Міністерство освіти та науки України
Луцький національний технічний університет

ДИСКРЕТНА МАТЕМАТИКА

Конспект лекцій
для здобувачів першого (бакалаврського) рівня вищої освіти
освітньої програми «Інформаційні системи
та технології охорони і безпеки»
галузі знань F Інформаційні технології
спеціальності F6 Інформаційні системи та технології
денної та заочної форм навчання

Луцьк 2025

004.896
Д26

Рекомендовано до видання вченою радою факультету КІТ ЛНТУ
протокол №________від «_____»__________________20_______року.
Голова вченої ради факультету КІТ ________________________ Інна КОНДІУС
(підпис)
Електронна копія друкованого видання передана для внесення в репозитарій Луцького НТУ

директор бібліотеки ____________ Наталія ПОЛІЩУК
(підпис)
Затверджено вченою радою факультету КІТ,
протокол № ______ від «____» ______________ 20__ року.

Розглянуто і схвалено на засіданні кафедри комп’ютерних наук Луцького НТУ,

протокол №_______ від «____» ____________________ __20______ року.

Завідувач кафедри комп’ютерних наук ________________________ Валерій ЛІЩИНА
(підпис)

Укладачі: ___________ (підпис) Ніна ЗДОЛБІЦЬКА, кандидат технічних наук, доцент кафедри комп’ютерних наук ЛНТУ

Рецензент: ___________ (підпис) Катерина МЕЛЬНИК, кандидат технічних наук, доцент кафедри комп’ютерної інженерії та безпеки ЛНТУ

Відповідальний за випуск ___________ (підпис) Валерій ЛІЩИНА, кандидат технічних наук, доцент кафедри комп’ютерних наук ЛНТУ

Д26 Дискретна математика: Конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти вищої освіти освітньої програми «Інформаційні системи та технології охорони і безпеки» галузі знань F Інформаційні технології спеціальності F6 Інформаційні системи та технології денної та заочної форм навчання / уклад. Н.В. Здолбіцька, Луцьк, ЛНТУ, 2025. 116 с.

Конспект лекцій складено відповідно до діючої програми курсу Дискретна математика з метою використання студентами спеціальності F6 Інформаційні системи та технології при вивченні даної дисципліни.

Н.В. Здолбіцька

ЗМІСТ

ВСТУП

Дискретна математика є фундаментальною основою сучасних інформаційних технологій і відіграє критично важливу роль у всіх сферах, де інформація обробляється, передається та зберігається у цифровому вигляді. На відміну від неперервної математики (наприклад, числення), яка вивчає неперервні процеси, дискретна математика займається дискретними, скінченними структурами – елементами, які можна перерахувати, такими як цілі числа, графи, множини та логічні висловлювання.

Дискретна математика забезпечує теоретичний апарат для розуміння та розробки ключових концепцій в галузі комп'ютерних наук. Її принципи лежать в основі алгоритмів, які керують програмним забезпеченням, структур даних для ефективного зберігання інформації, криптографії для забезпечення безпеки даних, а також логіки програмування та проектування комп'ютерних архітектур. Вивчення дискретної математики формує алгоритмічне та логічне мислення, необхідне кожному фахівцю для ефективного вирішення складних, практичних завдань сьогодення та майбутнього. Це пояснюється необхідністю створення і експлуатації сучасних ПК, засобів передачі та обробки інформації, автоматизованих систем управління та проектування.

Розподіл математики на «неперервну» і «дискретну» умовний, тому що вся математика єдина і пронизана глибокими аналогіями. З одного боку, відбувається обмін ідеями і методами між ними, а з іншого – часто виникає необхідність дослідження моделей, що володіють як дискретними, так і безперервними властивостями одночасно. Наприклад, апарат теорії множин і теорії графів використовується при вивченні не тільки дискретних, але і неперервних об'єктів. Наявність знань з дискретної математики у інженерів-програмістів важко переоцінити, бо їх глибоке ґрунтовне знання грає велику роль у фундаментальній математичній підготовці – як в плані формування у студентів певного рівня математичної культури, так і в плані формування в них наукового світогляду, особливо з таких компонентів, як розуміння сутності прикладної і практичної спрямованості навчання, оволодіння методом математичного моделювання, вмінням здійснювати міжпредметні зв'язки.

ТЕМА 1. МНОЖИНИ. ОПЕРАЦІЇ НАД МНОЖИНАМИ

1.1 Основи теорії множин

Поняття множини є одним із найважливіших у математиці. Німецький математик Георг Кантор – основоположник теорії множин – визначив множину як сукупність об'єктів, які добре розпізнаються за певною ознакою. Складена із окремих елементів сукупність – множина є вже дещо ціле. Об'єкти, з яких складається множина, називаються її елементами. Найвищий рівень абстракції, притаманний математиці, проявляється перш за все в понятті множини: ми можемо вивчати загальні властивості множин, не фіксуючи особливих властивостей елементів, які належать їм, абстрагуючись від сутності елементів. Множини можна (і в деяких випадках необхідно) конкретизувати, і, звичайно, можна розрізняти множини, наприклад, натуральних чисел, неперервних функцій на відрізку, різних комп'ютерних зображень деякого формату тощо.

Існує два підходи до побудови теорії множин, у яких поняття множини вводять по різному:

– конструктивний;

– формалістичний.

У конструктивній теорії множин поняття множини означують. У формалістичній теорії поняття множини буде первісним, тобто таким, що його не означують. Ми будемо розглядати поняття множини як первісне, тобто таким, що його не означують.

Поняття множини

Множиною називають сукупність деяких об'єктів, які можна розглядати як єдине ціле. Множини позначають великими буквами

$$A, B, C.$$

Об'єкти, які складають множину, називають її елементами, позначають їх малими буквами a, b, c. Якщо елемент належить множині, то записують $a \in A$, у противному разі $a \notin A$.

Множину можна задати:

1. перерахуванням її елементів:

2. за допомогою характеристик її елементів;

3. графічно.

1. Належність елементів $a_1, a_2, ..., a_n$ до множини А позначають так

$$A = \{a_1, a_2, ..., a_n\}, \text{ або } A = \{a_n\} \text{ де } n = 1, 2, ..., N.$$

Наприклад, множина A = $\{a,e,i,o,u\}$ містить елементи a,e,i,o,u і лише ці елементи. Множина не може містити двох однакових елементів, а порядок її елементів не фіксують.

2. Задати множину можна, вказавши спільну властивість всіх її елементів, увівши таке позначення: $A = \{x | P(x)\}$, яке читають так: "А – це множина об'єктів х, які мають властивість Р(х) "

Наприклад , $A = \{x | x \in N_0, x < 7\}$ – це множина {0.1,2,3,4.5,6}.

Іноді замість вертикальної риски використовують дві крапки, тобто $A = \{x : P(x)\}$

3. Множини часто задаються графічно, за допомогою діаграм Ейлера-Венна. Наприклад, на рис. 1.1 зображено множини A = {1; 2} та B = {1; 4; 5 } в універсальному просторі U.

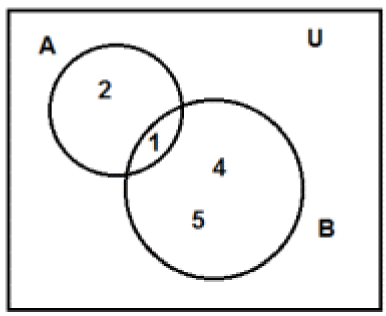


Рисунок 1.1 – Діаграма Ейлера-Венна

Замкнену лінію, що обмежує елементи множини, називають кругом Ейлера. Прямокутник U в якому розміщені множини А та В – універсальним простором (множиною).

1.2 Види множин

Множину, яка не має жодного елемента, називають порожньою і позначають символом

$$\varnothing.$$

Множину, яка складається із скінченного числа елементів, називають скінченною. Число елементів множини А називають потужністю множини і позначають:

$$|A|.$$

Деякі множини мають загально прийняті позначення:

$N = \{1,2,\ldots,n,\ldots\}$ – множина натуральних чисел;

$Z = \{\ldots,-n,\ldots,-2,-1,0,1,2,\ldots,n,\ldots\}$ – множина цілих чисел;

$N_0 = \{0,1,2,\ldots,n,\ldots\}$ – множина цілих невід'ємних чисел;

$Q = \left\{ \frac{p}{q} \middle| p \in Z, q \in N \right\}$ – множина раціональних чисел;

$R = \{x | x = \pm a_1, a_2, ..., a_n\}$ – множина дійсних чисел. $\varnothing$

Множину B називають підмножиною множини A, якщо кожний елемент множини B належить множині A, тобто

$B \subset A \leftrightarrow (b \in B \to b \in A)$

де $\subset$ - знак належності.

Множина A називається надмножиною.

Якщо B є підмножиною A, а A в свою чергу є підмножиною B, то множини A і B називаються рівними $A = B$.

Якщо $B \subseteq A$ і $B \neq A$, то B називають власною підмножиною множини A.

Відмітим, що $\forall A \quad A \subset A$,

$\forall A \quad \varnothing \subset A$.

Щоб розрізнити власні і невласні підмножини то використовують знак $\subset$ для власних підмножин і знак $\subseteq$ - для невласних .

Якщо $A \subseteq B$, $B \subseteq C$, то $A \subseteq C$.

Не треба плутати відношення належності і включення .

Якщо $1 \in \{1\}$, $\{1\} \in \{\{1\}\}$,то не правильно що $1 \in \{\{1\}\}$, так як єдиним елементом множини $\{\{1\}\}$ є $\{1\}$.

Множину всіх підмножин множини A називають булеаном і позначають 2^A. $2^A = \{A | A \subset M\}$.

Потужність множини позначають $|A|$. Для скінченних множин потужніть – це число елементів множини ,рівне 2^A. Теорема $|2^A| = 2^{|A|}$

Наприклад для $|\varnothing| = 0$, але $|\{\varnothing\}| = 1$.

Якщо $A = B$, то множини називаються рівнопотужними.

1.3 Операції над множинами

Об'єднання $A \cup B = \{x | x \in A \vee x \in B\}$.

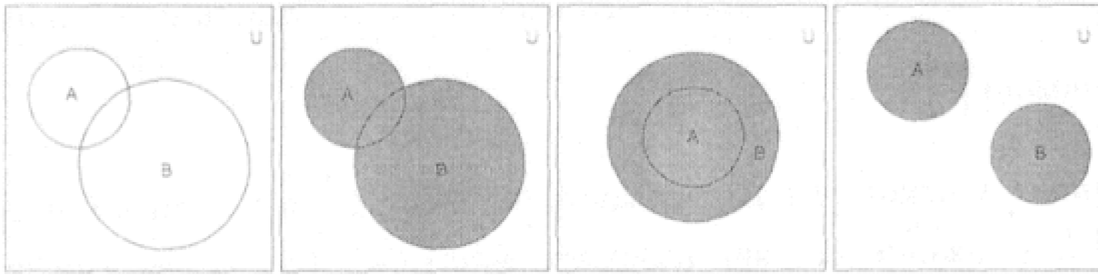


Рисунок 1.2 – Зображення об'єднання множин за допомогою діаграм Ейлера-Венна

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

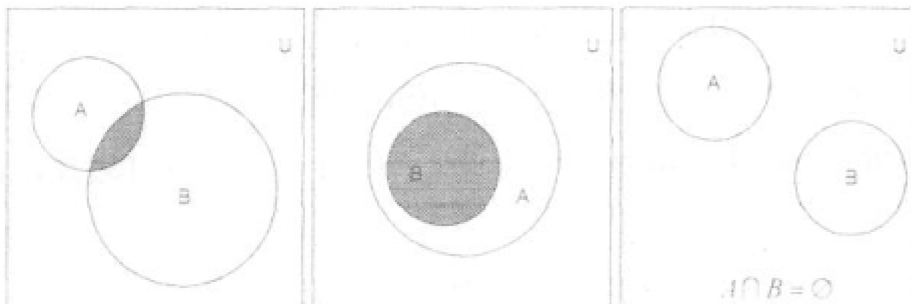


Рисунок 1.3 – Зображення перетину множин за допомогою діаграм Ейлера-Венна

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

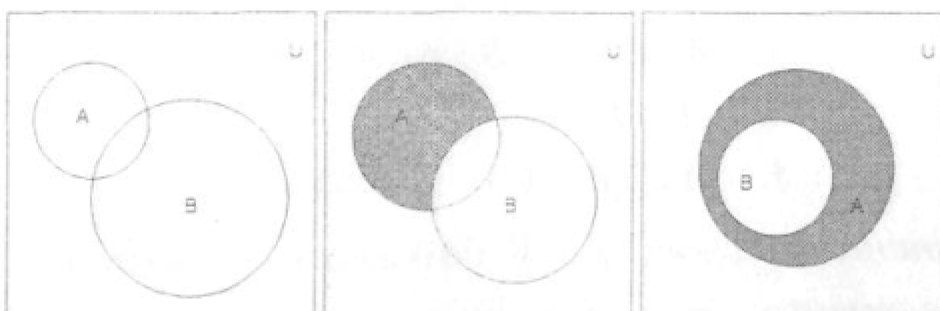
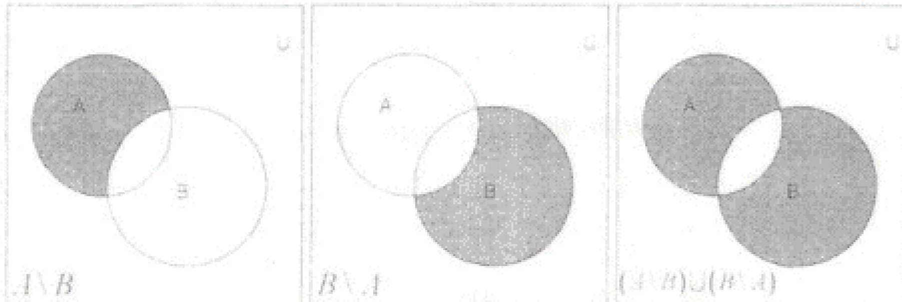


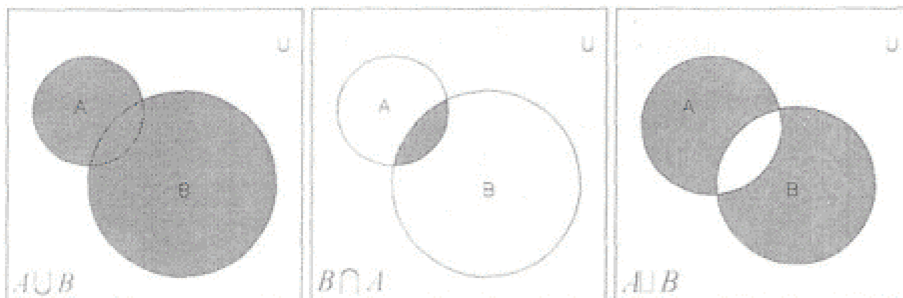
Рисунок 1.4 – Зображення різниці множин за допомогою діаграм Ейлера-Венна

Симетрична різниця

$$A \oplus B = (A \cup B) \setminus (A \cap B) = (A \setminus B) \cup (B \setminus A) =$$
$$\{x | (x \in A \wedge x \notin B) \vee (x \in B \wedge x \notin A)\}.$$



а



б

Рисунок 1.5 – Зображення перетину множин за допомогою діаграм Ейлера-Венна різними способами

Доповнення $\overline{A} = \{x | x \notin A\}$.

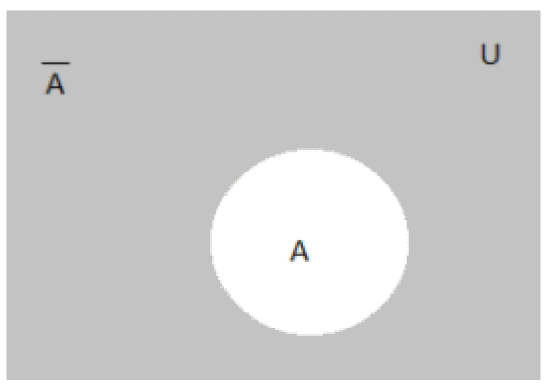


Рисунок 1.6 – Зображення доповнення множини за допомогою діаграм Ейлера-Венна

Існує універсальна множина U: $\overline{A} = U \setminus A$.

Операції об'єднання допускають узагальнення:

Нехай I **– деяка множина, елементами якої є індекси, і нехай для будь-якого** $\forall i \in I$ **визначена множина** A_i**. Тоді**

$$\bigcup_{i\in I} A_i = \{x \mid \exists i \in I \;\; x \in A_i\}, \;\; \bigcap_{i\in I} A_i = \{x \mid \forall i \in I \;\; x \in A_i\}$$

1.4 Розбиття і покриття

Нехай $\varepsilon = \{E_i\}_{i\in I}$ – деяка сукупність підмножин множини A, $E_i \subset A$. Сукупність ε називається покриттям множини A, якщо кожен елемент A належить хоча б одному з E_i:

$$A \subset \bigcup_{i\in I} E_i \leftrightarrow \forall x \in A \;\; \exists i \in I \;\; x \in E_i .$$

Сукупність ε називається диз'юнктивною, якщо елементи цієї сукупності попарно не перетинаються, тобто кожний елемент множини A належить не більше ніж одній з множин E_i:

$$\forall i, j \in I \; i \neq j \rightarrow E_i \cap E_j = \varnothing .$$

Диз'юнктивне покриття ε називається розбиттям множини A.

Приклад.

Нехай $A = \{1,2,3\}$. Тоді $\{\{1,2\},\{2,3\},\{3,1\}\}$ є покриттям, але не розбиттям, $\{\{1\},\{2\},\{3\}\}$ є розбиттям (і покриттям), а сукупність $\{\{1\},\{2\}\}$ є диз'юнктивною, але не є покриттям, ні розбиттям.

1.5 Властивості операцій над множинами

1. ідемпотентність

$A \cup A = A, \;\; A \cap A = A$

2. комутативність

$A \cup B = B \cup A, \;\; A \cap B = B \cap A$

3. асоціативність

$A \cup (B \cup C) = (A \cup B) \cup C, \; A \cap (B \cap C) = (A \cap B) \cap C$

4. дистрибутивність

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

5. поглинання

$(A \cap B) \cup A = A, \; (A \cup B) \cap A = A$

6. властивість нуля

$A \cup \varnothing = A, \; A \cap \varnothing = \varnothing$

7. властивість одиниці

$A \cup U = U, \; A \cap U = A$

8. інволютивність (закон заперечення заперечення)

$\overline{\overline{A}} = A$

9. закони де Моргана:

$\overline{A \cap B} = \overline{A} \cup \overline{B}$, $\overline{A \cup B} = \overline{A} \cap \overline{B}$

10. властивість доповнення

$A \cup \overline{A} = U$, $A \cap \overline{A} = \varnothing$

11. вираз для різниці

$A \setminus B = A \cap \overline{B}$

1.6 Ком'ютерне зображення множин

Для комп'ютерного зображення множин у використовують такі способи:

– накопичення елементів множини у невпорядкованому вигляді;

– за допомогою бітових рядків.

Якщо використовувати спосіб 1, то операції із множинами вимагатимуть значних витрат часу через необхідність щоразу здійснювати перегляд елементів.

Упорядкуємо довільним способом елементи універсальної множини. Нехай універсальна множина U містить n елементів.

Тоді $U = \{a_1, a_2, ..., a_n\}$. Множину $A \in U$ зображають у комп'ютері рядком із 0 та 1 довжини n так: якщо $a \in A$, то i-й біт дорівнює 1. якщо $a \notin A$ то i-й біт дорівнює 0.

Приклад. Нехай $U = \{a, b, c, d, e, f, m, n, p, q\}$, $A = \{b, m, n, q\}$, $B = \{a, b, f, m, q\}$. Знайти об'єднання множин $A \cup B$. Множину А зобразимо рядком 0100001101, множину В – рядком 1100011001.

Неважко переконатись, що перетину множин відповідає порозрядна кон'юнкція над бітовими рядками, об'єднанню множин – порозрядна диз'юнкція над бітовими рядками.

Логічні операції наведені в табл. 1, де символ 1 відповідає значенню TRUE (істина), символ 0 – відповідає значенню FALSE (хибно).

Комп'ютери відображають інформацію за допомогою бітів. Біт має два можливі значення – 0 (нуль) і 1 (одиниця). Його можна використовувати для подання значень істинності Т й F. Зазвичай 1 використовують для зображення Т й 0 – для зображення F. Змінну називають булевою, якщо її значення – T чи F. Отже, булеві змінні можна подати за допомогою бітів.

Комп'ютерні операції над бітами відповідають логічним операціям. Замінивши T на 1, а F – на 0 у таблицях істинності для логічних операцій $\vee$, $\wedge$ та $\oplus$, отримаємо таблиці відповідних операцій над бітами. Ми будемо також використовувати нотацію OR, AND і XOR відповідно для логічних операцій $\vee$, $\wedge$ та $\oplus$, як у багатьох мовах програмування. Значення операцій OR, AND та XOR над бітами наведено в табл. 1.1

Таблиця 1.1 – Значення операцій OR, AND та XOR

x	y	OR	AND	XOR
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Приклад. Знайдемо результати операцій порозрядного OR, порозрядного AND і порозрядного XOR бітових рядків 1011000011 і 1101010101. Одержимо

1011000011

1101010101

1111010111 – порозрядне OR

1001000001 – порозрядне AND

0110010110 – порозрядне XOR

Якщо універсальна множина U має велику потужність, а підмножини універсальної множини не дуже потужні, то зображення за допомогою бітових рядків не є ефективним з точки зору витрат нам'яті. У такому разі для зображення множин доцільно використовувати інші структури даних - як правило, зв'язані списки або хеш-таблиці. У деяких задачах доцільно використовувати спеціальні методи зображення множин, в основі яких є використання дерев.

Таблиця 1.2 – Значення операцій

a_i	b_i	$a_i \wedge b_i$	$a_i \vee b_i$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Література: [3, 4, 6, 7, 8, 10, 11].

ТЕМА 2. НЕЧІТКІ МНОЖИНИ

2.1 Поняття нечіткої множини.

Нехай U – універсальна множина, x – елемент U, а G – деяка властивість. Звичайна (чітка) підмножина A універсальної множини U, елементи якої мають властивість G, визначається як множина впорядкованих пар {<μA(x)|x>}, де μA(x) – характеристична функція приналежності, що приймає значення 1, якщо x має властивість G, та 0 – у протилежному випадку.

Нечітка підмножина відрізняється від звичайної тим, що для елементів u з U немає однозначної відповіді «ні» або «так» щодо властивості G.

У зв'язку з цим нечітка підмножина A універсальної множини U визначається як множина впорядкованих пар $A = \{<\mu_A(x)|x>\}$, де $\mu_A(x)$ – характеристична функція приналежності (або просто функція приналежності), що приймає значення в деякій цілком впорядкованій множині M (наприклад, $M = [0; 1]$).

Функція приналежності вказує ступінь приналежності елемента u нечіткій підмножині A.

Множину M називають множиною приналежностей. Якщо $M=\{0, 1\}$, то нечітка підмножина A може розглядатися як чітка множина. Чітку множину A можна розглядати як граничний випадок нечіткої множини A, функція приналежності якої $\mu_A(x)$ набуває лише бінарних значень.

Приклади

Приклад 2.1. Представити у вигляді нечіткої множини поняття «Чоловік середнього зросту») на універсальній множині U={155,160,165,170,175,180,185,190}.

Одне з можливих рішень виглядає так:

A=(0/155, 0,1/160, 0,3/165, 0,8/170, 1/175, 1/180, 0,5/185, 0/190).

Нечітка змінна визначається як <a, U, A>, де a – найменування змінної, U={u} – область визначення змінної (набір можливих значень u), A={<μA(u)|u>} – нечітка множина, що описує обмеження на можливі значення змінної a (семантику).

Нечітка змінна – це теж ж, що і нечітке число, тільки з додаванням імені, яким формалізується поняття, що описується цим числом. Для людини зручніше задавати значення змінної не числами, а словами. Щодня ми приймаємо рішення на основі лінгвістичної інформації типу: «дуже висока температура»; «утомлива поїздка»; «швидка відповідь»; «красивий букет»; «гармонійний смак» і тому подібне. Психологи встановили, що в людському мозку майже вся числова інформація вербально перекодується і зберігається у вигляді слів.

Лінгвістична змінна – це множина нечітких змінних, вона використовується для того, щоб дати словесний опис деякому нечіткому числу, отриманому в результаті деяких операцій.

Лінгвістична змінна визначається як $<x, L, U, G, M>$, де x – найменування змінної, L – множина її значень (базова терм-множина), що складається з найменувань нечітких змінних, областю визначення кожної з яких є множина U; G – синтаксична процедура (граматика), що дозволяє оперувати елементами терм-множини L, зокрема – генерувати нові осмислені терми; $L' = L \cup G(L)$ задає розширену терм-множину ($\cup$ – знак об'єднання); M – семантична процедура, що дозволяє приписати кожному новому значенню лінгвістичної змінної нечітку семантику, шляхом формування нової нечіткої множини.

Терм-множина – це множина всіх можливих значень лінгвістичної змінної.

Терм – будь-який елемент терм-множини. У теорії нечітких множин терм формалізується нечіткою множиною за допомогою

функції приналежності.

Наприклад, змінна «швидкість автомобіля» може набувати значень «низька», «середня», «висока» і «дуже висока». В цьому випадку лінгвістичною змінною є «швидкість автомобіля», термами – лінгвістичні оцінки «низька», «середня», «висока» і «дуже висока», які і складають терм-множину.

Нечіткий терм – це нечітка множина, яка має властивість, якій відповідає певне поняття.

Універсум – це множина, яка містить всі можливі елементи в рамках контексту визначеної предметної області. Універсум, зазвичай, позначається літерою X.

Нечіткою множиною $\bar{A}$ називається множина впорядкованих пар або кортежів вигляду $<\mu A(x)|x>$, де x – елемент універсуму X, $\mu_{\bar{A}}(x): X \rightarrow [0,1]$ – функція належності, яка ставить у відповідність кожному елементу $x \in X$ дійсне число з інтервалу $[0,1]$, що характеризує ступінь належності елемента x до нечіткої множини $\bar{A}$. Чим більшим є значення функції належності $\mu_{\bar{A}}(x)$, тим більше елемент універсальної множини x відповідає властивостям нечіткої множини $\bar{A}$.

Скінченну нечітку множину $\bar{A}$ будемо записувати у вигляді

$\bar{A}=\{<x_1; \mu_{\bar{A}}(x_1)>, \ldots, <x_n; \mu_{\bar{A}}(x_n)>\}$, або $\bar{A}=\{<x; \mu_{\bar{A}}(x)>\}$, де n – кількість елементів нечіткої множини v. Крім даних позначень використовують такі форми запису: $\bar{A}=\{(\mu_{\bar{A}}(x_1);x_1),\ldots,(\mu_{\bar{A}}(x_n);x_n)\}$,

$$\bar{A}=\left\{\frac{x1}{\mu_{\bar{A}}(x1)}+\cdots+\frac{xn}{\mu_{\bar{A}}(xn)}\right\} \text{ та } \bar{A}=\left\{\frac{\mu_{\bar{A}}(x1)}{x1}+\cdots+\frac{\mu_{\bar{A}}(xn)}{xn}\right\}=\sum_{i=1}^{n}\frac{\mu_{\bar{A}}(xi)}{xi}.$$

При цьому горизонтальна (або похила) лінії є розділовими знаками, а знак "+" позначає теоретико-множинне об'єднання окремих елементів. Неперервні нечіткі множини, як правило, записують у вигляді:

$\bar{A}=\int_X \frac{\mu_{\bar{A}}(xi)}{xi} dx$.

Порожня нечітка множина позначається символом Ø і визначає нечітку множину, функція належності якої тотожньо дорівнює нулю (;$\mu_{\emptyset}$ =0).

Носієм нечіткої множини $\bar{A}$ називається звичайна множина $\bar{A}_s$, яка містить ті елементи універсуму X, значення функції належності яких не дорівнюють нулю, тобто $\bar{A}_s = \{x \in X | \mu_{\bar{A}}(x) > 0\}$.

Нечіткі множини задаються трьома способами:

1. У формі списку з переліком всіх елементів нечіткої множини та відповідних їм значень функції належності: $\bar{A}$={<x_1; $\mu_{\bar{A}}$ (x_1)>, ..., <x_n; $\mu_{\bar{A}}$ (x_n)>}. При цьому елементи з нульовими значеннями функції належності, зазвичай, не вказуються в даному списку. Цей спосіб застосовують для визначення нечітких множин, які мають скінченний дискретний носій.

Наприклад, розглянемо в якості універсума X множину натуральних чисел. Тоді нечітку множину $\bar{A}$, яка описує "невелике натуральне число", можна визначити наступним чином:

$\bar{A}$={<1;1,0>, <2;1,0>, <3;0,9>, <4;0,8>, <5;0,6>, <6;0,5>, <7;0,4>, <8;0,2>, <9;0,1>}.

2. Аналітично – у формі математичного виразу для відповідної функції належності: $\bar{A}$={<x_i; $\mu_{\bar{A}}$(x_i)}. Цей спосіб використовують для визначення нечітких множин як зі скінченним, так і з нескінченним носієм.

3. Графічно – у формі деякої кривої або сукупності окремих точок у двовимірному просторі. При цьому одна координата відповідає елементам універсуму X, а друга – значенням функції належності цих елементів.

Основні види функцій належності. Існує багато кривих для визначення функцій належності. Найбільш розповсюдженими функціями належності є трикутна, трапецієвидна, функція Гаусса та Z- (або S-) подібні.

Трикутна функція належності визначається трійкою чисел (a,b,c), а її значення в точці x обчислюється за формулою:

$$\mu(x)=\begin{cases}\dfrac{x-a}{b-a}, & a\le x\le b\\ \dfrac{c-x}{c-b}, & b\le x\le c\\ 0, & \text{в інших випадках}\end{cases}$$

При (b-a)=(c-b) маємо симетричну трикутну функцію належності, яка однозначно задається двома параметрами з трійки (a,b,c).

Для визначення трапецієвидної функції належності потрібні чотири числа (a,b,c d), а її значення в точці x обчислюється за формулою:

$$\mu(x)=\begin{cases}\dfrac{x-a}{b-a}, & a\le x\le b\\ 1, & b\le x\le c\\ \dfrac{d-x}{d-c}, & c\le x\le d\\ 0, & \text{в інших випадках}\end{cases}$$

При (b-a)=(d-c) ця функція належності приймає симетричний вигляд. Функція належності Гаусса (рис. 2.1), зазвичай, описується формулою $\mu(x)=e^{-\frac{(x-c)^2}{2\sigma^2}}$ і визначається параметрами (c, s). Z- і S-подібні функції належності одержали свою назву у зв'язку з виглядом кривих, які зображують їхні графіки (рис. 2.2).

