КОМУНАЛЬНИЙ ЗАКЛАД ВИЩОЇ ОСВІТИ
«ЛУЦЬКИЙ ПЕДАГОГІЧНИЙ ІНСТИТУТ»
ВОЛИНСЬКОЇ ОБЛАСНОЇ РАДИ

Луцький педагогічний фаховий коледж
ЦК природничо-математичних дисциплін

ДИСКРЕТНА МАТЕМАТИКА

курс лекцій

Навчальний посібник
для студентів фахових коледжів
спеціальності «Комп'ютерні науки»

Луцьк – 2026

УДК 519.854(075.8)
Д 49

Рекомендовано до друку методичною радою
Луцького педагогічного фахового коледжу
Комунального закладу вищої освіти «Луцький педагогічний інститут»
Волинської обласної ради
(протокол №11 від 24.06.2026)

Рецензенти:

Собчук О. М. – кандидатка педагогічних наук, доцентка кафедри загальної математики та методики навчання інформатики Волинського національного університету імені Лесі Українки
Констанкевич Л. Г. – викладачка-методистка циклової комісії природничо-математичних дисциплін Комунального закладу вищої освіти «Луцький педагогічний інститут» Волинської обласної ради.

Дискретна математика: курс лекцій. Навчальний посібник для студентів фахових коледжів спеціальності «Комп'ютерні науки» / упоряд. Т. Г. Четверикова, Н. Ю. Нарихнюк, В. О. Корнелюк. Луцьк: Комунальний заклад вищої освіти «Луцький педагогічний інститут» Волинської обласної ради, 2026. 127 с.

У навчальному посібнику подано курс лекцій з дискретної математики, адаптований для студентів фахових коледжів, які навчаються за спеціальністю «Комп'ютерні науки». Матеріал охоплює два змістові модулі: теорію множин і математичну логіку, а також комбінаторику, теорію графів, автомати та формальні мови. Посібник містить теоретичний матеріал, розбір практичних прикладів, алгоритми, завдання для самостійного опрацювання та запитання для самоконтролю. Посібник буде корисний студентам, викладачам фахових коледжів та всім, хто вивчає основи комп'ютерних наук.

ЗМІСТ

ПЕРЕДМОВА..... 4
Змістовий модуль І. ТЕОРІЯ МНОЖИН І МАТЕМАТИЧНА ЛОГІКА.....5
Тема 1. Основні поняття теорії множин, способи визначення множини.....5
Тема 2. Операції над множинами.....10
Тема 3. Алгебра множин.....15
Тема 4. Декартовий добуток множин. Бінарні відношення. Способи подання бінарних відношень.....17
Тема 5. Операції над бінарними відношеннями. Властивості бінарних відношень.....27
Тема 6. Функціональні відношення.....37
Тема 7. Логіка висловлювань. Алгебра логіки. Основні поняття. Формули логіки висловлювань.....43
Тема 8. Закони логіки висловлень. Рівносильні перетворення формул. Спеціальні форми подання булевих функцій.....49
Тема 9. Числення предикатів.....54
Змістовий модуль ІІ. КОМБІНАТОРИКА. ТЕОРІЯ ГРАФІВ, АВТОМАТИ ТА ФОРМАЛЬНІ МОВИ..... 60
Тема 10. Основні комбінаторні конфігурації та правила підрахунку..... 60
Тема 11. Спеціальні методи комбінаторики та біноміальна теорема.....64
Тема 12. Основні поняття теорії графів.....68
Тема 13. Способи визначення та типи графів.....78
Тема 14. Операції над графами. Зв’язність графа. Маршрути, шляхи, ланцюги, цикли.....84
Тема 15. Алгоритми з графами.....90
Тема 16. Формальні мови.....100
Тема 17. Формальні граматики..... 111
Тема 18. Автомати та способи їх задання..... 120
СПИСОК ЛІТЕРАТУРИ.....127

ПЕРЕДМОВА

Сучасний етап розвитку інформаційних технологій та комп’ютерних наук вимагає від майбутніх фахівців глибоких фундаментальних знань, які є базою для опанування програмування, проєктування складних систем та алгоритмізації. Основою такої підготовки є дискретна математика, яка забезпечує теоретичний фундамент для побудови систем керування базами даних, організації комп’ютерних мереж та розробки програмного забезпечення.

Навчальний посібник «Дискретна математика: курс лекцій» створений відповідно до освітньо-професійних програм підготовки студентів фахових коледжів спеціальності «Комп’ютерні науки». Головна мета видання – сформувати у майбутніх ІТ-спеціалістів системне математичне мислення, ознайомити з базовим понятійним апаратом дискретного аналізу та навчити застосовувати теоретичні знання для розв'язання практичних прикладних задач.

