$$

(оскільки загальна кількість перестановок дорівнює $n!$).

За допомогою формули включень та виключень можна розв'язати таку задачу: «Знайти кількість перестановок із n елементів, при яких дані r елементів зміщені (а інші можуть бути або на своїх місцях, або зміщеними)». Ця кількість дорівнює

$$n! - C_r^1(n-1)! + C_r^2(n-2)! - \ldots + (-1)^r(n-r)!.$$

Аргументація здійснюється за зразком виводу формули для D_n: від першого кроку — виключення з P_n кількості перестановок $C_r^1 P_{n-1}$, в яких один з r елементів не змінює свого місця, до r-го кроку — врахування кількості $C_r^r P_{n-r} = P_{n-r}$ перестановок, в яких r вказаних елементів залишається на своїх місцях:

$$P_n - C_r^1 P_{n-1} + C_r^2 P_{n-2} - \ldots + (-1)^r C_r^r P_{n-r}.$$

Задачі про розміщення предметів

Розв'язуючи задачі про розміщення (розподіл предметів за урнами), необхідно проаналізувати зміст задачі, фізичні властивості предметів, щоб правильно розв'язати питання про те, вважати предмети однаковими чи різними, враховувати чи ні порядок вибірки предметів, розрізняти чи ні урни одну від одної. Якщо задача формулюється так, що неможливо однозначно відповісти на подібні питання, необхідно прийняти додаткові припущення.

Розподіл n різних предметів за k урнами. Кількість розміщень за k урнами n різних предметів за умовою, щоб у першу урну попало n_1 предметів, у другу — n_2, ..., в останню — n_k предметів, дорівнює:

$$P(n_1, n_2, \ldots, n_k) = \frac{n!}{n_1! n_2! \ldots n_k!}, \qquad (10.20)$$

$$n = n_1 + n_2 + \ldots + n_k.$$

Завдання

Виходячи із збігу формул (10.20) і (10.6), встановити зв'язок між задачами про n-перестановки з повтореннями (п. 10.3) і розподіл предметів між урнами.

Вказівка. Номери будь-якого з n_i предметів i-ї урни відповідають номерам місць, які займають у перестановках повторювані елементи i-го типу.

Розподіл n однакових предметів за k урнами

Встановимо, що n однакових предметів між k особами можна розділити

$$P(n, k-1) = C^{n}_{n+k-1} = C^{k-1}_{n+k-1}$$

способами.

□ Вишикуємо в один рядок n однакових предметів і утворимо додатково множину з $(k - 1)$ предметів другого типу («роздільників»). Вставляючи всі додаткові предмети на різні місця у рядку між початковими предметами, можна розділити останні на «k» послідовних ланцюжків, які вручаються відповідним особам. Зрозуміло, що кількість шуканих розподілів n однакових предметів між k **особами** дорівнює кількості перестановок із $(n + k - 1)$ предметів з повтореннями n предметів першого типу і $(k - 1)$ предметів другого типу, тобто дорівнює $P(n, k - 1)$. Якщо при цьому m роздільників передують всім n предметам першого типу, тоді особи 1, 2, ..., m не одержують жодного предмету при розділі, $m \in \{1, \ldots, k - 1\}$. ■

Якщо ділити справедливо, з рівнем справедливості r, то кожний учасник розподілу повинен одержати мінімум r предметів. Почнемо з того, що кожному роздамо по r предметів. Після цього залишається $(n - kr)$ предметів, які розподілимо довільно. Це можна зробити

$$C^{k-1}_{n-kr+k-1} = C^{k-1}_{n-k(r-1)-1}$$

способами.

Розподіл різних предметів без урахування порядку предметів в урнах

Необхідно розподілити n різних предметів між k урнами, місткість яких не обмежена (кожна людина, яка бере участь у розподілі, може забрати собі все). Перший предмет можна покласти у будь-яку з k урн, другий (незалежно від першого) також у k урн і т. д. За правилом добутку предмети можна розподілити k^n способами.

Розподіл різних предметів між однаковими урнами за умови, що урни не порожні

Позначимо S_n^k кількість способів розподілу n різних предметів між k однаковими урнами так, щоб кожна урна була не порожньою. Із кожного такого розподілу можна одержати $k!$ аналогічних розподілів за різними урнами (позначити k урн можна $k!$ способами). Таким чином, *кількість* $M(n, k)$ *розподілів* n *різних предметів між* k *різними урнами з використанням кожної урни у кожному розподілі* («не порожні урни») дорівнює $M(n, k) = k!S_n^k$.

Кількості $M(n, k)$ носять назву ***чисел Моргана.*** Формула

$$S_n^k = \frac{1}{k!}\left[k^n - C_k^1(k-1)^n + C_k^2(k-2)^n + \ldots + (-1)^{k-1}C_k^{k-1}1^n\right] \quad (10.21)$$

випливає з формули включень і виключень.

Числа S_n^k називаються ***числами Стірлінга другого роду.*** Визначимо:

$S_n^k = 0$, якщо $k > n$ або $k = 0$. Також $S_0^0 = 1$, $S_n^n = 1$.

Числа Стірлінга S_n^k задовольняють рекурсивне співвідношення:

$$S_{n+1}^k = kS_n^k + S_n^{k-1}.$$

Числа Белла

Числа Белла B_n — це кількість всіх способів розбиття множини з n елементів в об'єднання непорожніх підмножин, які не перетинаються. Зрозуміло, що

$$B_n = \sum_{k=0}^{n} S_n^k,$$

де $B_0 = 1$, $S_n^0 = 0$.

Перевіряється, що

$$B_{n+1} = \sum_{k=0}^{n} C_n^k B_k.$$

Розподіл різних предметів з урахуванням їх порядку в урнах

Якщо не обмежувати кількість предметів в урнах і якщо предмети не розрізняти між собою, то кількість таких розподілів дорівнює C_{n+k-1}^{k-1}.

Кожному способу розподілу однакових предметів між урнами відповідає $n!$ способів розподілу різних предметів з урахуванням їх порядку, які одержуються за допомогою перестановок предметів між собою без зміни кількості предметів в урнах. За правилом добутку одержуємо шукану кількість розподілів:

$$C_{n+k-1}^{k-1} \cdot n! = \frac{(n+k-1)!}{(k-1)!} = A_{n+k-1}^{n}.$$

10.5. Біноміальна і поліноміальна формули

Біном Ньютона. Формулу

$$(a+b)^n = C_n^0 a^0 b^n + C_n^1 a^1 b^{n-1} + \ldots + C_n^k a^k b^{n-k} + \ldots + C_n^n a^n b^0 \qquad (10.22)$$

доведемо комбінаторними міркуваннями. Якщо вираз $(a+b)^n$ записати у вигляді добутку чинників $(a+b)(a+b)\ \ldots\ (a+b)$, то після їх перемноження без зміни порядку ми одержимо подібні доданки, до кожного з яких у різному порядку входять k множників a і $(n-k)$ множників $b (k = n, n-1, \ldots, 0)$. Кількість подібних членів $a^k b^{n-k}$ з фіксованими степенями k і $n-k$ дорівнює кількості варіантів запису букви a на k позиціях із n можливих позицій, тобто дорівнює C_n^k. Звідси випливає формула (10.22).

Поліноміальна формула

$$(a_1 + a_2 + \ldots + a_m)^n = \sum_{k_i=0}^{n} P(k_1, k_2, \ldots, k_m) \cdot a_1^{k_1} a_2^{k_2} \ldots a_m^{k_m}, \qquad (10.23)$$

де $k_1 + k_2 + \ldots + k_m = n$, є узагальнення формули біному Ньютона (10.22). Доведення одразу ж випливає з того, що кількість подібних членів $a_1^{k_1} \cdot a_2^{k_2} \cdot \ldots \cdot a_m^{k_m}$, що одержані після перемноження n множників

$$(a_1 + a_2 + \ldots + a_m)\,(a_1 + a_2 + \ldots + a_m)\ \ldots\ (a_1 + a_2 + \ldots + a_m),$$

дорівнює числу розбиття n місць на m груп для k_1 множників a_1, k_2 множників a_2, ..., k_m множників a_m. Загальна кількість варіантів такого розбиття дорівнює $P(k_1, k_2, \ldots, k_m)$ (див. (10.20)).

10.6 Комбінаторні задачі і теорія чисел

Решето Ератосфена

Одна з великих загадок математики — розміщення простих чисел у натуральному ряді. *Ератосфен* запропонував розв'язок задачі про знаходження всіх простих чисел серед натуральних від 1 до N (зараз 1 вважається особливим числом). Спосіб Ератосфена: спочатку викреслюють всі числа, які діляться на 2, залишаючи саме число 2, потім беруть наступне число — 3, зрозуміло, що воно просте, і викреслюють всі наступні числа, які діляться на 3 і т. д. Числа, які залишилися після викреслювання, і є прості.

Підрахуємо, скільки простих чисел у першій сотні. Позначимо цю кількість через $\pi(100)$. Прості числа, менші, ніж $\sqrt{100}=10$, є числа 2, 3, 5, 7. Застосуємо алгоритм решета Ератосфена до послідовності 2, 3, 4, 5, 6, 7, 8, ..., 100 і викреслимо всі числа, які мають хоча б один дільник 2, 3, 5, 7, а самі ці дільники залишаємо у послідовності. Тоді залишається $\pi(100)$ простих чисел. Позначимо властивості:

α_1 — подільність на 2;

α_2 — подільність на 3;

α_3 — подільність на 5;

α_4 — подільність на 7.

Зрозуміло, що у послідовності 1, 2, 3, ..., 100 кількість чисел $N(\alpha_1', \alpha_2', \alpha_3', \alpha_4')$, які не мають жодної властивості $\alpha_i (i = 1, 2, 3, 4)$, дорівнює $N(\alpha_1', \alpha_2', \alpha_3', \alpha_4') = \pi(100) + 1 - 4 = \pi(100) - 3$. Тут додана 1 — кількість особливих чисел (одиниця і відняте 4 — кількість перших чотирьох простих чисел (2, 3, 5, 7), кожне з яких має точно одну з властивостей α_i.

Нагадаємо, що $N(\alpha_1) = [\frac{100}{2}] = 50$ — кількість чисел від 1-го до 100, які діляться на 2; $N(\alpha_1\alpha_2) = [\frac{100}{2\cdot 3}] = 16$ — діляться на 2 і 3 одночасно; $N(\alpha_2, \alpha_3, \alpha_4) = [\frac{100}{3\cdot 5\cdot 7}] = 0$ — діляться на 3, 5, 7 одночасно; $N(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = [\frac{100}{2\cdot 3\cdot 5\cdot 7}] = 0$ — діляться на будь-яке з чисел 2, 3, 5, 7. Тут $[k]$ є позначення цілої частини числа k.

За формулою включень та виключень маємо:

$$\pi(100) - 3 = N(\alpha_1', \alpha_2', \alpha_3', \alpha_4') = 100 - N(\alpha_1) - N(\alpha_2) -$$
$$- N(\alpha_3) - N(\alpha_4) + N(\alpha_1, \alpha_2) + N(\alpha_1, \alpha_3) + N(\alpha_1, \alpha_4) + N(\alpha_2, \alpha_3) +$$
$$+ N(\alpha_2, \alpha_4) + N(\alpha_3, \alpha_4) - N(\alpha_1, \alpha_2, \alpha_3) - N(\alpha_1, \alpha_2, \alpha_4) - N(\alpha_1, \alpha_3, \alpha_4) -$$
$$- N(\alpha_2, \alpha_3, \alpha_4) + N(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = 100 - 50 - 33 - 20 - 14 + 16 +$$
$$+ 10 + 7 + 6 + 4 + 2 - 3 - 2 - 1 - 0 + 0 = 22.$$

Остаточно одержуємо, що у першій сотні вміщується $\pi(100) = 25$ простих чисел.

Завдання

Навести всі 25 простих чисел від 2 до 100. Скільки серед них налічується «пар близнюків» — пар простих чисел, які відрізняються на 2? Скільки простих чисел попаде у кожний десяток — перший, другий, ..., десятий?