2.2 Приклади нечітких множин

Розглянемо приклади нечітких множин.

Приклад 2.1. Побудуємо нечітку множину, яка змістовно описує вихідні дні тижня. В термінології класичних множин ситуація зрозуміла: дні тижня з понеділка по п'ятницю є робочими, а субота та неділя – вихідними. Зазначимо, що мова йде про традиційний календарний тиждень. Таким чином, множина вихідних днів складається з двох елементів {субота, неділя}.

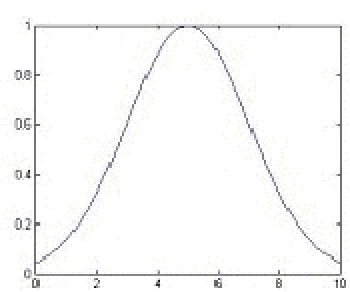
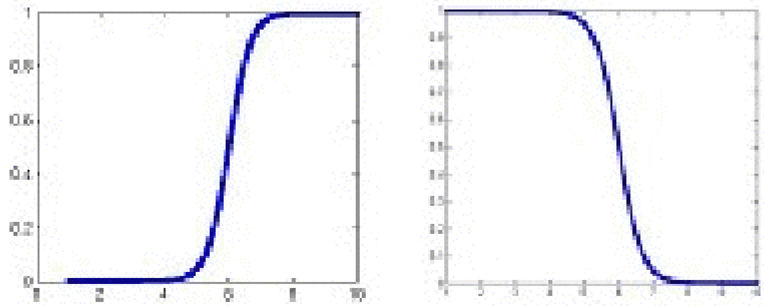


Рисунок 2.1 – Графік функції належності Гаусса при c=5, s =2



а б

Рисунок 2.2 – Графіки сигмоїдальної функції належності при (а) a=3, b=6; (б) a=-3, b=6

Спробуємо оцінити ступінь нашого емоційного ставлення до різних днів тижня, розглядаючи їх з точки зору можливого відпочинку. Що стосується днів з понеділка по четвер, то ставлення до них як до робочих днів навряд чи зміниться. А от п’ятниця для багатьох асоціюється з наближенням відпочинку і високим ступенем позитивних емоцій. Субота є, безумовно, вихідним днем, протягом якого можна забути всі службові турботи. Що стосується неділі, то ближче до вечора ситуація змінюється – нерідко приходить думка: «Завтра потрібно рано вставати та йти на роботу» і настрій псується.

Отже, нечітку множину $\bar{A}$, яка описує вихідні дні тижня можна задати таким чином: $\bar{A}$={<понеділок;0>, <вівторок;0>, <середа;0>, <четвер;0>, <п’ятниця;0,5>, <субота;1,0>, <неділя;0,8>}. При цьому, в якості елементів універсума розглядаються всі дні тижня, а саме: X={понеділок, вівторок, середа, четвер, п’ятниця, субота, неділя}, а функція належності задається перерахуванням числових значень: чим ближчим є її значення до 1, тим більше відповідає той чи інший день тижня нашому ставленню до нього як до вихідного дня. Зобразимо нечітку множину $\bar{A}$ графічно (рис. 2.3). Для цього на горизонтальній осі відкладемо окремі значення елементів універсума X, а на вертикальній – відповідні їм значення функції належності $\mu_{\bar{A}}(x)$.

Зазначимо, що нечітка множина $\bar{A}$ може мати й такий вигляд: $\bar{A}$={<понеділок;0>, <вівторок;0>, <середа;0>, <четвер;0,1>, <п’ятниця;0,6>, <субота;1,0>, <неділя;0,7>}.

Приклад 2.2. Визначимо нечітку множину $\overline{B}$, яка описує “гарячу каву”. В якості універсума в цьому випадку, природно розглянути шкалу температур $X=\{x|0°C<x<100°C\}$. Очевидно, що окрема чашка кави, скажімо x1, з температурою 10°C не може бути визнана гарячою, тому для неї значення функції належності множини $\overline{B}$ буде дорівнювати нулю, тобто $\mu_{\overline{B}}(x_1)=0$. З іншого боку, чашка кави x_2 з температурою 90°C цілком може бути визнана гарячою, тому для неї значення функції належності множини $\overline{B}$ буде дорівнювати 1 (тобто $\mu_{\overline{B}}(x_2)=1$).

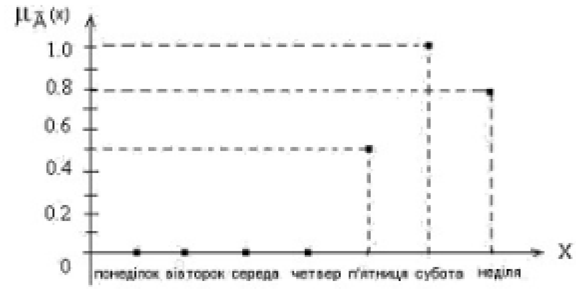


Рисунок 2.3 – Графічне зображення нечіткої множини $\bar{A}$

Ситуація, яка стосується температур, розташованих між значеннями 10°С та 90°С, є винятково суб'єктивною й невизначеною, оскільки, наприклад, чашка кави з температурою 55°С для однієї людини може виявитися гарячою, а для іншої – ні. Саме в цьому й проявляється "нечіткість" визначення відповідної множини. Проте можна з впевненістю стверджувати, що відповідна функція належності є монотонно зростаючою.

Отже, в якості нечіткої множини $\overline{B}=\{x;\ \mu_{\overline{B}}(x)\}$ можна розглянути нечітку множину. Такі фрази, як «висока температура», «високий тиск», «швидкісний автомобіль», «високооплачувана робота», «щедрі чайові», «престижний район», «смачна вечеря» породжують нечіткі множини.

Основні характеристики нечітких множин.

Нехай $\bar{A}$ – нечітка множина, задана на універсумі X.

Множиною α-рівня (α-розрізом) нечіткої множини $\bar{A}$ називається чітка множина $A\alpha=\{x\in X \mid \mu_{\bar{A}}(x)=\alpha\}$, де $\alpha \in(0,1]$.

Приклад 3. Нехай $X=\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Розглянемо нечітку множину $\bar{A}=\{\langle 1;1,0\rangle, \langle 2;1,0\rangle, \langle 3;0,9\rangle, \langle 4;0,8\rangle, \langle 5;0,6\rangle, \langle 6;0,5\rangle, \langle 7;0,4\rangle, \langle 8;0,2\rangle, \langle 9;0,1\rangle\}$, яка описує "невелике натуральне число".

Тоді множини α-рівня, які відповідають нечіткій множині $\bar{A}$, мають вигляд: $A_{1,0}=\{1, 2\}$, $A_{0,9}=\{1, 2, 3\}$, $A_{0,8}=\{1, 2, 3, 4\}$, $A_{0,6}=\{1, 2, 3, 4, 5\}$, $A_{0,5}=\{1, 2, 3, 4, 5, 6\}$, $A_{0,4} = \{1, 2, 3, 4, 5, 6, 7\}$, $A_{0,2} = \{1, 2, 3, 4, 5, 6, 7, 8\}$, $A_{0,1}=\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $A_0=\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Величина $h_{\bar{A}} = \sup_{x\in X}\{\mu_{\overline{B}}(x)\}$ називається висотою нечіткої множини $\bar{A}$. Наприклад, висота скінченної нечіткої множини $\bar{A}$, яка описує "невелике натуральне число" (див. приклад 3), дорівнює 1,0 і відповідає двом елементам універсума: 1 і 2.

Приклад 4. Розглянемо неперервну нечітку множину $\overline{C}$, яка описує «велике дійсне число» і має функцію належності:

$$\mu_{\bar{C}}(x) = \begin{cases} 0 \ \text{для} \ x \in [0, \ 1) \\ \dfrac{x-1}{x} \ \text{для} \ x \in R_+ \setminus [0, \ 1) \end{cases}$$

Висота нечітої множини $\bar{C}$ дорівнює 1,0, однак серед елементів універсума X=R_+ відсутні числа, для яких $\mu_{\bar{C}}$(x)=1,0.

Нечітка множина $\bar{A}$ називається нормальною, якщо максимальне значення її функції належності дорівнює 1, тобто $\exists x \in X$: $\mu_{\bar{A}}$(x)=1,0.

2.3 Властивості операцій над нечіткими множинами.

Рівність і домінування нечітких множин. Дві нечіткі множини $\bar{A}$ ={x; $\mu_{\bar{A}}$(x)} і $\bar{B}$={x; $\mu_{\bar{B}}$(x)} називаються рівними, якщо їхні функції належності приймають однакові значення на всьому універсумі X, тобто

$\forall x_i \in X$: $\mu_{\bar{A}}(x_i)$= $\mu_{\bar{B}}(x_i)$.

У цьому випадку використовують позначення $\bar{A} = \bar{B}$.

Нечітка множина $\bar{A}$ ={x_i; $\mu_{\bar{A}}(x_i)$} є нечіткою підмножиною нечіткої множини $\bar{B}$ ={x_i; $\mu_{\bar{B}}(x_i)$} якщо $\forall x \in X_i$: $\mu_{\bar{A}}(x_i) \leq \mu_{\bar{B}}(x_i)$.

У цьому випадку говорять, що нечітка множина $\bar{B}$ домінує над нечіткою множиною $\bar{A}$, а нечітка множина $\bar{A}$ міститься в нечіткій множині $\bar{B}$ і позначають $\bar{A} \leq \bar{B}$. Наприклад, для скінченних нечітких множин $\overline{A_1}$ й $\overline{A_2}$, які описують «невелике натуральне число» і мають вигляд:

$\overline{A_1}$={<1;1,0>, <2;1,0>, <3;0,9>, <4;0,8>, <5;0,6>, <6;0,5>, <7;0,4>, <8;0,2>, <9;0,1>} і $\overline{A_2}$={<1;1,0>, <2;0,9>, <3;0,8>, <4;0,7>, <5;0,5>, <6;0,4>, <7;0,3>, <8;0,2>, <9;0,1>},

має місце наступне відношення домінування: $\overline{A_2} \leq \overline{A_1}$.

Нехай $\bar{A}$={x; $\mu_{\bar{A}}$(x)}, $\bar{B}$={x; $\mu_{\bar{B}}$(x)} – довільні нечіткі множини, визначені на універсумі X.

Перетином нечітких множин $\bar{A}$ і $\bar{B}$ називається нечітка множина $\bar{C}$, задана на универсумі X, функція належності якої визначається за формулою:

$\forall x_i \in X$: $\mu_{\bar{C}}(x_i) = \min \{ \mu_{\bar{A}}(x_i), \mu_{\bar{B}}(x_i)\}$,

або $\forall x_i \in X$: $\mu_{\bar{A} \cup \bar{B}}(x_i) = \mu_{\bar{A}}(x_i) \wedge \mu_{\bar{B}}(x_i)$.

Позначають $\bar{C} = \bar{A} \cap \bar{B}$. Операцію перетину нечітких множин іноді називають min-перетином.

Приклад 2.3. Розглянемо нечітку множину $\bar{A}$, яка описує “невелике натуральне число”: $\bar{A}$ ={<1;1,0>, <2;1,0>, <3;0,9>, <4;0,8>, <5;0,6>, <6;0,5>, <7;0,4>, <8;0,2>, <9;0,1>}, і нечітку множину $\bar{B}$, яка описує “натуральне число, що приблизно дорівнює двом”: $\bar{B}$={<1;0,5>,

<2;1,0>, <3;0,6>, <4;0,4>, <5;0,2>, <6;0>, <7;0>, <8;0>, <9;0>}. Тоді нечітка множина $\bar{C}=\bar{A}\cap\bar{B}$ має вигляд: $\bar{C}$={<1;0,5>, <2;1,0>, <3;0,6>, <4;0,4>, <5;0,2>, <6;0>, <7;0>, <8;0>, <9;0>}. Змістовно нечітка множина $\bar{C}$ може описувати «невелике натуральне число, що приблизно дорівнює двом».

Об'єднанням нечітких множин $\bar{A}$ і $\bar{B}$ називається нечітка множина $\bar{C}$, задана на универсумі X, функція належності якої визначається за формулою:

$\forall x_i \in X$: $\mu_{\bar{C}}(x_i) = max\{\mu_{\bar{A}}(x_i), \mu_{\bar{B}}(x_i)\}$,

або $\forall x_i \in X$: $\mu_{\bar{A}\cup\bar{B}}(x_i) = \mu_{\bar{A}}(x_i) \vee \mu_{\bar{B}}(x_i)$.

Позначають $\bar{C}=\bar{A}\cup\bar{B}$. Операцію об'єднання нечітких множин іноді називають max-об'єднанням.

Приклад 2.4. Розглянемо нечіткі множини $\bar{A}$ і $\bar{B}$ з попереднього прикладу. Тоді нечітка множина $\bar{C}=\bar{A}\cup\bar{B}$ має вигляд: $\bar{C}$={<1;1,0>, <2;1,0>, <3;0,9>, <4;0,8>, <5;0,6>, <6;0,5>,<7;0,4>,<8;0,2>;<9;0,1>}. Змістовно нечітка множина $\bar{C}$ може описувати "невелике натуральне число або натуральне число, що приблизно дорівнює двом".

Існують й інші (альтернативні) способи визначення операцій перетину/об'єднання нечітких множин, серед яких можна виділити алгебраїчний перетин/об'єднання та граничний перетин/об'єднання нечітких множин.

Алгебраїчним добутком нечітких множин $\bar{A}$ і $\bar{C}$ називається нечітка множина $\bar{C}$, задана на універсумі X, функція належності якої визначається за формулою:

$\forall x_i \in X$: $\mu_{\bar{C}}(x_i) = \mu_{\bar{A}}(x_i) \cdot \mu_{\bar{B}}(x_i)$.

Позначають $\bar{C} = \bar{A} \cdot \bar{B}$.

Приклад 2.5. Розглянемо нечіткі множини $\bar{A}$ і $\bar{B}$ з попереднього прикладу. Тоді нечітка множина $\bar{C} = \bar{A} \cdot \bar{B}$ має вигляд: $\bar{C}$ = {<1;0,5>, <2;1,0>, <3;0,54>, <4;0,32>, <5;0,12>, <6;0>, <7;0>, <8;0>, <9;0>}.

Алгебраїчною сумою нечітких множин $\bar{A}$ і $\bar{B}$ називається нечітка множина $\bar{C}$, задана на універсумі X, функція належності якої визначається за формулою:

$\forall x_i \in X$: $\mu_{\bar{C}}(x_i) = \mu_{\bar{A}}(x_i) + \mu_{\bar{B}}(x_i) - \mu_{\bar{A}}(x_i) \cdot \mu_{\bar{B}}(x_i)$.

Позначають $\bar{C} = \bar{A} + \bar{B}$.

Приклад 2.6. Розглянемо нечіткі множини $\bar{A}$ і $\bar{B}$. Тоді нечітка множина

$\bar{C} = \bar{A} + \bar{B}$ має вигляд: $\bar{C}$ ={<1;1,0>, <2;1,0>, <3;0,96>, <4;0,88>, <5;0,68>, <6;05>, <7;04>, <8;02>, <9;01>}.

Доповненням нечіткої множини $\bar{A}$ називається нечітка множина $\bar{A}^{\neg}$ ={x_i; $\mu_{\bar{A}^{\neg}}(x_i)$}, функція належності якої визначається за формулою:

$\mu_{\bar{\acute{A}}}(x_i) = 1 - \mu_{\bar{A}}(x_i)$

Приклад 2.7. Розглянемо нечіткі множини $\bar{A}$ і $\bar{B}$. Тоді їхні доповнення будуть мати вигляд: $\bar{\acute{A}}$={<1;0>, <2;0>, <3;0,1>, <4;0,2>, <5;0,4>, <6;0,5>, <7;0,6>, <8;0,8>, <9;0,9>} і $\bar{\acute{B}}$={<1;0,5>, <2;0>, <3;0,4>, <4;0,6>, <5;0,8>, <6;1,0>, <7;1,0>, <8;1,0>, <9;1,0>}. Змістовно нечітка множина $\bar{\acute{A}}$ описує "натуральне число, яке не є невеликим", а нечітка множина $\bar{\acute{B}}$ – "натуральне число, яке не дорівнює приблизно двом".

Операцією концентрування CON($\bar{A}$) називається операція, результатом якої є нечітка множина $\bar{C}$, функція належності якої визначається за формулою:

$\forall x_i \in X$: $\mu_{\bar{C}}(x_i) = (\mu_{\bar{A}}(x_i))^2$.

Приклад 2.8. Розглянемо нечітку множину $\bar{A}$. Тоді CON($\bar{A}$)={<1;1,0>, <2;1,0>, <3;0,81>, <4;0,64>, <5;0,36>, <6;0,25>, <7;0,16>, <8;0,04>, <9;0,01>}.

Операцією розтягу DIL($\bar{A}$) називається операція, результатом якої є нечітка множина`C, функція належності якої визначається за формулою:

$\forall x_i \in X$: $\mu_{\bar{C}}(x_i) = (\mu_{\bar{A}}(x_i))^{1/2}$.

Приклад 2.9. Розглянемо нечітку множину $\bar{A}$.

Тоді DIL($\bar{A}$)={<1;1,0>, <2;1,0>, <3;0,949>, <4;0,894>, <5;0,775>, <6;0,707>, <7;0,632>, <8;0,447>, <9;0,316>}.

Різницею нечітких множин $\bar{A}$ і $\bar{B}$ називається нечітка множина $\bar{C}$, задана на універсумі X, функція належності якої визначається за формулою:

$\forall x_i \in X$: $\mu_{\bar{C}}(x_i) = \max\{\mu_{\bar{A}}(x_i) - \mu_{\bar{B}}(x_i), 0\}$.

Позначають $\bar{C} = \bar{A} \backslash \bar{B}$.

Приклад 2.10. Розглянемо нечіткі множини $\bar{A}$ і $\bar{B}$

Тоді нечітка множина $\bar{C} = \bar{A} \backslash \bar{B}$ має вигляд: $\bar{C}$={<1;0,5>, <2;0>, <3;0,3>, <4;0,4>, <5;0,4>, <6;0,5>, <7;0,4>, < 8;0,2>; <9;0,1>}. Змістовно нечітка множина C може описувати "невелике натуральне число, яке не є приблизно рівним двом".

Симетричною різницею нечітких множин $\bar{A}$ і $\bar{B}$ називається нечітка множина $\bar{C}$, задана на універсумі X, функція належності якої визначається за формулою:

$\forall x_i \in X$: $\mu_{\bar{C}}(x_i) = |\mu_{\bar{A}}(x_i) - \mu_{\bar{B}}(x_i)|$.

Позначають $\bar{C} = \overline{A \Theta B}$.

Позначають $\bar{C} = \overline{A \Theta B}$, причому $\overline{A \Theta B}$ =($\bar{A} \backslash \bar{B}$)∪ ($\bar{B} \backslash \bar{A}$).

Приклад 2.11. Розглянемо нечіткі множини $\bar{A}$ і $\bar{B}$. Тоді нечітка множина $\overline{C} = \overline{A\Theta B}$ має вигляд: $\overline{C}$={<1;0,5>, <2;0>, <3;0,3>, <4;0,4>, <5;0,4>, <6;0,5>, <7;0,4>, <8;0,2>; <9;0,1>}. Змістовно нечітка множина $\overline{C}$ може описувати «невелике натуральне число, яке не є приблизно рівним двом».

Існують й інші способи визначення розглянутих вище операцій над нечіткими множинами. Допускається також узагальнене зображення нечітких теоретико-множинних операцій через так звані нечіткі оператори. Ці оператори діють на множині значень функцій належності і тому їх можна застосовувати до функцій належності довільних нечітких множин.

Література: [3, 4, 6-8, 10, 11, 18-20].

ТЕМА 3. КОМБІНАТОРИКА

Комбінаторика – це наука, основним завданням якої є перерахунок і перелічення елементів у скінченних множинах.

Виникла комбінаторика у XVII ст. Але у самостійну наукову дисципліну сформувалася лише у XX ст.

Виділяють такі 3 проблеми комбінаторики:

– задачі на перелічення, в яких необхідно визначити кількість розміщень елементів скінченної множини, що задовольняють певні умови;

– задачі про існування та побудову. В задачах такого класу розглядається питання: чи має місце визначена конфігурація частин скінченних множин з деякими властивостями; якщо така конфігурація існує, то як її побудувати;

– задачі про вибір. Задачі такого типу досліджують умови, за яких можна здійснити вибір підмножини або деякої сукупності частин множини так, щоб задовольнити певні вимоги.

3.1 Класичні задачі комбінаторики

Магічний квадрат. Розмістити числа 1,2,3,4,5,6,7,8,9, у вигляді квадрату так, щоб сума чисел будь-якого із стовпців і діагоналей була однією й тією ж.

4	9	2
3	5	7
8	1	6

Шахові задачі. Добре відома задача про ферзі: Поставити на шахову дошку найбільшу кількість ферзів таким чином, щоб кожен з них не міг 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 (рис. 3.1). Задачу можна розв'язати прямим перебором варіантів, і виявиться, що 8 ферзів можна розмістити таким чином, причому всього є 92 варіанти такого розміщення.

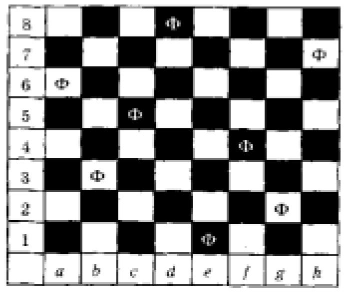


Рисунок 3.1 – Розміщення ферзів таким чином, що жоден з них не може взяти іншого

В загально прийнятих позначеннях вказаний варіант записується так:

(a,6), (b,3), (c,5), (d, 8), (e,1), (f,4), (g,2), (h,7).

Латинські квадрати. Квадрат розміром n рядків і n стовпців n*n, складений з n літер таким чином, щоб кожна буква входила лише один раз у кожний стовпець і кожний рядок, називають латинським.

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

3.2 Основні формули

Правило суми. Якщо об'єкт a може бути вибраний p способами, а об'єкт b – іншими q способами, то вибір «або a, або b» може бути здійснений $p+q$ способами.

Правило добутку. Якщо об'єкт a можна вибрати p способами і після кожного з таких виборів об'єкт b, у свою чергу, може бути вибраний q способами, то вибір «a та b» у вказаному порядку можна здійснити pq способами.

Це правило використовують тоді, коли способи вибору а та в – незалежні.

Доведення.

Нехай $a \in A$, $b \in B$, де $|A| = p$, $|B| = q$. Розглянемо множини $X_i\,(i = \overline{1, p})$ пар (x,b), де $x = a_i, b \in B$. Тоді множини X_i попарно не перетинаються, $|X_i| = q, i = \overline{1, p}$. Множина всіх пар $(x, y) \in X = \bigcup_{i=1}^{p} X_i$, і за правилом суми маємо $|X| = \left|\bigcup_{i=1}^{p} X_i\right| = \sum_{i=1}^{p} |X_i| = pq$.

Сформульоване та доведене правило добутку для послідовного вибору двох елементів може бути узагальнене на n елементів. Загалом правило добутку формулюється так:

Якщо об'єкт x_1 може бути вибраний n_1 способами, після чого об'єкт x_2 може бути вибраний n_2 способами, і для будь-якого i, де $2 \le i \le m-1$, після вибору об'єктів $x_1, x_2, \ldots, x_i$ об'єкт x_{i+1} може бути вибраний n_{i+1} способами, то вибір упорядкованої послідовності m об'єктів $(x_1, x_2, \ldots, x_m)$ може бути здійснений $n_1 \bullet n_2 \bullet \ldots \bullet n_m$ способами.

Набір елементів $x_{i_1}, x_{i_2}, \ldots, x_{i_r}$ із множини $X = \{x_1, x_2, \ldots, x_n\}$ називають вибіркою обсягом r, або r-вибіркою.

Вибірку r-елементів називають

r-перестановкою, якщо враховується порядок слідування елементів, і

r-сполученням, якщо беруться до уваги тільки елементи без урахування їхнього порядку;

n-перестановка з n різних елементів є просто перестановкою.

Приклад. Для множини $M = \{a,b,c\}$ розрізняють шість 3-перестановок, утворених з тих самих елементів: *abc, acb, bac, bca, cab, cba;* водночас ці вибірки є різними записами того самого 3-сполучення елементів *a,b,c*.

Вибірки можуть допускати і не допускати повторення елементів. У разі r-вибірок із повторенням розрізняють два випадки:

– запас елементів, що повторюються, обмежений і визначається специфікацією $\{n_1, n_2, \ldots, n_k\}$, де n_i – кількість елементів i-го вигляду.

Загальне число елементів початкової множини $n = n + n_2 + \ldots + n_k$ причому $k \le n$.

– запас елементів не обмежується й у вибірці з r-елементів допускається будь-яке число повторень, що не перевищує k. Початкову множину в цьому випадку можна розглядати як таку, що складається з різних елементів, але після вибірки деякого елемента він відновлюється в цій множині. Таку вибірку називають вибіркою з поверненням.

Розміщенням без повторень з n елементів по r називаються впорядковані r-вибірки без повторень. Їх число позначають A_n^r і обчислюються за формулою:

$$A_n^r = n(n-1)(n-2)\ldots(n-r+1) = \frac{n!}{(n-r)!}, r \le n$$

Доведемо це твердження. Кожна r-перестановка є впорядкованою послідовністю завдовжки r, члени якої – попарно різні й вибираються з n-елементної множини. Тоді перший член цієї послідовності може бути вибраний n способами, після кожного вибору першого члена послідовності другий – (n-1) способами і т. д. Відповідно після кожного вибору першого, другого і т. д. аж до (r–1)-го членів послідовності r-й член може бути вибраний $n-(r-1) = n-r+1$ способами, звідси за узагальненим правилом добутку дістаємо наведену вище формулу.

Звичайно розміщення без повторень з n елементів по n називаються перестановками з n елементів. Їх число обчислюється за формулою

$$P(n,n) = A_n^n = n!$$

Відмітимо, що цю задачу можна розв'язати інакше. Для вибору кольору першої полоси маємо 5 варіантів. Після зробленого вибору колір для другої полоси можна вибрати 4 способами з 4 що залишилися. Далі вибираємо колір для третьої полоси прапора з 3-х. Це можна зробити 3-ма способами. За правилом добутку маємо $5 \times 4 \times 3 = 60$ способів.

Розміщення з повтореннями з n елементів по r впорядковані r-вибірки з n елементів з повтореннями. Їх число позначається

$$\overline{A_n^r} = n^r \ .$$

Сполучення без повторень з n елементів по r називаються невпорядковані r-вибірки з n елементів без повторень. Їх число позначається C_n^r і обчислюється за формулою:

$$C_n^r = \frac{n!}{r!(n-r)!}, r \le n .$$

Сполучення без повторень з n елементів по r утворюють k-елементні підмножини вихідної множини потужності n.

Числа C_n^r називаються біноміальними коефіцієнтами.

Сполучення з повтореннями з n елементів по r називаються невпорядковані r-вибірки з n елементів з повтореннями.

Їх число позначається $\overline{C_n^r}$ і обчислюється за формулою:

$$\overline{C_n^r} = C_{n+r-1}^r = \frac{(n+r-1)!}{r!(n-1)!}, \ \forall r, n \in N .$$

Задача про цілочислові розв'язки

Цю задачу формулюють так: знайти кількість розв'язків рівняння $x_1 + x_2 + ... + x_r = n$ у цілих невід'ємних числах, де n – ціле невід'ємне число.

Узявши такі невід'ємні цілі числа $x_1, x_2, ..., x_n$, що $x_1 + x_2 + ... + x_r = n$, можна одержати сполучення з повтореннями з r елементів по n, а саме: елементів першого типу – x1 одиниць, другого – x2, ..., r-го – x_r. Навпаки, якщо є сполучення з повтореннями з r елементів по n, то кількість елементів кожного типу задовольняють вимоги рівняння $x_1 + x_2 + ... + x_r = n$ у цілих невід'ємних числах. Отже, кількість цілих невід'ємних розв'язків цього рівняння дорівнює:

$$H_r^n = C_{n+r-1}^n = \frac{(r+n-1)!}{n!(r-1)!}.$$

Перестановки з повтореннями. Розглянемо задачу: Маємо предмети r різних видів. Скільки різних комбінацій (перестановок) можна зробити з n_1 предметів 1-го виду, з n_2 предметів 2-го виду, ..., з n_r предметів r-го виду? Число предметів в кожній перестановці $n = n_1 + n_2 + ... + n_r$. Такі комбінації називаються перестановками з повтореннями. Їх число позначається $P(n_1, n_2, ..., n_r)$ і обчислюється за формулою:

$$P(n_1, n_2, ..., n_r) = \frac{n!}{n_1! n_2! ... n_r!}.$$

Приклад. Скількома способами можна розмістити в ряд 5 чорних, 4 білих і 3 червоних фішки?

Розв'язок. Ця задача на перестановки з повтореннями. Маємо фішки 3 різних видів: чорних $n_1 = 5$, білих $n_2 = 4$, червоних $n_3 = 3$. Всього фішок 12. Отже за формулою маємо $P(5,4,3) = \frac{12!}{5!4!3!} = 27720$ способів.

Зауваження. Якщо $n_1 = r$, $n_2 = n - r$, то маємо $P(r, n-r) = C_n^r$.

Біном Н'ютона. Числа C_n^r виникають як коефіцієнти при розкритті дужок в біномі $(a+b)^n$.

Наприклад

$$\begin{aligned}(a+b)^3 &= (a+b)(a+b)(a+b) = \\ &= aaa + aab + aba + abb + baa + bab + bba + bbb = \\ &= a^3 + 3a^2b + 3ab^2 + b^3\end{aligned}$$

Кожен з 8 доданків в 2-му рядочку перетворень отримали при перемноженні 3-х змінних, які вибираємо по одному з кожної дужки. Видно, що рівно 3 доданки містять одну змінну a і дві b. Це відбувається тому що у нас є $C_3^2 = 3$ способи вибору двох дужок із трьох, звідки ми беремо змінну b, а з тої що залишилася вибираємо a.

В загальному випадку, розкриваючи дужки в біномі $(a+b)^n$, ми будемо отримувати члени виду $a^{n-r}b^r$ (де r набуває кожне із значень від 0 до n при перемноженні символів b, взятих з r дужок, і a , взятих із (n-r) тих дужок, що залишилися. Так як є рівно C_n^r способів r дужок з n, то в нас буде в точності C_n^r членів виду $a^{n-r}b^r$ при $r = 0,1,...,n$. Звідси

$$(a+b)^n = C_n^0 a^n + C_n^1 a^{n-1} b + C_n^2 a^{n-2} b^2 + ... + C_n^n b^n.$$

або $(a+b)^n = \sum_{r=0}^{n} C_n^r a^r b^{n-r}$.