Матеріал навчального посібника структуровано за модульним принципом і поділено на два змістові блоки, які охоплюють ключові розділи дисципліни:

Змістовий модуль І. Теорія множин і математична логіка. У цьому розділі розглядаються поняття теорії множин, операції над ними, алгебра множин, декартові добутки, а також бінарні та функціональні відношення, що є основою для розуміння структур даних. Значна увага приділена логіці висловлювань, алгебрі логіки, рівносильним перетворенням формул та численню предикатів, що безпосередньо застосовується в логічному програмуванні та побудові комп’ютерних схем.

Змістовий модуль ІІ. Комбінаторика. Теорія графів, автомати та формальні мови. У цьому розділі розглядається теорія графів, типи графів, операції над ними та базові графічні алгоритми, які широко використовуються в оптимізації маршрутизації та мережевих технологіях. Завершується курс вивченням теорії формальних мов і граматик, а також автоматів та способів їх задання, що є базою для проектування компіляторів та теорії обчислень.

Методична особливість цього курсу лекцій полягає у його чіткій орієнтації на самостійну та практичну роботу студента. Кожна тема містить визначену мету вивчення, структурований план, детальний лекційний матеріал із великою кількістю прикладів розв'язання типових задач, завдання для самостійного виконання та питання для самоконтролю. Окрім того, до кожної теми запропоновано список актуальної навчальної літератури та електронних ресурсів, що дозволяє студентам поглибити свої знання.

Упорядники сподіваються, що цей посібник стане надійним помічником для студентів у процесі освоєння дискретного аналізу, а викладачам допоможе в організації ефективного освітнього процесу.

Змістовий модуль І. ТЕОРІЯ МНОЖИН І МАТЕМАТИЧНА ЛОГІКА

Тема 1. Основні поняття теорії множин, способи визначення множини

Мета вивчення: сформувати поняття про дискретну математику; поняття множини та її елементів, способи задания множин, потужність множин, поняття множини та підмножини.

План

1. Дискретна математика.
2. Основні поняття теорії множин.
3. Способи задання множин.
4. Потужність множин.
5. Основні числові множини та підмножини.

Література:

1. Коцовський В. М. Дискретна математика та теорія алгоритмів. Ужгород : Видавництво УжНУ «Говерла», 2015. 95 с. URL: https://dspace.uzhnu.edu.ua/jspui/handle/lib/16302 (дата звернення: 01.05.2026).

2. Методи прикладної математики в транспортних технологіях. Елементи теорії множин : метод. рек. для студентів спеціальності «Транспортні технології» / уклад. І. І. Філімоніхіна. Кропивницький : ЦНТУ, 2022. 39 с. URL: https://dspace.kntu.kr.ua/server/api/core/bitstreams/e7e7c935-ff0b-492d-a7cc-3cd429cba0cc/content (дата звернення: 01.05.2026).

3. Дискретна математика: Конспект лекцій (Частина 1) : навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані та математичне моделювання» / уклад. О. Л. Темнікова ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2021. 154 с. URL: https://ela.kpi.ua/server/api/core/bitstreams/990893b6-f853-408a-8476-d3dd7c89d2a1/content (дата звернення: 01.05.2026).

Текст лекції

1. Дискретна математика

Дискретна математика (дискретний аналіз, скінченна математика) – це розділ сучасної математики, в якому вивчають властивості математичних об'єктів дискретного характеру.

Традиційно до курсу дискретної математики відносять такі розділи: теорія множин, комбінаторика, алгебраїчні структури, математична логіка, логіка висловлювань і предикатів, теорія графів, теорія алгоритмів і автоматів.

2. Основні поняття теорії множин

У повсякденному житті і практичній діяльності часто потрібно говорити про деякі сукупності різних об’єктів, предметів, понять, чисел, символів тощо.

Наприклад, *сукупність деталей механізму, аксіом геометрії, чисел натурального ряду, літер алфавіту.*

На основі інтуїтивних уявлень про такі сукупності сформувалося математичне поняття множини.

Великий внесок в теорію множин зробив Георг Кантор: «Множина – це багато, що мислиться як єдине ціле». Пізніше, завдяки його дослідженням, теорія множин стала повністю визначеним та обгрунтованим розділом математики, а у теперішній час набула фундаментального значення. Теорія множин є основою для усіх розділів дискретної математики та комп'ютерних наук в цілому, є однією з основ функціонального аналізу, топології, загальної алгебри та у теперішній час проводяться глибокі дослідження у самій теорії множин, що пов'язані з основами математики. Теорія множин разом із іншими розділами дискретної математики має багато корисних застосувань у програмуванні. Наприклад, вона використовується для побудови систем керування базами даних, при розбудові та організації роботи комп'ютерних мереж, мережі Інтернет.