Рекурентні співвідношення

Метод рекурентних співвідношень полягає у тому, що розв'язок комбінаторної задачі з n предметами виражається через розв'язок аналогічної задачі з меншою кількістю предметів за допомогою деякого співвідношення, яке називається рекурентним («зворотним»). Користуючись цим співвідношенням, шукану величину можна обчислити, виходячи з того, що для невеликої кількості предметів (одного, двох) розв'язок задачі легко знаходиться.

Проілюструємо метод рекурентних співвідношень на прикладі сполучень з повтореннями.

Позначимо кількість сполучень із n предметів $\{a_1, ..., a_n\}$ по k з повтореннями через f_n^k (у розділі 10.3 було застосовано позначення $\overline{C}_n^k$). Будь-яке із сполучень або містить, або не містить a_1. Кількість сполучень, які не містять a_1, дорівнює f_{n-1}^k (це сполучення з предметів $a_2, ..., a_n$ по k). Кожне сполучення, яке містить a_1 як мінімум один раз, може бути одержане приєднанням до a_1 деякого сполучення з n предметів по $(k - 1)$ (кількість таких сполучень дорівнює f_n^{k-1}). Отже,

$$f_n^k = f_{n-1}^k + f_n^{k-1}. \qquad (10.24)$$

Послідовно застосовуючи це рекурентне співвідношення, одержимо

$$f_n^k = f_n^{k-1} + f_{n-1}^k = f_n^{k-1} + (f_{n-1}^{k-1} + f_{n-2}^k) = \ldots = f_n^{k-1} + f_{n-1}^{k-1} + \ldots + f_2^{k-1} + f_1^k. \qquad (10.25)$$

Зрозуміло, що $f_n^1 = n$, $f_1^k = 1$.

Вважаючи $k = 2$, маємо (див. (10.17)):

$$f_n^2 = n + (n-1) + \ldots + 2 + 1 = n(n+1)/2 = C_{n+1}^2. \qquad (10.26)$$

Якщо $k = 3$, тоді з урахуванням (10.16) маємо:

$$f_n^3 = C_{n+1}^2 + C_n^2 + \ldots + C_3^2 + C_2^2 = C_{n+2}^3.$$

На $(k-1)$-му кроці одержимо:

$$f_n^k = C_{n+k-1}^k = P(k, n-1), \qquad (10.27)$$

що узгоджується з попереднім результатом (10.11).

Числа Фібоначчі

Леонардо Фібоначчі у 1202 р. розв'язував задачі за допомогою рекурентного співвідношення, в якому n-й член виражався за допомогою двох попередніх. Задача, яка розглядалася, має такий вигляд:

«Кожна пара дорослих кролів приносить щомісяця ще пару кроликів (самку і самця), які, у свою чергу, починають давати такий самий приплід через два місяці після свого народження. Скільки пар кроликів буде через рік, якщо на початку року була одна пара новонароджених кроликів і жоден з них за рік не загинув?»

Позначимо через u_n кількість пар кроликів на початку n-го місяця. Тоді за умовою $u_1 = 1$, $u_2 = 1$, $u_3 = 2$, оскільки на початку 3-го місяця з'являється приплід. Далі $u_4 = 3$, $u_5 = 5$, оскільки приплід дає як первинна пара, так і пара, що народилася наприкінці другого місяця. Легко бачити, що дані числа зв'язані між собою співвідношенням:

$$u_n = u_{n-1} + u_{n-2}, \qquad (10.28)$$

оскільки на початку n-го місяця маємо всі пари, які були на початку попереднього, і, крім того, приплід принесуть всі пари, що народилися за два місяці до даного і раніше.

Легко підрахувати, що $u_{13} = 233$, тобто через рік буде 233 пари кроликів.

Числа $u_1, u_2, \ldots, u_n$ називають ***числами Фібоначчі***. Ці числа зустрічаються у різних розділах математики, наприклад, при оптимальному знаходженні точок екстремуму методом проб.

Наведемо 14 членів послідовності $\{u_n\}$ з початковими членами $u_1 = 1$, $u_2 = 1$: $\{u_n\}_{n=1}^{\infty} = 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, \ldots$

Іноді послідовність чисел Фібоначчі починають з чисел $u_0 = 0$, $u_1 = 1$, доповнюючи натуральний індекс значенням $n = 0$:

$$\{u_n\}_{n=0}^{\infty} = 0, 1; 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots \quad (10.29)$$

Тоді рекурентне співвідношення (10.28) виконується для значень індексу $n \geq 2$.

В інших випадках розглядається послідовність чисел Фібоначчі f_n з початковими членами

$$f_0 = 1, f_1 = 2: \quad \{f_n\}_{n=0}^{\infty} = 1, 2; 3, 5, 8, 13, 21, 34, 55, 89, \ldots \quad (10.30)$$

Фібоначчі запропонував саме цю послідовність, виходячи з одної пари дорослих кроликів на передодні першого місяця ($f_0 = 1$) і одержуючи після першого народження дві пари кроликів на першому місяці ($f_1 = 2$). Рекурентне співвідношення (10.28) також виконується для послідовності (10.30), якщо $n \geq 2$:

$$f_n = f_{n-1} + f_{n-2}.$$

Зрозуміло, що послідовність чисел Фібоначчі (10.30) можна змістити вліво, наприклад, вибираючи початкові елементи $f_0 = 13$, $f_1 = 21$, або змістити вправо, наприклад, вибираючи

$$f_0 = -3, \quad f_1 = 2.$$

Тоді, задовольняючи рекурентне співвідношення (10.28), будемо мати відповідно послідовності

$$13, 21; 34, 55, 89, 144, 233, 377, \ldots$$

$$-3, 2; -1, 1, 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots \quad (10.31)$$

Перша з цих послідовностей є частиною послідовності Фібоначчі (10.30), друга — навпаки, містить послідовність (10.30) як свою частину.

Тим не менш, не будь-яка послідовність $\{f_n\}_0^{\infty}$, яка задовольняє рекурентну умову $f_n = f_{n-1} + f_{n-2}$, містить у своєму складі яку-небудь частину послідовних чисел Фібоначчі (10.30), наприклад,

$$2, 5; 7, 12, 19, 31, 50, 91, \ldots$$

$$-2, 6, 4, 10, 14, 24, 38, 62, \ldots$$

Наведемо явну формулу для обчислення довільного числа Фібоначчі, яку вперше одержав Біне. Для послідовності (10.29) формула має вигляд

$$u_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n, \quad n = 0, 1, 2, \ldots \quad (10.32)$$

Дійсно, для послідовності (10.32)

$$u_{n+2}-u_n=\frac{1}{\sqrt{5}}\frac{(1+\sqrt{5})^n}{2^n}\left[\frac{(1+\sqrt{5})^2}{2^2}-1\right]-\frac{1}{\sqrt{5}}\frac{(1-\sqrt{5})^n}{2^n}\left[\frac{(1-\sqrt{5})^2}{2^2}-1\right]=$$

$$=\frac{1}{\sqrt{5}}\frac{(1+\sqrt{5})^n}{2^n}\cdot\frac{1+\sqrt{5}}{2}-\frac{1}{\sqrt{5}}\frac{(1-\sqrt{5})^n}{2^n}\cdot\frac{1-\sqrt{5}}{2}=u_{n+1},$$

так що рекурентне співвідношення (10.28) виконується. Початкові значення $u_0 = 0$, $u_1 = 1$ одержуються з (10.32), якщо $n = 0$, $n = 1$ відповідно.

Враховуючи зміщення індексу на 2, для послідовності чисел Фібоначчі f_n (10.30) маємо такий вигляд формули Біне:

$$f_n=\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+2}-\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+2},\quad n=0,\ 1,\ 2,\ \ldots$$

Особлива роль чисел $\frac{1+\sqrt{5}}{2}$, $\frac{1-\sqrt{5}}{2}$ у формулі Біне пояснюється тим, що ці числа є коренями рівняння

$$\lambda^2-\lambda-1=0:\quad \lambda_1=\frac{1+\sqrt{5}}{2},\ \lambda_2=\frac{1-\sqrt{5}}{2}. \qquad (10.33)$$

Це рівняння носить назву ***характеристичного*** для рекурентного співвідношення Фібоначчі (10.28) $u_n - u_{n-1} - u_{n-2} = 0$ і виникає, якщо шукати розв'язок (10.28) у вигляді $u_n = \lambda^n$:

$$\lambda^n-\lambda^{n-1}-\lambda^{n-2}=0 \sim \lambda^{n-2}(\lambda^2-\lambda-1)=0.$$

Доведемо, що загальний розв'язок рекурентного співвідношення (10.28) — послідовність $\{u_n\}_{n=0}^{\infty}$ з будь-якими початковими значеннями u_0, u_1 — має вигляд

$$u_n = c_1\lambda_1^n + c_2\lambda_2^n,\quad n = 0,\ 1,\ 2,\ \ldots, \qquad (10.34)$$

де постійні c_1, c_2 залежать тільки від вибору початкових значень u_0, u_1.

Дійсно, перші два співвідношення у (10.34) утворюють систему двох лінійних рівнянь відносно постійних c_1, c_2:

$$\begin{cases} 1c_1+1c_2=u_0,\\ \lambda_1c_1+\lambda_2c_2=u_1. \end{cases}$$

Детермінант цієї системи дорівнює $\Delta = \lambda_2 - \lambda_1 = -\sqrt{5} \neq 0$. Система має єдиний розв'язок

$$c_1 = -\frac{\lambda_2}{\sqrt{5}}u_0 + \frac{1}{\sqrt{5}}u_1; \quad c_2 = \frac{\lambda_1}{\sqrt{5}}u_0 - \frac{1}{\sqrt{5}}u_1.$$

Зокрема, послідовності (10.29) відповідають у формулі (10.34) сталі

$$c_1 = \frac{1}{\sqrt{5}},\ c_2 = -\frac{1}{\sqrt{5}} \qquad (u_0 = 0,\ u_1 = 1),$$

послідовності (10.30) — сталі

$$c_1 = \frac{1}{2} + \frac{3}{2\sqrt{5}},\ c_2 = \frac{1}{2} - \frac{3}{2\sqrt{5}} \qquad (u_0 = f_0 = 1,\ u_1 = f_1 = 2).$$

І нарешті, послідовності з початковими значеннями (−2), 6 відповідають сталі

$$c_1 = \frac{7}{\sqrt{5}} - 1, \quad c_2 = -\frac{7}{\sqrt{5}} - 1.$$

Асимптотична поведінка загального розв'язку $\{u_n\}$ (10.34) рекурентного співвідношення (10.28) і чисел Фібоначчі u_n (10.29) за умови $n \to \infty$ з'ясовується після зауваження $|\lambda_2| = \left|\frac{1-\sqrt{5}}{2}\right| < 1$, з якого випливає збіжність $\lambda_2^n \to 0$, $n \to \infty$. Таким чином, $u_n \sim c_1\lambda_1^n (n \to \infty)$. Зокрема, для чисел Фібоначчі (10.29)

$$u_n \sim \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n, \quad n \to \infty. \tag{10.35}$$

Більш того, число Фібоначчі u_n є найближче до $\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$ ціле число.

До чисел Фібоначчі приводить, наприклад, така задача теорії інформації. Позначимо через f_n кількість повідомлень, які можні передати за допомогою n сигналів 0 і 1 так, щоб при цьому ніде не опинилися поряд два сигнали 0. Зрозуміло, що $f_0 = 1$ (відсутність сигналу розглядається як одне «порожнє» повідомлення), $f_1 = 2$. Всі повідомлення «довжиною» $(n + 2)$ кількістю f_{n+2} можна поділити на два класи:

а) ті, в яких перший сигнал є 1; їх кількість дорівнює f_{n+1}, оскільки другий сигнал може бути як 0, так і 1;

б) ті, в яких перший сигнал є 0, і, отже, другий сигнал обов'язково є 1; їх кількість дорівнює f_n.