Ця формула називається біномом Н'ютона. Тому коефіцієнти C_n^r називають біноміальними коефіцієнтами.

Тепер узагальним біноміальну теорему для випадку знаходження коефіцієнтів розкладу $(x_1 + x_2 + x_3 + \ldots + x_m)^n$.

Теорема. Для заданого додатного числа n

$$(x_1 + x_2 + x_3 + \ldots + x_m)^n = \sum \frac{n!}{n_1! n_2! n_3! \ldots n_m!} x_1^{n_1} x_1^{n_2} x_1^{n_3} \ldots x_1^{n_m},$$

де сума взята по всім невід'ємним цілим числам $n_1, n_2, n_3, \ldots, n_m$, таким що $n_1 + n_2 + n_3 + \ldots + n_m = n$.

Арифметичний трикутник (трикутник Паскаля)

$$C_n^r = \frac{n!}{r!(n-r)}, \ r \le n \tag{3.1}$$

Послідовні значення C_n^r можна підрахувати за допомогою формул, які перевіряються безпосередньо за допомогою (3.1):

$$C_n^r = C_n^{n-r}, \ C_n^r = C_{n-1}^{r-1} + C_{n-1}^r \tag{3.2}$$

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. 2).

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 (2) $C_2^1 = C_1^0 + C_1^1$, тобто C_2^1 дорівнює сумі чисел попереднього рядка, що розміщені ліворуч та праворуч від нього: $C_2^1 = C_1^0 + C_1^1 = 1 + 2 = 2$.

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 рисунку 3.2.

$$
\begin{array}{ccccccccc}
 & & & & 1 & & & & \\
 & & & 1 & & 1 & & & \\
 & & 1 & & 2 & & 1 & & \\
 & 1 & & 3 & & 3 & & 1 & \\
1 & & 4 & & 6 & & 4 & & 1 \\
\_ & \_ & \_ & \_ & \_ & \_ & & &
\end{array}
$$

Рисунок 3.2 – 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.

Література: [1, 2, 4-6, 7, 9, 10].

ТЕМА 4. РЕКУРЕНТНІ СПІВВІДНОШЕННЯ

4.1 Розв'язування рекурентних рівнянь

Числову послідовність (a_n) можна задати рекурентним рівнянням (використовують також термін рекурентне співвідношення). Таке рівняння описує правило для знаходження елементів послідовності через один або декілька попередніх, при чому задано відповідну кількість початкових елементів.

Розв'язком рекурентного рівняння називають послідовність, яка задовольняє це рівняння. Інакше кажучи, послідовність задано рекурентною формулою, а потрібно знайти явний вираз для a_n через n.

Метод рекурентних рівнянь у комбінаториці полягає у зведенні комбінаторної задачі для меншої кількості об'єктів.

Приклад.

Розглянемо рекурентне співвідношення

$$a_n = a_{n-1} + 2a_{n-2}, \qquad n = 2,3,\dots \qquad a_0 = 2, \;\; a_1 = 7$$

Його розв'язок – послідовність $a_n = 3\cdot 2^n - (-1)^n$.

Справді

$a_0 = 3 - 1 = 2$, $a_1 = 6 + 1 = 7$

$$a_{n-1} + 2a_{n-2} = 3 \cdot 2^{n-1} - (-1)^{n-1} + 2 \cdot 3 \cdot 2^{n-2} - 2 \cdot (-1)^{n-2} =$$

$$= 3 \cdot 2^{n-1} - (-1)^{n-1} + 3 \cdot 2^{n-1} + 2 \cdot (-1)^{n-1} =$$

$$= 2 \cdot 3 \cdot 2^{n-1} + (-1)^{n-1} = 3 \cdot 2^n - (-1)^n$$

4.2 Числа Фібоначчі

Цю задачу розв'язував у 1202 ст. Леонардо Пізанський, відомий як Фібоначчі.

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:

$u_n = u_{n-1} + u_{n-2}$,

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 $u_0 = 0$, $u_1 = 1$:

0,1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, ... (*)

Наведемо явну формулу для обчислення довільного числа Фібоначчі, яку вперше одержав Біне. Для послідовності (*) формула має вигляд

$$u_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n, \ n = 0,1,2,\ldots$$

До чисел Фібоначчі приводить, наприклад, така задача теорії інформації. Позначимо через u_n, кількість повідомлень, які можна передати за допомогою n сигналів 0 і 1 так, щоб при цьому ніде не опинилися поряд два сигнали 0. Зрозуміло, що $u_0 = 1$ (відсутність сигналу розглядається як одне «порожнє» повідомлення), $u_1 = 1$. Всі повідомлення «довжиною» (n+2) кількістю u_{n+2} можна поділити на два класи:

а) ті, в яких перший сигнал є 1; їх кількість дорівнює u_{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 u_n.

Отже, $u_{n+2} = u_{n+1} + u_n$, тому u_n є числами Фібоначчі.

Загального методу розв'язування рекурентних співвідношень немає. Проте певний клас рівнянь можна розв'язувати однаковим методом.

Рекурентне рівняння називають лінійним однорідним порядку k зі сталими коефіцієнтами, якщо воно має вигляд

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}$$

де $c_1, c_2, \ldots, c_k$ – дійсні числа та $c_k \neq 0$.

Приклад.

$a_n = 1{,}11 a_{n-1}$ – лінійне однорідне першого порядку

$a_n = a_{n-1} + a_{n-2}$ – лінійне однорідне другого порядку

$a_n = a_{n-5}$ – лінійне однорідне п'ятого порядку

$a_n = a_{n-1} + a_{n-1}^2$ – нелінійне

$a_n = a_{n-1} + a_{n-2} a_{n-3}$ – нелінійне

$a_n = 2a_{n-1} + 1$ – лінійне неоднорідне

4.3 Розв'язок рекурентного рівняння k порядку

Розв'язок рекурентного рівняння k порядку називають загальним, якщо він залежить від k довільних сталих $B_1, B_2, ..., B_k$ і будь-який його розв'язок можна одержати підбором цих сталих.

Щоб рекурентне рівняння визначало конкретну послідовність, достатньо задати k початкових умов: $a_0 = A_0, a_1 = A_1, ..., a_{k-1} = A_{k-1}$. Із цих умов і визначають сталі $B_1, B_2, ..., B_k$.

Теорема. Якщо послідовності $a_n^{(1)}, a_n^{(2)}, ..., a_n^{(p)}$ – це розв'язки рекурентного рівняння $a_n = c_1 a_{n-1} + c_2 a_{n-2} + ... + c_k a_{n-k}$, то для довільних чисел $B_1, B_2, ..., B_p$ послідовність $a_n = B_1 a_n^{(1)} + B_2 a_n^{(2)} + ... + B_p a_n^{(p)}$ також являє собою розв'язок цього рівняння.

Теорема. Якщо число r_1 – корінь рівняння $r^k = c_1 r^{k-1} + c_2 r^{k-2} + ... + c_k$, то послідовність $r_1^n (n = 1, 2, ...)$ – розв'язок рекурентного рівняння.

<u>Приклад</u>. Послідовність чисел Фібоначчі задає рекурентне співвідношення другого порядку $u_n = u_{n-1} + u_{n-2}$ з початковими умовами

$u_0 = 0$, $u_1 - 1$. Характеристичне рівняння $r^2 = r + 1$, тобто $r^2 - r - 1 = 0$, звідки випливає, що

$$r_1 = \frac{1+\sqrt{5}}{2}, \quad r_2 = -\frac{1-\sqrt{5}}{2}.$$

Отже $u_n = B_1 \left(\frac{1+\sqrt{5}}{2}\right)^n - B_1 \left(\frac{1-\sqrt{5}}{2}\right)^n$.

Для визначення констант B_1 і B_2 скористаємося початковими умовами

$$\begin{cases} B_1 + B_2 = 0 \\ B_1 \left(\frac{1+\sqrt{5}}{2}\right) + B_2 \left(\frac{1-\sqrt{5}}{2}\right) = 1 \end{cases}$$

Отримаємо $B_1 = \frac{1}{\sqrt{5}}$, $B_2 = -\frac{1}{\sqrt{5}}$.

Отже $u_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$, $n = 0,1,2,\ldots$.

Література: [2, 4, 6, 7, 10].

ТЕМА 5. ОСНОВИ ТЕОРІЇ ВІДНОШЕНЬ

5.1 Основні поняття теорії відношень

Відношення – одне з основних понять сучасної математики. Мову відношень використовують для опису зв'язків між об'єктами та поняттями. Найзручніший спосіб задати зв'язок між елементами двох множин – записати впорядковані пари елементів, що перебувають у цьому зв'язку.

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.

Декартовим добутком множин А та В (позначають $A \times B$) називають множину всіх пар (a,b), таких, що $a \in A$, $b \in B$. Зокрема, якщо А=В, то обидві компоненти належать до А. Такий добуток позначають через A^2 і називають декартовим квадратом множини А. Аналогічно, декартовим добутком множин $X_1,\ldots,X_n$ (позначають $X_1 \times \ldots \times X_n$) називають множину всіх впорядкованих наборів (кортежів) $x_1,\ldots,x_n$ довжини n таких, що $x_1 \in X_1,\ldots,x_n \in X_n$.

Частковий випадок $X \times \ldots \times X$ позначають X^n і називають *n-м* степенем множини *X*.

Приклад.

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

Тоді

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

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

$$\begin{aligned} A \times B \times C = \{&(a_1, b_1, c_1), (a_1, b_1, c_2), (a_1, b_2, c_1), (a_1, b_2, c_2),\\ &(a_2, b_1, c_1), (a_2, b_1, c_2), (a_2, b_2, c_1), (a_2, b_2, c_2),\\ &(a_3, b_1, c_1), (a_3, b_1, c_2), (a_3, b_2, c_1), (a_3, b_2, c_2)\}\end{aligned}$$

n-арне відношення *R* на множинах $X_1 \times \ldots \times X_n$ – це підмножина декартового добутку цих *n* множин: $R \subseteq X_1 \times \ldots \times X_n$.

Якщо набір елементів $(x_1, x_2, \ldots, x_n)$ належить відношенню *R*, то стверджують, що елементи $x_1, \ldots, x_n$ знаходяться у відношенні *R*.

Під n- арним відношенням R на множині *X* розуміється підмножина *n-го* степеня цієї множини: $R \subseteq X^n$.

Якщо $n = 1$, то відношення називається унарним, якщо $n = 2$ – бінарним. Зауважимо, що унарне відношення *R* на множині *X* - це підмножина в самому *X*: $R \subseteq X$.

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 декартового добутку множин.

5.2 Способи задання бінарних відношень

Якщо R – бінарне відношення на множинах $X,У,$ 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.

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, $x \in A$, $y \in B$.

Приклад. 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 У може бути задане за допомогою матриці $(W = W(R))$, рядки якої відповідають елементам множини X, стовпці – елементам множини У. Якщо $n = |X|$, $m = |Y|$ – кількість елементів множин X і Y відповідно, то довільна матриця W має розмір $n \times m$. Елемент w_{ij} матриці відповідає парі $(x_i, y_j) \in A \times B$ декартового 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, y_j елементи множин X і Y. Якщо пара (x_i, y_j) належить відношенню R, з'єднаємо точки x, y 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.

Бінарні відношення можна задавати графічно за допомогою діаграм Хассе.

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. 5.1,б. Може трапитися, що $R = \varnothing$, – таке відношення називається порожнім. При $|A| = 5$ граф зображено на рис. 5.1,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. 5.1,a для $|A| = 5$.

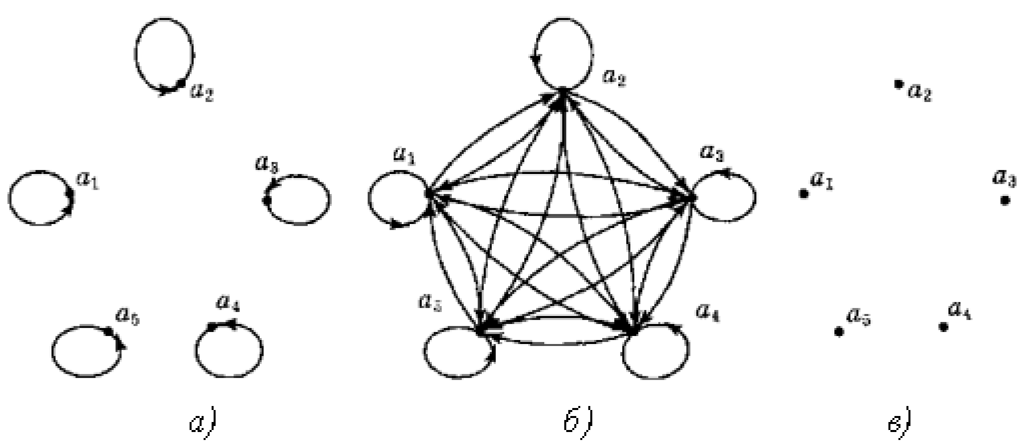


Рисунок 5.1 – Граф підношення: а) тотожного: б) повного: в)порожнього.

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.

Графік відношення

Кожній точці на площині відповідають її координати – впорядкована пара (x_i, y_j). Отже відношення на множині R можна відобразити на площині деякою конфігурацією чи множиною точок.

На множині дійсних чисел R множину точок, що відповідають елементам відношення називають графіком цього відношення (рис. 5.2).

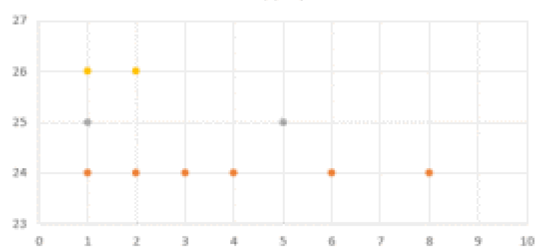


Рисунок 5.2 – Графік відношення

5.3 Операції над відношеннями

Областю визначення бінарного відношення називають множину $X_R = \{x | \exists y : xRy\}$.

Областю значень бінарного відношення називають множину $Y_R = \{x | \exists x : yRx\}$.

Для бінарних відношень означено теоретико-множинні операції об'єднання $\cup$, перетин $\cap$, різниці.

$R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 \setminus R_2$.

Доповненням $\overline{R}$ до відношення R називають множину $\overline{R} = (A \times B) \setminus R$, тобто $x\overline{R}y$ виконується для всіх пар *(x,y)*, які не належать *R*.

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 елементам другої множини, розташовуються праворуч. Таким чином, дуги графа спрямовані зліва направо.

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 рисунку 5.3.

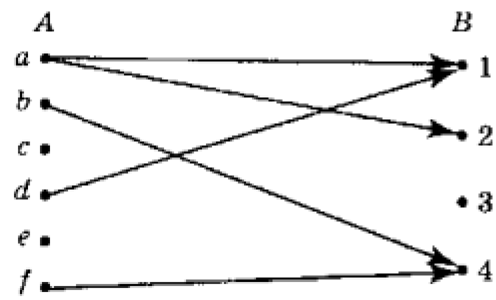


Рисунок 5.3 – Граф відношення $R = \{(a,1), (a,2), (b,4), (d,1), (f,4)\}$

Для того, щоб побудувати граф відношення R^{-1}, змінимо напрямки дуг (рис. 5.4).

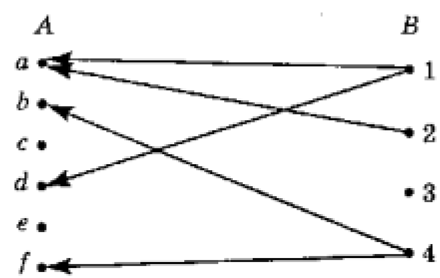


Рисунок 5.4 – **Граф відношення** $R = \{(1, a), (2, a), (4, b), (1, d), (4, f)\}$

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. 5.5. зобразивши на ньому крім графа відношення *R*, граф відношення S.

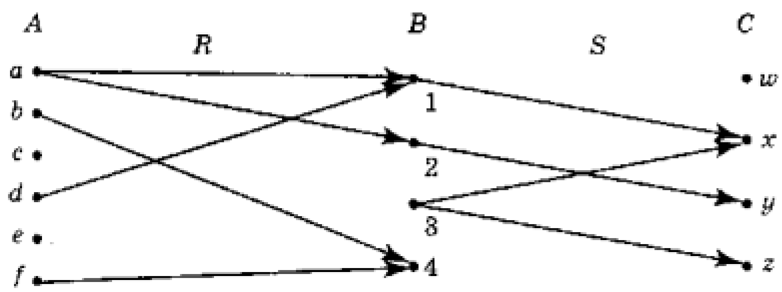


Рисунок 5.5 – Граф відношення R і відношення $S=\{(1,x),(2,y),(3,x),(3,z)\}$

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 $R\subseteq A\times B$, $S\subseteq B\times C$, зображених на рис. 6, композиція $S\circ R$ є відношення, що зображене на рис. 6 і є підмножиною декартового добутку $A\times C$.

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 $S\circ R$ (рис. 5.6).

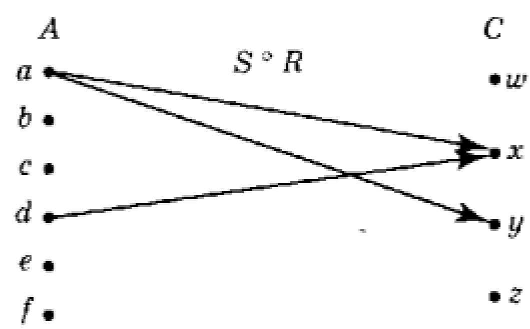


Рисунок 5.6 – Граф відношення $S\circ R$

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

Нехай R – деяке відношення, визначене на множині X: R. 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 $n = 1,2,...$

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

Нехай $R \subseteq X \times Y$ - 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 під множили 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 $A = \{a, b, c, d, e, f\}$, $B = \{1,2,3,4\}$, що задане графом (рис. 5.7).

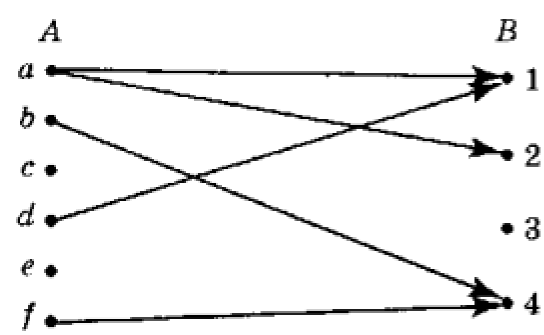


Рисунок 5.7 – Граф відношення $R = \{(a,1), (a,2), (b,4), (d,1), (f,4)\}$

Можна одержати такі перерізи:

$R(a) = \{1,2\}$, $R(b) = \{4\}$, $R(d) = \{1\}$, $R(e) = \varnothing$, $R(f) = \{4\}$.

Фактор-множина $B / R = \{\{1,2\}, \{4\}, \{1\}, \varnothing\}$.

5.4 Властивості бінарних відношень

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.

$$(R - рефлексивне) \leftrightarrow (\forall x \in X)(xRx)$$

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).

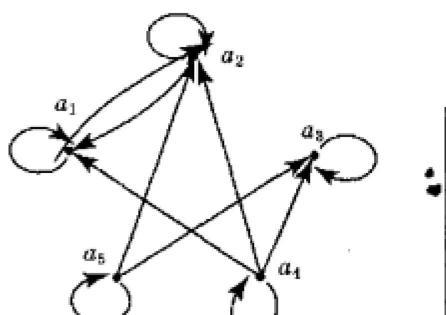
2. Антирефлексивність

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

$$(R - антирефлексивне) \leftrightarrow (\forall x \in X)(x\overline{R}x)$$

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).

Приклад. Рефлексивним є відношення, представлене на рис. 5.8.



	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

Рисунок 5.8 – Приклад графа і матриці рефлексивного відношення

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 $x_1 R x_2$ виходить $x_2 R x_1$ (тобто для будь-якої пари R виконується або в обидві боки, або не виконується взагалі).

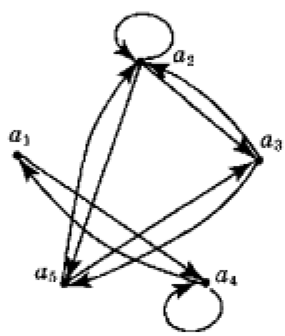
$$(R - симетричне) \leftrightarrow (\forall x_1, x_2 \in X)(x_1 R x_2 \to x_2 R x_1)$$

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.

Приклади:

1. рівності a=b, то b=a.
2. паралельність $a \| b \to b \| a$.
3. $a \perp b \to b \perp a$

Приклад. Приклад симетричного відношення зображено на рис. 5.9.



	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

Рисунок 5.9 – Приклад графа і матриці симетричного відношення

4. Несиметричність

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

$$(R - несиметричне) \leftrightarrow (\exists x_1, x_2 \in X)(x_1 R x_2 \to x_2 \overline{R} x_1)$$

5. Антисиметричність

Відношення R називається антисиметричним, якщо з $x_1 R x_2$ і $x_2 R x_1$ виходить, $x_2 = x_1$.

$$(R - антисиметричне) \leftrightarrow (\forall x_1, x_2 \in X)(x_1Rx_2 \wedge x_2Rx_1) \rightarrow (x_1 = x_2)$$

Приклад антисиметричного відношення – відношення $\le$ на множині дійсних чисел: якщо а $a \le b \wedge b \le a \rightarrow a - b$.

Відношення < на дійсній вісі є несиметричним, оскільки якщо $a < b \rightarrow b < a$ не виконується.

6. Транзитивність

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

$$(R - транзитивне) \leftrightarrow (\forall x_1, x_2, x_3 \in X)(x_1Rx_2 \wedge x_2Rx_3) \rightarrow (x_1Rx_3)$$

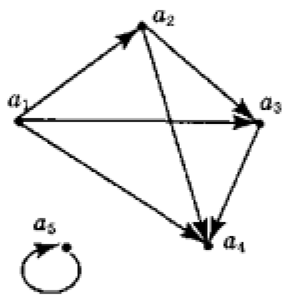
У графі, що задає транзитивне відношення B, для будь-якої пари дуг, таких, що кінець першої співпадає з початком другої, існує третя дуга, що має спільний початок з першою і спільний кінець з другою.

Приклад.

1. Відношення $\le$ і < на множині дійсних чисел транзитивні: якщо $a \le b \wedge b \le c \rightarrow a \le c$.

2. $a \| b \wedge b \| c \rightarrow a \| c$

Приклад. Приклад транзитивного відношення зображено на рис. 5.10.



	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

Рисунок 5.10 – Приклад графа і матриці транзитивного відношення

7. Антитранзитивність

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

$$(R - антитранзитивне) \leftrightarrow (\forall x_1, x_2, x_3 \in X)(x_1Rx_2 \wedge x_2Rx_3) \rightarrow (x_1\overline{R}x_3)$$

Приклад.

1. Відношення «рівність», «бути дільником», що задані на множині цілих чисел,

2. Відношення «жити в одному місті», що задане на множині людей, – транзитивні;

3. відношення «бути сином» – антитранзитивне.

4. $a \perp b \rightarrow b \perp c \rightarrow a \| c$

Відношення еквівалентності, порядку, толерантності

Розглянемо деякі найчастіше використовувані класи бінарних відношень.

Бінарне відношення, що має властивості рефлексивності, симетричності і транзитивності, називається відношенням еквівалентності (позначається $\approx$).

Нехай задана множина А 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 А, полягає в тому, що воно розбиває множину А на неперетинні підмножини, які називаються Класом еквівалентності породженим елементом a з множини А називається множина всіх елементів з А еквівалентних елементу a.

$[a] \leftrightarrow \{x \in A | x \approx a\}$

У випадку скінченної множини А розбиття її на класи еквівалентності відбувається таким чином. Нехай на множині А 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 А 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 А 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 А не залишиться жодного елемента, що не входить до одного з класів C_i. Вийде система класів $C_1, C_2, \ldots, C_n$. 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.

Відношення $a \| b$.

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

– рефлексивне ($a \le a$);

– антисиметричне (якщо $a \le b$ і $b \le a$, то $a = b$);

– транзитивне (якщо $a \le b$ і $b \le c$, то $a \le c$).

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 $a < b < c$, і елемент с 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. 5.11 зображено граф відношення включення на множині 2^A всіх підмножин триелементної множини $A = \{a, b, c\}$.

Із властивістю транзитивності пов'язане поняття шляху в графі.

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.

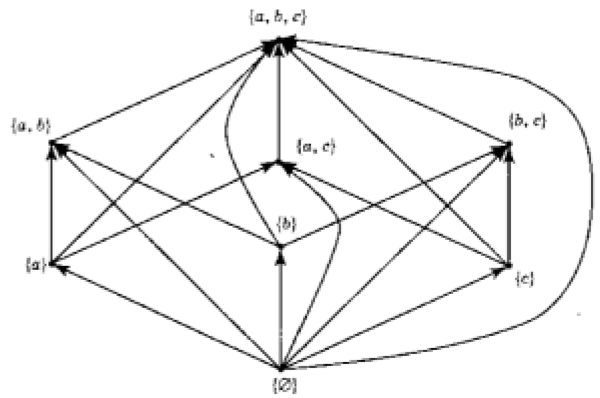


Рисунок 5.11 – Граф відношення включення множин

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. 4 можна зобразити за допомогою діаграми Хассе, як зображено на рис. 5.12.

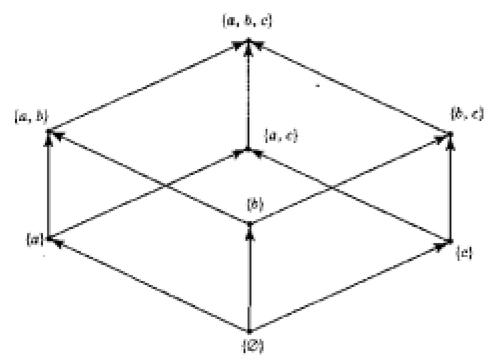


Рисунок 5.12 – Діаграма Хассе відношення включення множин

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

Елементи a, b називаються порівнянними у відношенні часткового порядку R, якщо виконується хоча б одне із співвідношень aRb або bRa. Множина А, 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 А 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. 5.13 розглянуту множину N задано за допомогою графа відношень і діаграм Хассе.

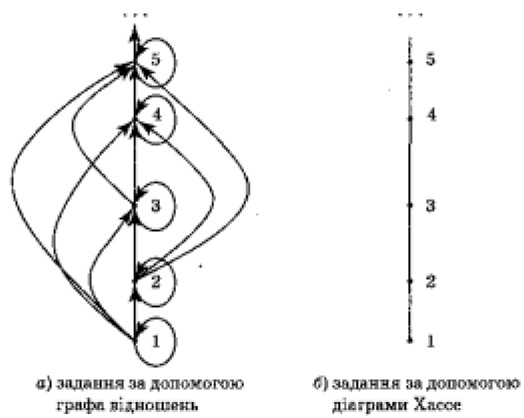


Рисунок 5.13 – Лінійно упорядкована множина натуральних чисел

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:

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

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

– 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.

Відношення називається відношенням толерантності, якщо воно: рефлексивне, симетричне, 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.

Література: [4, 6, 7, 10, 11].

ТЕМА 6. ЛОГІКА ВИСЛОВЛЮВАНЬ

6.1 Базові поняття логіки висловлювань

Логіка висловлювань застосовується до простих висловлювань, де базисні висловлювання – або істинні або хибні. Твердження, які мають одну або більше змінних, можуть бути істинними для деяких значень змінних і хибними для інших.

Предикатом називається твердження, в якому змінні приймають значення істинне чи хибне в залежності від значень змінних і позначають P(x).

Квантори «∀» і «∃» зв'язують перемінну x, перетворюючи одномісний предикат у висловлення.

Вирази «для всіх» і «існує» називаються *кванторами* і позначаються відповідно $\forall$ і $\exists$.

Правила застосування кванторів

1. Однойменні квантори можна переставляти місцями

$\forall x\ \forall y\ P(x, y) = \forall y\ \forall x\ P(x, y)$;

$\exists x\ \exists y\ P(x, y) = \exists y\ \exists x\ P(x, y)$

2. Різнойменні квантори в загальному випадку переставляти не можна

$\forall x\ \exists y\ P(x, y) \neq \exists y\ \forall x\ P(x, y)$

3. Між кванторами спільності "∀" і існування "∃" мають місце співвідношення, що узагальнюють закони де-Моргана.

$\neg(\forall x\ P(x)) = \exists x \neg(P(x))$; $\neg(\exists x\ P(x)) = \forall x\ \neg(P(x))$

«Завжди істинні» висловлювання (логічні закони) – висловлювання, істинність яких детермінується їх формою і не залежить від логічного значення простих висловлювань, їх складових.

«Завжди хибні» висловлювання (логічні суперечності) – висловлювання, хибність яких детермінується їх формою і не залежить від логічного значення їх складових.

Виконувані або невизначені висловлювання – висловлювання, логічне значення яких не можна визначити без врахування логічного значення (істинності чи хибності) простих висловлювань, які входять до їх складу. Таблиця істинності логічних зв'язок подано в табл. 6.1.

Таблиця 6.1 – Таблиця істинності логічних зв'язок

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$	$\overline{A}$
істинне	істинне	істинне	істинне	істинне	істинне	хибне
істинне	хибне	хибне	істинне	хибне	хибне	хибне
хибне	істинне	хибне	істинне	істинне	хибне	істинне
хибне	хибне	хибне	хибне	істинне	істинне	істинне

6.2 Логічні закони

Закон тотожності: $A \to A$ («Якщо *A*, то *A*») або $A \leftrightarrow A$ («*A* тоді і тільки тоді, коли *A*»).

Закон комутативності для кон'юнкції:

$(A \wedge B) \leftrightarrow (B \wedge A)$ («*A* і *B* тоді і тільки тоді, коли *B* і *A*).

Закон комутативності для диз'юнкції:

$$(A \vee B) \leftrightarrow (B \vee A)$$

(«*A* або *B* тоді і тільки тоді, коли *B* або *A*»).

Закон подвійного заперечення: $\overline{\overline{A}} \to A$

(«Коли хибно, що хибно А, то А»)

Закон ідемпотентності для кон'юнкції: $(A \wedge A) \leftrightarrow A$

(«*A* і *A* тоді і тільки тоді, коли *A*»).

Закон асоціативності для кон'юнкції: $((A \wedge B) \wedge C) \leftrightarrow (A \wedge (B \wedge C))$

«(*A* і *B*) *і C* тоді і тільки тоді, коли *A* і (*B* і *C*)»).

Закон асоціативності для диз'юнкції: $((A \vee B) \vee C) \leftrightarrow (A \vee (B \vee C))$

(«(*A* або *B*) або С тоді і тільки годі, коли *A* або (В або С)»).

Закон дистрибутивності кон'юнкції щодо диз'юнкції: $(A \wedge (B \vee C)) \leftrightarrow ((A \wedge B) \vee (A \wedge C))$ («*A* і (*B* або С), тоді і тільки тоді коли

(*A* і *B*) або (*A і C*)»).