Інтуїтивне поняття множини: ***множина – сукупність деяких різних об'єктів, яким властиве дещо спільне і які можна розглядати як єдине ціле.***

Об'єкти, з яких складається множина, називаються ***елементами множини. Множини***, як правило, ***позначають великими латинськими літерами A, B, X, Y***,…, а ***їх елементи – малими латинськими літерами a, b, x, y*** ….

Множина, що не містить жодного елемента, називається ***порожньою***, і позначається символом ∅.

Приклади порожніх множин: множина квадратних тричленів, що мають більше двох коренів; множина живих людей, вік яких більше 1000 років; множина слонів, що літають.

Задати множину – вказати правило, яке дозволяє з'ясувати, чи є певний об`єкт елементом даної множини.

3. Способи задання множини

1. Задання множини переліком її елементів

Цей спосіб на практиці використовується для задання скінченних множин, які містять відносно невелику кількість елементів. Він є непридатним для задання нескінченних множин, а також для задання скінченних множин, елементи яких важко перелічити. Це безпосередній перелік усіх її елементів всередині фігурних дужок через кому, наприклад,

$A = \{a_1, a_2, \cdots, a_n\}$, $B = \{1, 2, 3, 4, 5\}$, $N = \{1, 2, 3, \cdots\}$

При цьому порядок розташування елементів у множині ***неістотний.*** Наприклад, множина всіх десяткових цифр – {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, множина

всіх основних математичних операцій – $\{+,\ -,\ \div,\ \cdot\}$, множина розв'язків нерівності $(x-1)^2 \le 0$ – $\{1\}$.

2. ***Задання множини вказівкою властивостей її елементів (предикативний).*** Тут вказується ***характерна властивість***, якою володіють усі елементи множини. При цьому множину X з елементів x, що задовольняють деяку умову $p(x)$, позначають так: $X=\{x \mid p(x)\}$.

Наприклад,

$A=\{x \mid x$ – *просте число*$\}$ ($A = \{2, 3, 5, 7, 11, 13, 17,...\}$)

$B = \{x \mid x \in R, x^2 + 2x - 15 = 0\}$ ($B = \{-5,3\}$)

Завдання: Знайти елементи множини $A = \{x \mid x \in R, x^2 - 7x + 10 = 0\}$

3. ***Задання множини за допомогою процедури породження елементів.***

Приклад. Множина чисел Фібоначчі визначається наступним чином:

$f_1= 1, f_2= 1, f_{n+1} =f_n+f_{n-1}, n = 1, 2,...$

Вкажемо перші десять чисел Фібоначчі:

$f_1= 1, f_2= 1, f_3= 2, f_4= 3, f_5= 5, f_6= 8, f_7= 13, f_8= 21, f_9= 34, f_{10}= 55.$

Із заданням множин пов'язаний цілий ряд парадоксів. Розглянемо *парадокс перукаря*. Єдиний перукар у місті N визначає множину A мешканців, яких він повинен голити, як сукупність всіх тих мешканців N, які не голяться самі. Але тоді для самого перукаря виходить протиріччя і при включенні його до множини A, і при віднесенні його до мешканців, які голяться самі.

4. Потужність множин

Якщо елемент a входить до складу множини A (елемент a міститься у множині A), то символічно це записують $a \in A$, інакше $a \notin A$. Нехай, наприклад, $A = \{1, 2, 3\}$, тоді $3 \in A$, $7 \notin A$.

Розглядається також множина, яка не містить жодного елемента. Ця множина називається *порожньою* і позначається $\emptyset$.

Множина A називається ***підмножиною*** множини B (позначається $A \subset B$), якщо кожен елемент множини A належить також і множині B ($A \subset B$). Нехай, наприклад, $A = \{1, 3, 5\}$, $B = \{1, 2, 3, 4, 5\}$, тоді $A \subset B$.

Також запис $A \subset B$ може називатися *«включення»* і читатися «A міститься (входить) у B», «B містить A».

Приклад. Включення $A \subset B$ правильне для множин

$A = \{2, 4\}$, $B = \{1, 2, 3, 4\}$.

Приклад.

1) множина A складається з усіх чотирикутників;
2) множина B складається з усіх паралелограмів;
3) множина C складається з усіх прямокутників;
4) множина D складається з усіх квадратів.

Кожна наступна множина є підмножиною попередньої: A⊃ B⊃ C⊃ D.