Отже, $f_{n+2} = f_{n+1} + f_n$, тому f_n є числами Фібоначчі (10.30).

10.7. Композиції і розбиття

Композиції

При розв'язанні задач про розподіл n однакових предметів по k непорожнім коміркам можна говорити про розкладання числа n у суму натуральних доданків n_i:

$$n = n_1 + n_2 + \ldots + n_k, \quad \text{де } k \geq 1,\ n_i \geq 1. \tag{10.36}$$

Два таких розкладання числа n вважаються різними, якщо вони відрізняються хоча б одним доданком. У такому випадку кажуть про композиції числа n. Інакше, ***композиції числа n*** — це його розкладання у вигляді (10.36), де враховуються як *величини доданків* («частин») n_i, так і *порядок їх розташування у сумі* (10.36).

Випишемо, наприклад, всі композиції числа 3:

$$3 = 3,\ 3 = 2 + 1,\ 3 = 1 + 2,\ 3 = 1 + 1 + 1.$$

Композиції із обмеженнями на кількість доданків. Число композицій $\Psi(k, n)$ числа n з k доданками дорівнює числу розподілів n однакових предметів по k різних комірках за умови відсутності порожніх комірок. Число предметів, які потрапили у комірку з номером i, дає доданок n_i. Звідси випливає, що $\Psi(k, n) = C_{n-1}^{k-1}$. Дійсно, відповідно до п. 10.4 кількість розподілів n однакових предметів між k особами (у нашому випадку — між k *комірками*) з рівнем справедливості r дорівнює $C_{n-k(r-1)-1}^{k-1}$. За нашими умовами кожна комірка одержує не менше одного предмета, оскільки $n_i > 0$. Таким чином, слід встановити «рівень справедливості» $r = 1$, і ми одержуємо число розподілів $\psi(k, n) = C_{n-1}^{k-1}$. Зрештою залишається об'єднати числа композицій $\psi(k, n)$ за всіма можливими кількостями доданків $k = 1, 2, \ldots, n$. ***Число всіх композицій числа n*** складає:

$$\sum_{k=1}^{n} \Psi(k, n) = \sum_{k=1}^{n} C_{n-1}^{k-1} = 2^{n-1}, \tag{10.37}$$

де остання рівність обґрунтовується за допомогою тотожності (10.14).

Розбиття

Визначення

Розбиттям числа n *називається його зображення у вигляді* (10.36), *якщо порядок доданків* n_i *не враховується.*

Підрахунок кількості варіантів розбиття еквівалентний задачі про кількість розподілів n однакових предметів за однаковими непорожніми комірками. Як і для композицій, доданки n_i у розбитті називаються ***частинами***; кількість входжень кожної частини у суму (10.36) називається її ***кратністю***. Розбиття числа n на k доданків $n_1, n_2, \ldots, n_k$ можна розглядати як k-сполучення (з повтореннями) цих чисел, яке задовольняє умову (10.36). Іноді для пошуку композицій числа n (або обчислення їх кількості) зручно розв'язати цю ж задачу для його розбиття, а після цього перейти до композицій.

Оскільки у розбитті (на відміну від композицій) порядок частин (доданків) не враховується, домовилися у розбитті (10.36) розташовувати доданки так, щоб вони не зростали:

$$n_1 \geq n_2 \geq \ldots \geq n_k \quad (k \geq 1). \tag{10.38}$$

Кількість всіх варіантів розбиття числа n на k частин позначимо як $p_k(n)$, а кількість всіх варіантів розбиття числа n із всіма можливими значеннями числа частин $k (1 \leq k \leq n)$ — через $p(n)$. Зрозуміло, що за правилом суми

$$p(n) = \sum_{k=1}^{n} p_k(n), \tag{10.39}$$

причому $p_1(n) = 1$, $p_n(n) = 1$, $p_k(n) = 0$ для $n < k$.

Із рівності $n = \sum_{i=1}^{k} n_i$ (10.36) формально одержуємо

$$\sum_{i=1}^{k} (n_i - 1) = n - k, \quad (n_i - 1) \geq 0. \tag{10.40}$$

Рівність (10.40) можна вважати розбиттям числа $(n - k)$ на k_0 доданків, де $k_0 = \min \{i: n_i = 1\}$ (див. (10.38)). Якщо у розбитті (10.36) $n_i > 1$ для всіх частин, тоді $k_0 = k$. За правилом суми кількість розбиття числа $(n - k)$ на частини, яких не більше ніж k штук, дорівнює $\sum_{i=1}^{k} p_i(n-k)$. Кожному такому розбиттю числа $(n - k)$ на k_0 частин

$$n - k = m_1 + m_2 + \ldots + m_{k0}, \; 1 \leq k_0 \leq k$$

однозначно відповідає розбиття n на k частин вигляду

$$n = (m_1 + 1) + (m_2 + 1) + \ldots + (m_{k_0} + 1) + \underbrace{1 + 1 + \ldots + 1}_{(k - k_0) \text{ раз}}.$$

Тому

$$p_k(n) = \sum_{i=1}^{k} p_i(n-k). \tag{10.41}$$

Це рекурентне співвідношення визначає числа $p_k(n)$ за початкових умов $p_k(k) = 1$, $p_k(n) = 0$, якщо $n < k$. Для використовування зручніше переписати (10.41) у розгорнутому вигляді

$$p_k(n) = p_k(n - k) + p_{k-1}(n - k) + \ldots + p_1(n - k). \quad (10.42)$$

Послідовно маємо: $p_1(n) = 1$, $\forall n \geq 1$;

$$p_2(n) = p_2(n - 2) + p_1(n - 2) = p_2(n - 2) + 1,\ n \geq 2.$$

Таким чином,

$$p_2(1) = 0;\ p_2(2) = 1;\ p_2(3) = p_2(1) + 1 = 1;$$
$$p_2(4) = p_2(2) + 1 = 1 + 1 = 2;\ p_2(5) = p_2(3) + 1 = 1 + 1 = 2;$$
$$p_2(6) = p_2(4) + 1 = 2 + 1 = 3;\ p_2(7) = 3;\ p_2(8) = 4;$$
$$p_2(9) = 4;\ p_2(10) = 5 \text{ і т. д.}$$

Далі

$$p_3(n) = p_3(n - 3) + p_2(n - 3) + p_1(n - 3);$$
$$p_3(n) = p_3(n - 3) + p_2(n - 3) + 1,\ n \geq 3.$$

Отже,

$$p_3(3) = 1;\ p_3(4) = p_3(1) + p_2(1) + 1 = 0 + 0 + 1;$$
$$p_3(5) = p_3(2) + p_2(2) + 1 = 0 + 1 + 1 = 2;$$
$$p_3(6) = p_3(3) + p_2(3) + 1 = 1 + 1 + 1 = 3 \text{ і т. д.}$$

Значення числа варіантів розбиття $p_k(n)$ для $1 \leq k,\ n \leq 10$ наведено у таблиці 10.1.

Таблиця 10.1. Значення $p_k(n)$, $p(n)$

k \ n	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	2	2	3	3	4	4	5
3	0	0	1	1	2	3	4	5	7	8
4	0	0	0	1	1	2	3	5	6	9
5	0	0	0	0	1	1	2	3	5	7
6	0	0	0	0	0	1	1	2	3	5
7	0	0	0	0	0	0	1	1	2	3
8	0	0	0	0	0	0	0	1	1	2
9	0	0	0	0	0	0	0	0	1	1
10	0	0	0	0	0	0	0	0	0	1
$p(n)=\sum_k p_k(n)$	1	2	3	5	7	11	15	22	30	42

Сумуючи числа $p_k(n)$ в n-му стовпчику, ми одержуємо кількість варіантів розбиття $p(n)$ числа n, або кількість розподілів n однакових предметів за n однаковими комірками, частина яких може виявитися порожніми.

Властивості чисел розбиття $p(n)$, $p_k(n)$ вивчалися з середніх віків; вони важливі як для комбінаторики, так і для теорії чисел. Перші поглиблені результати для них одержав Л. Ейлер, і далі до останнього часу — К. Гаусс, Феррер, Дж. Сильвестр, П. Макмагон, Г. Харді, С. Рамануджан, І. Шур, Г. Радемахер, Г. Ендрюс та ін. Однією з нетривіальних задач є характеристика зростання числа $p(n)$ за великих $n \to \infty$. В нижньому рядку таблиці 10.1 наведено значення $p(1)$, $p(2)$, ..., $p(10)$. Зауважимо, що швидкість зростання $p(n)$ тим вище, чим більше n: якщо $p(1) = 1$, $p(5) = 7$, $p(10) = 42$, то $p(20) = 627$, $p(30) = 5604$, $p(40) = 37\,338$, $p(50) = 204\,226$, $p(60) = 966\,467$, $p(70) = 4\,087\,968$, $p(80) = 15\,796\,476$, $p(90) = 56\,634\,173$, $p(100) = 190\,569\,292$, $p(200) = 3\,972\,999\,029\,388$.

Асимптотично при $n \to \infty$ послідовність чисел $p(n)$ ($n = 1, 2, 3$ ...) еквівалентна такій послідовності чисел g_n:

$$g_n = \frac{1}{4\sqrt{3}} e^{\pi\sqrt{\frac{2}{3}}\sqrt{n}} \sim p(n): \quad \lim_{n\to\infty} \frac{p(n)}{g_n} = 1. \tag{10.42}$$

Таким чином, число $\frac{1}{4\sqrt{3}}(e^{\pi\sqrt{\frac{2}{3}}})^{\sqrt{n}}$ є наближене значення натурального числа $p(n)$ для великого n. Більш точні наближення для $p(n)$ знайшли Г. Харді і С. Рамануджан (індійський математичний геній-самоучка): урахування всього 8 членів у їх формулі наближеного обчислення значення $p(200) = 3\,972\,999\,029\,388$ — дуже великого числа — має помилку всього 0,004.

Більшість якісних висновків відносно чисел $p_k(n)$, $p(n)$ можливо одержати за допомогою наочного зображення розбиття, що назване діаграмою Ферре (або Феррера).

Діаграма Ферре розбиття (10.36) є n точок, розташованих рядками один під одним так, що перший рядок містить n_1 точок, другий рядок — n_2 точок, ..., і зрештою k-й рядок містить n_k точок. Наприклад, розбиття

$$15 = 5 + 5 + 3 + 2 \tag{10.43}$$

відображується діаграмою Ферре на рис. 10.7.

Кожному розбиттю числа n однозначно відповідає спряжене розбиття цього числа, яке визначається транспозицією діаграми Ферре (перетворенням рядків у стовпчики і стовпчиків у рядки). Наприклад, транспозиція діаграми рис. 10.7 є діаграма рис. 10.8, яка визначає розбиття

$$15 = 4 + 4 + 3 + 2 + 2. \tag{10.44}$$

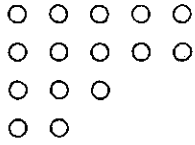


Рис. 10.7. Діаграма Ферре розбиття (10.43)

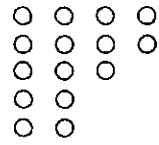


Рис. 10.8. Діаграма Ферре розбиття (10.44)

Зрозуміло, що для розбиття на k частин спряжене розбиття складається з частин, які не перевершують число k, і хоча б одна частина дорівнює k (для розбиття (10.43) $k = 4$).

Це спостереження доводить таке твердження.

Теорема 10.1

Кількість варіантів розбиття числа n на k частин дорівнює кількості варіантів розбиття числа n з найбільшою частиною, яка дорівнює k.

Завдання

Довести, що кількість варіантів розбиття числа n, в яких частини не перевершують k (можливо, не дорівнюючись k жодного разу), співпадає з кількістю варіантів розбиття числа $n + k$ на k частин (будь-яких), тобто співпадає з $p_k^{(n+k)}$.

За допомогою діаграм Ферре доводяться такі цікаві твердження.