Закон дистрибутивності диз'юнкції щодо кон'юнкції: $(A \vee (B \wedge C)) \leftrightarrow ((A \vee B) \wedge (A \vee C))$

(« А або (В і С), якщо і тільки якщо (А або В) і (А або С)).

Перший закон де Моргана: $(\overline{A \wedge B}) \leftrightarrow (\overline{A} \vee \overline{B})$ («Хибно, що А і В тоді і тільки тоді, коли хибно, що А, або хибно, що В »).

Другий закон де Моргана: $(\overline{A \vee B}) \leftrightarrow (\overline{A} \wedge \overline{B})$

(«Хибно, що A або B тоді і тільки тоді, коли хибно, що A і хибно, що B»).

Основні ЛЗЗ формули алгебри предикатів

1	$\forall xP(x) \to P(y)$	Правило універсальної конкретизазії
2.	$P(a) \to \exists xP(x)$	Правило екзистенціального узагальнення
3.	а) $\neg\forall \times P(x) \leftrightarrow \exists x\neg P(x)$	Закон де Моргана
	б) $\neg\exists xP(x) \leftrightarrow \forall x\neg P(x)$	
4.	а) $\forall x(P(x) \land Q(x)) \leftrightarrow (\forall xP(x) \land \forall xQ(x))$	Закон перенесення $\forall$ через $\land$
	б) $\exists x(P(x) \lor Q(x)) \leftrightarrow (\exists xP(x) \lor \exists xQ(x))$	Закон перенесення $\exists$ через $\lor$
5.	а) $(\forall xP(x) \lor \forall xQ(x)) \to \forall x(P(x) \lor Q(x))$	Закон перенесення $\forall$ через $\lor$
	б) $\exists x(P(x) \land Q(x)) \to (\exists xP(x) \land \exists xQ(x))$	Закон перенесення $\exists$ через $\land$
6.	а) $\forall x\big(P(x) \to Q(x)\big) \to \big(\forall xP(x) \to \forall xQ(x$	Закон перенесення $\forall$ через $\to$
	б) $\big(\exists xP(x) \to \exists xQ(x)\big) \to \exists x\big(P(x) \to Q(x)\big)$	Закон перенесення $\exists$ через $\to$
7.	а) $\forall x\big(P(x) \leftrightarrow Q(x)\big) \to \big(\forall xP(x) \leftrightarrow \forall xQ(x)\big)$	Закон перенесення $\forall$ через $\leftrightarrow$
	б) $\big(\exists xP(x) \leftrightarrow \exists xQ(x)\big) \to \exists x\big(P(x) \leftrightarrow Q(x)\big)$	Закон перенесення $\exists$ через $\leftrightarrow$
8.	а) $\forall x(P(x) \land B) \leftrightarrow (\forall xP(x) \land B)$	Формула B не містить входжень x
	б) $\forall x(P(x) \lor B) \leftrightarrow (\forall xP(x) \lor B)$	
	в) $\exists x(P(x) \land B) \leftrightarrow (\exists xP(x) \land B)$	Закон перестановки кванторів
	г) $\exists x(P(x) \lor B) \leftrightarrow (\exists xP(x) \lor B)$	
	д) $\forall x(P(x) \to B) \leftrightarrow (\exists xP(x) \to B)$	
	е) $\exists x(P(x) \to B) \leftrightarrow (\forall xP(x) \to B)$	
9.	а) $\forall x\forall yP(x,y) \leftrightarrow \forall y\forall xP(x,y)$	Жодне вільне входження y не стане зв'язаним в результаті заміни x на y
	б) $\exists x\exists yP(x,y) \leftrightarrow \exists y\exists xP(x,y)$	
	в) $\exists y\forall xP(x,y) \to \forall x\exists yP(x,y)$	
10.	а) $\forall xP(x) \leftrightarrow \forall yP(y)$	
	б) $\exists xP(x) \leftrightarrow \exists yP(y)$	

Нище наведено інтерпретації деяких логічних загально значимих формул.

$\forall xP(x) \to P(y)$ **Якщо всі люди смертні, то смертною є будь-яка людина.**

$P(a) \to \exists xP(x)$ **Якщо кішка а – сіра, то і існують сірі кішки.**

$\neg\forall \times P(x) \leftrightarrow \exists x\neg P(x)$ **Не всі кішки сірі** $\leftrightarrow$ **Існують не сірі кішки.**

$\neg\exists xP(x) \leftrightarrow \forall x\neg P(x)$ **Не існує сірих кішок** $\leftrightarrow$ **Всі кішки не сірі.**

$\forall x(P(x) \land Q(x)) \leftrightarrow (\forall xP(x) \land \forall xQ(x))$ **Всі кішки з вусами та хвостами** $\leftrightarrow$ **Кожна кішка має вуса і кожна кішка має хвіст.**

$(\exists x P(x) \leftrightarrow \exists x Q(x)) \to \exists x(P(x) \leftrightarrow Q(x))$ Деякі кішки білі або сірі ↔ Існує хоча б одна біла кішка або існує хоча б одна чорна кішка.

$\forall x(P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x))$ Якщо всі сторожові собаки злі, то якщо всі собаки – сторожові, то всі вони злі. Зворотнє не завжди правильне.

$(\exists x P(x) \to \exists x Q(x)) \to \exists x(P(x) \to Q(x))$ Якщо із того, що існують собаки, випливає, що існують гавкаючі істоти, то існують такі собаки, які гавкають. Зворотнє не завжди правильне.

Формула алгебри предикатів називається логічно загальнозначущою (ЛЗЗ), якщо вона правдива на всіх інтерпретаціях та при всіх наборах значень своїх вільних змінних.

Формула алгебри предикатів називається суперечливою, якщо вона хибна на всіх інтерпретаціях та при всіх наборах значень своїх вільних змінних.

Формула алгебри предикатів виконується, якщо вона правдива хоча б на одній інтерпретації та хоча б при одному наборі значень своїх вільних змінних.

Дві формули F1 та F2 називаються еквівалентними (логічно еквівалентними), якщо вони одночасно правдиві або хибні на кожній інтерпретації та при кожному наборі своїх вільних змінних (позначається F1 = F2, або F1 $\Leftrightarrow$ F2).

Зрозуміло, що F – ЛЗЗ формула тоді і тільки тоді, коли ¬ F суперечлива; F виконується тоді і тільки тоді, коли F не є суперечливою; F1 = F2 тоді і тільки тоді, коли F1 ↔ F2 – ЛЗЗ формула.

Література: [3, 7, 8, 10, 11, 14].

ТЕМА 7. БУЛЕВІ ФУНКЦІЇ. АЛГЕБРИ

7.1 Алгебраїчні операції та їх властивості

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

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.

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 (/).

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 використовується 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.

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 $\oplus$.

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 А з одиницею e і елементи $x, y \in A$ задовольняють рівності

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

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

Нехай n – довільне натуральне число. Додавання за модулем п 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 $a+b=k*n+c,\ 0<c<n;\ a,\ b,\ k \in$ Z+

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

Алгебраїчною структурою (короткою – структурою) називається множина разом із заданими операціями, визначеними і замкненими операціями 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, $9 \in Z_7$. Але (Z_7, $\oplus_7$) є алгебраїчною структурою, оскільки область значень операції $\oplus_7$ лежить у Z_7.

Структура А'-(А', $\oplus$') єпідструктурою алгебраїчної структури8 – (А, $\oplus$), якщо:

1. А' $\subset$ А.
2. $\oplus$' і $\oplus$ операції одного порядку і звуження операції

0 на підмножині А' співпадає з операцією $\oplus$' (наприклад, для бінарних операцій $a \oplus b = a \oplus' b$ для всіх $a, b \in A'$).

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

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

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

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

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 $x \in M$.

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 $\otimes$:

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

Твердження 1. Нехай – 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 є 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 $x' = x' \otimes e = x' \otimes (x \otimes x'') = (x' \otimes л) \otimes x'' = e \otimes x'' - x''$.

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

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

1. $\oplus$ асоціативна: $x \otimes (y \otimes z) = (x \otimes y) \otimes z$ для всіх $x, y, z \in R$.
2. $\oplus$ комутативна: $X \otimes y = y \oplus x$ 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 де, y, $z \in$ R.

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.

7.2 Алгебри булевих функцій

Розглянемо дві алгебри. Алгебру $\{P_2; \neg; \wedge; \vee\}$ з операціями заперечення, кон'юнкції та диз'юнкції називають алгеброю Буля, а алгубру $\{P_2; \vee; \oplus\}$ з операціями кон'юнкції та додавання за mod2 – алгеброю Жегалкіна.

Закони алгебри Буля

– закони асоціативності $(xy)z = x(yz) = xyz$,

$(x \vee y) \vee z = x \vee (y \vee z) = x \vee y \vee z$;

– закони комутативності $xy = yx$, $x \vee y = y \vee x$:

– дистрибутивний закон для кон'юнуції щодо дизюнкції

$(x \vee y)z = xz \vee yz$;

– дистрибутивний закон для диз'юнкції щодо кон'юнкції

$xy \vee z = (x \vee z)(y \vee z)$;

– закон подвійного заперечення $\bar{\bar{x}} = x$;

– закони де Моргана $\overline{xy} = \bar{x} \vee \bar{y}$, $\overline{x \vee y} = \bar{x}\bar{y}$;

– закони ідемпотентності $xx = x$, $x \vee x = x$;

– закони поглинання $x \vee xy = x$, $x(x \vee y) = x$;

– співвідношення для констант $\bar{1} = 0$, $\bar{0} = 1$, $1x = x$, $0x = 0$, $1 \vee x = 1$, $0 \vee x = x$;

– закон виключеного третього $x_1 \vee \bar{x} = 1$;

– закон суперечності $x\bar{x} = 0$.

Елементарні булеві функції наведено в табл. 7.1.

Таблиця 7.1 – Елементарні булеві функції

№ п/п	Позначення функції	Назва функції	a			
			0	0	1	1
			b			
			0	1	0	1
1.	$f_1 = a \wedge b$	Кон'юнкція	0	0	0	1
2.	$f_2 = a \vee b$	Диз'юнкція	0	1	1	1
3.	$f_3 = a \rightarrow b$	Імплікація	1	1	0	1
4.	$f_4 = a \leftarrow b$	Обернена імплікація	1	0	1	1
5.	$f_5 = a \sim b$	Рівносильність	1	0	0	1
6.	$f_6 = a \oplus b$	Нерівносильність (сума за модулем 2)	0	1	1	0
7.	$f_7 = a \vert b$	Функція Шеффера (інверсія кон'юнкції)	1	1	1	0
8.	$f_8 = a \downarrow b$	Функція стрілка Пірса-Вебба (інверсія диз'юнкції)	1	0	0	0
9.	$f_9 = a \overline{\rightarrow} b$	Інверсія імплікації	0	0	1	0
10.	$f_{10} = a \overline{\leftarrow} b$	Інверсія, обернена імплікації	0	1	0	0
11.	$f_{11} = a$	Повторення a (змінна a)	0	0	1	1
12.	$f_{12} = \overline{a}$	Інверсія a (функція НЕ a)	1	1	0	0
13.	$f_{13} = b$	Повторення b (змінна b)	0	1	0	1
14.	$f_{14} = \overline{b}$	Інверсія b (функція НЕ b)	1	0	1	0
15.	$f_{15} = 1$	Одинична функція (константа 1)	1	1	1	1
16.	$f_{16} = 0$	Нульова функція (константа 0)	0	0	0	0

Закони алгебри Жегалкіна:

– закони асоціативності $(xy)z = x(yz) = xyz$,

$(x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus y \oplus z$;

– закони комутативності $xy = yx$, $x \oplus y = y \oplus x$;

– дистрибутивний закон для кон'юнкції щодо додавання за mod2 $x(y \oplus z) = xy \oplus xz$;

– співвідношення для констант $1x = x$, $0x = 0$, $x \oplus 0 = x$;

– закон ідемпотентності для кон'юнкції $xx = x$;

– закон зведення подібних чденів у разі додавання за mod2 $x \oplus x = 0$.

Наведемо еквівалентності, які дають змогу перетворити будь-яку формулу булевої алгебри в рівносильну до неї формулу Жегалкіна й навпаки:

– $\overline{x} = 1 \oplus x$;

– $x \vee y = x \oplus y \oplus xy$;

– $x \oplus y = \overline{x}y \vee x\overline{y}$.

За допомогою законів алгебри Буля та Жигалкіна можна спрощувати різні формули в цих алгебрах.

<u>Диз'юнктивність нормальні формули</u>

Уведемо позначення $x^{\sigma} = x\sigma \vee \overline{x}\overline{\sigma}$, де σ – параметр, який дорівнює 0 чи 1. Очевидно, що

$$x^{\sigma} = \begin{cases} \overline{x}, \text{ якщо } \sigma = 0, \\ x, \text{ якщо } \sigma = 1. \end{cases}$$

Зауважимо, що $\sigma^{\sigma} = 1$.

Зафіксуємо множину змінних $X = \{x_1, x_2, \ldots, x_n\}$.

Елементарною кон'юнкцією називають вираз $k = x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} \ldots x_{i_r}^{\sigma_r}$, де x_{i_j} – змінні з множини X, причому всі x_{i_j} різна. Число r називають кон'юнкції. У разі $r = 0$ кон'юнкцію називають порожньою та вважають, що дорівнює 1.

<u>Приклад.</u> Елементарними кон'юнкції – це, наприклад $1, \overline{x_1}, x_1, x_2, \overline{x_1}\,\overline{x_2}x_3$, а вирази $0, x_1x_2x_1, \overline{x_1}x_1, \overline{x_1x_2}$ – не елементарні кон'юнкції.

Елементарну кон'юнкцію, яка містить усі змінні з множини X, називають конституентою одиниці. Інакше кажучи, конституента одиниці – це елементарна кон'юнкція з рангом n. Очевидно, що всі різних конституент одиниці для фіксованої множини n змінних $x_1, x_2, \ldots, x_n$ стільки, скільки двійкових наборів з n компонентами, тобто 2^n.

Диз'юнктивною нормальною формою (ДНФ) називають диз'юнкцію $D = k_1 \vee k_2 \vee \ldots \vee k_s$ елементарних кон'юнкцій k_j, у якій k_j попарно різні.

Є алгоритм, який дає змогу для будь-якої формули булевої алгебри на основі тотожних перетворень знайти рівносильну, побудовану зі змінних і їх заперечень за допомогою самих лише кон'юнкцій і диз'юнкцій (тобто заперечення можуть стояти лише над змінними). Для використовують закони де Моргана та закон подвійного заперечення.

На другому етапі домагаються, щоб усі кон'юкції виконувалися раніше, ніж диз'юнкції, для чого розкривають дужки на підставі дистрибутивного закону для кон'юнкції $(x \vee y)z = xz \vee yz$ або рівносильності $(x \vee y)(z \vee u) = xz \vee yz \vee yu$ (див приклад). Далі х використанням співвідношень для констант і закону суперечності вилучають нулі та, виходячи із законів ідемпотентності, об'днують рівні члени. На цьому процес отримання ДНФ закінчують.

Теорема. Будь-яку булеву функцію $f(x_1, \ldots, x_n) \neq 0$ можна єдиним способом подати в ДДНФ.

Для функції, заданої таблицею, ДДНФ будують так: для кожного набору, на якому функція набуває значення 1, будують віддповідно йому конституентну одиниці; диз'юнкція всіх цих конституент − це ДДНФ даної функції.

Приклад. Побудуємо ДДНФ для функції, заданої табл. 7.2

Таблиця 7.2 Значення функції

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

Функція набуває значення 1 на наборах (00) і (11), отже $f(x_1, x_2) = \overline{x_1}\,\overline{x_2} \vee x_1 x_2$.

Будь-яку ДНФ можна завести до ДДНФ розщепленням кон'юнкцій, які містять не всі змінні: якщо кон'юнкція k не містить змінної x, то

$$k = k(x \vee \overline{x}) = kx \vee k\overline{x}.$$

Приклад. Перетворимо ДНФ $\bar{x}\bar{z} \vee \bar{x}yz$ на досконалу. Застосувавши розщеплення для кон'юнкції $\bar{x}, \bar{z}$, одержемо

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

Кон'юнктивні нормальні форми

Двоїстим способом, замінюючи в означеннях нулі одиницями й навпаки, диз'юнкції кон'юнкціями й навпаки, означають поняття елементарної диз'юнкції, конституенти нуля, кон'юнктивної нормальної форми, досконалої кон'юнктивної нормальної форми.

Елементарною диз'юнкцією називають вираз $d = x_{i_1}^{\sigma_1} \vee x_{i_2}^{\sigma_1} \vee \ldots \vee x_{i_r}^{\sigma_r}$, у якому всі x_{i_j} різні, $x_{i_j} \in X$. Число r називають рангом диз'юнкції.

Кон'юнктивною нормальною формою (КНФ) називають кон'юнкцію $d_1 \wedge d_2 \wedge \ldots \wedge d_s$ елементарнихдиз'юнкцій d_j, у якій усі d_j різні.

Елементарну диз'юнкцію, яка містить усі змінні з множини X, називають ституентою нуля.

Досконалою кон'юнктивною нормальною формою (ДКНФ) називають КНФ, у якій кожна елементарна диз'юнкція $d_j\ (j = 1, \ldots, s)$ − конституента нуля.

Елементарну кон'юнкцію називають монотонною, якщо вона не містить заперечень змінних. Наприклад, $x_1x_2x_3, x_1, 1$ − монотонні кон'юнкції.

Формулу $P(\tilde{x}^n) = k_1 \oplus k_2 \oplus \ldots \oplus k_s$

де $k_1, k_2, \ldots, k_s$ − попарно різні монотонні кон'юнкції змінних із множини $X = \{x_1, x_2, \ldots, x_n\}$, називають поліномом Жегалкіном. Найбільший із рангів елементарнихкон'юнкцій, що входять у поліном, називабть степенем полінома. За окремим означенням 0 також уватимемо поліномом Жегалкіна.

Приклад. Формули $x, 1, xyz \oplus xy \oplus z \oplus 1$ − поліномом Жегалкіна, а $xx, xy \oplus yx \oplus x \oplus 1$ − ні.

Теорема. Будь-яку булеву функцію можна єдиним способом подати у вигляді полінома Жегалкіна.

Метод невизначених коефіцієнтів. Для функції $f(x_1,\ldots x_n)$ записують найзагальніший вигляд полінома Жегалкіна $P(x_1,\ldots,x_n)$ із невизначеними коефіцієнтами (їх 2^n). Зокрема, поліном від двох змінних має загальний вигляд

$$P(x,y)=c_0\oplus c_1x\oplus c_2y\oplus c_3xy,$$

а від трьох змінних –

$$P(x,y,z)=c_0\oplus c_1x\oplus c_2y\oplus c_3z\oplus c_4xy\oplus c_5xz\oplus c_6yz\oplus c_7xyz$$

.

Для кодного двійкового набору $(a_1,\ldots,a_n)$ значень змінних записують 2^n рівнянь $f(a_1,\ldots,a_n)=P(a_1,\ldots,a_n)$. Розв'язавши їх, отримують коефіцієнти полінома $P(x_1,\ldots,x_n)$.

Приклад. Побудуємо поліном Жигалкіна для функції $f(x,y)=x\sim y$. Прирівняємо значення функції та полінома на всіх чотирьох наборах значень змінних і одердимо систему рівнянь відносно неозначених коефіцієнтів:

$$f(0,0)=1=c_0,$$
$$f(0,1)=0=c_0\oplus c_2,$$
$$f(1,0)=0=c_0\oplus c_1,$$
$$f(1,1)=1=c_0\oplus c_1\oplus c_2\oplus c_3.$$

Розв'язавши її, визначаємо, що $c_0=1, c_1=1, c_2=1, c_3=0$ й, отже, $x\sim y=1\oplus x\oplus y$.

Побудова полінома Жегалкіна на основі рівносильних перетворень. Спочатку будують рівносильну фрмулу, к якій є лише операції та заперечення, а потім всюди замінюють $\overline{x}$ на $1\oplus x$. Після цього тривіальними перетвореннями отримують поліном Жегалкіна.

Приклад. Побудуємо поліном Жегалкіна для функції $f(x,y)=x\to y$. Використовуючи введені раніше еквівалентності, одержимо

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

Поліном Жегалкін має цікаву властивість, зручну для знаходження істотних змінних функцій.

Теорема. Усі змінні булевої функції,які входить у її поліном Жегалкіна, істотні.

Література: [1, 2, 3, 4-6, 7, 12].

ТЕМА 8. ОСНОВНІ ПОНЯТТЯ НЕОРІЄНТОВАНИХ ТА ОРІЄНТОВАНИХ ГРАФІВ

8.1. Основні означення та властивості

Термін «граф» уперше з'явився в книзі видатного угорського математика Д. Кеніга 1936р., хоча перші задачі теорії графів пов'язані ще з іменем Л. Ейлера (XVIII ст.).

Простим графом називають пару $G=(V,E)$, де V – непорожня скінченна множина елементів, називаних вершинами, E – множина невпорядкованих пар різних елементів з V. Елементи множини E (невпорядковані пари різних вершин) називають ребрами.

На рисунку 8.1 зображено простий граф G з множиною вершин $V=\{v_1,v_2,v_3,v_4\}$ і множиною ребер $E=\{\{v_1,v_2\},\{v_2,v_3\},\{v_3,v_4\}\}$.

Говорять, що ребро $\{u,v\}$ з'єднує вершини u та v. Оскільки E – множина, то в простому графі пару вершин може з'єднувати не більше ніж одне ребро.

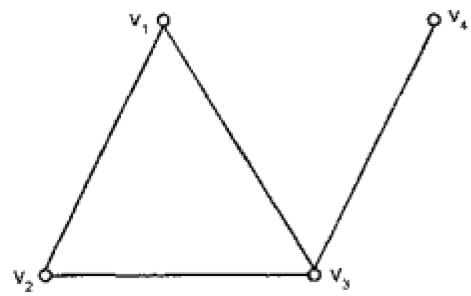


Рисунок 8.1 – Простий граф

Іноді розглядають графи, у яких дві вершини можуть бути з'єднані більше ніж одним ребром. Так виникає поняття мультиграфа. Мультиграфом називають пару (V,E) де V – скінченна непорожня множина вершин, E – сім'я невпорядкованих пар різних елементів з множини V. Тут застосовано термін «сім'я» замість «множина», бо елементив E (ребра) можуть повторюватись. Ребра, що з'єднують одну й ту саму пару вершин, називають кратними (або паралельними) ребрами.

Подальше узагальнення полягає в тому, що окрім кратних ребер розглядають ще й петлі, тобто ребра, які з'єднують вершину саму із собою. Псевдографом називають пару (V,E) де V – скінченна непорожня множина вершин, а E – сім'я невпорядкованих пар не обов'язково різних вершин.

Приклад. На рисунку 8.2 зображено (а) – мультиграф і (б) – псевдограф.

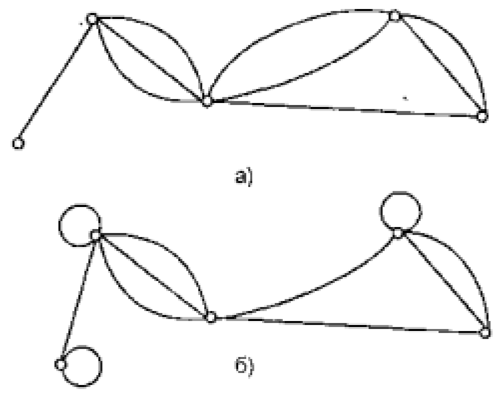


Рисунок 8.2 – а) мультиграф, б) – псевдограф

Розглянуті три типи графів називають неорієнтованими. Псевдограф – це найзагальніший тип неорієнтованого графа, бо він може містити петлі й кратні ребра. Мультиграф – це неорієнтований граф, який може містити кратні ребра, але не може містити петель. Нарешті, простий граф – це неорієнтований граф без кратних ребер і без петель.

Розглядають також орієнтовані графи. Орієнтованим графом називають пару (V,E) де V – скінченна непорожня множина вершин, а E – множина впорядкованих пар елементів множини V. Елементи множини E в орієнтованому графі називають дугами (або орієнтованими ребрами). Дугу (v,v) називають петлею.

Приклад 8.1. На рис. 8.3 зображено орієнтований граф із множиною вершин $V=\{v_1, v_2, v_3, v_4\}$ і множиною дуг

$E=\{\{v_2,v_1\},\{v_2,v_3\},\{v_3,v_2\},\{v_3,v_4\},\{v_4,v_3\},\{v_4,v_5\},\{v_5,v_5\}\}$.

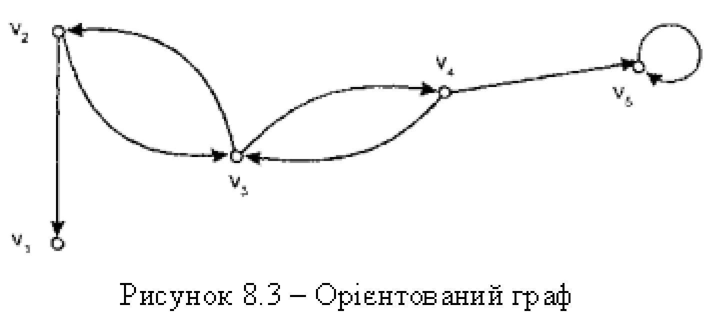


Рисунок 8.3 – Орієнтований граф

Зазначимо, що дуга – це впорядкована пара вершин (записують у круглих дужках), тому в графі на рис. 8.3 дуги (v_2, v_3) та (v_3, v_2) – різні. На рисунках дуги позначають стрілками.

Орієнтованим мультиграфом називають пару (V, E) де V – скінченна не порожня множина вершин, а E – сім'я впорядкованих пар елементів з V.

Отже, елементи (дуги) в E в разі орієнтованого мультиграфа можуть повторюватись, такі дуги називають кратними. Підкреслимо, що кратні дуги з'єднують одну пару вершин і однаково напрямлені.

Приклад. На рисунку 8.4 наведено приклад орієнтованого мультиграфа. Дуги e_2, та e_3 – кратні, а дуги e_5, e_6 – ні.

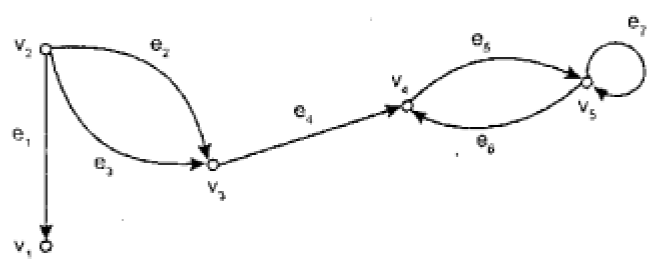


Рисунок 8.4 – Орієнтований мультиграф

Надалі ми будемо використовувати термін «граф» для опису довільних графів – орієнтованих і неорієнтованих, із петлями та кратними ребрами чи без них. Означення різних типів графів зведено в таблиці 8.1.

Таблиця 8.1 – Типи графів

Тип графа	Ребра	Кратні ребра дозволені?	Петлі дозволені?
Простий граф	Неорієнтовані	Ні	Ні
Мультиграф	Неорієнтовані	Так	Ні
Псевдограф	Неорієнтовані	Так	Так
Орієнтований граф	Орієнтовані (дуги)	Ні	Так
Орієнтований мультиграф	Орієнтовані (дуги)	Гак	Так

Дві вершини u та v в неорієнтованому графі G називають суміжними, якщо існує ребро $\{u, v\}$, тобто $\{u, v\} \in E$. Якщо $e = \{u, v\}$ – ребро, то вершини u та v називають його кінцями. Два ребра називають суміжними, якщо вони мають спільний кінець. Вершину v та ребро e називають інцидентними, якщо вершина v – кінець ребра e . Зазначимо, що суміжність – це зв'язок між однорідними елементами графа, а інцидентність – зв'язок між його різнорідними елементами.

Степінь вершини в неорієнтованому графі – це кількість ребер, інцидентних цій вершині, причому петлю враховують двічі. Степінь вершини V позначають $\deg(v)$. Якщо $\deg(v)$=0. то вершину v називають ізольованою: якщо $\deg(v)$=1 - висячою. або кінцевою.

Приклад. У неорієнтованому графі на рис. 8.5 степені вершин такі: $\deg(v_1)$=4, $\deg(v_2)$=4, $\deg(v_3)$=6, $\deg(v_4)$=1, $\deg(v_5)$=3, $\deg(v_6)$=0. Отже, вершина v_6 – ізольована, а v_4 – висяча.

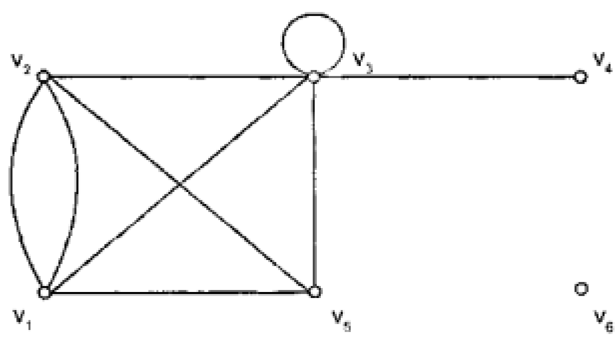


Рисунок 8.5 – Неорієнтований граф

Зв'язок між степенями вершин неорієнтованого графа та кількістю його ребер дає така теорема.

Теорема 1. Нехай $G=(V,E)$ неорієнтований граф з m ребрами. Тоді

$$\sum_{v\in V}\deg(v)=2m$$

Зазначимо, що це твердження стосується будь-якого неорієнтованого графа, зокрема, з петлями й кратними ребрами.

Доведення. Кожне ребро додає по одиниці до степенів двох вершин, або двійку до степеня однієї вершини у випадку петлі. З цього випливає, що сума степенів вершин удвічі більша від кількості ребер. Теорему доведено.

Теорема 2. Неорієнтований граф має парну кількість вершин непарного степеня.

Доведення. Позначимо як V_1 множину вершин непарного степеня, як V_2 – парного, а як m - кількість ребер графа. Тоді

$$2m=\sum_{v\in V}\deg(v)=\sum_{v\in V_1}\deg(v)+\sum_{v\in V_2}\deg(v).$$

Оскільки $\deg(v)$ – парне для $v \in V_2$, то другий доданок у правій частині рівності парний. Більше того, сума двох доданків справа – парна, бо дорівнює $2m$. І тому перший доданок $\sum_{v \in V_1} \deg(v)$ – також парний. Він являє собою суму, у якій всі доданки – непарні числа.

Отже, кількість цих доданків має бути парною. Звідси випливає, що кількість вершин з непарним степенем – парна. Теорему доведено.