Приклад. $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$

З визначення підмножини бачимо, що ***будь-яка множина є підмножиною самої себе***: A⊂ A. Порожня множина Ø є підмножиною будь-якої множини: Ø ⊂ A.

Виключивши ці граничні випадки (тобто Ø, A), ми отримаємо так звані ***власні підмножини*** множини A, тобто такі, що не є порожніми та не збігаються з A.

Для включення множин справедлива ***властивість транзитивності:***

якщо $A \subset B$ і $B \subset C$, то $A \subset C$.

Якщо всі множини, що розглядаються в процесі вивчення певного питання, є підмножинами деякої іншої множини *U*, то вона називається ***універсальною множиною*** у даному процесі.

Наприклад, при вивченні числових множин роль універсальної множини може виконувати поле комплексних чисел $\mathbb{C}$ (або деякі його підмножини). У теорії чисел універсальна множина збігається або із множиною всіх цілих чисел $\mathbb{Z}$, або з множиною натуральних чисел $\mathbb{N}$; в математичному аналізі універсальна множина – це множина дійсних чисел $\mathbb{R}$.

Універсальну множину домовимось надалі завжди позначають символом *U*. Для будь-якої множини *X: X* ⊂ *U*.

Множини A і B називаються ***рівними*** (позначається $\boldsymbol{A = B}$), якщо вони складаються з однакових елементів. Якщо множини A і B не рівні, то це позначають $\boldsymbol{A \neq B}$.

Нехай, наприклад, A = {1, 3, 5}, B = {3, 1, 5}, тоді A = B.

Очевидно, що A = B тоді і тільки тоді, коли A⊂ B і B⊂ A.

Властивості відношення рівності множин:

1) A = A;
2) якщо A = B, то B = A;
3) якщо A = B і B = C, то A = C .

Наприклад, множина {a} – одноелементна множина, елементом якої є елемент a.

Множини {1, 2, 3, 4} та {3, 2, 4, 1}є рівними, оскільки вони складаються із тих самих елементів. Множини {1, 2, 3, 4}та {{1, 2},{3, 4}} не є рівними, оскільки перша складається з чотирьох елементів, а друга – з двох.

Множина називається ***скінченною***, якщо кількість її елементів виражається деяким числом. ***Зліченна*** множина – це множина, всі елементи якої можна пронумерувати: $\{a_1, a_2, a_3, ...\}$.

Потужністю множини називається кількість елементів скінченної множини і позначаться $|A|$.

Приклад.

1)$|\emptyset| = 0$; 2) $|\{a, b, c, d\}| = 4$; 3) N – нескінченна зліченна множина.

Множина всіх підмножин даної множини називається булеаном і позначається B(A). Наприклад, якщо множина

A = {1, 2, 3}, то $B(A)$ = {{Ø},{1},{2},{3},{1, 2},{1, 3},{2, 3},{1, 2, 3}}.

Потужність булеану визначається за формулою: $|B(A)| = 2^{|A|}$.

5. Основні числові множини та підмножини

До ***основних числових множин,*** які розглядаються у математиці, відносять:

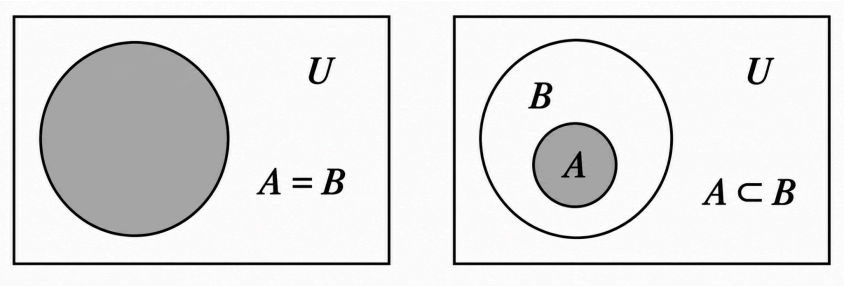
1) Множина натуральних чисел: $\mathbb{N}$ = {1, 2, 3,··· }.

2) Множина цілих чисел: $\mathbb{Z}$ = {··· , - 3, - 2, -1, 0, 1, 2, 3, ···}.

3) Множина раціональних чисел: $Q = \left\{\frac{a}{b} | \ a \in \mathbb{Z},\ b \in \mathbb{N}\right\}$

4) Множина дійсних чисел $\mathbb{R}$ (множина всіх чисел числової прямої).