Теорема 10.2

Кількість варіантів розбиття числа n на частини, з яких кожні дві різні, дорівнює кількості варіантів розбиття числа n на непарні частини.

Теорема 10.3

*Кількість варіантів самоспряженого розбиття числа n (**розбиття самоспряжене**, якщо воно співпадає зі своїм спряженням) дорівнює кількості варіантів такого розбиття числа n на непарні частини, в яких кожні дві частини різні.*

Теорема 10.4 ***(Л. Ейлер)***

Кількість варіантів розбиття числа n на не більш ніж k частин дорівнює кількості варіантів розбиття числа $n + \frac{k(k+1)}{2}$ на k нерівних частин.

Теорема 10.5

Кількість варіантів розбиття числа n на непарні частини дорівнює кількості варіантів розбиття числа n, в яких кожний доданок, крім найбільшого, присутній парну кількість разів, а найбільший доданок — непарну кількість разів.

10.8. Продуктивні функції

Поняття про продуктивні функції

Нехай дано деяку послідовність чисел $a_0, a_1, a_2, ..., a_n, ...$ Якщо степеневий ряд

$$a_0 + a_1x + a_2x^2 + ... + a_nx^n + ... \qquad (10.45)$$

співпадає за деяких x з функцією $f(x)$, то $f(x)$ називається ***продуктивною функцією*** для послідовності $a_0, a_1, a_2, ..., a_n, ...$.

З послідовністю $a_0, a_1, a_2, ..., a_n, ...$ можна зв'язати ще один степеневий ряд:

$$a_0 + a_1x + a_2\frac{x^2}{2!} + ... + a_n\frac{x^n}{n!} + ... \qquad (10.46)$$

Якщо цей ряд співпадає з функцією $\varphi(x)$, то $\varphi(x)$ називається ***експоненціальною продуктивною функцією*** для послідовності $a_0, a_1, a_2, ..., a_n, ...$ Ця назва пояснюється тим, що для послідовності $1, a, a^2, ... a^n, ...$ експоненціальною продуктивною функцією (10.46) є експонента e^{ax}:

$$e^{ax} = 1 + ax + \frac{a^2x^2}{2!} + ... + \frac{a^nx^n}{n!} + ... \qquad (10.47)$$

Для тієї ж послідовності звичайна продуктивна функція є $\frac{1}{1-ax}$:

$$(1 - ax)^{-1} = 1 + ax + a^2x^2 + ... + a^nx^n + ... \qquad (10.48)$$

Дійсно, якщо $q = ax$, $|q| < 1$, то є $|x| < \frac{1}{|a|}$, тоді за формулою суми нескінченно спадної геометричної прогресії (10.48) одержуємо:

$$\frac{1}{1-q} = 1 + q + q^2 + ... = 1 + ax + a^2x^2 + ...$$

Зокрема, якщо $a = 1$, тоді

$$\frac{1}{1-x} = 1 + x + x^2 + ... + x^n + ..., \quad |x| < 1. \qquad (10.49)$$

Диференціювання (10.49) приводить до рівності

$$\frac{1}{(1-x)^2} = 1 + 2x + 3x^2 + ... + nx^{n-1} + ..., \quad |x| < 1, \qquad (10.50)$$

а множення (10.50) на x дає

$$\frac{x}{(1-x)^2} = x + 2x^2 + 3x^3 + \ldots + nx^n + \ldots, \quad |x| < 1. \qquad (10.51).$$

Таким чином, $\frac{1}{1-x}$ є продуктивна функція для послідовності 1, 1, 1, 1, ... (10.49), $\frac{1}{(1-x)^2}$ — продуктивна функція для послідовності чисел натурального ряду 1, 2, 3, 4, ... (10.50), $\frac{x}{(1-x)^2}$ — продуктивна функція для послідовності 0, 1, 2, 3, ... (10.51).

Функцію $f(x)$, яка має похідні як завгодно високого порядку при $t = 0$, можна вважати експоненціальною продуктивною функцією для послідовності своїх похідних $f(0)$, $f'(0)$, $f''(0)$, ..., $f^{(n)}(0)$.... Дійсно, ми маємо розкладання у ряд Тейлора:

$$f(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2!} + \ldots + f^{(n)}(0)\frac{x^n}{n!} + \ldots$$

Розглянемо продуктивні функції для послідовностей, зв'язаних з комбінаторними задачами.

Операції над послідовностями та продуктивними функціями

Завдяки останнім прикладам ми можемо припустити, що операціям над продуктивними функціями повинні відповідати певні операції над відповідними послідовностями. Дійсно, нехай $f(x)$ є продуктивна функція послідовності a_0, a_1, a_2, ..., і далі $g(x)$ — продуктивна функція послідовності b_0, b_1, b_2, ..., тобто

$$f(x) = \sum_{k=0}^{\infty} a_k x^k, \quad g(x) = \sum_{k=0}^{\infty} b_k x^k.$$

Тоді лінійна комбінація функцій $\alpha f(x) + \beta g(x) = \varphi(x)$ є продуктивна функція послідовності $\{c_k = \alpha a_k + \beta b_k\}_{k=0}^{\infty}$, яка одержується як відповідна лінійна комбінація послідовностей $\{a_k\}$, $\{b_k\}$:

$$\varphi(x) = \sum_{k=0}^{\infty} c_k x^k.$$

Позначений зв'язок між операціями у класі продуктивних функцій і в класі послідовностей символічно можна записати так: якщо $f(x) \sim \{a_k\}$, $g(x) \sim \{b_k\}$, тоді

$$\alpha f(x) + \beta g(x) \sim \alpha\{a_k\} + \beta\{b_k\}. \qquad (10.52).$$

Введемо для послідовностей бінарну операцію згортки за правилом:

$$\{a_k\} * \{b_k\}_0^\infty = \{d_k\}_{k=0}^\infty, \qquad d_k = a_0 b_k + a_1 b_{k-1} + \ldots + a_k b_0.$$

Із правила множення степеневих рядів маємо, що згортці двох послідовностей відповідає добуток їх продуктивних функцій:

$$f(x)\ g(x) \sim \{a_k\} * \{b_k\}. \qquad (10.53).$$

Відповідності (10.52), (10.53) між послідовностями та їх продуктивними функціями виявляються ефективними і корисними у комбінаторних обчисленнях. Як зразок наведемо формулу для суми квадратів $1 + 2^2 + \ldots + n^n$ перших n чисел натурального ряду.

Нескінченна послідовність квадратів $\{n^2\}_{n=1}^\infty$ має продуктивну функцію $\dfrac{d}{dx}\dfrac{x}{(1-x)^2}$; в цьому можна переконатися за допомогою диференціювання рівності (10.51). Тому функція $x\dfrac{d}{dx}\dfrac{x}{(1-x)^2}$ відповідає послідовності $\{n^2\}_{n=0}^\infty$ з першим членом 0, оскільки, якщо $f(x) \sim \{a_0, a_1, a_2, \ldots\}$, то $xf(x) \sim \{0, a_0, a_1, a_2, \ldots\}$. Оскільки $\dfrac{1}{1-x} \sim \{1, 1, 1, \ldots\}$ (див. (10.49)), то

$$\frac{1}{1-x}x\frac{d}{dx}\frac{1}{(1-x)^2} \sim \{1,1,1,\ldots\} * \{n^2\}_{n=0}^\infty = \left\{\alpha_n = \sum_{k=0}^{n} k^2\right\}_{n=0}^\infty.$$

Таким чином, шукані числа $\alpha_n = 1 + 2^2 + \ldots + n^2$ повинні бути коефіцієнтами степеневого ряду для функції

$$\frac{x}{1-x}\frac{d}{dx}\frac{x}{(1-x)^2} = \frac{x(x+1)}{(1-x)^4} = \sum_{n=0}^{\infty}\alpha_n x^n, \quad \alpha_0 = 0.$$

Спочатку одержимо розкладання функції $\dfrac{1}{(1-x)^4}$ в степеневий ряд. Коефіцієнт ряду дорівнює

$$a_k = \frac{1\cdot d^k}{k!dx^k}(1-x)^{-4}\bigg|_{x=0} = \frac{1}{k!}4(4+1)\cdot\ldots\cdot(4+k-1)(1-x)^{-4-k}\bigg|_{x=0} =$$

$$= \frac{1}{k!}[4]^k(1-x)^{-4-k}\big|_{x=0} = \frac{[4]^k}{k!},$$

де використані позначення

$$[m]^k = m(m+1)\ldots(m+k-1). \qquad (10.54)$$

Звідси одержуємо послідовно

$$\frac{x(x+1)}{(1-x)^4} = (x^2+x)(1-x)^{-4} = (x+x^2)\sum_{k=0}^{\infty}\frac{[4]^k}{k!}x^k =$$

$$= \sum_{n=1}^{\infty}\frac{[4]^{n-1}}{(n-1)!}x^n + \sum_{n=2}^{\infty}\frac{[4]^{n-2}}{(n-2)!}x^n = \sum_{n=1}^{\infty}\alpha_n x^n,$$

де

$$\alpha_1 = \frac{[4]^0}{0!} = 1; \quad \alpha_n = \frac{[4]^{n-1}}{(n-1)!} + \frac{[4]^{n-2}}{(n-2)!},\ n \geq 2.$$

Враховуючи значення $[4]^{n-1}$, $[4]^{n-2}$ згідно з (10.54), відразу одержуємо, що

$$\alpha_n = \frac{n(n+1)(2n+1)}{1\cdot 2\cdot 3}.$$

Оскільки α_n є сума квадратів, шукана формула має вигляд:

$$1^2 + 2^2 + \ldots + n^2 = \frac{1}{6}n(n+1)(2n+1). \qquad (10.55)$$

Зауваження. Згортка довільної послідовності $\{a_n\}_0^\infty$ з послідовністю {1, 1, 1, ...} дорівнює $\{a_n\}_0^\infty * \{1, 1, 1, \ldots\} = \{a_0, a_0 + a_1, a_0 + a_1 + a_2, \ldots\}$, тому послідовність {1, 1, 1, ...} носить назву ***суматора.*** Нагадаємо, що суматор має продуктивну функцію $\frac{1}{1-x}$ (10.49) і що властивостями суматора ми скористалися вище для підсумовування квадратів $1^2 + 2^2 + \ldots + n^2$. При підсумовуванні скінченних послідовностей $\{b_0, b_1, b_2, \ldots, b_n, 0, 0, 0 \ldots\}$ елементи згортки стабілізуються, починаючи з визначеного номера. Зокрема, для функції

$$(1 - 2x^3 + 3x^7 + 4x^{25} + 5x^{45})\frac{1}{1-x}$$

коефіцієнти степеневого ряду при x^{45}, x^{46}, ... дорівнюють числу 11 (= 1 − 2 + 3 + 4 + 5), коефіцієнти при x^7, x^8, ..., x^{24} дорівнюють 2 (= 1 − 2 + 3).

Продуктивні функції сполучень

Знайдемо продуктивну функцію для послідовності

$$\{C_n^0, C_n^1, C_n^2, \ldots, C_n^n, 0, 0, 0, \ldots\}. \qquad (10.56)$$

Утворимо для змінних x, x_1, x_2, ..., x_n многочлен

$$(1 + x_1x)(1 + x_2x)...(1 + x_nx) = 1 + (x_1 + x_2 + ... + x_n)x +$$
$$+ (x_1x_2 + x_1x_3 + ... + x_{n-1}x_n)x^2 + ... + (x_1x_2 ... x_n)x^n. \quad (10.57)$$

Зрозуміло, що коефіцієнт перед $x^k (0 \le k \le n)$ має вигляд суми всіх k-добутків з елементів n-множини $X = \{x_1, x_2, ..., x_n\}$. Оскільки будь-який k-добуток у сумі взаємно однозначно відповідає визначеному k-сполученню з n-множини X, то кількість доданків (k-добутків) у сумі перед x^k дорівнює саме кількості k-сполучень (без повторень) з n-множини, тобто C_n^k.