Тепер розглянемо орієнтований мультиграф $G = (V, E)$. Якщо $(u, v) \in E$. то вершину u називають початковою (ініціальною). а вершину v – кінцевою {термінальною) вершиною дуги $e = (u, v)$. Петля має початок і кінець в одній і тій самій вершині. Вершини орієнтованого графа називають суміжними, якщо одна з них – початкова, а інша – кінцева для якоїсь дуги. Дуги називають суміжними, якщо вони мають спільну вершину. Вершину u називають інцидентною дузі e, якщо u – початкова чи кінцева вершина цієї дуги.

Дія орієнтованого графа означення степеня вершини інше. В орієнтованому мультиграфі напівстепенем входу вершини V називають кількість дуг, для яких вершина v кінцева: позначають $\deg^-(v)$. Напівстепенем виходу вершини v називають кількість дуг. Для яких вершина V початкова; позначають $\deg^+(v)$.

Приклад. Для графа, зображеного на рис. 8.6 напівстепені вершин такі:

$\deg(v_1) = 0, \deg^+(v_1) = 1, \deg^-(v_2) = 2, \deg^-(v_2) = 0, \deg^-(v_3) = 1, \deg^+(v_3) = 2$

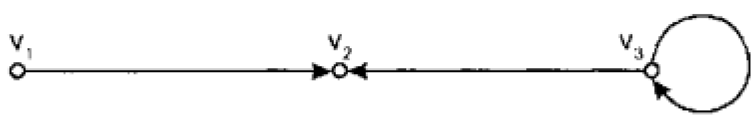


Рисунок 8.6 – Орієнтований граф

Теорема 3. Нехай $G = (V, E)$ – орієнтований мультиграф, який має m дуг. Тоді

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = m$$

Доведення. Оскільки кожна дуга має початкову й кінцеву вершини, то суми напівстепенів входу й виходу однакові. Кожна з цих сум, очевидно, дорівнює кількості дуг. Теорему доведено.

Неорієнтований граф H називають підграфом неорієнтованого графа $G = (V, E)$, якщо всі вершини графа H належать V, а всі його ребра належать E. Якщо графи $H = (W, F)$ і $G = (V, E)$ – прості та H – підграф графа G, то розглядають іще два окремих випадки. Підграф H називають каркасним підграфом (або фактором), якщо $W = V$. Якщо $W \subseteq V$, а F – множина всіх ребер із E, які мають кінці в W, то підграф H називають породженим (індукованим) множиною W і позначають як $G(W)$.

Приклад. На рис. 8.7 зображено граф G та три його підграфа H_1, H_2, H_3 серед яких H_2 – породжений, а H_3 – каркасний.

Об'єднанням двох простих графів $G_1 = (V_1, E_1)$ та $G_2 = (V_2, E_2)$ називають простий граф $G = (V, E)$ такий, що $V = V_1 \cup V_2, E = E_1 \cup E_2$.

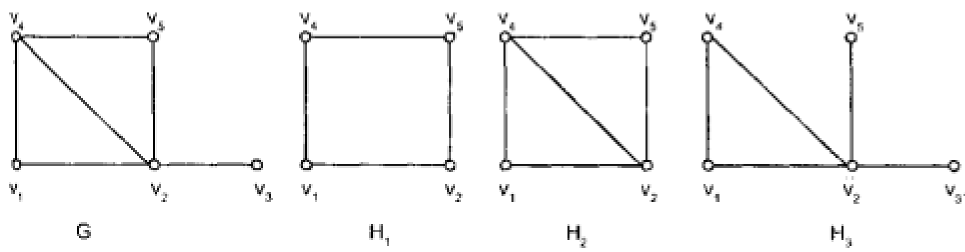


Рисунок 8.7 – Підграфи H_1, H_2, H_3 графа G

Приклад. На рис. 8.8 наведено приклад об'єднання двох простих графів. Знаки «$\cup$» та «=» тут мають символічний зміст.

Для алгоритмів на графах як міру обсягу вхідної інформації природно вибирати n чи так $\max(m, n)$, де n – кількість вершин, а m – кількість ребер графа.

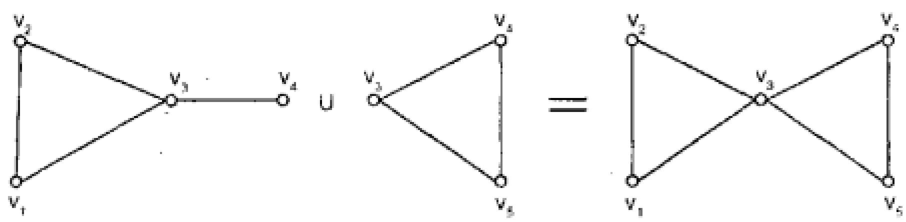


Рисунок 8.8 – Об'єднання двох простих графів

8.2 Деякі спеціальні класи простих графів

Розглянемо деякі спеціальні класи простих графів, які часто використовують як приклади й широко застосовувані.

Повний граф з n вершинами (позначають як K_n) – це граф, у якого будь-яку пару вершин з'єднано точно одним ребром.

Кількість ребер у графі K_n дорівнює $C_m^2 = n(n-1)/2$.

Приклад. На рис. 8.9 зображено графи K_n для $n = 1,2,3,4,5$.

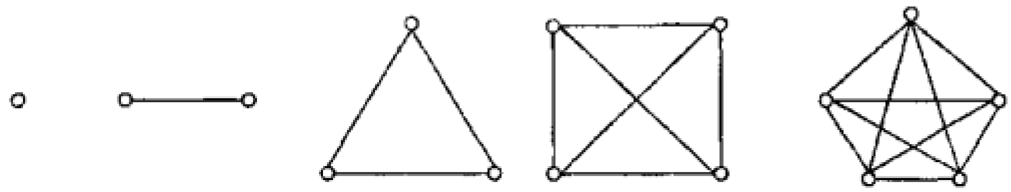


Рисунок 8.9 – Графи K_n для $n = 1,2,3,4,5$

Граф K_5 простий повний граф.

Граф називають порожнім, якщо $E = \varnothing$, тобто такий граф не має ребер. Порожній граф з n вершинами позначають як O_n

Граф $G = (V, E)$ називають дводольним, якщо множину його вершин V можна розбити на дві під множини V_1 і V_2 що не перетинаються так, що кожне ребро з'єднує вершину з V_1 і вершину з V_2 Іноді дводольний граф позначають як $G = (V_1 \cup V_2, E)$. Дводольний граф називають повним дводольним графом, якщо кожну вершину з V_1 з єднано ребром із кожною вершиною з V_2 . Повний дводольний граф позначають як $K_{m,n}$, де $m = |V_1|, n = |V_2|$ Граф $K_{1,n}$ називають зіркою.

Граф $K_{m,n}$ має $n + m$ вершин та $n \cdot m$ ребер.

Приклад. На рис. 8.10 наведено повні дводольні графи $K_{1,5}$, $K_{2,3}$, $K_{3,3}$

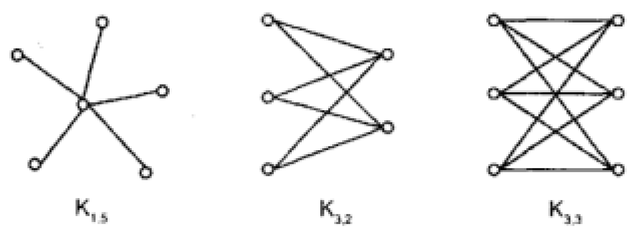


Рисунок 8.10 – Дводольні графи $K_{1,5}$, $K_{2,3}$, $K_{3,3}$

Циклом C_n, $n \geq 3$, називають граф із множиною вершин $V = \{v_1, v_1, ... v_n\}$ і множиною ребер $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$.

Приклад. Па рис. 8.11 зображено цикли C_3, C_4, C_5 і C_6.

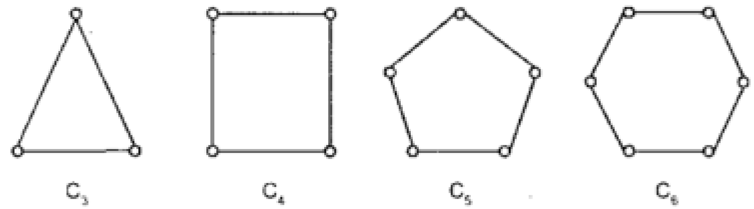


Рисунок 8.11 – Цикли C_3, C_4, C_5, C_6

Колесом W_n називають граф. який одержують із циклу C_n додаванням іще однієї вершини, яку з'єднують з усіма n вершинами в C_n новими ребрами.

Приклад. На рис. 8.12 зображено колеса W_3, W_4, W_5, W_6.

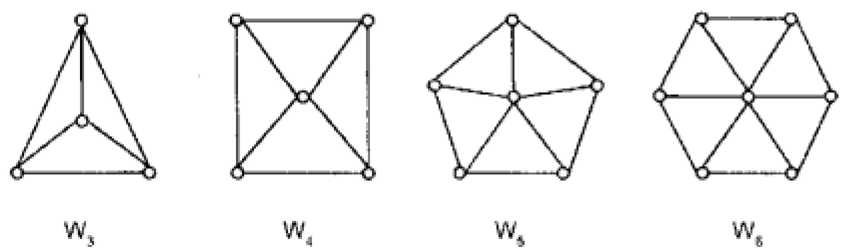


Рисунок 8.12 – Колеса W_3, W_4, W_5, W_6

ТЕМА 9. СПОСОБИ ПОДАННЯ ГРАФІВ

Найзрозуміліший і корисний для людини спосіб подання (зображення) графів – це рисунок на площині у вигляді точок і ліній, які з'єднують ці точки. Проте цей спосіб подання абсолютно непридатний, якщо потрібно розв'язувати на комп'ютері задачі з графами.

Розглянемо декілька інших способів подання графів. Для спрощення розглядатимемо два найбільш важливих типи графів: простий граф (рис. 9.1) і орієнтований граф (рис. 9.2).

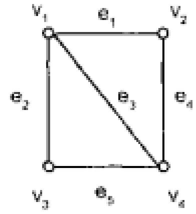


Рисунок 9.1 – G_1

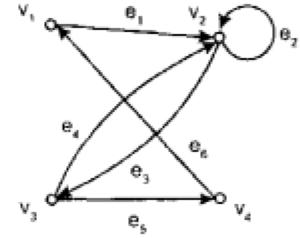


Рисунок 9.2 – G_2

Матрицю, кожний елемент якої дорівнює 0 або 1, називають булевою.

9.1 Матриця інцидентності

Нехай $G=(V,E)$ простий граф із множиною вершин $V=\{v_1,v_1,...v_n\}$ і множиною ребер $E=\{e_1,e_1,...e_n\}$.

Матрицею інцидентності графа G яка відповідає заданій нумерації вершин і ребер, називають булеву $n \times m$ матрицю M з елементами m ($i=1...,n, j=1,..m$), де

$$m_{ij}=\begin{cases}1, \text{якщо вершина } v_1 \text{ та ребро } e_1 \text{ інцидентні,}\\ 0 \text{ у протилежному випадку.}\end{cases}$$

Приклад. Для графа, зображеного на рис. 9.1, матриця інцидентності має вигляд.

$$\begin{array}{c|c} & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Отже, для простого графа в матриці інцидентності в кожному стовпці точно дві одиниці і немає однакових стовпців. Матрицю інцидентності можна використовувати й для подання мультиграфа. Тоді з'являться однакові стовпці (вони відповідають кратним ребрам). Для подання псевдографа петлю e_i у вершині v_i. зображають значенням m_{ij} =2 (у цьому разі матриця інцидентності, очевидно, не булева).

За допомогою матриці інцидентності можна подавати й орієнтовані графи. Для таких графів вона також ие булева. Нехай x_i – орієнтований граф із множиною вершин (x_j) і множиною дуг x_j .

Матрицею інцидентності орієнтованого графа G , яка відповідає заданій нумерації вершин і дуг, називають $n \times m$ матрицю M з елементами m_{ij} $(i=1...,n, j=1,..m)$, де

$$m_{ij}=\begin{cases}1, \text{якщо дуга } e_1 \text{ виходить з вершини } v_1,\\ -1, \text{якщо дуга } e_1 \text{ входить у вершину } v_1,\\ 2, \text{якщо дуга } e_1 - \text{це петля у вершині } v_1,\\ 0 \text{ в інших випадках.}\end{cases}$$

Приклад. Для графа, зображеного на рис. 9.2, матриця інцидентності має вигляд.

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 2 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array}$$

З алгоритмічної точки зору матриця інцидентності не є добрим вибором для комп'ютерних застосувань. По-перше, вона вимагає $n \times m$ комірок пам'яті, більшість із яких зайнята нулями. По-друге, незручний доступ до інформації. Щоб отримати відповідь на елементарні запитання (наприклад, чи існує дуга (v_i, v_j), до яких вершин ведуть дуги з v_i), у найгіршому випадку потрібно перебрати всі стовпці матриці, тобто виконати m кроків.

9.2 Матриця суміжності

Нехай $G = (V, E)$ – простий граф, $|V| = n$. Припустимо, що вершини графа G занумеровані: $v_1, v_1, ... v_n$. Матрицею суміжності графа G (яка відповідає даній нумерації вершин) називають булеву $n \times n$ матрицю A з елементами $a_{ij} \, (i, j = 1, ..., n)$, де

$$a_{ij} = \begin{cases} 1, \text{ якщо } \{v_i, v_j\} \in E \\ 0 \text{ у протилежному випадку} \end{cases}$$

Цілком очевидно, що для неорієнтованого графа $a_{ij} = a_{ji}$ тобто матриця суміжності неорієнтованого графа симетрична. Більше того, позаяк у простому графі немає петель, то для нього в матриці суміжності $a_{ii} = 0 \, (i = 1, ..., n)$.

Матрицю суміжності можна використовувати також для подання псевдографа. Тоді це не булева матриця: елемент a_{ij} дорівнює кількості ребер, що з'єднують v_i та v_j. Петлю у вершині v_i подають значенням діагонального елемента a_{ii} =1.

Приклад. Матриця суміжності для графа, зображеного на рис. 9.1, має вигляд.

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Для подання орієнтованих графів також використовують матрицю суміжності. Це булева $n \times n$ матриця A з елементами $a_{ij}\,(i, j = 1,\ldots,n)$, де

$$a_{ij} = \begin{cases} 1, \text{ якщо } \{v_i, v_j\} \in E \\ 0 \text{ у протилежному випадку} \end{cases}$$

Приклад. Матриця суміжності для графа, зображеного на рис. 9.2, має вигляд

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Зазначимо, що матриця суміжності орієнтованого графа, загалом кажучи, несиметрична.

Матрицю суміжності можна використовувати й для подання орієнтованого мультиграфа. У такому разі це не булева матриця: елемент a_{ij} дорівнює кількості дуг. які мають v_i початковою вершиною, а v_j – кінцевою.

Великою перевагою матриці суміжності як способу подання графа є швидкий доступ до інформації: за один крок можна одержати відповідь на запитання, чи існує ребро (дуга) з v_i у v_j. Недоліком є те, що незалежно від кількості ребер обсяг пам’яті становить n^2 комірок. Як іще один аргумент проти використання матриці суміжності можна зазначити, що деякі алгоритми, які в разі використання матриці суміжності мають оцінку складності O(n^2).

Нарешті, розглянемо ще два способи подання графів у пам’яті комп’ютера. Будь-яку скінченну послідовність елементів довільної природи будемо називати списком, а кількість елементів списку – його довжиною.

9.3 Подання графа списком пар (списком ребер)

Економнішим щодо пам'яті, особливо якщо m (кількість ребер) значно менша, ніж n^2 (n – кількість вершин), є метод подання графа списком пар, які відповідають його ребрам (або дугам). Пара $[u,v]$ відповідає ребру $\{u,v\}$, якщо граф неорієнтований, і дузі (u,v), якщо граф орієнтований.

Для графів, зображених на рис. 9.1 та 9.2, списки пар подані відповідно на рис. 9.3,а та 9.3,б.

v_1	v_2
v_1	v_3
v_1	v_4
v_2	v_4
v_3	v_4

v_1	v_2
v_2	v_2
v_2	v_3
v_3	v_2
v_3	v_4
v_4	v_1

Рисунок 9.3 Списки пар а) – G_1, б) – G_2

Очевидно, що обсяг пам'яті в разі використання цього способу подання дорівнює $2m$ – це найекономніший щодо пам'яті спосіб. Недолік – велика (порядку m) кількість кроків для знаходження множини вершин, до яких ідуть ребра або дуги із заданої вершини. Ситуацію можна значно поліпшити, упорядкувавши множину пар лексикографічно та застосувавши двійковий пошук.

9.4 Подання графа списками суміжності

Орієнтовпний граф G (без кратних дуг. але, можливо, з петлями) можна подати, указавши скінченну непорожню множину вершин V і відповідність Г. котра вказує. Як зв'язані між собою вершини. Відповідність Г - багатозначне відображення множини V у V а граф у такому разі позначають парою $G=(V,\Gamma)$.

Приклад. Для орієнтованого графа, зображеного на рис. 9.1 відповідність Г задано таблиці 9.2.

Цей спосіб подання можна використовувати й для неорієнтовакнх простих графів, якщо кожне ребро умовно замінити двома протилежно напрямленими дугами.

Приклад. Для простого графа, зображеного на рисунку 9.1. відповідність Г задано таблиці 9.2.

Таблиця 9.1

v	$\Gamma(v)$ (термінальні вершини)
v_1	v_2
v_2	v_2, v_3
v_3	v_2, v_4
v_4	v_1

Таблиця 9.2

v	$\Gamma(v)$ (суміжні з v вершин)
v_1	v_2, v_3, v_4
v_2	v_1, v_4
v_3	v_1, v_4
v_4	v_1, v_2, v_3

Якщо відображення Г діє не на одну вершину, а на множину вершин $A=\{x_1, x_2, ..., x_n\}$ то під $\Gamma(A)$ розуміють об'єднання множин

$$\Gamma(A)=\Gamma(x_1)\cup\Gamma(x_2)\cup...\cup\Gamma(x_n)$$

Розглянемо спосіб комп'ютерного подання графа списками суміжності. Для цього використовують масив Adj із $n=|V|$ списків – по одному на кожну вершину. Для кожної вершини $u\in V$ список $Adj\ [u]$ містить у довільному порядку (вказівники на) всі вершини множини Г(u).

Приклад. На рисунку 9.4 за допомогою списків суміжності подано неорієнтований граф з рис. 9.1.

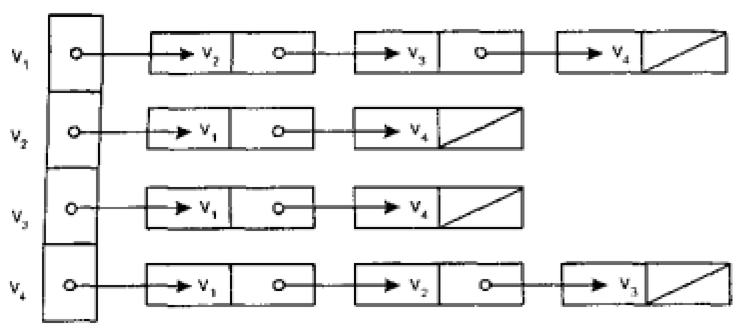


Рисунок 9.4 – Списки суміжності графа G_1

На рис. 9.5 показано аналогічне подання для орієнтованого графа з рис. 9.2.

Для орієнтованого графа сума довжин усіх списків суміжних вершин дорівнює загальній кількості дуг: дузі (u, v)-відповідає елемент v зі списку $Adj\ [u]$.

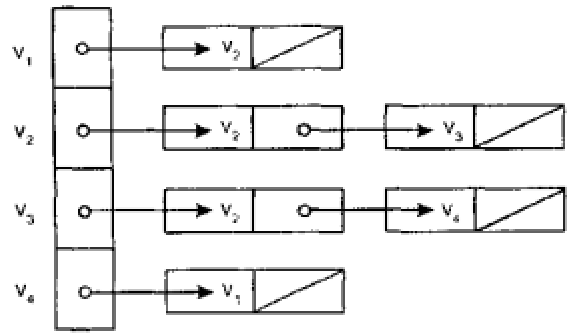


Рисунок 9.5 – Списки суміжності графа G_2

Для неорієнтованого графа ця сума дорівнює подвоєній кількості ребер, бо ребро $\{u, v\}$ породжує елемент у списку суміжних вершин як для вершини u, так і для вершини v.

Література: [4, 6, 7, 10, 11].

ТЕМА 10. ШЛЯХИ ТА ЦИКЛИ. ЗВ'ЯЗНІСТЬ. ІЗОМОРФІЗМ ГРАФІВ

10.1 Шляхи та цикли

Шляхом довжиною r із вершини u в вершину v в неорієнтованому графі називають послідовність ребер $e_1 = \{x_0, x_1\}$, $e_2 = \{x_1, x_2\}, .. e_r = \{x_{r-1}, x_r\}$, де $x_0 = u$, $x_r = v$, r – натуральне число. Отже, шлях довжиною r має r ребер, причому ребро враховують стільки разів, скільки воно входить у шлях. Вершини u та v називають крайніми, а решту вершин шляху – внутрішніми.

Циклом у неорієнтованому графі називають шлях, який з'єднує вершину саму із собою, тобто $u = v$.

У простому графі шлях можна задати послідовністю вершин, через які він проходить:

$$x_0, x_1, x_2, \ldots, x_{r-1}, x_r.$$

Шлях або цикл називають простим, якщо він не містить повторюваних ребер.

Говорять, що шлях із крайніми вершинами u та v з'єднує ці вершини. Шлях, що з'єднує вершини u та v, позначають як $\langle u, v \rangle$ та називають $\langle u, v \rangle$ – шляхом.

Приклад. На рис. 10.1 зображено простий граф. У ньому a, d, c, f, e – простий шлях довжиною 4, оскільки пари {a,d},{d,c},{c,f},{f,e} -ребра. Однак d,e,c,b – не шлях, бо пара {e,c} – не ребро. Зазначимо що b,c,f,e,b – простий цикл довжиною 4, позаяк {b,c},{c,f},{f,e} та {e,b} ребра та цей шлях починається і закінчується в одній і тій самій вершині b.

Шлях a, b, e, d, a, b довжина якого дорівнює 5 (рис. 10.1), не простий, оскільки він двічі проходить через ребро {a,b}.

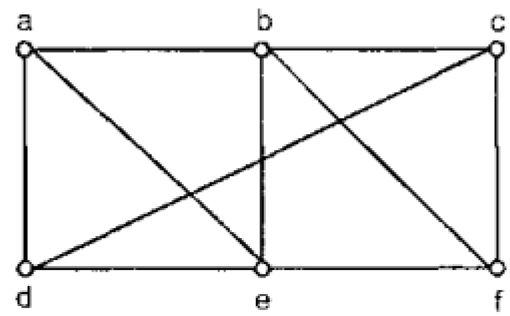


Рисунок 10.1 – Простий граф

Неорієнтований граф називають зв'язним, якщо будь-які дві його вершини з'єднані шляхом. Граф називають незв'язним, якщо він не є зв'язним. Незв'язний граф складається з двох або більше зв'язних підграфів,кожна пара з яких не має спільних вершин. Ці зв'язні підграфи називають компонентами зв'язності або просто компонентами графа.

10.2 Зв'язні та незв'язні графи. Компоненти зв'язності

Приклад. Граф G на рис. 10.2 зв'язний, а граф H – не зв'язний, оскільки не існує шляху (u,v). Граф має дві компоненти.

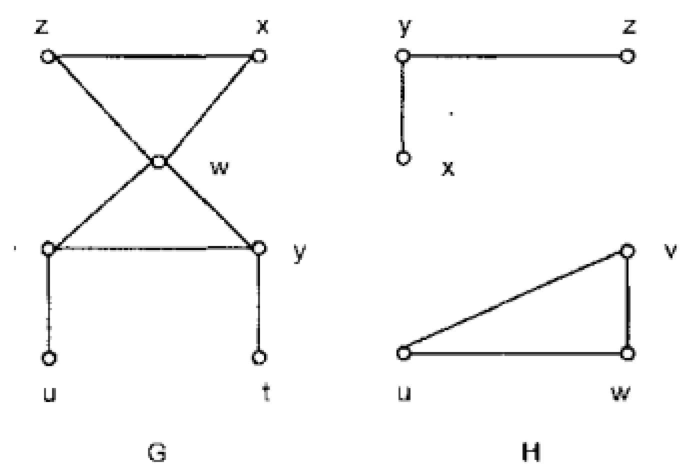


Рисунок 10.2 – Графи G та граф H

Віддалю d(u,v) між вершинами u та v у неорієнтованому графі G довжину найкоротшого (u,v) шляху, а сам цей шлях називають геодизичним.

Теорема 1. Між кожною парою різних вершин зв'язного неорієнтованого графа $-$ зв'язний, то існує принаймі один шлях $G = (V, E)$. Оскільки граф G між u та v.

Для орієнтованого графа вводять поняття орієнтованого шляху (або просто шляху) з вершину v. Це скінченна послідовність дуг $e_1 = (x_0, x_1), e_2 = (x_1, x_2), e_r = (x_{r-1}, x_r)$, де $x_0 = u, x_r = v$. Вершини u та v називають, як і в неорієнтованому графі, крайніми, а решту вершин шляху-внутрішніми. Довжиною шляху називають кількість дуг,з яких він складається. Орієнтованим циклом називають орієнтований шлях, який з'єднує вершину саму із собою, тобто $u = v$. Орієнтований шлях або цикл називають простим, якщо він не містить повторюваних дуг.

Орієнтований граф називають сильно зв'язним, якщо для будь – яких його різних вершин u та v існують орієнтовані шляхи від u до v та від v до u. Отже, для сильної зв'язаності орієнтованого графа має існувати послідовність дуг з урахування орієнтації від будь-якої іншої.

Орієнтований граф може не бути сильно зв'язним, але може бути, так би мовити в одному цілому. У зв'язку із цим дамо таке означення. Орієнтовний граф називають слабко зв'язним, якщо існує шлях між будь-якими двома різними вершинами у відповідному йому неорієнтованому графі.

Приклад. На рис. 10.3 зображено графи G й H. Граф G – сильно зв'язаний. Граф H слабко зв'язаний, він не сильно зв'язаний, бо не існує орієнтованого шляху від w_1 до w_2.

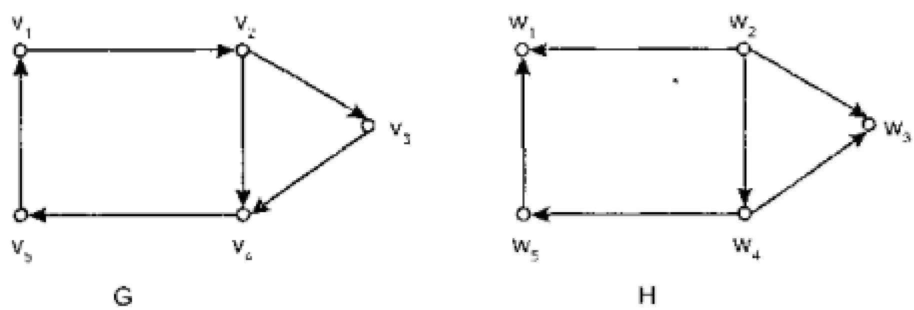


Рисунок 10.3 – Графи G та H

Кількість різних шляхів між двома довільними вершинами графа можна підрахувати за допомогою матриці суміжності.

Теорема 2. Нехай G – граф його матриця суміжності А, яка відповідає, заданій нумерації вершин $v_1, v_2 \dots v_n$. Тоді кількість різних шляхів довжиною r (r – натуральне) з вершини v_i *та* v_j дорівнює сумі елементів матриці A^r.

Характеристики зв'язності простого графа

Розглянемо неорієнтований графи K_n та C_n. Обидва ці графи зв'язні, проте інтуїтивно зрозуміло, що для $n > 3$ граф K_n сильніше зв'язаний, ніж граф C_n.

Розглянемо два поняття, які характеризують міру зв'язності простого графа.

Числом вершинної зв'язності (або просто числом зв'язності) $k(G)$ простого графа G називають найменшу кількість вершин, вилучення яких дає незв'яний або одновершинний граф. Зазначимо, що вершину вилучають разом з інцидентними їй ребрами.

Наприклад. $k(K_1) = 0, k(K_n) = n - 1, k(C_n) = 2.$ Граф G, зображений на рис. 10.4 зв'язний, але його зв'язність можна порушити вилученням вершин u.

Отже, $k(G) = 1$. Якщо ж спробувати порушити зв'язність цього графа вилученням ребер (а не вершин) , то потрібно вилучити не менше ніж три ребра.

Нехай G – простий граф з $n > 1$вершинами. Числом реберної зв'язаності $\lambda(G)$ графа G називають найменшу кількість ребер, вилучення яких дає незв'язний граф. Число реберної зв'язності одновершинного графа вважають таким,що дорівнює 0. Для графа G зображеного на рис. 10.4 $\lambda(G) = 3$

Вершину u простого графа графа G називають точкою з'єднання, якщо граф G в разі її вилучення матиме більш компонент, ніж даний граф G. Зокрема,якщо G – зв'язний граф і u – точка з'єднання, то G без вершини u –незв'язний.

Приклад. Граф H із рис. 10.4 має три точки з'єднання t, s , p та один міст $\{t, s\}$.

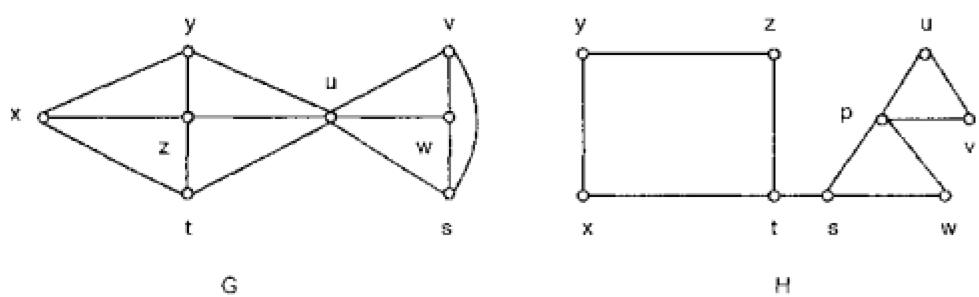


Рисунок 10.4 – Графи G та H

Позначимо як $\delta(G)$ мінімальний степінь вершин графа G. Можна показати, що $k(G) \leq \lambda(G) \leq \delta(G)$.

Простий граф G називають t–зв'язним, якщо $k(G) \geq t$ і реберно $t -$ вязним якщо $\lambda(G) \geq 1$. Отже відмінний від K_1 граф, однозв'язний тоді й лише тоді, коли він зв'язний. Двозв'язні графи – це зв'язні графи без точок з'єднання, відмінні від графа K_1.