Для наочного зображення множин використовують спеціальні графічні ***діаграми Ейлера-Венна****:* універсальну множину U зображають у вигляді прямокутника, а інші множини – у вигляді кругів всередині цього прямокутника. Наприклад, діаграми Ейлера-Венна для ілюстрації понять $A = B$ і $A \subset B$ мають вигляд:



Завдання для самостійного розв'язання

1. Поставте замість * знак $\in$ або $\notin$ так, щоб отримати правильне твердження:

1) $5 * N$; 3) $-5 * Q$; 5) $4,15 * Q$;

2) $0 * N$; 4) $-\frac{1}{2} * Z$; 6) $\pi * Q$.

2. Які з наведених тверджень правильні:

1) $1 \in \{1, 2, 3\}$; 3) $\{1\} \in \{1, 2\}$;

2) $1 \notin \{1\}$; 4) $\{1\} \in \{\{1\}\}$?

3. Запишіть множину коренів рівняння:

1) $x(x - 1) = 0$; 3) $x = 2$;

2) $(x - 2)(x^2 - 4) = 0$; 4) $x^2 + 3 = 0$.

4. Задайте за допомогою переліку елементів множину:

1) правильних дробів зі знаменником 7;

2) правильних дробів, знаменник яких не більший за 4;

3) букв слова «математика»;

4) цифр числа 5555.

5. Задайте за допомогою переліку елементів множину:

1) $A = \{x | \, x \in N, \, x^2 - 1 = 0\}$;

2) $B = \{x | \, x \in Z, \, |x| < 3\}$;

3) $C = \{x | \, x \in N, \, x \leq 15, \, x = 7k, \, k \in Z\}$;

4) $D = \{x | \, x \in Z, \, x(2|x| - 1) = 0\}$;

5) $K = \{x | \, x \in N, \ -3 \leq x < 2\}$.

6. Яке з тверджень є правильним:

1) $\{a\} \in \{a, b\}$; 3) $a \subset \{a, b\}$;

2) $\{a\} \subset \{a, b\}$ 4) $\{a, b\} \in \{a, b\}$?

7. Дано множини $\{7\}$, $\{11\}$, $\{19\}$, $\{7, 11\}, \{7, 19\}, \{11, 19\}, \varnothing$, які є всіма власними підмножинами деякої множини А. Запишіть множину А.

8. Запишіть усі підмножини множини $\{1, 2\}$.

Питання для самоконтролю

1. Що таке множина? Які існують способи її задання?
2. Що таке потужність множини? Як вона визначається для скінченних та нескінченних множин?
3. Назвіть основні числові множини та їхні підмножини.

Тема 2. Операції над множинами

Мета вивчення: ознайомити з основними операціями над множинами та їх властивостями, розглянути подання основних операцій над множинами з використанням діаграм Венна.

План

1. Операції над множинами та їх властивості.
2. Діаграми Венна.

Література:

1. Коцовський В. М. Дискретна математика та теорія алгоритмів. Ужгород : Видавництво УжНУ «Говерла», 2015. 95 с. URL: https://dspace.uzhnu.edu.ua/jspui/handle/lib/16302 (дата звернення: 01.05.2026).

2. Методи прикладної математики в транспортних технологіях. Елементи теорії множин : метод. рек. для студентів спеціальності «Транспортні технології» / уклад. І. І. Філімоніхіна. Кропивницький : ЦНТУ, 2022. 39 с. URL:

https://dspace.kntu.kr.ua/server/api/core/bitstreams/e7e7c935-ff0b-492d-a7cc-3cd429cba0cc/content (дата звернення: 01.05.2026).

3. Дискретна математика: Конспект лекцій (Частина 1) : навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані та математичне моделювання» / уклад. О. Л. Темнікова ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2021. 154 с. URL: https://ela.kpi.ua/server/api/core/bitstreams/990893b6-f853-408a-8476-d3dd7c89d2a1/content (дата звернення: 01.05.2026).

Текст лекції

1. Основні операції над множинами

До основних операцій над множинами відносяться доповнення, перетин, об'єднання, різниця, та симетрична різниця, для позначення яких використовуються відповідно символи $\bar{}$, $\cap$, $\cup$, \, Δ.

Об'єднання множин A і B (позначається через $\boldsymbol{A \cup B}$ і читається як «об'єднання A і B») є множина всіх предметів, які є елементами множин A або B, тобто

$$\boldsymbol{A \cup B = \{x \mid x \in A \text{ або } x \in B\}.}$$

Тут мається на увазі невиключний зміст слова «або». Таким чином, за визначенням, $x \in A \cup B$ тоді і тільки тоді, коли x є елемент хоча б однієї з множин A і B.

Наприклад: $\{1, 2, 3\} \cup \{1, 3, 4\} = \{1, 2, 3, 4\}$.