Вираз (10.57) містить явний перелік всіх індивідуальних k-сполучень. Але якщо нам треба зафіксувати їх загальну кількість C_n^k, покладемо у (10.57) $x_1 = x_2 = ... x_n = 1$ і одержимо продуктивну функцію для послідовності чисел сполучень (10.7):

$$(1 + x)^n = \sum_{k=0}^{n} C_n^k x^k. \quad (10.58)$$

Безперечно, це інший погляд на добре відому біноміальну формулу. Залишаючись на новому шляху, диференціюємо рівність (10.58) k разів, покладемо після цього $x = 0$ і одержимо:

$n(n - 1) ... (n - k + 1) = C_n^k k(k - 1)... \cdot 2 \cdot 1$, звідки

$$C_n^k = \frac{n(n-1)...(n-k+1)}{k!} = \frac{n!}{k!(n-k)!} \text{ (див. (10.7) п. 10.3).}$$

Одержимо тепер продуктивну функцію для послідовності

$$\left\{\overline{C_n^0}, \overline{C_n^1}, \overline{C_n^2}, ..., \overline{C_n^k}, ..., \overline{C_n^{n+1}}, ...\right\}, \quad (10.59)$$

де $\overline{C_n^k}$ — кількість k-сполучень з повтореннями без обмежень з елементів n типів. Узагальнюючи багаточлен (10.57), для зображення індивідуальних k-сполучень з n елементів з повтореннями утворимо n-добуток формальних рядів з такими ж змінними x, x_1, x_2, ..., x_n:

$$(1 + x_1x + x_1^2x^2 + ...)(1 + x_2x + x_2^2x^2 + ...)...(1 + x_nx + x_n^2x^2 + ...) =$$
$$= 1 + (x_1 + x_2 + ... + x_n)x + (x_1x_2 + ... + x_{n-1}x_n + x_1^2 + x_2^2 + ... + x_n^2)x^2 + ... +$$
$$+ (x_1x_2...x_k + x_1^2x_2...x_{k-1} + ... + x_n^k)x^k + ... \quad (10.60)$$

В сумі, яка виділена дужками перед x^k, кожний доданок (k-добуток) зображує відповідне індивідуальне k-сполучення

з повтореннями. Щоб одержати загальну кількість таких k-сполучень $\overline{C_n^k}$, покладемо у (10.60) $x_1 = x_2 = \ldots = x_n = 1$:

$$(1+x+x^2+\ldots)^n = \sum_{k=0}^{\infty} \overline{C_n^k} x^k, \quad \overline{C_n^0} = 1.$$

Замінимо ряд $\sum x^k$ його сумою (10.49) і одержимо:

$$\frac{1}{(1-x)^n} = \sum_{k=0}^{\infty} \overline{C_n^k} x^k, \quad |x| < 1. \tag{10.61}$$

Таким чином, продуктивною для послідовності (10.59) є функція $(1-x)^{-n}$. За загальним правилом k-й коефіцієнт $\overline{C_n^k}$ степеневого ряду (10.61) може бути обчислений диференціюванням лівої частини:

$$\overline{C_n^k} = \frac{1}{k!}\frac{d^k}{dx^k}(1-x)^{-n}\Big|_{x=0} = \frac{(-n)(-n-1)\ldots(-n-k+1)(-1)^k}{k!},$$

$$\overline{C_n^k} = \frac{n(n+1)\ldots(n+k-1)}{k!} = \frac{(n+k-1)!}{(n-1)!k!} = C_{n+k-1}^k.$$

Вдруге одержаний вираз $\overline{C_n^k} = C_{n+k-1}^k$ для кількості k-сполучень з повтореннями через кількість k-сполучень без повторень, створюваних на поширеній множині елементів.

Аналогічно будуються продуктивні функції для сполучень, які задовольняють різні обмеження на використання будь-якого з елементів — індивідуальним або загальним.

Наприклад, якщо будь-який з n елементів може повторюватися у кожному сполученні не більше ніж m разів, тоді розглядається n-добуток багаточленів

$$(1+x_1x+\ldots+x_1^mx^m)(1+x_2x+\ldots+x_2^mx^m)\ldots(1+x_nx+\ldots+x_n^mx^m) =$$

$$= \sum_{k=0}^{mn}(x_1^{k_1}x_2^{k_2}\ldots x^{k_n}+\ldots)x^k;$$

$$0 \le k_i \le m, \quad k_1 + k_2 + \ldots + k_n = k.$$

Покладемо тут $x_1 = x_2 = \ldots = x_n = 1$:

$$(1+x+x^2+\ldots+x^m)^n = \sum_{k\ge 0}^{mn} C(n, k; m)x^k, \tag{10.62}$$

де через $C(n; k; m)$ позначена кількість таких k-сполучень з повтореннями з елементів n типів, в яких кожний елемент може бути використаний (повторений) від 0 до m разів. Продуктивна функція таких сполучень є ліва частина (10.62).

Продуктивні функції розміщень та перестановок

Для індивідуального переліку *упорядкованих вибірок* (перестановок, розміщень) більш зручною частіше виявляється *експоненціальна продуктивна функція*, оскільки звичайна (степенева) функція трактує добутки x_1x_2 і x_2x_1 як два однакових.

Біноміальну формулу (10.58), де $C_n^k = \frac{A_n^k}{k!}$, тобто

$$(1+x)^n = \sum_{k=0}^{n} A_n^k \cdot \frac{x^k}{k!}, \qquad (10.63)$$

одночасно можна вважати розкладанням у ряд експоненціальної продуктивної функції $(1 + x)^n$ для послідовності $\{A_n^0, A_n^1, \ldots, A_n^n; 0, 0, 0, \ldots\}$ чисел k-розміщень (або k-перестановок) n-множини. Тут маються на увазі k-перестановки без повторень.

Якщо будь-який з n елементів у k-перестановці може повторюватися від 0 до m разів, тоді експоненціальна продуктивна функція для чисел $C_e(n, k; m)$ таких k-перестановок має вигляд:

$$\left(1 + x + \frac{x^2}{2!} + \ldots + \frac{x^m}{m!}\right)^n = \sum_{k=0}^{mn} C_e(n, k; m)\frac{x^k}{k!}.$$

Припустимо, у k-перестановці можна використовувати (повторювати) елемент i-го типу не більше ніж m_i разів, $i = 1, 2, \ldots, n$. Експоненціальна продуктивна функція таких перестановок

$$\prod_{i=1}^{n}\left(1 + x + \frac{x^2}{2!} + \ldots + \frac{x^{m_i}}{m_i!}\right) —$$

багаточлен степеня $(m_1 + m_2 + \ldots + m_n)$.

Завдання

1. Розглянути k-перестановки з необмеженими повтореннями із n різних типів елементів. Чи правильна наведена формула для експоненціальної продуктивної функції таких перестановок, і як інтерпретуються коефіцієнти n^k у правій частині:

$$\left(1 + x + \frac{x^2}{2!} + \ldots\right)^n = \sum_{k=0}^{\infty} n^k \cdot \frac{x^k}{k!}.$$

Інакше кажучи, чи є ліва частина формули експоненціальною продуктивною функцією для послідовності $\{n^k\}_{k=0}^{\infty}$?

2. Перевірити, що функція $\frac{1}{\ln(1-x)}$ є продуктивною (звичайною) для послідовності

$$\left\{\frac{1}{k+1}\right\}_{k=0}^{\infty} = \left\{1, \frac{1}{2}, \frac{1}{3}, \ldots\right\}.$$

3. Перевірити, що

$$\frac{1}{k!}(e^x - 1)^k = \sum_{n=k}^{\infty} S_n^k \cdot \frac{x^n}{n!},$$

тобто $\dfrac{(e^x-1)^k}{k!}$ — експоненціальна продуктивна функція для чисел Стірлінга другого роду $\{S_n^k\}_{k=0}^{\infty}$, де $S_n^k = 0$, якщо $n < k$ (див. п. 10.4).

Продуктивна функції для розбиття чисел

Розбиття (10.36) числа n на частини n_i

$$n = n_1 + n_2 + \ldots + n_k$$

іноді задається за допомогою ***векторної специфікації***

$$\delta = (\delta_1, \delta_2, \ldots, \delta_n), \quad 0 \le \delta_i \le n, \tag{10.64}$$

компонента якої δ_i дорівнює *кількості входжень натурального числа i в розбиття* (10.36), $i = 1, 2, \ldots, n$. Інакше кажучи, невід'ємні цілі числа δ_i утворюють специфікацію (10.64) деякого розбиття числа n, якщо

$$n = 1 \cdot \delta_1 + 2 \cdot \delta_2 + \ldots + n \cdot \delta_n = \sum_{i=1}^{n} i \cdot \delta_i. \tag{10.65}$$

Зрозуміло, що кількість k частин розбиття (10.36) дорівнює сумі компонент специфікації:

$$\delta_1 + \delta_2 + \ldots + \delta_n = k, \quad 1 \le k \le n. \tag{10.66}$$

Частина компонент δ_i складається з нулів. Використовується також ***мультиплікативний запис специфікації***:

$$(1^{\delta_1}, 2^{\delta_2}, \ldots, n^{\delta_n}). \tag{10.67}$$

Він добре узгоджується із записом розбиття у вигляді «зведеної» суми (10.65). На відміну від векторної специфікації (10.64), мультиплікативний запис можна скорочувати за рахунок пар i^{δ_i}, у яких $\delta_i = 0$. Наприклад, для розбиття $7 = 2 + 2 + 1 + 1 + 1$ наведене розбивання (10.65) є $7 = 1 \cdot 3 + 2 \cdot 2 = 1 \cdot \delta_1 + 2 \cdot \delta_2$, специфікація (10.64) має вигляд $\delta = (3, 2, 0, 0, 0, 0, 0)$, а мультиплікативний запис специфікації більш зручний: $(1^3, 2^2)$.

Відповідно для розбиття $7 = 5 + 1 + 1$ маємо $\delta = (2, 0, 0, 0, 1, 0, 0)$ и $(1^2, 5^1)$. Якщо $7 = 7$, тоді $\delta = (0, 0, 0, 0, 0, 0, 1)$ і мультиплікативний запис є (7^1).

Нагадаємо, що $p_k(n)$ позначає кількість варіантів k-розбиття числа n (розбиття (10.36) з k членами), $p(n)$ — кількість варіантів будь-якого розбиття числа n. Через $p(n, n_i \le r)$ позначимо кількість варіантів розбиття числа n на частини (доданки) n_i, які *не перевершують* r. Маючи на увазі тривіальне «розбиття» $0 = 0$ з тривіальною «специфікацією» $\delta_i = 0$ $(\forall i)$, домовимося, що $p(0, n_i \le r) = 1$. Розглянемо формальний добуток двох рядів

$$(1 + x^{1\cdot 1} + x^{1\cdot 2} + x^{1\cdot 3} + x^{1\cdot 4} + \ldots)(1 + x^{2\cdot 1} + x^{2\cdot 2} + x^{2\cdot 3} + \ldots) =$$
$$= 1 + 1\cdot(x^{1\cdot 1}) + (x^{1\cdot 2} + x^{2\cdot 1}) + (x^{1\cdot 3} + x^{1\cdot 1 + 2\cdot 1}) + (x^{1\cdot 4} + x^{2\cdot 2} +$$
$$+ x^{1\cdot 2 + 2\cdot 1}) + \ldots = 1 + 1\cdot x + 2x^2 + 2x^3 + 3x^4 + 3x^5 + \ldots =$$
$$= p(0, n_i \le 2) + p(1, n_i \le 2)\cdot x + p(2, n_i \le 2)\cdot x^2 +$$
$$+ p(3, n_i \le 2)\cdot x^3 + p(4, n_i \le 2)\cdot x^4 + \ldots + p(n, n_i \le 2)\cdot x^n + \ldots$$