Приклад. Граф G зображений на рис. 10.4 однозв'язний і реберно 3- зв'язний.

Вершинами перевищує кількість ребер у графі K_n тобто n(n-1)/2.

Теорема 3. Якщо простий граф G має n вершин і k компонент, то кількість m задовільняє нерівності

$$n-k \le m \le \left(\frac{1}{2}\right)(n-k)(n-k+1)$$

Теорема 4. Критерій дводольності графа. (Кеніг, 1936 р.). Для того, щоб граф С був дводольним, необхідно й достатньо, щоб він не містив простих циклів із непарною довжиною.

10.3 Матриці досяжностей і контрдосяжностей

Матриця досяжностей $R=[r_{ij}]$ визначаються наступним чином:

$$r_{ij} = \begin{cases} 1, \text{ якщо вершина } x_j \text{ досяжна із } x_i, \\ 0 \text{ в противному випадку.} \end{cases}$$

Множини вершин $R=(x_i)$ графа G, досяжних із заданої вершини x_i, складається з таких елементів x_j, для яких (i,j)-й елемент в матриці досяжностей рівний 1. Очевидно що всі діагональні елементи в матриці R рівні 1, оскільки кожна вершина досяжна із себе самої з допомогою шляху 0.

Оскільки $\Gamma(x_i)$ являється множиною таких вершин x_j, які досяжні із x_i з використанням шляхів довжиною 1 (тобто $\Gamma(x_i)$ – така множина вершин, для яких в графі існують дуги (x_i, x_j)) і оскільки Г (x_j) являється множиною вершин, досяжних із x_j з допомогою шляхів довжиною 1, то множина $\Gamma(\Gamma(x_i)) = \Gamma^2\ (x_i)$ складається із вершин досяжних із x_i з використанням шляхів довжиною 2. Аналогічно Γ^p (x_i) являється множиною вершин, які досяжні і x_i з допомогою шляхів довжиною р.

Так як будь-яка вершина графа G, яка досяжна із x_i, повинна бути досяжна з використанням шляху (або шляхів) довжиною 0, або 1, або 2,…, або р (з деякими кінцевими, але, можливо достатньо більшим значенням р), то множина вершин, досяжних із x_i, можна представити в вигляді (8.1).

$$R(x_i)=\{x_i\}\cup \Gamma^2(x_i)\cup \Gamma^2(x_i)\cup ...\cup \Gamma^p(x_i). \qquad (10.1)$$

Таким чином, множина $R(x_i)$ може бути получено отримано послідовним виконанням (зліва на право) операція об'єднання в співвідношенні (1), до тих пір, поки «протікаюче» не перестане збільшуватися по розміру при наступній операції об'єднання. З цього моменту наступні операції не будуть давати нових членів множині і, таким чином буде утворено досяжна множина $R(x_i)$. Число об'єднань, яке потрібно виконати, залежить від графа, але, очевидно, що число p менше числа вершин в графі.

Матрицю досяжностей можна вистроїти так. Знаходимо досяжні вершини $R(x_i)$ для всіх вершин $x_i \in X$ способом, наведеним вище покладемо $r_{ij}=1$, якщо $x_j \in R(x_i)$, і $r_{ij}=0$ в протилежному випадку. Отримана таким образом матриця R являється матрицею досяжностей.

Матриця контрдосяжностей Q=$[q_{ij}]$:

$$q_{ij}=\begin{cases}1, \text{ якщо із вершини } x_j \text{ можна досягнути вершину } x_i. \\ 0 \text{ в противному випадку.}\end{cases}$$

Контрдосяжною множиною Q(x_i) графа G являється множина таких вершин, що із будь-якої вершини цієї множини можна досягнути вершину x_i. Аналогічно побудови досяжної множини $R(x_i)$ на основі відношення (10.1) можна «сформулювати» множину Q(x_i), використовуючи наступний вираз:

$$Q(x_i)=\{x_i\}\cup \Gamma^{-1}(x_i)\cup \Gamma^{-2}(x_i)\cup ...\cup \Gamma^{-p}(x_i). \qquad (10.2)$$

де $\Gamma^{-2}(x_i)=\Gamma^{-1}(\Gamma^{-1}(x_i))$ і т. д.

Операції виконуються зліва на право до тих пір, поки наступна операція об'єднання не перестане змінювати «поточну» множину Q(x_i).

Очевидно, що стовбець x_i матриці Q (в якому $q_{ij}=1$, якщо $x_j \in Q(x_i)$, і $q_{ij}=0$ в противному випадку) співпадає із рядком x_i матриці R, тобто $Q=R^t$, де R^t – матриця транспонована до матриці досяжностей R.

Приклад. Знайти матриці досяжностей і зворотніх досяжностей для графа G, приведеного на рис. 10.5.

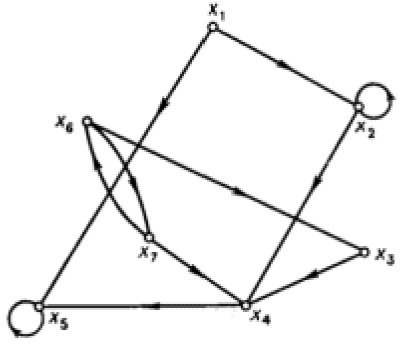


Рисунок 10.5 – Граф G

Множини досяжностей визначаються з співвідношення (10.1):

$R(x_1) = \{x_1\} \cup \{x_2, x_5\} \cup \{x_2, x_4, x_5\} \cup \{x_2, x_4, x_5\} = \{x_1, x_2, x_4, x_5\}$,

$R(x_2) = \{x_2\} \cup \{x_2, x_4\} \cup \{x_2, x_4, x_5\} \cup \{x_2, x_4, x_5\} = \{x_2, x_4, x_5\}$,

$R(x_3) = \{x_3\} \cup \{x_4\} \cup \{x_5\} \cup \{x_5\} = \{x_3, x_4, x_5\}$,

$R(x_4) = \{x_4\} \cup \{x_5\} \cup \{x_5\} \cup \{x_4, x_5\}$,

$R(x_5) = \{x_5\} \cup \{x_5\} = \{x_5\}$,

$R(x_6) = \{x_6\} \cup \{x_3, x_7\} \cup \{x_4, x_6\} \cup \{x_3, x_5, x_7\} \cup \{x_4, x_5, x_6\} =$
$= \{x_3, x_4, x_5, x_6, x_7\}$,

$R(x_7) = \{x_7\} \cup \{x_4, x_6\} \cup \{x_3, x_5, x_7\} \cup \{x_4, x_5, x_6\} = \{x_3, x_4, x_5, x_6, x_7\}$.

Матриця суміжності графа G має вид

$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

Матриця досяжностей буде мати вигляд

$$R = \begin{array}{c|ccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \hline x_1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ x_2 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ x_3 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ x_4 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ x_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ x_6 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ x_7 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}.$$

Матриця обернених досяжностей така:

$$Q = R^t = \begin{array}{c|ccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \hline x_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ x_2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_3 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ x_4 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ x_5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_6 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ x_7 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}.$$

Так як $R(x_i)$ є множиною вершин, досяжних із x_i, а $Q(x_j)$ – множина вершин із яких можна досягнути x_j, то $R(x_i) \cap Q(x_j)$ – множина таких вершин кожна із яких належить як мінімум до одного шляху слідуючому від x_i до x_j. Ці вершини називаються істотними або не від'ємними від відносно двох кінцевих вершин x_i і x_j. Всі інші вершини $x_k \in R(x_i) \cap Q(x_j)$ називаються несуттєвими або надлишковими, оскільки їх видалення впливає на шляхи від x_i до x_j.

Матриці досяжностей і зворотніх досяжностей, визначені вище, є повними в тому сенсі, що на довжини шляхів від x_i до x_j не накладались ніякі обмеження. З іншої сторони можна визначити матриці обмежених досяжностей і контрдосяжностей – потрібно вимагати щоб довжини шляхів не перевищували деякого заданого числа.

10.4 Ізоморфізм графів

Нехай $G_1 = (V_1, E_1)$ та $G_2 = (V_2, E_2)$ – прості графи, а $\varphi : V_1 \to V_1$ – бієкція. Якщо для будь-яких вершин u та v графа G_1 їх образи $\varphi(u)$ та $\varphi(v)$ суміжні в G_2 тоді й лише тоді, коли $\varphi : V_1 \to V_1$ суміжні в G_2, то цю бієкцію називають ізоморфізмом графа G_1, на граф G_2. Якщо такий ізоморфізм існує, то графи G_1 та G_2 називають ізоморфними та записують $G_1 \cong G_2$. Інакше кажучи, прості графи G_1 та G_2 називають ізоморфними, якщо існує така бієкція $\varphi : V_1 \to V_1$, що вершини $\varphi : V_1 \to V_1$ суміжні в G_1 тоді й лише тоді, коли вершини $\varphi(u)$ та $\varphi(v)$ суміжні G_2 для всіх $u, v \in V_1$ (у такому разі говорять, що ця бієкція зберігає суміжність вершин).

Приклад. Графи на рис. 10.6 ізоморфні, бієкцію можна задати так $\varphi(x_1) = y_1; \varphi(x_2) = y_2; \varphi(x_3) = y_3; \varphi(x_4) = y_4$.

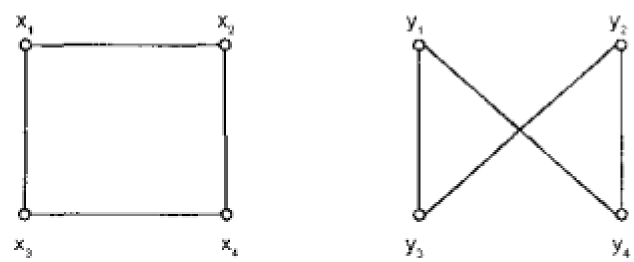


Рисунок 10.6 – Графи G_1 та G_2

Приклад. Усі три графи, зображені на рис. 10.7 ізоморфні.

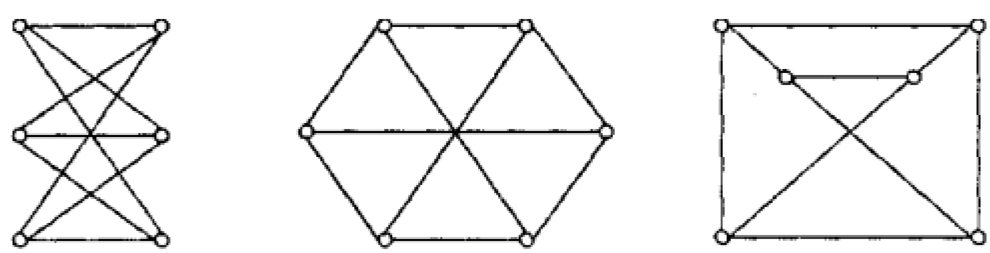


Рисунок 10.7 – Ізоморфні графи

Література: [4, 5, 6, 10].

ТЕМА 11. ЕЙЛЕРІВ ТА ГАМІЛЬТОНІВ ШЛЯХ ТА ЦИКЛ У ГРАФІ

11.1 Ейлерів цикл у графі

Початок теорії графів як розділу математики пов'язують із задачею про кенігсберзькі мости. Сім мостів міста Кенігсберга було розміщено на річці Прегель так, як зображено на рис. 11.1. Чи можна, починаючи з якоїсь точки міста, пройти через усі мости точно по одному разу й повернутися у початкову точку? Швейцарський математик Л. Ейлер розв'язав цю задачу. Його розв'язання опубліковане 1736 р. було першим явним застосуванням теорії графів. Ейлер поставив у відповідність плану міста мультограф G, вершини якого відповідають чотирьом частинам A, B, C, D міста, а ребра – мостам . Цей мультиграф зображено на рис. 11.1.

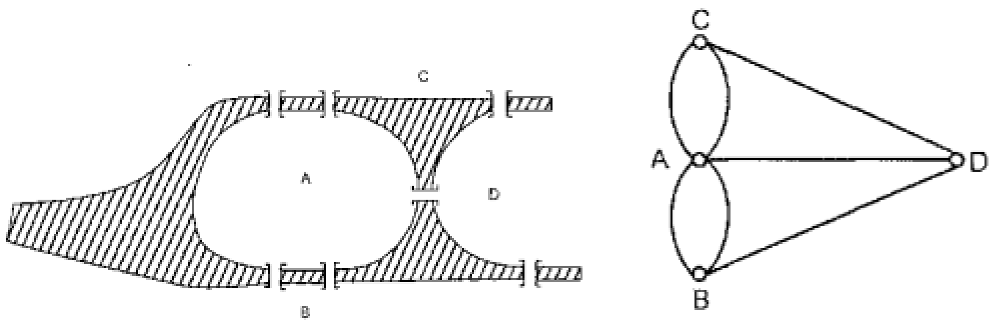


Рисунок 11.1 – Сім мостів та мультиграф, що їм відповідає

Отже задачу про кенігсберзькі мости мовою теорії графів можна сформулювати так: чи існує в мультографі G простий цикл, який містить усі ребра цього мультографа? Ейлер довів нерозв'язність задачі про кенігсберзькі мости. Нагадаємо, що в простому циклі ребра не повторюється, а вершини можуть повторюватись.

Ейлеровим циклом у зв'язному мультиграфі G називають простий цикл, який містить усі ребра графа. Ейлеровим шляхом у зв'язному мультиграфі G називають простий шлях, який містить усі ребра графа.

Приклад. На рис. 11.2 проілюстровано концепцію ейлеревих циклів і шляхів. Граф G_1 має ейлерів цикл, наприклад a,c,d,e,b,d,a; *граф* G_3 не має ейлеревого циклу, але має ейлеровий шлях a,c,d,e,b,d,a,b, *граф* G_2 не має ейлеревого циклу, ні ейлеревого шляху.

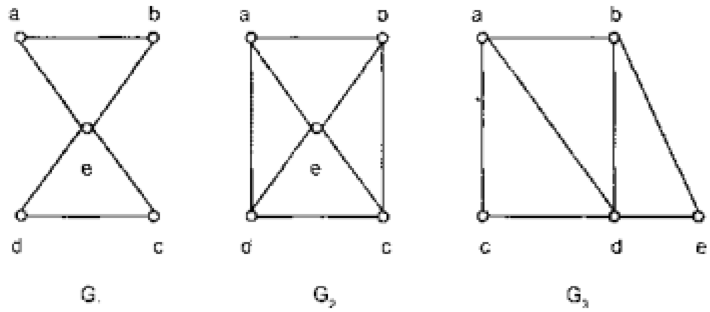


Рисунок 11.2 – Графи G_1, G_1, G_3

Існує простий критерій (необхідна і достатня умова) для виявлення, чи має граф ейлерів цикл.

Теорема 1. Зв'язний мультиграф має ейлерів цикл тоді і лише тоді, коли степені всіх його вершин парні.

Алгоритм Флері побудови ейлеревого циклу

Робота полягає в нумерації ребер у процесі побудови ейлеревого циклу.

Крок 1 (початок). Починаємо з довільної вершини u та присвоюємо довільному ребру $\{u, v\}$ номер 1. Викреслюємо ребро $\{u, v\}$ і переходимо у вершину v.

Крок 2 (ітерація). Нехай w – вершина, у яку ми перейшли на попередній ітерації, k – останній присвоєний номер. Вибираємо довільне ребро, інцидент не вершині , причому міст вибираємо лише тоді, коли немає інших можливостей. Присвоюємо вибраному ребру номер (k+1) і викреслюємо його.

Крок 3 (закінчення). Цей процес закінчуємо, коли всі ребра графа викреслено та пронумеровано ці номери задають послідовність ребер в ейлеровому циклі.

Повертаючись до задачі про кенігсберзькі мости, виявляємо, що мультограф на рис. 9.1 має всі вершини непарного степеня. Отже, цей мультограф не має ейлеревого циклу, тому неможливо пройти кожний міст по одному разу й повернутись у початкову точку шляху.

Теорема 2. Зв'язний мультиграф має ейлерів шлях, але не має ейлеревого циклу тоді й лише тоді, коли він має точно дві вершини непарного степеня.

Ейлеровим циклом у слабко зв'язному орієнтованому мультографі називають орієнтований простий цикл, який містить усі дуги графа.

Теорема 3. Орієнтований слабко зв'язний мультограф має ейлерів цикл тоді й лише тоді, коли напівстепінь входу кожної вершини дорівнює її напівстепеню виходу.

11.2 Гамільтонів цикл у графі

Шлях $x_0, x_1 \ldots, x_{n-1}$ у неорієнтованому зв'язному графі G=(V, E), V=$\{v_1, \ldots, v_n\}$ називають гамільтоновим шляхом якщо $x_i \in V$ i $x_i \neq x_j$ для $0 \leq i < j \leq n-1$. Цикл x_0, x_1, …,x_{n-1}, x_0 (тут $n \geq 3$) у графі G називають гамільтоновим циклом, якщо $x_0, x_1, \ldots, x_{n-1}$ – гамільтонів шлях.

Інакше кажучи, гамільтонів цикл і гамільтонів шлях проходять через кожну вершину графа точно один раз (у циклі окрім першої й останньої вершин). Зазначимо, що гамільтонові цикл і шлях, узагалі кажучи, не містять усіх ребер графа. За означенням гамільтонові цикли розглядають лише для графів, які мають не менше трьох вершин (отже, довжина гамільтонового циклу не менша трьох).

Термін «гамільтонів» у цих означеннях походить від імені відомого ірландського математика. У. Гамільтон (W. Hamilton), який 1857р. запропонував гру «Навколосвітня подорож». Кожній із двадцяти вершин додекаедра (правильного дванадцятигранника, грані якого – п'ятикутники) приписано назву одного з великих міст світу. Потрібно, розпочавши з довільного міста, відвідати решту 19 міст точно один раз, і повернутись у початкове місто. Перехід дозволено ребрами додекаедра.

Приклад. Цю задачу можна зобразити й на площині (рис. 11.3). Вона зводиться до відшукання в графі гамільтонового циклу. Один із можливих розв'язків показано потовщеними лініями.

Теорема 3. (Г. Дірак, 1952 р.) Якщо G – зв'язний простий граф з $n \geq 3$ вершинами і для кожної вершини v виконується нерівність $\deg(v)$ $n/2$, то граф G має гамільтонів цикл.

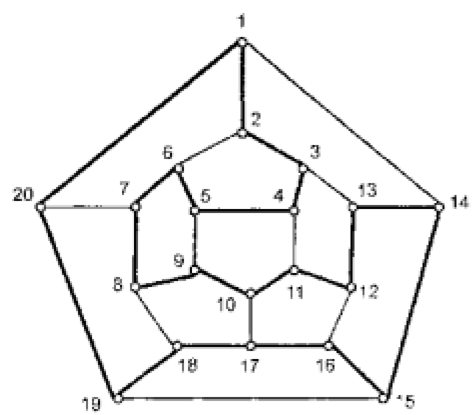


Рисунок 11.3 – Гамільтонів шлях

Література: [1, 4, 6, 7, 10, 11, 14].

ТЕМА 12. ЗВАЖЕНІ ГРАФИ ТА АЛГОРИТМИ ПОШУКУ НАЙКОРОТШИХ ШЛЯХІВ

12.1 Зважені графи

У реальних задачах на графах часто потрібно брати до уваги додаткову інформацію – фактично віддаль між окремими пунктами, вартість проїзду, час проїзду тощо. Для цього використовують поняття зваженого графа.

Зваженим називають простий граф, кожному ребру e якого приписане дійсне число w(e). Це число називають вагою ребра e.

Аналогічно означають зважений орієнтований граф: це такий орієнтований граф, кожній дузі e з якого приписане дійсне число w(e), назване вагою дуги.

Розглянемо три способи зберігання зваженого графа $G = (V, E)$в пам'яті комп'ютера. Нехай $|V| = n. |E| = m$. Перший – подання графа матрицею ваг W, яка являє собою аналог матриці суміжностей .

Її елементи w_{ij}=(v_i, v_j), якщо ребро {v_i, v_j }ϵ E (у разі орієнтованого графа-дуга $(v_i, v_j) \in E$). Якщо ж ребро $\{v_i, v_j\}$∈E (або дуга $(v_i, v_j) \in$ E), то w_{ij}=∞ залежно від розв'язуваної задачі.

Другий спосіб-подання графа списком ребер. Для зваженого графа під кожний елемент списку E можна відвести три комірки- дві для ребра й одну для його ваги, тобто всього потрібно 3 m комірок.

Третій спосіб-подання графа списками суміжності. Для зваженого графа кожний список Adj$[u]$ містить окрім вказівників на всі вершини v з множини Г (u) ще й числа w(u,v).

Довжиною шляху в зваженому графі називають суму ваг ребер (дуг), які утворюють цей шлях. Якщо граф не зважений, то вагу кожного ребра (кожної дуги) уважають рівною одиниці й отримують раніше введене поняття довжини шляху як кількості ребер (дуг) у ньому.

Задача про найкоротший шлях полягає в знаходженні найкоротшого шляху від заданої початкової вершини a до заданої вершини z. Наступні дві задачі – безпосередні узагальнення сформульованої задачі про найкоротший шлях.

1. Для заданої початкової вершини a знайти найкоротші шляхи від a до всіх інших вершин.

2. Знайти найкоротші шляхи між усіма парами вершини.

Виявляється, що майже всі методи розв'язання задачі про найкоротший шлях від заданої початкової вершини a до заданої вершини z також знаходять у процесі розв'язування й найкоротші шляхи від вершини a до всіх інших вершин графа. Отже, за їх допомогою можна розв'язати задачу 1 із невеликими додатковими обчислювальними витратами. З іншого боку задачу 2 можна розв'язати або n разів застосувавши алгоритм задачі 1 із різними початковими вершинами, або один раз застосувавши спеціальний алгоритм.

Розглянемо два алгоритми: перший алгоритм призначений для розв'язування задачі 1, другий – для задачі 2.

Найефективнішим алгоритмом визначення довжини найкоротшого шляху від фіксованої вершини до будь-якої іншої є алгоритм запропонований 1959р. датським математиком Е. Дейкстрою (E. Dijkstra). Цей алгоритм застосований лише у випадку, коли вага кожного ребра (дуги) додатна. Опишемо докладно цей алгоритм для орієнтованого графа.

Нехай G=(V,E)-зважений орієнтований граф, w(v_i, v_j) – вага дуги (v_i, v_j). Почавши з вершини a знаходимо віддаль від a до кожної із суміжних із нею вершин. Вибираємо вершину, віддаль від якої до вершини a найменша; нехай це буде вершина v^*. Далі знаходимо віддалі від вершини a до кожної суміжної з v^* вершини вздовж шляху, який проходить через вершину v^*. Якщо для якоїсь із таких вершин ця віддаль менша від поточної, то заміняємо нею поточну віддаль. Знову вибираємо вершину, найближчу до a, але таку, що не вибиралась раніше;повторюємо процес.

Описаний процес зручно виконувати за допомогою присвоювання вершинам міток.

Є мітки двох типів – тимчасові та постійні. Вершини з постійними мітками групують у множину М, яку називають *множиною позначених вершин*. Решта вершин має тимчасові мітки, і множину таких вершин позначаємо як Т, де T=V\M. Позначатимемо мітку (тимчасову чи постійну) вершини v як l(v).

Значення постійної мітки l(v) дорівню довжині найкоротшого шляху від вершини a до вершини v, тимчасової - довжині найкоротшого шляху, який проходить лише через вершини з постійними мітками.

Фіксованою початковою вершиною вважаємо вершину a; довжину найкоротшого шляху шукаємо до вершини z (або до всіх вершин графа).

12.2 Алгоритми пошуку найкоротших шляхів

Тепер опишемо алгоритм Дейкстри формально.

Опис Алгоритму Дейкстри

Присвоєння початкових значень

Крок 1. Виконати I(a):=0 і вважати цю мітку постійною. Виконати I(v):=∞ для всіх v≠a і вважати ці мітки тичасовими. Виконати x:=a, M:={a}.

Оновлення міток

Крок 2. Для кожної вершини $v \in \Gamma(x)\backslash M$ замінити мітки $I(v):=\min\{I(v);I(x)+w(x,v)\}$, тобто оновлювати тимчасові мітки вершин, у які з вершини x іде дуга.

Перетворення мітки в постійну.

Крок 3. Серед усіх вершин з тимчасовими мітками знайти вершину з мінімально міткою, тобто знайти вершину v^* за умови $I(v^*)=\min\{I(v)\}$, $v \in T$, де $T=V\backslash M$.

Крок 4.Уважати мітку вершини v^* постійною і виконати $M:=M\cup(v);x:= v^*$ (вершину v^* включено в множину M).

Закінчення

Крок 5. а) Для пошуку шляху від a до z: якщо x=z, то I(z)-довжина найкоротшого шляху від a до z, зупинитись, якщо $x\neq z$, то перейти до кроку2.

б) Для пошуку шляхів від a до всіх інших вершин: якщо всі вершини отримали постійні мітки(включені в множину M), то ці мітки дорівнюють довжинам найкоротших шляхів, зупинитись; якщо деякі вершини мають тимчасові мітки, то перейти до кроку 2.

Приклад. Знайдемо довжину найкоротшого шляху від початкової вершини *a* до вершини *z* у графі на рис. 12.1,*а*.

Послідовність дій зображено на рис. 12.1,*б – е,* мітки записано в дужках біля вершин. Вершини, які включено в множину *M,* обведено кружечками; мітки таких вершин оголошують постійними.

У процесі роботи алгоритму будують два вектори: вектор l постійних міток (довжини найкоротших шляхів від вершини *a* до даної вершини) і вектор θ вершин, з яких у дану вершину безпосередньо потрапляє найкоротший шлях.

У табл. 12.1 в першому рядку містяться довільно впорядковані вершини графа, у другому – відповідні постійні мітки (компоненти вектора l). а в третьому – компоненти вектора. Постійна мітка вершини z дорівнює 13. Отже, довжина найкоротшого шляху від a до z дорівнює 13. Сам шлях знаходять за допомогою першого й третього рядків таблиці та будують у зворотному порядку. Кінцева вершина z; у неї потрапляємо з вершини е (див. вектор у табл. 12.1). У вершину е потрапляємо з вершини d. у вершину d потрапляємо з b та продовжуємо цей процес до вершини a: a. Отже, найкоротший шлях такий a,c,b,d,e,z.

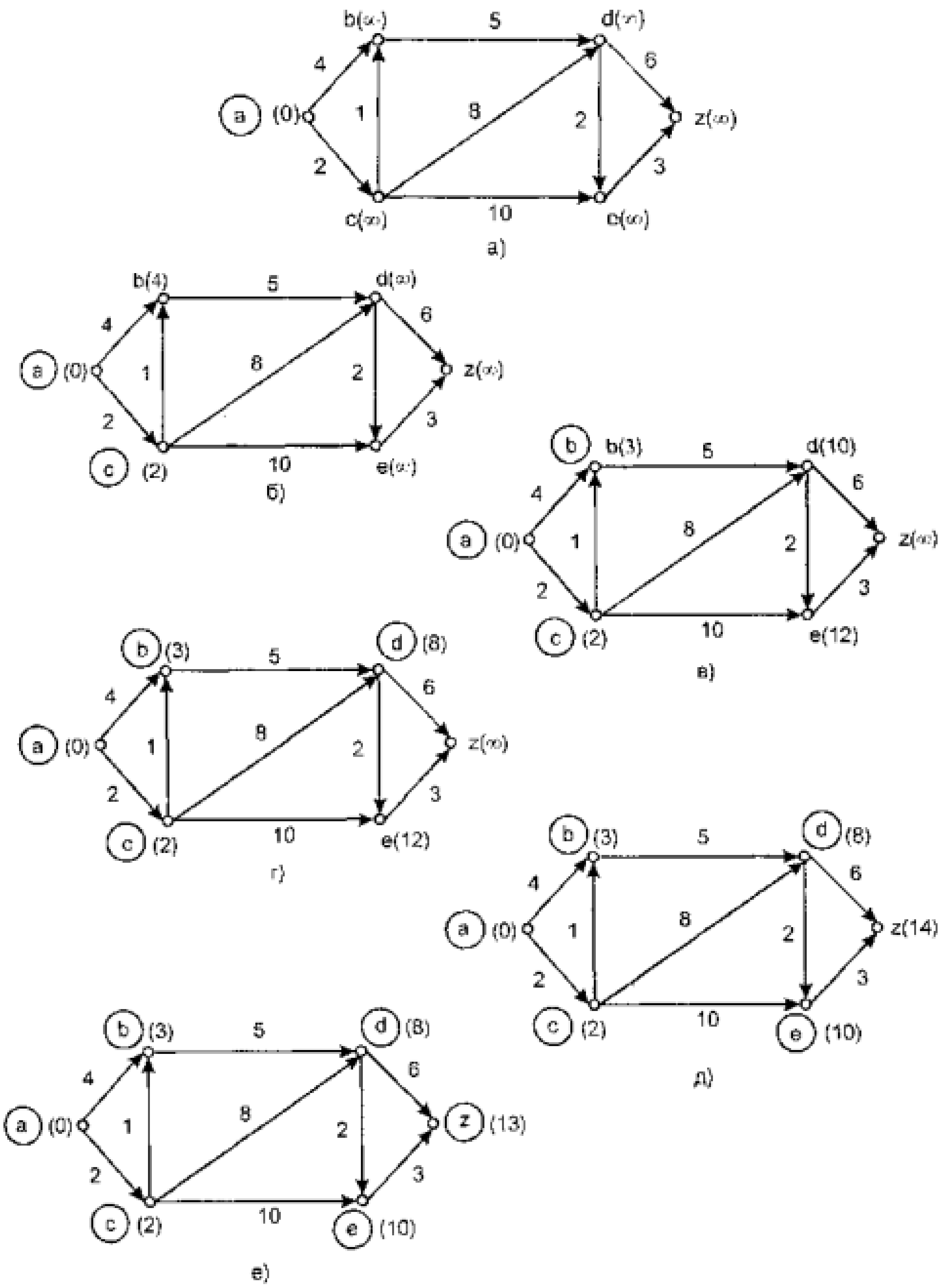


Рисунок 12.1 – Знаходження найкоротшого шляху

Таблиця 12.1 – Знаходження найкоротшого шляху

Вершини графа (елементи множини V)	a	b	c	d	e	z
Вектор l (постійні мітки вершин)	0	3	2	8	10	13
Вектор θ (вершини, з яких у дану вершину заходить найкоротший шлях)	-	c	a	b	d	e

Алгоритм Флойда

Ми розглянули задачу знаходження на графі найкоротшого шляху від якоїсь виділеної (початкової) вершини до будь-якої іншої. Розглянемо задачу пошуку на графі найкоротших шляхів між кожною парою вершин. Звичайно, цю загальнішу задачу можна розв'язати багатократним застосуванням алгоритму Дейкстри з послідовним вибором кожної вершини графа як початкової. Проте є й прямий спосіб розв'язування цієї задачі алгоритм Флойда. У цьому алгоритмі для довжин дуг допускаються від'ємні значення, проте не допускаються цикли від'ємної довжини.