Перетин множин A і B (позначається через $\boldsymbol{A \cap B}$ і читається як «перетин A і B») є множина всіх предметів, які є елементами обох множин A і B, тобто

$$\boldsymbol{A \cap B = \{x \mid x \in A \; i \; x \in B\}.}$$

Таким чином, за визначенням, $x \in A \cap B$ тоді і тільки тоді, коли x є одночасно елементом множини A і елементом множини B.

Наприклад: $\{1, 2, 3\} \cap \{1, 3, 4\} = \{1, 3\}$.

Для будь-якої пари множин A і B мають місце такі включення:

$\varnothing \subseteq A \cap B \subseteq A \subseteq A \cup B$.

Дві множини A і B називаються ***непересічними*** (або диз'юнктивними), якщо $A \cap B = \varnothing$, і ***пересічними***, якщо $A \cap B \neq \varnothing$. Система множин називається ***розчленованою***, якщо будь-яка пара її різних елементів є непересічною.

Доповнення множин A (позначається через $\bar{}$ або A' або $\neg A$) − це множина всіх елементів, що не є елементами множини A: $\overline{A} = \{x \mid x \notin A\}$

Різниця множин *(відносне доповнення* множини A до множини B*)* − це множина $\boldsymbol{A \backslash B}$, що читається як «$A$ мінус B».

Таким чином $\boldsymbol{A \backslash B = \{x \in A \mid x \notin B\}}$, тобто це множина тих елементів множини A, які не є елементами множини B.

Наприклад: {1, 2, 3}\{1, 3, 4} = {2}, а {1, 3, 4}\{1, 2, 3} = {4}

Симетрична різниця множин *A* і *B*, що позначається через ***A÷B*** (іноді використовуються позначення ***AΔB*** або ***A+B***), визначається наступним чином: $x \in A \div B$ тоді і тільки тоді, коли *x* належить рівно одній з множин *A або B*:

$A \Delta B = (A \setminus B) \cup (B \setminus A)$ − симетрична різниця множин A та B

Якщо всі множини, що розглядаються в ході будь-якого міркування, є підмножинами деякої множини ***U***, то цю множину ***U*** називають ***універсальною*** або ***універсумом*** (для цього міркування). Наприклад, для елементарної арифметики універсальним служить ***Z***, а для аналітичної геометрії площини − множина всіх упорядкованих пар дійсних чисел.

Тоді абсолютне доповнення множини *A* можна розглядати як $A' = U \setminus A$.

Приклад. Нехай U = {5 , 6, ..., 30} − універсальна множина,

A ={6, 8, 9, 10, 12, 14, 20, 21, 22, 27},

B = {3x − 6 | x − непарне, x ∈N},

C = {x | x + 4 − просте число}.

Вказати перелік елементів множини $D = B \setminus (A \Delta \overline{C})$.

Розв'язання.

1. Задамо множини B та C переліком їх елементів:

B = {9, 15, 21, 27}, C = {7, 9, 13, 15, 19, 25, 27}.

2. Послідовно виконаємо операції над множинами:

а) $\overline{C}$ = {5, 6, 8, 10, 11, 12, 14, 16, 17, 18, 20, 21, 22, 23, 24, 26, 28, 29,30}.

б) Знайдемо симетричну різницю множини A та $\overline{C}$ за формулою:

$A \Delta \overline{C} = (A \setminus \overline{C}) \cup (\overline{C} \setminus A)$.

Оскільки $A \setminus \overline{C}$ = {9, 27}, $C \setminus \overline{A}$ = {5, 11, 16, 17, 18, 23, 24, 26, 28, 29,30}, то

$A \Delta \overline{C}$ = {5, 9, 11, 16, 17, 18, 23, 24, 26, 27, 28, 29,30}.

в) Знайдемо різницю множин B та $A \Delta \overline{C}$: $B \setminus (A \Delta \overline{C})$ = {15, 21}.