Дійсно, (1^1) — єдина мультиплікативна специфікація розбиття числа 1; (1^2) і (2^1) — всі мультиплікативні специфікації розбиття числа 2; (1^3) і $(1^1, 2^1)$ — специфікації всіх варіантів розбиття числа 3 виду $3 = 1 + 1 + 1$, $3 = 2 + 1$, таких, що їх частини не перевершують числа $r = 2$; (1^4), (2^2), $(1^2, 2^1)$ — специфікації всіх тих варіантів розбиття числа 4 виду $4 = 1 + 1 + 1 + 1$, $4 = 2 + 2$, $4 = 2 + 1 + 1$, частини яких не перевершують числа 2. Тому сума $f(x)$ ряду

$$1 + 1\cdot x + 2x^2 + 2x^3 + 3x^4 + 3x^5 + \ldots = \sum_{n=0}^{\infty} p(n, n_i \le 2)\cdot x^n = f(x)$$

повинна бути продуктивною функцією послідовності

$$(1, 1, 2, 2, 3, 3, \ldots) = \{p(0, n_i \le 2),\ p(1, n_i \le 2),$$
$$p(2, n_i \le 2),\ \ldots,\ p(n, n_i \le 2),\ \ldots\}. \qquad (10.68)$$

Зауважимо, що

$$\frac{1}{1 - x^2} = 1 + x^{2\cdot 1} + x^{2\cdot 2} + x^{2\cdot 3} + \ldots,\ |x| < 1$$

є продуктивна функція послідовності

$$(1, 0, 1, 0, 1, 0, 1, 0, \ldots).$$

Оскільки згортка цієї послідовності із суматором

$$(1, 1, 1, 1, \ldots) * (1, 0, 1, 0, 1, 0, \ldots)$$

співпадає з послідовністю (10.68), то послідовність (10.68) має продуктивну функцію

$$f(x)=\frac{1}{(1-x)(1-x^2)}=\sum_{n=0}^{\infty}p(n,\ n_i\le 2)\cdot x^n.$$

Для довільного натурального r, яке обмежує частини розбиття, відповідно маємо продуктивну функцію

$$\frac{1}{(1-x^1)(1-x^2)\ldots(1-x^r)}=\sum_{n=0}^{\infty}p(n,\ n_i\le r)\cdot x^n. \qquad (10.69)$$

За допомогою символу добутку продуктивну функцію (10.69) записують як

$$\prod_{k=1}^{r}\frac{1}{1-x^k}.$$

Якщо у (10.69) $r\to\infty$, тоді обмеження на частини розбиття n_i зникає, і слід очікувати, що

$$\prod_{k=1}^{\infty}\frac{1}{1-x^k}=\sum_{n=0}^{\infty}p(n)\cdot x^n \qquad (10.70)$$

є продуктивна функція для послідовності чисел розбиття $\{p(n)\}_{n=1}^{\infty}$. Це твердження складає вміст відомої теореми комбінаторного аналізу. Безпосереднє доведення формули (10.70) одержується шляхом послідовного запису членів степеневого ряду, який формально дорівнює нескінченному добутку

$$\prod_{k=1}^{\infty}(1+x^{k\cdot 1}+x^{k\cdot 2}+x^{k\cdot 3}+\ldots)=1+a_1\cdot x+a_2\cdot x^2+\ldots$$

рядів $\frac{1}{1-x^k}=1+x^{k\cdot 1}+x^{k\cdot 2}+x^{k\cdot 3}+\ldots$, $|x|<1$, аналогічно тому, як це було зроблено вище для добутку двох рядів.

Завдання

1. Через $p(n,\ dif)$ позначимо кількість варіантів розбиття числа n на попарно різні частини.
   Перевірити, що продуктивна функція послідовності $\{p(n,\ dif)\}_{n=0}^{\infty}$ є:
   $$R(x)\equiv(1+x)(1+x^2)(1+x^3)\ldots(1+x^k)\ldots=\sum_{n=0}^{\infty}p(n,dif)x^n.$$
2. Продуктивна функція для чисел розбиття на непарні частини дорівнює:
   $$N(x)\equiv\frac{1}{1-x}\cdot\frac{1}{1-x^3}\cdot\frac{1}{1-x^5}\cdot\ldots\cdot\frac{1}{(1-x^{2k-1})}\cdot\ldots$$

Зауваження. Продуктивні функції $R(x)$ і $N(x)$ співпадають. Дійсно, скористаємося співвідношенням:

$$1 + x^k = \frac{1 - x^{2k}}{1 - x^k}, \quad k = 1, 2, \ldots$$

Маємо

$$R(x) = \frac{1 - x^2}{1 - x} \cdot \frac{1 - x^4}{1 - x^2} \cdot \frac{1 - x^6}{1 - x^3} \cdot \ldots = \frac{1}{1 - x} \cdot \frac{1}{1 - x^3} \cdot \ldots = N(x).$$

Таким чином, ми одержуємо доведення теореми 10.2 за допомогою продуктивних функцій.

10.9. Асимптотичні оцінки та формули

Взагалі комбінаторні характеристики (функції) залежать від числа n елементів початкової множини, з якої здійснюються комбінаторні вибірки. Асимптотичні оцінки та формули характеризують значення відповідної комбінаторної функції при дуже великих значеннях n, як, наприклад, співвідношення (10.35) для чисел Фібоначчі u_n.

При запису асимптотичних оцінок та формул використовуються такі позначення:

1) $f(x) = o(g(x))$ при $x \to \infty$ (читається «є $f(x)$ o-мале від $g(x)$») тоді і тільки тоді, якщо $\lim\limits_{x \to \infty} |f(x)/g(x)| = 0$;
2) $f(x) \sim g(x)$ при $x \to \infty$ (читається «$f(x)$ і $g(x)$ асимптотично рівні» або еквівалентні) тоді і тільки тоді, коли $\lim\limits_{x \to \infty} |f(x)/g(x)| = 1$;
3) $f(x) = O(g(x))$ при $x \to \infty$ (читається «є $f(x)$ O-велике від $g(x)$») тоді і тільки тоді, якщо існують константи $C > 0$, $R > 0$, такі, що $|f(x)| \le C|g(x)|$ при $|x| > R$.

Визначенням еквівалентності $\sim$ ми вже користувалися для асимптотичної оцінки (10.35) чисел Фібоначчі u_n.

Звернемося до формули кількості D_n зміщень — переставлень із n елементів, у яких жодний елемент не залишається на первинному місці:

$$D_n = n!\left(1 - \frac{1}{1!} + \frac{1}{2!} + \ldots + \frac{(-1)^n}{n!}\right).$$

Порівняємо D_n з величиною $n!e^{-1}$, скориставшись розкладанням у ряд

$$e^{-1} = 1 - \frac{1}{1!} + \frac{1}{2!} + \ldots + \frac{(-1)^n}{n!} + \ldots$$

і оцінкою $|a_1 - a_2 + a_3 - a_4 + ...| < |a_1|$ для суми збіжного знакозмінного ряду:

$$\left|D_n - n!e^{-1}\right| = \left|(-1)^{n+1} n!\left(\frac{1}{(n+1)!} - \frac{1}{(n+2)!} + \ldots\right)\right| =$$

$$= \left|\frac{1}{n+1} - \frac{1}{(n+1)(n+2)} + \ldots\right| \le \frac{1}{n+1} < \frac{1}{n}.$$

Таким чином, справедливі асимптотичні формули

$$D_n = n!e^{-1} + O(\tfrac{1}{n}), \text{ або } D_n \sim \frac{n!}{e}, \ n \to \infty.$$

Для довідки далі наведено деякі комбінаторні асимптотичні формули:

$$\sum_{k=1}^{n} \frac{1}{k} = \ln n + O(1), \quad n \to \infty. \tag{10.71}$$

$$\ln(n!) = \sum_{k=1}^{n} \ln k = n\ln n - n + O(\ln n), \quad n \to \infty. \tag{10.72}$$

$$\ln(n!) = (n+\frac{1}{2})\ln n - n + \ln\sqrt{2\pi} + \frac{\theta}{12n}, \ 0 < \theta < 1, \ n \to \infty. \tag{10.73}$$

Формула (10.73) точніша за формулу (10.72). Скориставшись тотожністю $n! = e^{\ln n!}$, одержуємо еквівалентне формулі (10.73) співвідношення (10.74):

$$n! = \sqrt{2\pi n} \cdot (\tfrac{n}{e})^n \cdot e^{\frac{\theta}{12n}}, \quad 0 < \theta < 1, \ n \to \infty. \tag{10.74}$$

Факторіальна функція $\Gamma(n+1) = n!$ дуже важлива, тому для неї одержано багато апроксимацій і формул наближення. Формула (10.74) часто носить назву ***формули Стірлінга.*** Наведемо ще більш точну ***«апроксимацію Стірлінга»***:

$$n! = \sqrt{2\pi n} \cdot (\tfrac{n}{e})^n \cdot e^{\frac{1}{12n} - \frac{1}{360n^3} + \frac{q}{1260n^5}}, \quad 0 < q < 1. \tag{10.75}$$

Якщо у формулі (10.75) відкинути відповідні доданки у показнику степеня експоненти, відразу одержимо досить тонкі оцінки значень факторіальної функції при $n \to \infty$:

$$\sqrt{2\pi n} \cdot (\tfrac{n}{e})^n \cdot e^{\frac{1}{12n} - \frac{1}{360n^3}} \le n! \le \sqrt{2\pi n} \cdot (\tfrac{n}{e})^n \cdot e^{\frac{1}{12n}}. \tag{10.76}$$

Для чисел Стірлінга другого роду S_n^k (10.21) виконується асимптотична формула:

$$S_n^k \sim \frac{k^n}{k!}, \quad n \to \infty. \tag{10.77}$$

На завершення нагадаємо, що асимптотична поведінка числа $p(n)$ всіх варіантів розбиття числа n за умови $n \to \infty$ розглядалася вище у п. 10.7 (див. (10.42)).

Завдання

1. Одержати формулу (10.74), виходячи з формули (10.75) або оцінок (10.76).
2. Одержати формулу (10.72) як наслідок формули (10.73).
3. Довести, що з будь-якої формули (10.72), (10.73), (10.74), (10.75) або оцінок (10.76) випливає еквівалентність факторіальної функції:

   $$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \quad n \to \infty.$$

4. Довести (або перевірити для яких-небудь значень n) оцінку

   $$n^{n/2} \le n! \le \frac{(n+1)^n}{2^n}, \quad n > 2.$$

5. Довести, що

   $$\sum_{k=1}^{n} k^2 = O(n^3), \quad n \to \infty,$$

   точніше

   $$\sum_{k=1}^{n} k^2 \approx \frac{1}{3}n^3, \quad n \to \infty.$$

6. Порівняти зростання послідовностей $\{a_n\}_{n=1}^{\infty}$, $\{b_n\}_{n=1}^{\infty}$, где $a_n = n^{\ln n}$, $b_n = (\ln n)^n$, $n \to \infty$.
7. Довести, що

   $$1 + \frac{2}{n} + O(n^{-2}) = (1 + \frac{2}{n})(1 + O(n^{-2})), \quad n \to \infty.$$

8. За допомогою формули Стірлінга довести:

   $$C_{2n}^{n} \sim \frac{1}{\sqrt{\pi n}} 4^n, \quad n \to \infty.$$

9. Довести нерівності $(2n)! < (n(n+1))^n$, $\left(\frac{n}{3}\right)^n < n!$.

Комбінаторні завдання для контролю знань та самостійної роботи

1. З Києва до Чернігова можна дістатися пароплавом, поїздом, автобусом, літаком; із Чернігова до Новгород-Сіверського — пароплавом або автобусом. Скількома способами

можна здійснити подорож за маршрутом Київ — Чернігів — Новгород-Сіверський?