Нехай *G=(V,* Г) – орієнтований граф. Нагадаємо, що коли $a, x_1, x_2 \ldots x_{m-1}, b$ – шлях у графі *G,* то внутрішніми вершинами шляху є $x_1, x_2 \ldots x_{m-1}$ Наприклад, внутрішніми вершинами шляху a, c, d, a, f, d є вершини c, d , a, f Пронумеруємо вершини графа цілими числами від 1 до *n*. Позначимо як $w_{ij}^{(k)}$ довжину найкоротшого шляху з вершини у вершину і, у якому як внутрішні можуть бути лише перші *k* вершин графа *G*. Якщо між вершинами і та j не існує жодного такого шляху, то умовно вважатимемо, що $w_{ij}^{(k)} = \infty$. Зі сказаного випливає, що $w_{ij}^{(0)}$ – це вага дуги (і j). а якщо такої дуги немає, то $w_{ij}^{(0)}$. Для довільної вершини і вважатимемо $w_{ii}^{(0)}$. Отже. $w_{ii}^{(n)}$ дорівнює довжині найкоротшого шляху з вершини *і у* вершину j. Позначимо як $W^{(K)}$ матрицю розміру *nxn, (і, j)*-й елемент якої дорівнює $w_{ij}^{(k)}$. Якщо в заданому орієнтованому графі *G* відома вага кожної дуги, то грунтуючись на попередніх міркуваннях можна сформувати матрицю $W^{(0)}$ – це матриця ваг цього графа. Наша мета тепер полягає у знаходженні матриці елементи якої дорівнюють довжинам найкоротших шляхів між усіма парами вершин графа G.

В алгоритмі Флойда як початкову беруть матрицю $W^{(0)}$. Спочатку за нею обчислюють матрицю $W^{(1)}$. потім за $W^{(1)}$ обчислюють матрицю $W^{(n)}$ і процес повторюють доти, доки за матрицею $W^{(n-1)}$ не буде обчислено матрицю $W^{(n)}$. Розглянемо ідею, на якій грунтується алгоритм Флойда. Припустимо, що відомі:

1) найкоротший шлях із вершини і у вершину *k,* у якому як внутрішні використано лише перші *(k-1)* вершин;

2) найкоротший шлях із вершини *k* у вершину j, у якому як внутрішні використано лише перші *(k-1)* вершин;

3) найкоротший шлях із вершини i у вершину j, у якому як внутрішні використано лише перші (k-1) вершин.

Оскільки за припущенням граф G не містить циклів із від'ємною довжиною, то один із двох шляхів - шлях із п. 3 чи об'єднання шляхів із пп. 1 та 2 - найкоротший шлях із вершини і у вершину j. у якому в якому як внутрішні використано лише перші *(k-1)* вершин. Отже,

$$w_{ij}^{(k)} = \min\left\{w_{ik}^{(k-1)} + w_{kj}^{(k-1)}, w_{ij}^{(k-1)}\right\} \tag{12.1}$$

Зі співвідношення (1) видно, що для обчислення елементів матриці $W^{(K)}$ потрібні тільки елементи матриці $W^{(K)-1}$. Тепер ми можемо формально описати алгоритм Флойда для знаходження в графі найкоротших шляхів між усіма парами вершин.

Опис Алгоритму Флойда

Крок 1. Присвоювання початкових значень. Пронумерувати вершини графа G цілими числами від 1 до n. Побудувати матрицю $W^{(0)}$ задавши кожний її (i, j)-й елемент рівним вазі дуги з вершини I у вершину j. Якщо в графі G такої дуги немає, то $w_{ij}^{(0)} := \infty$. Крім того, для всіх і виконати $w_{ij}^{(0)} := 0$

Крок 2. Ітерація. Для k, що послідовно набуває значення 1, 2,.... n. визначити за елементами матриці $W^{(k-1)}$ елементи матриці $W^{(k)}$, використовуючи рекурентне співвідношення (1).

Після закінчення цієї процедури величина елемента (i,j) матриці дорівнює довжині найкоротшого шляху з вершини і у вершину j.

Якщо під час роботи алгоритму для якихось k та і виявиться, що $w_{ii}^{(k)} \prec 0$, то в графі G існує цикл із від'ємною довжиною, який містить вершину i. Тоді роботу алгоритму потрібно припинити.

Якщо заздалегідь відомо, що в графі G немає циклів із від'ємною довжиною, то обсяг обчислень можна дещо зменшити. У цьому разі для всіх i та всіх k має бути $w_{ij}^{(k)} = 0$. Тому не потрібно обчислювати діагональні елементи матриць $W^{(1)}, W^{(2)} .. W^{(N)}$ Окрім того, для всіх i = 1, 2 ....n справджуються співвідношення $w_{ik}^{(k-1)} = w_{ik}^{(k)}, w_{ki}^{(k-1)} = w_{ki}^{(k)}$. Ці співвідношення випливають із того, що коли немає циклів із від'ємною довжиною, вершина k може бути внутрішньою в будь-яких найкоротших шляхах, котрі починаються чи закінчуються в самій вершині k. Отже, обчислюючи матрицю , немає потреби переобчислювати елементи *k-ого* рядка й *k-ого* стовпця матриці. Звідси випливає, то в матриці необхідно обчислювати за формулою (1) лише

n^2-3n+2 елементів, до яких не входять діагональні елементи, а також елементи k-го рядка й k-го стовпця. Очевидно, що складність алгоритму Флойда становить $O(n^3)$.

Щоб після закінчення роботи алгоритму Флойда можна було швидко знайти найкоротший шлях між будь-якою парою вершин, будемо на кожній ітерації разом із матрицею $W^{(k)}$ будувати матрицю $\Theta^{(k)}=\left[\theta_{ij}^{(k)}\right]$. Спочатку $\theta_{ig}^{(o)}:=i$ для всіх i j = 1,2. Далі на k-й ітерації $\theta_{ij}^{(k)}:=\theta_{ij}^{(k-1)}$. якщо $w_{ij}^{(k)}=w_{ij}^{(k-1)}$ і $\theta_{ij}^{(k)}:=\theta_{ij}^{(k-1)}$ якщо $w_{ij}^{(k)}=w_{ik}^{(k-1)}+w_{kj}^{(k-1)}$ Отже. $\theta_{ij}^{(k)}$ – номер вершини, яка є перед вершиною j у поточному (i, j)-шляху. Під поточним (I j)-шляхом розуміємо найкоротший (i j)-шлях. Усі внутрішні вершини якого містяться в множині $\{1,2...k\}$. Матриця $\Theta^{(n)}=\left[\theta_{ij}^{(n)}\right]$ відіграє ту саму роль, що й вектор θ в алгоритмі Дейкстри. За допомогою матриці $\Theta^{(n)}$ вершини, через які проходить найкоротший (i,j)-шлях визначають так $i,..j_3,j_2,j_3$.

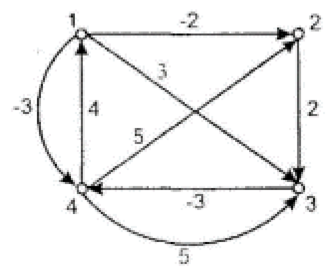


Рисунок 12.2 – Граф

Приклад. Знайдемо найкоротші шляхи між усіма парами вершин орієнтованого графа на рис. 12.2. Нижче наведено результати виконання кожної із чотирьох ітерацій алгоритму Флойда.

$$W^{(0)}=W=\begin{pmatrix}0&-2&3&-3\\\infty&0&2&\infty\\\infty&\infty&0&-3\\4&5&5&0\end{pmatrix}\qquad \Theta^{(0)}=\begin{pmatrix}0&1&1&1\\2&0&2&2\\3&3&0&3\\4&4&4&0\end{pmatrix}$$

$$W^{(1)}=\begin{pmatrix}0&-2&3&-3\\\infty&0&2&\infty\\\infty&\infty&0&-3\\4&2&5&0\end{pmatrix}\qquad \Theta^{(1)}=\begin{pmatrix}0&1&1&1\\2&0&2&2\\3&3&0&3\\4&1&4&0\end{pmatrix}$$

$$W^{(2)}=\begin{pmatrix}0 & -2 & 0 & -3\\ \infty & 0 & 2 & \infty\\ \infty & \infty & 0 & -3\\ 4 & 2 & 4 & 0\end{pmatrix} \qquad \Theta^{(2)}=\begin{pmatrix}0 & 1 & 2 & 1\\ 2 & 0 & 2 & 2\\ 3 & 3 & 0 & 3\\ 4 & 1 & 2 & 0\end{pmatrix}$$

$$W^{(3)}=\begin{pmatrix}0 & -2 & 0 & -3\\ \infty & 0 & 2 & -1\\ \infty & \infty & 0 & -3\\ 4 & 2 & 4 & 0\end{pmatrix} \qquad \Theta^{(3)}=\begin{pmatrix}0 & 1 & 2 & 1\\ 2 & 0 & 2 & 3\\ 3 & 3 & 0 & 3\\ 4 & 1 & 2 & 0\end{pmatrix}$$

$$W^{(4)}=\begin{pmatrix}0 & -2 & 0 & -3\\ 3 & 0 & 2 & -1\\ 1 & -1 & 0 & -3\\ 4 & 2 & 4 & 0\end{pmatrix} \qquad \Theta^{(4)}=\begin{pmatrix}0 & 1 & 2 & 1\\ 4 & 0 & 2 & 3\\ 4 & 1 & 0 & 3\\ 4 & 1 & 2 & 0\end{pmatrix}$$

Матриця $W^{(4)}$ дає довжини найкоротших шляхів між усіма парами вершин графа на рис. 12.2. Зокрема, $w_{21}^{(4)}=3$, тобто довжина найкоротшого шляху від вершини 2 до вершини 1 дорівнює 3. Для знаходження самого шляху скористаємось матрицею $\Theta^{(4)}$ і випишемо у зворотному порядку вершини, через які він проходить: $\theta_{21}^{(4)}=4, \theta_{24}^{(4)}=3, \theta_{23}^{(4)}=2$. Отже, найкоротший шлях із вершини 2 у вершину 1 проходить через вершини 2,3,4,1.

Література: [5, 6, 14, 15, 17].

ТЕМА 13. ОБХІД ГРАФІВ. ПЛАНАРНІ ГРАФИ

Існує багато алгоритмів на графах, які ґрунтуються на систематичному переборі їх вершин або обході вершин, під час якого кожна вершина одержує унікальний порядковий номер. Алгоритми обходу вершин графа називають методами пошуку.

13.1 Обхід графа пошуком вглиб

Опишемо метод пошуку в простому зв'язному графі. Цей метод називають пошуком углиб, або DFS-методом (від англ. Depth First Search).

Нехай G=(V, E) – простий зв'язний граф, усі вершини якого позначені попарно різними символами. У процесі пошуку вглиб вершинам графа G надають номери (DFS-номери). та певним чином позначають ребра. У ході роботи алгоритму використовують структуру даних для збереження множин, яку називають стеком (англ. stack –

стег). Зі стеку можна вилучити тільки той елемент, котрий було додано до нього останнім: стек працює за принципом «останнім прийшов – першим вийшов» (last in, first out – скорочено LIFO). Інакше кажучи, додавання й вилучення елементів у стеку відбувається з одного кінця, який називають верхівкою стеку. DFS-номер вершини x позначають як DFS(x).

Алгоритм пошуку вглиб у простому зв'язному графі

Крок 1. Почати з довільної вершини v_s. Виконати DFS(v):=1. Включити цю вершину в стек.

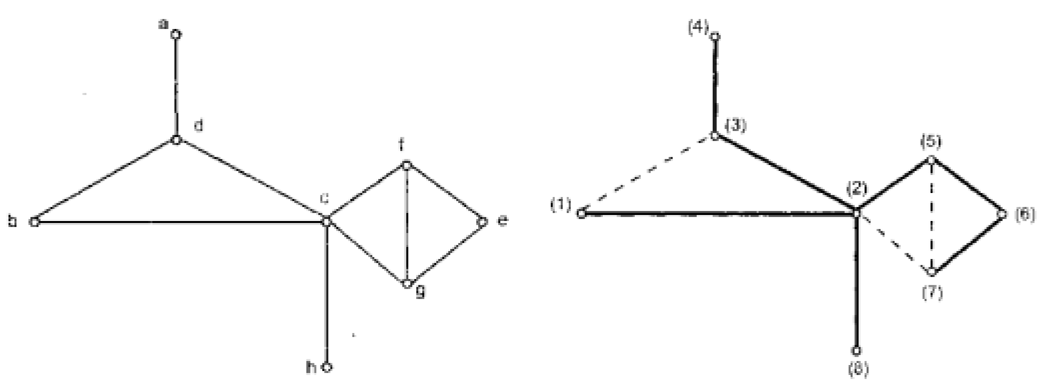
Крок 2. Розглянути вершину у верхівці стеку: нехай це буде вершина x. Якщо всі ребра, інциденти вершині x, позначено, то перейти до кроку 4, інакше до кроку 3.

Крок 3. Нехай {x, y} – непозначене ребро. Якщо DFS(y) уже визначено, то позначити ребро (x, y) штриховою лінією та перейти до кроку 2. Якщо DFS(y) не визначено, то позначити ребро (x, y) потовщеною суцільною лінією, визначити DFS(y) як черговий DFS-номер, включити цю вершину в стек і перейти до кроку 2.

Крок 4. Виключити вершину x зі стеку. Якщо стек порожній, то зупинитись, інакше – перейти до кроку 2.

Щоб вибір номерів був однозначним, доцільно домовитись, що вершини, суміжні з тією, яка вже отримала DFS-номер. аналізують за зростанням їх нумерації (або в алфавітному порядку). Динаміку роботи алгоритму зручно відображати за допомогою таблиці з трьома стовпцями: вершина, DFS-номер, уміст стеку. її називають протоколом обходу графа пошуком вглиб.

Приклад 13.1. Виконаємо обхід графа на рис. 13.1а пошуком углиб, починаючи з вершини b. Розв'язок подано на рис. 13.1б протокол пошуку вглиб подано в табл. 13.1. У цій таблиці в третьому стовпці вважаємо, що верхівка стеку праворуч.



а б

Рисунок 13.1 – Граф

Ребра, які позначено потовщеною суцільною лінією, називають «прямими», а пунктирною – «зворотними». Їх використовують у різних алгоритмах на графах, заснованих на алгоритмі пошуку вглиб.

Таблиця 13.1 – Протокол пошуку вглиб

Вершина	DFS-номер	Вміст стеку
b	1	*b*
c	2	*bc*
d	3	*bcd*
a	4	*bcda*
-	-	*bcd*
-	-	*bc*
f	5	*bcf*
e	6	*bcfe*
g	7	*bcfeg*
-	-	*bcfe*
-	-	*bcf*
-	-	*bc*
h	8	*bch*
-	-	*bc*
-	-	*b*
-	-	$\varnothing$

13.2 Обхід графа пошуком вшир

У процесі пошуку вшир вершини графа проглядають в іншій послідовності, ніж у методі пошуку вглиб, і їм надають BFS-номери (від англ. Breadth First Search). BFS-номер вершини v позначають як BFS(x). Назва пояснюється тим, що під час пошуку рухаються вшир, а не вглиб: спочатку проглядаються всі сусідні вершини, після цього - сусіди сусідів і так далі.

У ході реалізації алгоритму використовують структуру даних для збереження множин, яку називають чергою (англ. queue). Із черги можна вилучити тільки той елемент, який перебував у ній найдовше: працює принцип «першим прийшов - першим вийшов» (first in, first out - скорочено FIFO). Елемент включається в хвіст черги, а виключається з її голови. Пошук ушир, узагалі кажучи, відрізняється від пошуку вглиб заміною стеку на чергу. Після такої модифікації що раніше відвідується вершина (включається в чергу), то раніше вона використовується (і виключається із черги). Використання вершини полягає в перегляді одразу всіх іще не відвіданих її сусідів. Алгоритм пошуку вшир у простому зв'язному графі.

Крок 1. Почати з довільної вершини v_s. Виконати BFS(v_s):=1. Включити вершину v_s у чергу.

Крок 2. Розглянути вершину, яка знаходиться на початку черги, нехай це буде вершина x. Якщо для всіх вершин, суміжних із вершиною x, уже визначені BFS-номери, то перейти до кроку 4, інакше до кроку 3.

Крок 3. Нехай {x, y} – ребро, у якому номер BFS(y) не визначений. Позначити це ребро потовщеною суцільною лінією, визначити BFS(y) як черговий BFS-номер, включаємо вершину у чергу й перейти до кроку 2.

Крок 4. Виключити вершину x із черги. Якщо черга порожня, то зупинитись, інакше - перейти до кроку 2.

Щоб результат виконання алгоритму був однозначним, вершини, які суміжні з вершиною x, аналізують за зростанням їх нумерації (або в алфавітному порядку). Динаміку роботи алгоритму пошуку вшир також зручно відображати за допомогою протоколу обходу, це уміст черги (вважаємо, що голова черги ліворуч, а хвіст - праворуч).

Приклад 13.2. Виконаємо обхід графа на рис. 13.2 а пошуком вшир, починаючи з вершини b. Розв'язок зображено на рис. 13.2 б; протокол пошуку вшир подано в табл. 13.2.

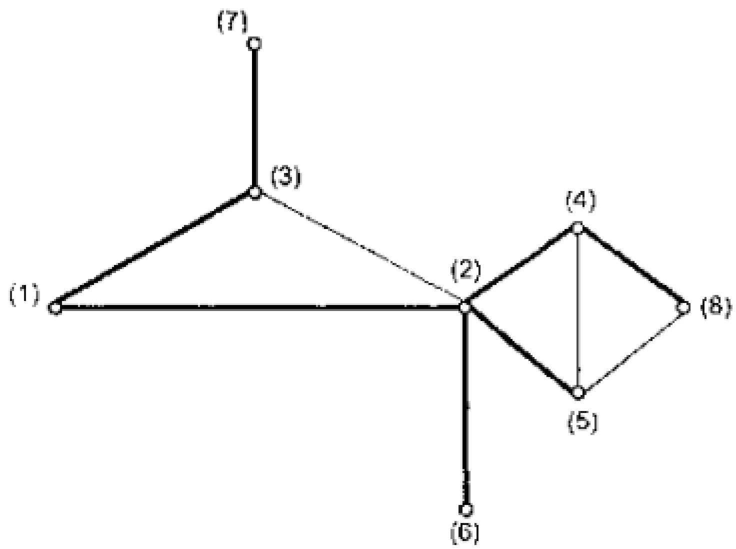


Рисунок 13.2 – Граф

Обчислювальна складність обох алгоритмів обходу однакова й у разі подання графа списками суміжності становить 0(m+n). де m— кількість ребер, а n – кількість вершин графа. Отже, наведені алгоритми мають лінійну складність, тобто їх ефективність дуже висока.

Таблиця 13.2 – Протокол пошуку вшир

Вершина	BFS -номер	Вміст черги
b	1	*b*
c	2	*bc*
d	3	*bcd*
-	-	*cd*
f	4	*cdf*
g	5	*cdfg*
h	6	*cdfgh*
-	-	*dfgh*
a	7	*dfgha*
e	-	*fgha*
-	8	*fghae*
-	-	*ghae*
-	-	*hae*
-	-	*ae*
-	-	*e*
-	-	∅

13.3 Планарні графи

Розглянемо неорієнтовані графи (кратні ребра і петлі дозволені).

Плоским називають граф, зображений на площині так. що ніякі два його ребра геометрично не перетинаються ніде, окрім інцидентних їм вершин. Граф, ізоморфний до плоского графа, називають планарним.

Приклад. Усі три графи на рис. 13.3 планарні, але лише другий і третій із них плоскі.

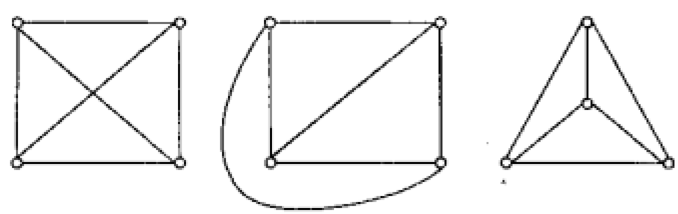


Рисунок 13.3 – Графи

Теорема Ейлера пов'язує кількість вершин, ребер і граней зв'язного плоского графа. Попередньо дамо означення поняття грані плоского зв'язного графа.

Внутрішньою гранню плоского зв'язного графа називають область площини, обмежену циклом і таку, що не містить всередині себе ні вершин, ні ребер графа. Цикл, що обмежує грань, називають межею грані. Частину площини, що складається з точок, які не належать ні графу, ні жодній із його внутрішніх граней, називають зовнішньою (або необмеженою) гранню.

Приклад 13.2. На рис. 13.4 зображено плоский граф. Він має 4 грані: r_1, r_2, r_3, r_4, грань r_4 – зовнішня.

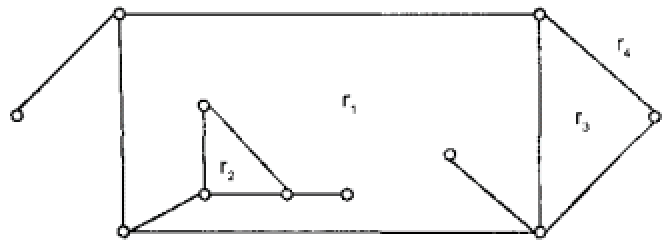


Рисунок 13.4 – Грані графа

Якщо плоский граф двозв'язний (тобто зв'язний і не має точок з'єднання), то межа грані являє собою цикл без повторюваних вершин (окрім першої та останньої вершин циклу).

Теорема 1 (теорема Ейлера про плоскі графи). Нехай зв'язний плоский граф G має n вершин, m ребер та r граней. Тоді

$n + r = m + 2.$

Теорема 2. Графи K_5 і $K_{3,3}$, непланарні.

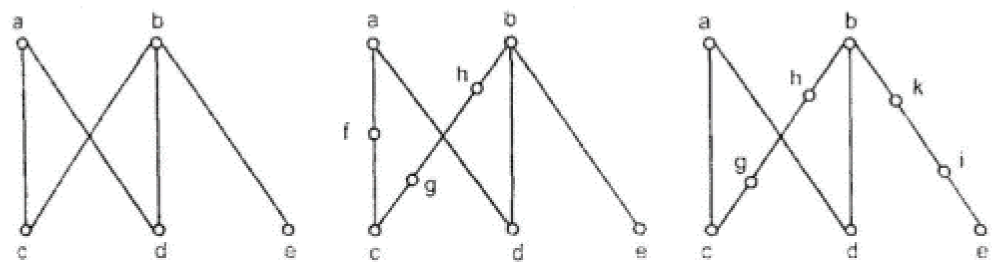


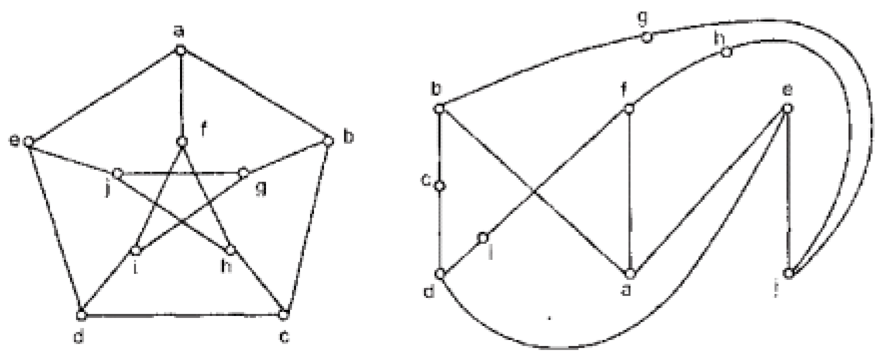
Рисунок 13.5 – Графи

Два графи називають гомеоморфними (або тотожними і точністю до вершин степеня 2), якщо вони можуть бути отримані з одного графа включенням у його ребра нових вершин степеня 2. Наприклад, графи, зображені на рис. 13.5 гомеоморфні, і те ж можна сказати про будь-які два цикли C_n.

Зрозуміло, що введення терміну «гомеоморфні графи» зручне лише з технічних міркувань, бо включення чи виключення вершин степеня 2 ніяк не пов'язано з планарністю. Проте це дає нам змогу сформулювати результат, відомий як теорема Куратовського. Ця теорема дає критерій (необхідну й достатню умову) планарності графа.

Теорема 3 (Куратовськнй, 1930 р.). Граф пленарний тоді й лише тоді. коли він не містить підграфів, гомеоморфних жодному з графів K_5 і $K_{3,3}$.

Приклад 13.3. На рисунку 13.6 а зображено граф Петерсена, а на рис 13.6 б – його підграф. гомеоморфний $K_{3,3}$. Отже, за теоремою Куратовського граф Петерсена неплапарний.



а б

Рисунок 13.6 – Граф Петерсена

Граф K_5 простий повний граф.

За теоремою 2 Граф K_5 непланарний.

Граф K_5 і $K_{3,3}$. неплоскі графи.

Розглянемо задачі про з'єднання n будинків та m пунктів обслуговування з допомогою комунікацій.

Відома задача «про три будинки та три колодязі», в якій жильці будинків хотіли б ходити во воду до колодязів так, щоб ніколи не перетинатися зі своїми сусідами. Для цього необхідно щоб їх шляхи не перетиналися (рис. 13.7).

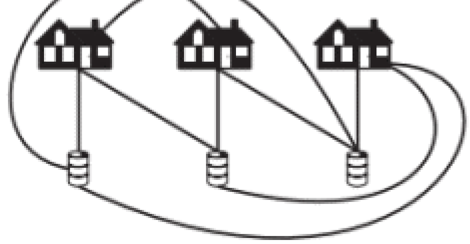


Рисунок 13.7 – Задача «про три будинки та три колодязі»

Література: [5, 6, 9, 14,15].

ТЕМА 14. РОЗФАРБОВУВАННЯ ГРАФІВ. НЕЗАЛЕЖНІ МНОЖИНИ. КЛІКИ

14.1 Розфарбування графів

Розфарбуванням простого графа *G* називають таке приписування кольорів (або натуральних чисел) його вершинам, що ніякі дві суміжні вершини не набувають однакового кольору. Найменшу можливу кількість кольорів у розфарбуванні називають хроматичним числам і позначають $x(G)$. Очевидно, що існує розфарбування графа в k кольорів (використовують термін k- розфарбування) для будь-якого $к$ в діапазоні $x(G)<k<n$, де n - кількість вершин графа. Множину вершин, розфарбованих в один колір, називають одноколірним класом. Такі класи утворюють незалежні множини вершин, тобто ніякі дві вершини в одноколірному класі не суміжні.

Очевидно, що $x(k_n)$=n отже, легко побудувати графи з як завгодно великим хроматичним числом. З іншого боку, $x(G)$ = 1 тоді й лише тоді, коли G- порожній граф. x(G) = 2 тоді й лише тоді, коли G- дводольний граф (зважаючи на це, дводольний граф називають біхроматичним).

Теорема 1. Якщо найбільший зі степенів вершин графа *G* дорівнює р. то цей граф можна розфарбувати в р + 1 колір.

Теорема 2 (Брукс, 1941). Якщо G-зв'язний граф, який не є повним, і найбільший зі степенів його вершин дорівнює р (р>3). то цей граф можна розфарбувати в р кольорів.

Із середини XIX століття відкритою залишалась проблема, відома під назвою гіпотези чотирьох фарб. її формулюють так: будь-який планарни граф можна розфарбувати в чотири кольори, тобто x(G) <4 для будь-якого планарного графа *G*.

Теорема 3 (Гейвуд). Будь-який пленарний граф можна розфарбувати в п'ять кольорів, тобто x(C) < 5 для будь-якого планарного графа G.

Американські математики Апель (Appel) і Гайкен (Haken) 1976 р. довели гіпотезу чотирьох фарб, великою мірою використавши комп'ютерні обчислення. Це перший випадок, коли настільки відому математичну проблему було розв'язано за допомогою комп'ютера. Спочатку проблему було зведено до розгляду скінченної (хоча й великої) кількості випадків. Надалі список «підозрілих» графів було зменшено й за допомогою комп'ютера розфарбовано в чотири кольори пленарні графи з цього списку. Для цього в 1976 р. за різними джерелами було потрібно від 1500 до 2000 годин роботи потужного комп'ютера. Отже, проблему чотирьох фарб було розв'язано. Хоча сам по собі метод доведення являє собою видатне досягнення й цілком достатній, щоб припинити пошуки контр прикладу, було б добре, щоб хтось знайшов елегантніше доведення гіпотези. Найцікавішим у цьому доведенні є те. що воно розширило наше уявлення про математичне доведення. Проте не всі поділяють цю думку. Опоненти вважають, що важко погодитись із таким доведенням, бо й зведення загального випадку до скінченної множини графів, і розфарбування останніх дуже важко повторити.

Хроматичні поліноми

Нехай *G* – простий граф, а *PG(k)* дорівнює кількості способів, якими можна розфарбувати вершини графа *G* в *k* кольорів. Функцію *PG(k)* називають хроматичною функцією графа *G*. Для повного графа *K* виконується рівність $PG(k)= k(k\text{-}1)(k\text{-}2)...(k\text{-}n+1)$. Справді, для першої вершини є *k* варіантів кольору. Оскільки друга вершина суміжна з першою, то її можна розфарбувати в один з *k*- 1 кольорів. Третя вершина суміжна з першою та другою, тому для неї є *k-2* варіантів вибору. Продовжуючи ці міркування, доходимо висновку, що для останньої вершини залишиться *k-(n-1)* варіантів вибору кольору.

Якщо $k<x(G)$, то $P(\kappa)=0$, а для $k>x(G)$ виконується нерівність $P(\kappa)>0$. Гіпотеза чотирьох фарб еквівалентна твердженню: якщо *G* - простий пленарний граф, то $P(4)>0$. Якщо задано довільний простий граф, то в загальному випадку важко отримати його хроматичну функцію за допомогою простих міркувань. Наступне теорема йнаслідок із неї дають систематичний метод одержання хроматичної функції простого графа у вигляді суми хроматичних функцій повних графів

Розглянемо деякі практичні задачі, які зводяться до розфарбування графів.

Задача складання розкладу. Припустимо, що потрібно прочитати декілька лекцій у найкоротший термін. Кожна лекція окремо займає одну годину, але деякі лекції т не можна читати одночасно (наприклад, якщо це робить один і той самий лектор). Побудуємо граф *G,* вершини якого взаємно однозначно відповідають лекціям, і дві вершини суміжні тоді й лише тоді, коли відповідні лекції не можна читати одночасно. Очевидно, що будь-яке розфарбування цього графа задає можливий розклад: лекції, які відповідають вершинам одноколірного класу, читають одночасно. Навпаки, будь-який можливий розклад задає розфарбування графа *G*. Оптимальні розклади відповідають розфарбуванням мінімальною кількістю кольорів, а час. необхідний для читання всіх лекцій, дорівнює $x(G)$.