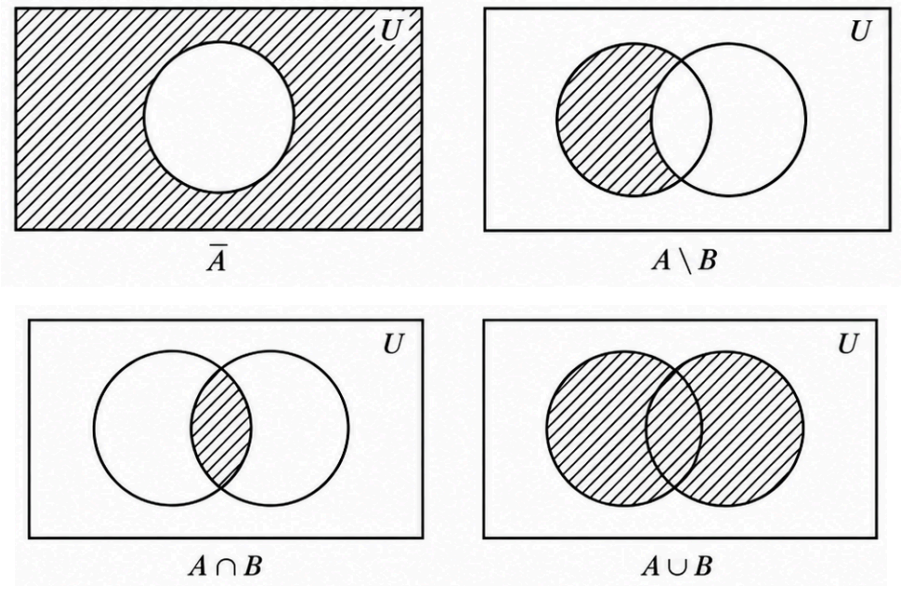
Відповідь: D = {15, 21}.

2. Діаграми Венна (круги Ейлера)

Для графічної ілюстрації відношень, які можуть мати місце між підмножинами будь-якої універсальної множини ***U***, часто використовують так звані діаграми Венна – за іменем англійського священика Джона Венна (1834 – 1923), що застосовував їх в своїх дослідженнях з логіки. Діаграма Венна є схематичним зображенням множин у вигляді точкових множин: універсальна множина ***U*** зображується множиною точок деякого прямокутника, а його підмножина *A* – у вигляді круга або який-небудь інший простій області всередині цього прямокутника. Правильніше, проте, було б назвати їх діаграмами Ейлера, оскільки задовго до Венна їх вживав знаменитий

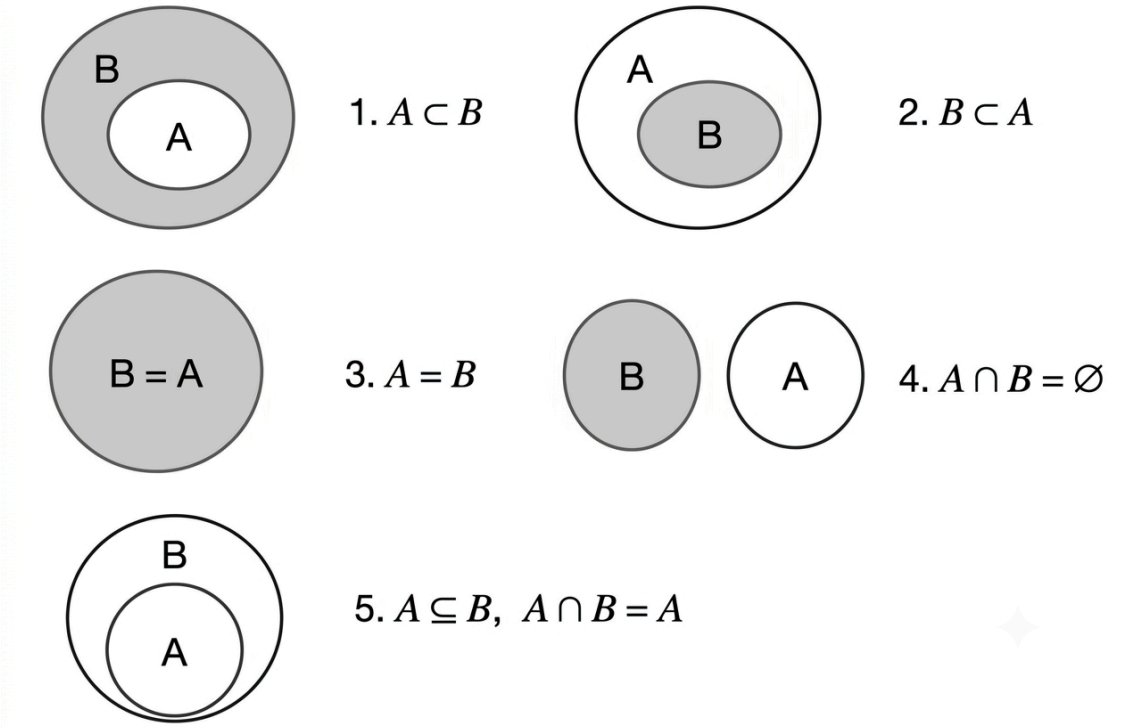
швейцарський математик Леонард Ейлер (1707 – 783), Нижче на рис. показані деякі операції над множинами.