2. Скільки чотиризначних чисел можна скласти з цифр 1, 2, 3, 4, 5, якщо:
   а) жодна з цифр не повторюється більше одного разу;
   б) цифри можуть повторюватися;
   в) числа повинні бути непарними (цифри можуть повторюватися)?
3. В кошику 12 яблук і 10 апельсинів. Брат вибирає яблуко або апельсин, після чого сестра вибирає із фруктів, які залишилися, і яблуко, і апельсин. Скільки можливостей таких виборів? При якому виборі брата сестра має більше можливостей вибору?
4. Є 5 видів конвертів без марок і 4 види марок. Скількома способами можна вибрати конверт і марку для листа?
5. Скількома способами можна вибрати голосну і приголосну у слові «*паркет*»?
   Відповідь: $n = 8$.
6. Скількома способами можна вказати на шаховій дошці два квадрати — білий та чорний? Розв'яжіть цю задачу, якщо немає обмежень на колір квадрату. Розв'яжіть її, якщо треба вибрати 2 білих квадрати.
7. Скількома способами можна вибрати на шаховій дошці білий та чорний квадрати, які не лежать на одній горизонталі або на одній вертикалі?
8. Із 3 екземплярів підручників алгебри, 7 геометрії і 6 фізики треба вибрати комплект, який містить по одному підручнику кожного предмета. Скількома способами це можні зробити?
   Відповідь: $n = 3 \cdot 7 \cdot 6 = 126$.
9. Доведіть, що 77 телефонів не можна зв'язати між собою так, щоб кожний з них був зв'язаний рівно з 15 іншими.
10. Скільки треба взяти елементів, щоб число перестановок, утворених з них, дорівнювало 5040?
    Відповідь: $n = 7$ ($7! = 5040$).
11. В одинадцятому класі 35 учнів. Вони обмінялися фотографіями. Скільки всього фотографій було роздано?
    Розв'язок. Кожний учень одержить 34 фотографії. Тому $n = 35 \cdot 34$.

12. Скільки різних прямих можна провести через 10 точок площини, з яких ніякі 3 не лежать на одній прямій?
13. Доведіть, що серед будь-яких 5 грибів у лісі, з яких ніякі 3 гриби не розміщені на одній прямій, завжди можна знайти 4 таких, які є вершинами опуклого чотирикутника.
14. На площині взяті 9 точок, розміщених у вигляді квадрата 3×3. Скільки існує трикутників, у яких одна вершина знаходиться у фіксованій точці A, а дві інші — серед інших 8 точок?
15. Деяка комісія збиралася 40 разів. Кожний раз на засіданнях було по 10 людей, причому 2 її члени не були присутніми більше одного разу. Довести, що кількість членів комісії більше 60.
16. В деякій установі 25 співробітників. Довести, що із них не можна скласти понад 30 комісій по 5 людин у кожній так, щоб будь-які 2 комісії не мали більше одного загального члена.
17. Двоє грають у таку гру: не побачивши номеру автомашини, яка наближається, перший «бере» дві будь-яких цифри номеру (наприклад, першу і третю або другу і четверту), а другому дістануться дві інші; якщо номер стає відомим, обидва гравця складають цифри і перемагає той, у якого в сумі цифр одиниць більше. Скільки серед номерів від 0001 до 9999 таких чисел, що гра закінчується внічию незалежно від вибору, зробленого першим гравцем?
18. Для пофарбування однієї грані кубика треба 5 с. За який найменший час 3 людини можуть пофарбувати 188 кубиків? (Припускається, що дві людини не можуть разом фарбувати один кубик).
19. В змаганнях з гімнастики дві команди мали однакову кількість учасників. У результаті загальна сума балів, одержаних всіма учасниками, дорівнювала 156. Скільки було учасників, якщо кожний з них одержав оцінку 8 або 9 балів?
20. Яких чисел більше серед першого мільйона: тих, в запису яких зустрічається 1, або тих, в запису яких її немає?
Розв'язок. Підрахуємо кількість чисел від 0 до 999999, в запису яких немає одиниць, тобто скільки можна скласти чисел шести знаків із цифр 0, 2, 3, 4, ..., 9 (якщо число має менше шести цифр, домовимося дописувати зліва нулі). На першому місці у такому числі може

стояти будь-яка з дев'яти цифр, до будь-якої з них можна приписати з правого боку будь-яку з тих ж дев'яти: 0, 2, 3, 4, ..., 9; отже, одержимо 81 двозначне число с 0, 2, 3, 4, ..., 9. Продовжуючи, одержимо 9^6 шестизначних чисел, із них треба виключити 000000. Таким чином, показано, що серед першого мільйона існують $9^6 - 1$ чисел, в запису яких немає одиниць, тобто $9^6 - 1 = 531371$, що більше решти чисел у першому мільйоні.

21. За пересилання бандеролі слід сплатити 18 грн. Скількома способами можна сплатити її марками вартістю у 4, 6, 10 грн., якщо два способи, які відрізняються порядком марок, вважаються різними?

Відповідь: марки можна наклеювати 8-ю способами:

1) (10, 4, 4); 3) (4, 4, 10); 5) (4, 6, 4, 4); 7) (4, 4, 4, 6);
2) (4, 10, 4); 4) (6, 4, 4, 4); 6) (4, 4, 6, 4); 8) (6, 6, 6).

22. За круглий стіл сідає $n (n > 2)$ людей. Два розміщення за столом будемо вважати співпадаючими, якщо кожна людина має одних і тих же сусідів в обох випадках. Скільки існує способів сісти за стіл?
23. Скількома способами можна посадити за круглий стіл n чоловіків і n жінок таким чином, щоб ніякі дві особи одного полу не сиділи поряд?
24. Скільки можна скласти перестановок з n елементів, в яких задані m елементи не розміщені поряд у будь-якому порядку?
25. Маємо набір із 16 карток. На чотирьох написана буква А, на чотирьох — Б, на чотирьох — В, на чотирьох — Г. Скільки різних слів можні одержати, вибираючи з набору 4 картки і розміщуючи їх у деякому порядку?
26. Скількома способами можна обтягти 6 стільців тканиною, якщо вона є шести різних кольорів, і всі стільці повинні бути різнокольоровими?

Відповідь: $n = P_6 = 6! = 720$.

27. Скількома способами можуть розміститися у турнірній таблиці 10 футбольних команд, якщо відомо, що ніякі 2 не набрали однакової кількості очок?

Відповідь: $n = P_{10} = 10!$.

28. Скільки чотиризначних чисел можна утворити з цифр 0, 1, 2, 3, не повторюючи їх?

29. На зборах повинні виступити 5 людей: А, Б, В, Г, Д. Скількома способами можна їх розмістити у списку ораторів, якщо:
    1) Б не повинен виступати перед А;
    2) Б повинен виступити одразу за А?
30. Скількома способами можна скласти трьохколірний смугастий прапор, якщо є тканини п'яти різних кольорів? Розв'яжіть ту ж саму задачу за умови, коли одна смуга повинна бути червоною.
31. Є 8 токарів. Скількома способами можна доручити 3 з них виготовлення трьох різних деталей (за одним видом на кожного)?
    Розв'язок. Мова йде про вибір 3 токарів із 8 з подальшим розміщенням їх біля трьох верстатів, які виготовлюють різні деталі. Тому $n = A_8^3 = 336$.
32. В профактив факультету обрано 9 людин. З них треба обрати голову, його заступника, секретаря і культробітника. Скількома способами це можна зробити?
    Відповідь: $n = A_9^4 = \frac{9!}{4!} = 3024$.
33. Скількома способами можна опустити 5 листів у 11 поштових скриньок, якщо у кожну з них кидається не більше одного листа?
    Відповідь: $n = A_{11}^5 = ?$
34. Скільки різних натуральних чисел можна утворити з цифр 0, 1, 2, 3, 4, якщо кожне число містить будь-яку з даних цифр не більше одного разу?
35. Із колоди у 52 карти витягнули 10 карт. У скількох випадках серед цих карт виявиться:
    а) хоча б один туз;
    б) рівно один туз;
    в) не менше двох тузів;
    г) рівно два тузи?
36. Будівельна фірма формує бригаду з 5 робочих. В організації 20 робочих, в тому числі 5 малярів, 4 теслі і 2 штукатури. Скільком числом способів можна укомплектувати бригаду, щоб вона складалася з робочих всіх спеціальностей по одному?
37. Знайдіть суму чотиризначних чисел, які одержуються за допомогою всіляких перестановок цифр 1, 1, 4, 4. Те ж саме для 0, 0, 4, 4.

38. Скількома способами можна скласти три пари із n шахістів?
39. Скількома способами можна вибрати 6 карт із колоди у 52 карти таким чином, щоб серед них були карти кожної масті?
40. Скількома способами у грі «Спортлото» можна:
    а) вибрати п'ять куль з 36;
    б) вгадати 5, 4, 3 номери.
41. Скількома способами можна розмістити 12 різних деталей у 3-х ящиках?
42. Скільки існує автомобільних п'ятизначних номерів,
    а) складених з цифр 2, 3, 5, 7;
    б) що не містять цифру 8;
    в) що не містять цифр 0 и 8?
43. Знайти кількість способів розподілу n однакових куль за:
    а) двома; б) трьома; в) i нерізними урнами.
44. Скількома способами можна розмістити n_1 білих, n_2 чорних і n_3 блакитних куль за m різними урнами?
45. В лабораторії науково-дослідного інституту працює кілька людей, причому кожна з них знає хоча б одну іноземну мову, 6 — англійську, 6 — німецьку, 7 — французьку, 4 — англійську і німецьку, 3 — німецьку і французьку, 2 — французьку і англійську, одна особа — всі три мови. Скільки людей працює в лабораторії? Скільки з них знає лише англійську мову? Скільки осіб знає лише одну мову?
46. Скільки чисел серед першої тисячі натуральних чисел не діляться ані на 2, ані на 3, ані на 5, ані на 7?
47. Скільки шестизначних чисел можна скласти з цифр 1, 2, 3, ..., 9, якщо будь-яке число повинно складатися з трьох парних і трьох непарних цифр, причому ніякі 2 цифри в числі не повторюються?
48. 20 пасажирів збираються подорожувати поїздом. У касі є 12 білетів на нижні полки і 8 — на верхні. При цьому 4 пасажири не бажають їхати внизу, а 5 пасажирів — вверху. Скількома способами їх можна розмістити у поїзді, якщо:
    а) порядок розміщення пасажирів як внизу, так і вверху не враховується;
    б) порядок розміщення враховується як внизу, так і вверху;
    в) враховується порядок розміщення лише внизу?

49. Скільки слів можна одержати, переставляючи букви слів:
а) «*парабола*»; б) «*метаморфоза*»?
50. Для премій на математичній олімпіаді виділено 3 екземпляри одної книги, 4 — другої і 8 — третьої. Скількома способами можна розділити ці премії між 30 учасниками, якщо будь-якому вручають не більше однієї книги?
51. Абітурієнт повинен скласти 4 іспити. Він вважає, що для зачислення достатньо набрати 17 очок. Скількома способами він зможе скласти іспити, набравши не менше 17 очок і не одержавши жодної двійки?
52. Скільки слів, які містять не менше однієї букви, можна скласти з двох букв А, п'яти букв Б і дев'яти букв В?
53. Скільки слів з п'яти букв, будь-яке з яких складається з трьох приголосних і двох голосних, можна скласти з букв слова «рівняння»?
54. Учасники «Генуезької лотереї» купують білети, на яких розміщені числа від 1 до 90. На деяких білетах вказані одразу 2, 3, 4 або 5 чисел. У день розиграшу довільно обирають 5 жетонів з номерами від 1 до 90. Перемагають ті учасники, в яких всі номери на білетах будуть серед обраних на жетонах. Яка ймовірність виграшу у випадку купівлі білету з одним числом? З k числами ($1 \le k \le 5$)?
55. Скількома способами можна переставити букви слова «*обороноздатність*», щоб дві букви «о» не йшли підряд?
56. Знайдіть найбільший коефіцієнт у розкладанні:
а) $(a + b + c)^{10}$;
б) $(a + b + c + d)^{11}$.
57. Скільки існує трикутників, довжини боків яких приймають одне з таких значень: 4, 5, 6, 7 см?
58. Скільки можна побудувати прямокутних паралелепіпедів, довжини ребер яких виражаються натуральними числами від 1 до 10?
59. Скількома способами 9 пасажирів можна розмістити у трьох вагонах? Для скількох розміщень у перший вагон сядуть 3 з них? Для скількох розміщень у кожний вагон сядуть 3 з них? Для скількох розміщень в один вагон сяде 4, у другий — 3, а у третій — 2 пасажира?
60. Скількома способами можна відібрати кілька фруктів із 7 яблук, 4 лимонів, 9 апельсинів (вважаємо, що фрукти одного виду не відрізняються один від одного)?