Задача розподілу обладнання. Задано множини $V = \{v_1, \dots, v_n\}$ та $S = \{s_1, \dots s_n\}$ відповідно робіт і механізмів. Для виконання кожної роботи потрібен певний час. однаковий для всіх робіт, і якісь механізми. Жоден із механізмів не може бути зайнятий на декількох роботах одночасно. Потрібно розподілити механізми так. щоб загальний час виконання всіх робіт був мінімальним. Побудуємо граф G з множиною вершин V, вершини v_i та v_j ($i \neq j$) якого суміжні тоді й лише тоді, коли для виконання робіт v_i та v_j потрібний хоча б один спільний механізм. Розфарбуємо граф G. Роботи, що відповідають вершинам одного кольору, можна виконувати одночасно, а мінімальний час виконання всіх робіт відповідає розфарбуванню мінімальною кількістю кольорів.

Задача призначення телевізійних каналів. Передавальні станції зображають вершинами графа. Якщо віддаль між будь-якими двома станціями менша за певну величину l, то відповідні вершини графа з'єднують ребром. Граф розфарбовують і зіставляють різним кольорам вершин різні капали. Мінімальна кількість каналів відповідає розфарбуванню графа мінімальною кількістю кольорів.

14.2 Незалежні множини вершин. Кліки

Нехай G – простий граф. Множину його вершин називають незалежною (або внутрішньо стійкою), якщо ніякі вершини цієї множини не суміжні. Незалежну множину називають максимальною, якщо вона не є підмножиною жодної іншої незалежної множини з більшою кількістю вершин.

Найпотужнішу максимальну незалежну множину називають найбільшою. Кількість вершин у найбільшій незалежній множині графа G називають числом незалежності (або числом внутрішньої стійкості, нещільністю) і позначають як $a(G)$.

Приклад. У графі, зображеному на рис. 13.8 розглянемо множини вершин: $Y_1 = \{v_2, v_5, v_7, v_8\}$, $Y_2 = \{v_2, v_7, v_8\}$, $Y_3 = \{v_1, v_3, v_7, \}$, $Y_4 = \{v_4, v_6, \}$.

Максимальні незалежні множини – Y_1, Y_3, Y_4. Множина вершин Y_2 незалежна, але не максимальна. Найбільша незалежна множина це $Y_1 = \{v_2, v_5, v_7, v_8\}$. Отже. $a(G)=4$.

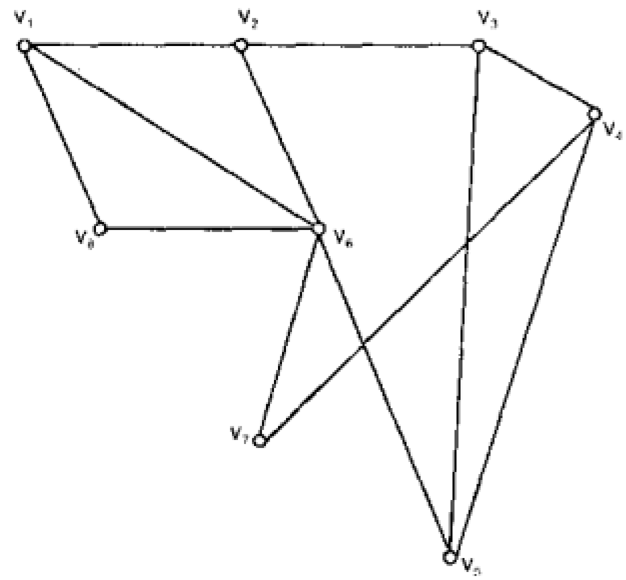


Рисунок 14.1 – Граф

Із поняттям незалежності в графі пов'язане поняття домінування. Підмножину V' вершин графа G=(V, E) називають домінантною (або зовнішньо стійкою), якщо кожна вершина з $V \setminus V'$ суміжна з якоюсь вершиною з V'. інакше кажучи, кожна вершина графа віддалена від домінантної множини не більше, ніж на одиницю. Домінантна множина має назву мінімальної, якщо жодна з її власних підмножин не домінантна. Домінантну множину з найменшою потужністю називають найменшою. У відшуканні в графі найменшої домінантної множини полягає зміст багатьох практичних задач. Типова задача така. Є множина населених пунктів, зв'язаних дорогами. У деяких із них потрібно розмістити підприємства обслуговування так, щоб віддаль від кожного з населених пунктів до будь-якого підприємства не перевищувала заданої величини. Розміщення слід виконати так, щоб обійтися мінімальною кількістю підприємств. Населеним пунктам поставимо у відповідність вершини графа, у якому дві вершини з'єднано ребром тоді і лише тоді, коли віддаль між відповідними пунктами не перевищує заданої величини. Тоді задача зводиться до побудови в цьому графі найменшої домінантної множини.

Уведемо ще одне поняття, пов'язане з поняттям незалежності. Говорять, що вершина й ребро графа покривають одне одного, якщо вони інциденті. Отже, ребро $e = \{u, v\}$ покриває вершини u та v, а кожна з цих вершин покриває ребро e. Підмножину вершин $V' \subset V$ називають покриттям (або вершинним покриттям, опорою) графа G=(V, *E*), якщо кожне ребро з *E* інцидентне принаймні одній вершині з V'.

Покриття графа *G* називають мінімальним, якщо воно не містить покриття з меншою кількістю вершин, і найменшим, якщо кількість вершин у ньому найменша серед усіх покриттів графа G. Кількість вершин у найменшому покритті графа *G* називають числом покриття (або числом вершинного покриття) графа G, і позначають як *b(G)*.

Приклад. У графі, зображеному на рис. 38, кожна з множин $X_1 = \{v_1, v_3, v_4, v_6\}$, $X_2 = \{v_1, v_3, v_4, v_5, v_6\}$, $X_3 = \{v_2, v_4, v_5, v_6, v_8\}$, $X_4 = \{v_1, v_2, v_3, v_5, v_7, v_8\}$ являє собою покриття, причому *X1 X3 X4* – мінімальні покриття. Покриття *X1* - найменше.

Наступна теорема свідчить про тісний зв'язок між покриттями та незалежними множинами графа.

Теорема 4. Множина X вершин графа G=(V, *E)* являє собою (найменше, мінімальне) покриття тоді й лише тоді, коли Y=V\X - (найбільша, максимальна) незалежна множина. Отже. a(G) + b(G)=|V|

Теорема 5. Підмножина вершин графа *G* являє собою кліку тоді й лише тоді, коли вона незалежна в доповнювальному графі *G*. Отже, $\varphi(G) = \alpha(G)$.

З теореми 5 випливає, що побудову максимальної кліки можна звести до побудови максимальної незалежної множини вершин.

Пароспoлучення в графах. Теорема Голла

Теорема, яку сформулював Ф. Голл має багато різних застосувань, два з яких ми розглянемо.

Задача про весілля. Розглянемо деяку множину юнаків, кожний з яких знайомий із кількома дівчатами (табл. 14.1). Потрібно визначити умови, за яких кожен з юнаків міг би одружитися зі знайомою йому дівчиною.

Таблиця 14.1 – Задача про весілля

Юнак	Дівчата, з якими знайомий юнак
b_1	g_1, g_4, g_5
b_2	g_1
b_3	g_2, g_3, g_4
b_4	g_2, g_4

Досконале пароспoлучення в дводольному графі. Пароспoлученням (або незалежною множиною ребер) *у* простому графі G= (*V, E)* називають множину ребер, у якій ніякі два ребра не суміжні.

Паросполучення графа G називають максимальним, якщо воно не міститься и жодному паросполучені з більшою кількістю ребер.

Нехай $G = (V_1 \cup V_2, E)$ – дводольний граф ($V_1 \cap V_2 = 0$ і кінці кожного ребра належать різним множинам V_1 та V_2). Досконалим паросполученням із V_1 у V_2 називають паросполучення. яке покриває вершини V_1 (це означає, що всі вершини з V_1 інцидентні ребрам, які утворюють паросполучення). За яких умов існує досконале паросполучення з V_1 у V_2.

Можна довести, що задачі 1 та 2 – це по суті, одна й та сама задача. Справді, нехай V_1 – множина юнаків. V_2 – множина дівчат, ребра – знайомства юнаків із дівчатами (рис. 14.2). У такому разі досконале паросполучення в дводольному графі – це шукана множина весіль (один із можливих розв'язків на рисунку зображено потовщеними лініями).

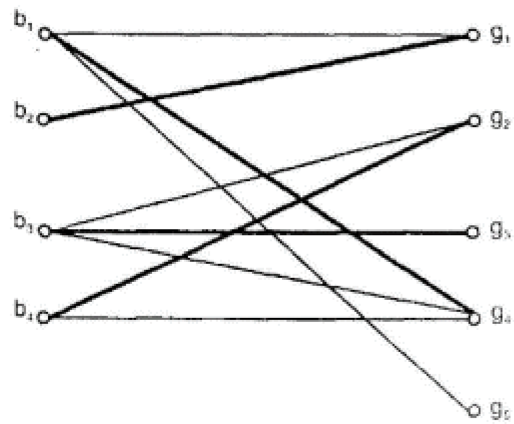


Рисунок 14.2 – Граф

Теореми 6 (Голл, 1935 р.). Розв'язок задачі про весілля існує тоді й лише тоді, коли будь-які k юнаків із даної множини знайомі в сукупності принаймні з k дівчатами (для як m позначено загальну кількість юнаків).

Література: [1, 4, 5, 6, 7, 10].

ТЕМА 15. ДЕРЕВА ТА ЇХ ЗАСТОСУВАННЯ

15.1 Основні означення та властивості

Деревам називають зв'язний граф без простих циклів. Граф, який не містить простих циклів і складається з k компонент, називають лісом із k дерев.

Приклад 15.1. На рис. 15.1 зображено приклади дерев. Граф, який зображено на рисунку 15.2 – не дерево, бо він незв'язний.

З означення випливає, що дерева й ліси являють собою прості графи.

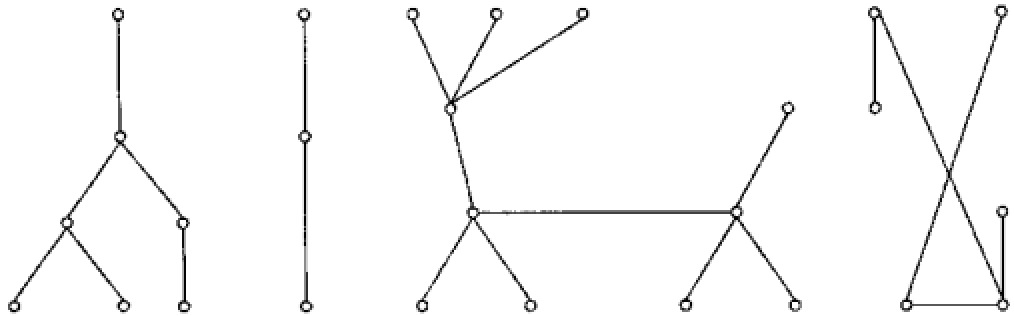


Рисунок 15.1 – Приклади дерев

Теорема 1. Нехай граф Г має n вершин. Тоді такі твердження еквівалентні:

I. граф Т- дерево;
II. граф Г не містить простих циклів і має (n-1) ребро;
III. граф Г зв'язний і має (n-1) ребро;
IV. граф Т зв'язний, але вилучення довільного ребра робить його незв'язним;
V. довільні дві вершини графа Г з'єднані точно одним простим шляхом;
VI. граф Т не містить простих циклів, але, додавши до нього довільне нове ребро (без додавання вершин), ми отримаємо точно один простий цикл.

Наслідок. Ліс із к дерев, який містить n вершин, має (*n-к*) ребер.

Розглянемо кореневе дерево. У багатьох застосуваннях певну вершину дерева означають як корінь. Тоді можна природно приписати напрямок кожному ребру. Оскільки існує єдиний простий шлях від кореня до кожної вершини графа, то можна орієнтувати кожне ребро в напрямку від кореня. Отже, дерево разом із виділеним коренем утворює орієнтований граф, який називають кореневим деревом.

Зазначимо, що різні способи вибору кореня утворюють різні кореневі дерева.

Приклад 15.2. На рис. 15.2, а зображено дерево, а на рис. 15.2, б, в – кореневі дерева з коренями відповідно у вершинах а та с.

Нехай Т – кореневе дерево. Якщо v – його вершина, відмінна від кореня, то батько v – це єдина вершина и така, що є орієнтоване ребро (u, v). Якщо u – батько, то v – син. Аналогічно за генеалогічною термінологією можна означити інших предків і нащадків вершини v. Вершини дерева, які не мають синів, називають листками. Вершини, які мають синів, називають внутрішніми. Зазначимо, що корінь належить до внутрішніх вершин.

Якщо a – вершина дерева, то піддерево з коренем a – це підграф, що містить a та всі вершини – нащадки вершини a, а також інцидентні їм ребра.

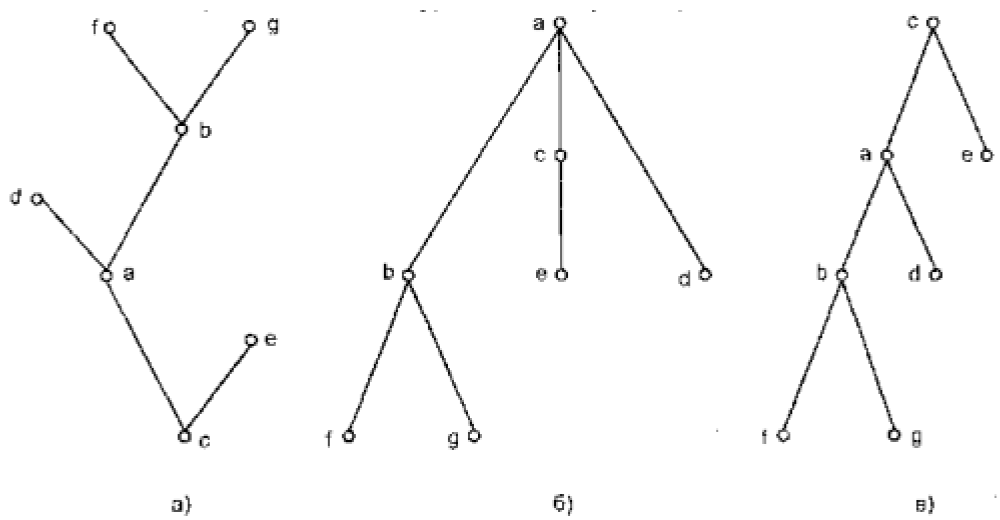


Рисунок 15.2 – Приклади дерев

Кореневе дерево називають m-арним, якщо кожна його внутрішня вершина має не більше ніж m синів. Дерево називають повніш m-арним, якщо кожна його внутрішня вершина має точно m синів. У разі m=2 дерево називають бінарним.

Упорядковане кореневе дерево – це кореневе дерево, у якому сини кожної внутрішньої вершини впорядковано. На рисунку таке дерево зображають так, щоб сини кожної вершини були розміщені зліва направо.

Якщо внутрішня вершина впорядкованого бінарного дерева має двох синів, то першого називають лівим сином, а другого – правим. Піддерево з коренем у вершині, яка являє собою лівого сина вершини v, називають лівим піддеревом у вершині v. Якщо корінь піддерева – правий син вершини v, то таке піддерево називають правим піддеревом у вершині v.

Приклад 15.3. У дереві, зображеному на рис. 15.3. Л і П – відповідно ліве та праве піддерева у вершині с.

Теорема 2. Повне m-арне дерево з i внутрішніми вершинами містить $n = mi + 1$ вершин.

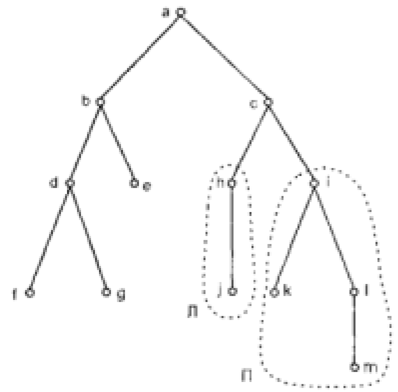


Рисунок 15.3 – Приклади піддерев

Рівнем вершини v в кореневому дереві називають довжину простого шляху від кореня до цієї вершини (цей шлях, очевидно, єдиний). Рівень кореня вважають нульовим. Висотою кореневого дерева називають максимальний із рівнів його вершини. Інакше кажучи, висота кореневого дерева – це довжина найдовшого простого шляху від кореня до будь-якої вершини.

Повне m-арне дерево, у якого всі листки на одному рівні, називають завершеним m-арним дертвом.

Кореневе m-арне дерево з висотою h називають збалансованим , якщо всі його листки на рівнях h або h–1.

Теорема 3. Нехай m-арне дерево має висоту h. Тоді в ньому не більше ніж m^h листків.

15.2 Рекурсія. Обхід дерев. Префіксна та постфіксна форми запису виразів

Об’єкт називають рекурсивним, якщо він містить сам себе, або означений за допомогою самого себе. Рекурсія – потужний засіб у математичних означеннях.

Приклад 4. Означимо повне бінарне дерево рекурсивно:

а) о (ізольована вершина) – повне бінарне дерево;

б) якщо A та B – повні бінарні дерева, то конструкція, зображена на рис 4 – повне бінарне дерево.

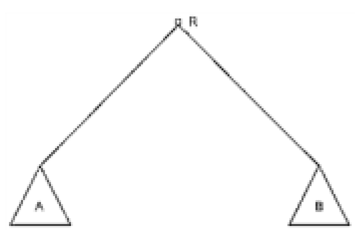


Рисунок 15.4 – Бінарне дерево

Приклад. Рекурсивне означення функції n\ для невід'ємних цілих чисел має такий вигляд:

а) 0! = 1;

б) якщо n>0, то $n! = n(n-1)!$

Очевидно, що потужність рекурсії пов'язана з тим, що вона дозволяє означити нескінченну множину об'єктів за допомогою скінченного висловлювання. Так само нескінченні обчислення можна описати за допомогою скінченної рекурсивної програми, навіть якщо ця програма не містить явних циклів. Проте, краще за все використовувати рекурсивні алгоритми в тих випадках, коли задача, яку розв'язують, або функція, яку обчислюють, або дані, які обробляють, задано за допомогою рекурсії.

Чимало задач можна моделювати з використанням кореневих дерев. Поширене таке загальне формулювання задачі: виконати задану операцію обробити з кожною вершиною дерева. Тут обробити – параметр загальнішої задачі відвідування всіх вершин, або так званого обходу дерева. Розглядаючи розв'язування цієї задачі як єдиний послідовний процес відвідування вершини дерева в певному порядку, можна вважати їх розміщеними одна за одною. Опис багатьох алгоритмів істотно спрощується, якщо можна говорити про наступну вершину дерева, маючи на увазі якесь упорядкування. Є три принципи впорядкування вершин, які природно випливають зі структури дерева. Як і саму деревоподібну структуру, їх зручно формулювати за допомогою рекурсії.

Звертаючись до бінарного дерева, де *R* – корінь, *A* та *B* – ліве та праве піддерева, можна означити такі впорядкування.

1. Обхід у прямому порядку або зверху вниз: *R, A, B* (корінь відвідують до обходу піддерев);

2. Обхід у внутрішньому порядку або зліва направо*: A, R, B:*

3. Обхід у зворотному порядку або знизу вверх *A, B, R* (тобто корінь відвідують після обходу піддерев).

Нижче наведено рекурсивні алгоритми обходу бінарних дерев. У кожному способі обходу використано команду обробити *(v),* де *v*-вершина. Цей термін залишено невизначеним, бо його значення залежить від того, що ми хочемо зробити під час проходження вершин. Для наших цілей достатньо використати звичайну команду друку символу.

Алгоритм обходу в прямому порядку позначено як ОПП *(x).* у внутрішньому – як ОВП (*x*) і в зворотному - як ОЗП (*x*). У всіх трьох алгоритмах x – це корінь дерева, яке обходять. Лівого сина вершини v позначено як лс (v), а правого - як пс (v).

Алгоритм обходу дерева в прямому порядку – ОПП (корінь).

Обробити(корінь)

Якщо лс (корінь) існує, то ОПП (лс (корінь))

Якщо пс (корінь) існує, то ОПП(пс(корінь))

Алгоритм обходу дерева у внутрішньому порядку - ОВП (корінь)

Якщо пс (корінь) існує, то ОВП {пс (корінь))

Обробити(корінь)

Якщо пс(корінь) існує, то ОВП (пс(корінь)

Алгоритм обходу дерева у зворотному порядку - ОЗП (корінь)

Якщо лс (корінь) існує, то ОЗП (лс (корінь))

Якщо пс (корінь) існує, то ОЗП (пс (корінь))

Обробити (корінь)

Приклад. На рис. 15.5 зображено бінарне дерево. Різні обходи дадуть такі послідовності вершин:

обхід у прямому порядку: abdehocfmpq;

обхід у внутрішньому порядку: abheoafcpmq;

обхід у зворотному порядку: dhoebfpqmca.

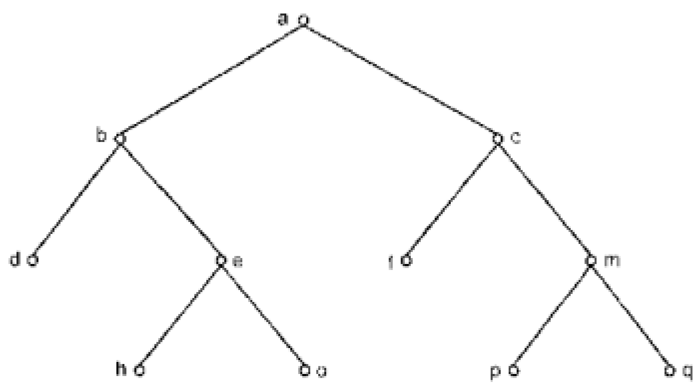


Рисунок 15.5 – Бінарне дерево

Зазначені способи обходу бінарних дерев можна узагальнити й на довільні m-арні дерева.

Надзвичайно поширене застосування в інформатиці обходу дерев-зіставлення виразам (арифметичним, логічним тощо) дерев і побудова на цій основі різних форм запису виразів. Суть справи зручно пояснити на прикладі. Розглянемо арифметичний вираз (зірочкою позначено операцію множення)

$$\left(a+\frac{b}{c}\right)*(d-e*f)$$

Подамо його у вигляді дерева. Послідовність дій відтворено на рис. 15.6. Рамкою на ньому обведено дерево, яке відповідає заданому арифметичному виразу. Внутрішнім вершинам цього дерева відповідають символи операцій, а листкам – операнди.

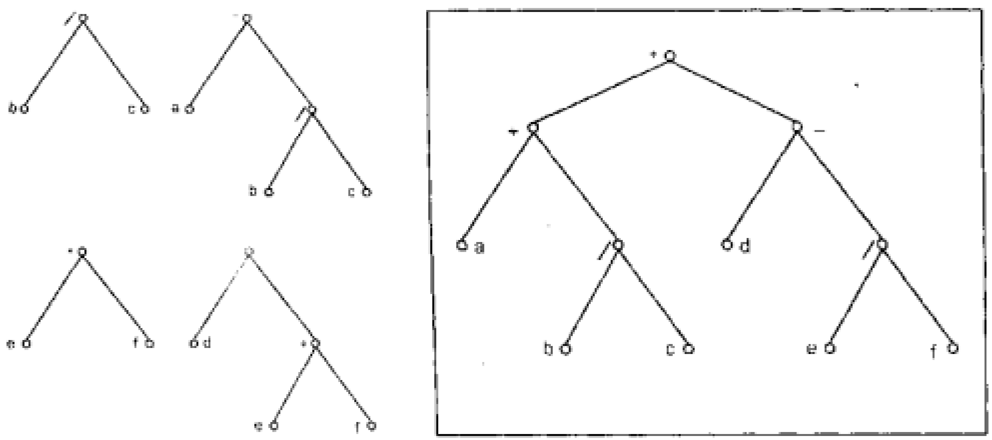


Рисунок 15.6 – Бінарне дерево

Обійдемо це дерево, записуючи символи у вершинах у тому порядку, у якому вони зустрічаються в разі заданого способу обходу. Тоді отримаємо такі три послідовності:

– у разі обходу в прямому порядку – префіксний або польський запис:

$$*+a/bc-d*ef\ ;$$

– у разі обходу у внутрішньому порядку – інфіксний запис (поки що без дужок, потрібних для визначення порядку операцій):

$$a+b/c*d-e*f\ ;$$

– у разі обходу в зворотному порядку - постфіксний або зворотний польський запис:

$$abc/+def*-*\ .$$

Для обчислення значення виразу в польському записі його проглядають справа наліво та знаходять два операнди разом зі знаком операції перед ними. Ці операнди та знак операції вилучають із запису, виконують операцію, а її результат записують на місце вилучених символів. Для обчислення значення виразу в зворотному польському записі його проглядають зліва направо та виділяють два операнди разом зі знаком операції після них. Ці операнди та знак операції вилучають із запису, виконують операцію, а її результат записують на місце вилучених символів.

Література: [1, 6, 10].

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Борисенко О. А. Дискретна математика: підручник для студентів вищих навчальних закладів. Суми: Університетська книга, 2023. 255 с.
2. Темнікова О. Л., Тавров Д. Ю. Дискретна математика. Частина 1. Основи дискретної математики. Практикум. Київ: КПІ ім. Ігоря Сікорського, 2023. 121 с.
3. Темнікова О. Л., Тавров Д. Ю. Дискретна математика. Частина 2. Булева алгебра. Практикум. Київ: КПІ ім. Ігоря Сікорського, 2024. 185 с.
4. Ємець О. О., Парфьонова Т. О. Дискретна математика: навчальний посібник для самостійного вивчення навчальної дисципліни студентами денної форми навчання спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра. Вид. 3-тє, допов. і перероб. Полтава: ПУЕТ, 2023. 282 с.
5. Кучинська Н В., Матійко А. А., Шолохов С. М. Дискретна математика: навч. посіб. для здобувачів освітнього ступеня бакалавра за ОПП «Безпека державних інформаційних ресурсів» спец. 125 Кібербезпека та захист інформації. Київ: КПІ ім. Ігоря Сікорського, 2025. 280 с.
6. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика: підручник; за наук. ред. д-ра техн. наук, проф. В. В. Пасічника. 7-ме видання, випр. та допов. Львів: ПП «Магнолія 2006»; ЛНУ ім. Івана Франка, 2024. 432 с.
7. Слесарев В. В., Новицький І. В., Ус С. А. Дискретна математика: навч. посіб. М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». Дніпро: НТУ «ДП», 2023. 183 с.
8. Матвієнко М. П., Шаповалов С.П. Математична логіка та теорія алгоритмів. Навчальний посібник. К.: Видавництво Ліра-К, 2021. 212 с.
9. Олійник А. С., Петравчук А. П. Дискретна математика. Навчальний посібник для студентів механіко-математичного факультету. К., 2024. 177 с.
10. Харченко В. М. Практикум з дискретної математики. Ніжин: НДУ ім. М. Гоголя, 2022. 148 с.
11. Желдак Т. А., Коряшкіна Л. С., Ус С. А. Елементи теорії нечітких множин. Дніпро: НТУ «ДП», 2022. 46 с.
12. Черницька О. В. Комп'ютерна дискретна математика. Конспект лекцій Дніпро Ліра, 2022. 116 с.
13. Discrete Mathematics with Applications, Metric Edition 5th edition Publisher: Cengage Learning Custom Publishing. URL: https://dokumen.pub/discrete-mathematics-with-applications-metric-edition-5nbsped-0357114086-9780357114087.html (дата звернення: 20.05.2025).
14. Лавренчук С. В., Здолбіцька Н. В., Хамула Н. М. Програмний комплекс для візуалізації алгоритмів на графах. Вісник Хмельницького національного університету. серія: Технічні науки (6), 2021. 81-85 с.
15. Лавренчук С. В., Здолбіцька Н. В., Хамула Н. М. Реалізація алгоритмів на графах засобами DHTML. Інформаційні технології і автоматизація – 2021. Матеріали XIV Міжнародної науково-практичної конференції. Одеса, 21-22 жовтня 2021 р. Одеса: Видавництво ОНАХТ, 2021. С. 213-215.
16. Семенюк П. А., Гунько Н. В., Здолбіцька Н. В. Відтворення шифрувальної машини Енігми засобами DHTML. Інформаційна безпека та

комп'ютерні технології: Матеріали V Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології»: тези доповідей, 19–20 травня 2022 р. Кропивницький: ЦНТУ, 2022. 66 с.

17. Здолбіцька Н. В., Сулім В. О., Вознюк А. В. Мультиагентна система маршрутизації на основі алгоритмів пошуку найкоротшого шляху в графі. The 5th International scientific and practical conference "Science and innovation of modern world" (January 25-27, 2023) Cognum Publishing House, London, United Kingdom. 2023. С. 155-157

18. Здолбіцька Н., Здолбіцький А., Давидюк Д. Машинне навчання для прогнозування попиту на товари онлайн-замовлень. Проблеми комп'ютерних наук, програмного моделювання та безпеки цифрових систем: матеріали II міжнар. науково-практ. конф. Луцьк, 9-11 червня 2025 р. Луцьк, 2025. С. 8-9.

19. Zdolbitska N., Bas D., Zdolbitsky S. Training neural networks to sort: a new approach to classical algorithms. Trends, Issues, and Challenges in Modern Science: Proceedings of the 2nd International Scientific Conference (Cambridge, United Kingdom, 5 September 2025). Lulu Press, Inc., 2025. P. 93-95.

20. Zdolbitska N., Ostapchuk O., Lavrenchuk S., Terletskyi T., Kaidyk O., O. Zhyharevych, Business information system for forecasting raw material stocks for the production of flexible packaging. 2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 2024. P. 1-8.

Д26 Дискретна математика: Конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти вищої освіти освітньої програми «Інформаційні системи та технології охорони і безпеки» галузі знань F Інформаційні технології спеціальності F6 Інформаційні системи та технології денної та заочної форм навчання / уклад. Н. В. Здолбіцька, Луцьк: ЛНТУ, 2025. 116 с.

Комп'ютерний набір і верстка: Здолбіцька Н. В.

Редактор Здолбіцька Н. В.

Підпис до друку ______20__ р. Формат 60х84/16. Папір офс.
Гарн. Таймс. Ум. друк. арк. . Обл.-вид. арк. .
Тираж 10 прим. Зам. №______

Відділ іміджу та промоцій
Луцького національного технічного університету
43018, м. Луцьк, вул. Львівська, 75