61. Дві команди А і Б грають серію матчів у баскетбол доти, поки одна з них не одержить чотири перемоги (нічиїх немає). Скільки різних серій таких матчів може бути?
62. Сейф відчиняється за допомогою цифрового коду, циферблат якого складається з 100 клавіш з числами, розміщеними по колу. Для того, щоб відчинити сейф, треба натиснути якісь 3 клавіші, причому відомо, що між будь-якими двома шуканими клавішами розміщується не менше ніж 10 інших. Скільки комбінацій із 3 клавіш треба перепробувати, щоб відчинити сейф, якщо:
    а) порядок натискання не має значення;
    б) порядок натискання має значення?
63. Для кожного натурального n знайти суму
$$S_n = 1 \cdot 1! + 2 \cdot 2! + 3 \cdot 3! + \ldots + n \cdot n!.$$
Вказівка. Перевірити $S_n = (n + 1)! - 1$.
64. Доведіть, що $(C_n^0)^2 + (C_n^1)^2 + \ldots + (C_n^n)^2 = C_{2n}^n$.
Розв'язок. C_{2n}^n — кількість n-елементних підмножин $2n$-елементної множини B можна підрахувати так. Розіб'ємо множину B на дві підмножини (B_1 і B_2) за n елементами у кожній. Будь-які із шуканих n-елементних підмножин складаються з k елементів множин B_1 і $(n - k)$ елементів множини B_2 ($k = 0, 1, 2, \ldots, n$). Для будь-якої з C_n^k вибірок k елементів множини B_1 решту $(n - k)$ елементів із множини B_2 можна вибрати $C_n^{n-k} = C_n^k$ способами. За правилом добутку кількість шуканих підмножин з k елементами множини B_1 буде $(C_n^k)^2$. Оскільки
$$k = 0, 1, 2, \ldots, n, \quad \text{то } C_{2n}^n = (C_n^0)^2 + (C_n^1)^2 + \ldots + (C_n^n)^2.$$
65. Доведіть таку тотожність для чисел C_m^k:
$$C_p^0 \cdot C_{n-p}^m + C_p^1 \cdot C_{n-p}^{m-1} + \ldots + C_p^k \cdot C_{n-p}^{m-k} \ldots + C_p^m \cdot C_{n-p}^0 = C_n^m.$$
Розв'язок. C_n^m — кількість m-елементних підмножин Y множини X з n елементів. Розіб'ємо множину X на дві підмножини X_p із p елементів і X_{n-p} з решти $(n - p)$ елементів. Множини Y можні будувати, обираючи спочатку 0 елементів множини X_p і m елементів множини X_{n-p} (що можна зробити $C_p^0 \cdot C_{n-p}^m$ способами), і так далі, обираючи k елементів множини X_p і $(m - k)$ елементів множини X_{n-p}, що можна зробити $C_p^k \cdot C_{n-p}^{m-k}$ способами. Оскільки для різних значень k будемо мати різні підмножини, а інших

підмножин бути не може, то звідси випливає справедливість записаної вище формули.

Тотожність із задачі 65 частіше зустрічається в одній з таких форм:

$$\sum_{k=-m}^{k} C_r^{m+k} C_s^{n-k} = C_{r+s}^{m+n}; \quad \sum_{p=0}^{n-k} C_n^{k+p} C_m^p = C_{m+n}^{n-k}.$$

Ця тотожність носить назву *згортки А. Вандермонда* (кінець XVIII ст.), однак у 1303 р. вона була відома Чжу Шидзе у Китаї.

ПІСЛЯМОВА

В одній книзі важко надати розгорнутий виклад всіх розділів дискретної математики, розглянутих у цьому підручнику. Тут докладно розглянуто відношення, булеві алгебри і функції, графи, матеріал яких може бути використаний під час виконання індивідуальних завдань і вивчення спецкурсів з відповідних тем. Досить ємко викладено множини, математична логіка і комбінаторика. В той же час для більш поглибленого вивчення матеріалу цих розділів і використання їх у спецкурсах ми рекомендуємо користуватися додатковою літературою: Куратовський К., Мостовський А. (23), Натансон І. П. (26) — з теорії множин; Клини С. (16), Мендельсон Е. (28), Колмогоров А. Н., Драгалін А. Г. (17), Чень Ч., Лі Р. (37) — з математичної логіки; Андерсон Дж. (1), Віленкін Н. Я. (8, 9), Липський В. (24), Ріордан Дж. (30), Грехєм Р., Кнут Д., Поташнік О. (13), Холл М. (36) — з комбінаторики.

Автори змушені були більш конспективно викласти такі розділи, як «Мови та граматики», «Алгоритми та автомати». Звичайно ці розділи читаються у вигляді окремих дисциплін на багатьох спеціальностях у вищих навчальних закладах. Можна порекомендувати цілу низку книг з прекрасним викладанням цих розділів: Ахо А., Ульман Дж. (2, 3), Капітонова Ю. В. та ін. (15), Глушков В. М. (12), Корман Т., Лейзерсон Ч., Рівест Р. (18), Кук Д., Бейз Г. (22).

В наведеній літературі є багато модельних і змістовних прикладів, які корисні та повчальні. Деякими з них автори з подякою скористалися.

СПИСОК ЛІТЕРАТУРИ

1. *Андерсон Дж.* Дискретная математика и комбинаторика.— Москва — С. Петербург — Киев.: Издат. дом «Вильямс», 2003.— 958 с.
2. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т. 1. Синтаксический анализ.— М.: Мир, 1978.— 612 с.
3. *Ахо А., Хопкфорт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов.— М.: Мир, 1979.— 536 с.
4. *Бардачев Ю. Н., Соколова Н. А., Ходаков В. Е.* Основы дискретной математики.— Херсон, ХГТУ, 2000.— 356 с.
5. *Белоусов А. И., Ткачев С. Б.* Дискретная математика.— М.: Издательство МГТУ им. Н. Э. Баумана, 2001.— 744 с.
6. *Білоус Н. В.* та ін. Основи комбінаторного аналізу/Н. В. Білоус, З. В. Дудар, Н. С. Лєсна, І. Ю.Шубін.— Харків: ХТУРЕ, 1999.— 96 с.
7. *Бондаренко М. Ф.* та ін. Збірник тестових завдань з дискретної математики/М. Ф. Бондаренко, Н. В. Білоус, І. Ю. Шубін.— Харків: ХТУРЕ, 2000.— 156 с.
8. *Виленкин Н. Я.* Комбинаторика.— М.: Наука, 1969.— 328 с.
9. *Виленкин Н. Я.* Популярная комбинаторика.— М.: Наука, 1975.— 208 с.
10. *Гаврилов Г. П., Сапоженко А. А.* Сборник задач по дискретной математике.— М.: Наука, 1977.— 368 с.
11. *Горбатов В. А.* Основы дискретной математики.— М.: Высшая школа, 1986.— 312 с.
12. *Глушков В. М.* Синтез цифровых автоматов.— М.: Физматгиз, 1962.— 476 с.
13. *Грэхем Р.* та ін. Конкретная математика, основание информатики/ Р. Грэхем, Д. Кнут, О. Поташник.— М.: Мир, 1998.— 704 с.
14. *Иванов Б. Н.* Дискретная математика (Алгоритмы и программы).— М.: Лаборатория базовых знаний, 2001.— 288 с.
15. *Капітонова Ю. В.* та ін. Основи дискретної математики/Ю. В. Капітонова, С. Л. Кривий, О. А. Летичевский та ін.— К.: Наукова думка, 2002.— 578 с.
16. *Клини С.* Математическая логика.— М.: Мир, 1973.— 480 с.
17. *Колмогоров А. Н., Драгалин А. Г.* Введение в математическую логику.— М.: МГУ, 1982.— 120 с.
18. *Корман Т.* та ін. Алгоритмы: построение и анализ/Т. Корман, Ч. Лейзерсон, Р. Ривест.— М.: МЦНМО, 2001.— 960 с.
19. *Криницкий Н. А.* Аналитическая теория алгоритмов.— М.: Физматлит, 1994.— 352 с.
20. *Кристофидес Н.* Теория графов (Алгоритмический подход).— М.: Мир, 1978.— 432 с.

21. *Кузнецов О. П., Адельсон-Вельский Г. М.* Дискретная математика для инженера.— М.: Энергия, 1980.— 344 с.
22. *Кук Д., Бейз Г.* Компьютерная математика.— М.: Наука, 1990.— 384 с.
23. *Куратовский К., Мостовский А.* Теория множеств.— М.: Мир, 1970.— 416 с.
24. *Липский В.* Комбинаторика для программистов.— М.: Мир, 1998.— 214 с.
25. *Мальцев А. И.* Алгебраические системы.— М.: Наука, 1970.— 370 с.
26. *Натансон И. П.* Теория функций вещественной переменной.— М.: Наука, 1974.— 480 с.
27. *Новиков Ф. А.* Дискретная математика для программистов.— СПб.: Питер, 2001.— 304 с.
28. *Мендельсон Э.* Введение в математическую логику.— М.: Наука, 1976.— 320 с.
29. *Оре О.* Теория графов.— М.: Наука, 1968.— 352 с.
30. *Риордан Дж.* Введение в комбинаторный анализ.— М.: ИИЛ, 1963.— 288 с.
31. *Рутткас А. Г.* Введение в теорию графов.— Харьков.: Принтал, 1993.— 63 с.
32. *Уилсон Р.* Введение в теорию графов.— М.: Мир, 1977.— 205 с.
33. *Филлипс Д., ГарсиаДиас А.* Методы анализа сетей.— М.: Мир, 1964.— 496 с.
34. *Форд Л., Фалкерсон Д.* Потоки в сетях.— М.: Мир, 1966.— 276 с.
35. *Харари Ф.* Теория графов.— М.: Мир, 1973.— 304 с.
36. *Холл М.* Комбинаторика.— М.: Мир, 1970.— 424 с.
37. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем.— М.: Наука, 1983,— 256 с.
38. *Яблонский С. В.* Введение в дискретную математику.— М.: Наука, 1986.— 384 с.

ПРЕДМЕТНИЙ ПОКАЖЧИК

Г

Ґ

Д

Е

З

Р

С

ЗМІСТ

Навчальне видання

БОНДАРЕНКО Михайло Федорович
БІЛОУС Наталія Валентинівна
РУТКАС Анатолій Георгійович

Комп'ютерна дискретна математика

Підручник для студентів
вищих навчальних закладів

Редактор *Драган Н. О.*
Комп'ютерна верстка *Кризський А. В.*
Художник обкладинки *Денисенко А. О.*
Технічний редактор *Марамзіна Л. М.*

Підписано до друку 18.11.2004. Формат 60 × 90 $^{1}/_{16}$. Друк офсетний.
Папір офсетний № 1. Гарнітура SchoolBookC. Умов. друк. арк. 30.
Тираж 6500 прим. Зам. № 22/12
Видано за рахунок державних коштів. Продаж заборонено

Видавництво «Компанія СМІТ».
61166, м. Харків, просп. Леніна, 14.
Тел.: 8-(057)-717-54-94, 702-08-16
Факс: 8-(057)-702-13-07
E-mail: book@smit.kharkov.ua
http://www.smit-book.com

Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців, виготівників і розповсюджувачів видавничої продукції ДК № 435 від 26.04.2001.

Надруковано в друкарні ТОВ «СІМ».
61012, м. Харків, пров. Лопанський, 3-а.
Тел. 8-(057)-719-19-30.
E-mail: tipa_graf@pisem.net