Діаграми Венна та круги Ейлера:



Доведення тверджень або теорем за допомогою діаграм Венна не є легітимним доведенням з огляду на те, що малюнки не вичерпують всі можливі взаємозв'язки між множинами. Якщо ми розглянемо тільки дві множини A і B, то можна представити зв'язок між ними у таких варіантах (рис.):

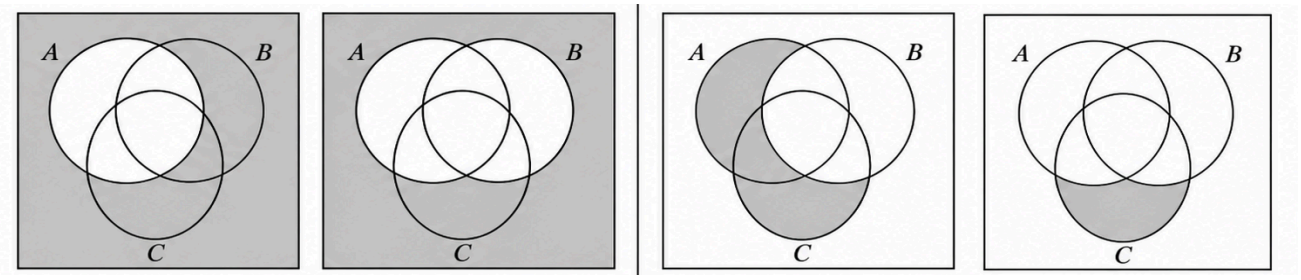
Взаємовідношення між множинами



Приклад. На діаграмі Венна зобразити множину $\overline{C} \setminus \overline{\overline{A} \cup B}$.

Розв'язання.

Зобразимо діаграми Венна для всіх операцій алгебри множин, які входять у формулу у порядку їх виконання.



$\overline{A}$ $\overline{A} \cup B$ $\overline{A \cup B}$ $C \backslash \overline{A \cup B}$

Завдання для самостійного розв'язання

1. Чи вірно, що:

1) $\{1,\ 4,\ 5\} = \{5, 1, 4\}$;

2) $\varnothing = \{\varnothing\}$; $\varnothing \subset \{\varnothing\}$; $\varnothing \subset \varnothing$; $\varnothing \subseteq \varnothing$;

3) $\{\{a, b\}, \{c, d, e\}\} = \{a, b, c, d, e\}$; $\{a, c, e\} \subset \{a, \{b, c\}, d, e\}$

2. Чи вірно, що:

1) $\{1,\ 3\} \in \{3, 2, 4, 5, 1\}$;

2) $a = \{a\}$; $a \subseteq \{a\}$; $\varnothing \subset \{a\}$; $(\{a\} \backslash \varnothing) \subset (\{a\} \cup \varnothing)$

3. Знайти $A \triangle (\overline{B} \cap C)$ за умови, що

$U = \{1, 2, \ldots, 20\}$, $A = \{x - 2 | 5x + 1 -$ квадрат цілого числа$\}$,

$B = \left\{x | x^2 > 60\right\}$, $C = \{x | (3x - 1) -$ просте число$\}$.

4. На діаграмі Ейлера-Венна зобразити множини:

а) $\overline{A \triangle \overline{C}} \backslash (B \cup A)$; б) $(\overline{A} \cap B) \cup (B \triangle C)$

5. $A = \{0, 3, 5, \{6, 9\}\}$, $B = \{2, 3, 9\}$, $C = \{0, 2, 3, 6\}$. Визначити наступні множини:

а) $B \backslash C$; б) $A \cap B$; в) $A \div C$; г) $(A \cup B) \backslash C$

6. Для кожної з наведених нижче множин використайте діаграми Ейлера-Венна і заштрихуйте ті її частини, які зображують задані множини:

а) $A \backslash (B \cap C)$; в) $(A \cup B \cup C) \backslash (A \cap B \cap C)$;

б) $(A \cap B) \div C$; г) $(A \cap B) \cup (A \cap C) \cup (B \cap C)$

Питання для самоконтролю

1. Які основні операції над множинами ви знаєте? Дайте математичне означення та поясніть суть операцій об'єднання, перетину, різниці та симетричної різниці множин.

2. Що таке діаграми Венна (круги Ейлера)? Яка їхня роль у комп'ютерній дискретній математиці та як за їх допомогою зображують універсальну множину і її підмножини?

3. Як за допомогою діаграм Венна графічно зобразити результат виконання таких операцій: перетин двох множин, що частково перетинаються;

різниця множин $A \setminus B$ та $B \setminus A$ (у чому різниця між цими областями?); абсолютне доповнення множини $\overline{A}$.

Тема 3. Алгебра множин

Мета вивчення: вивчення основних законів алгебри множин і формування вміння застосовувати закони алгебри множин для аналітичного доведення тотожностей і мінімізації складних виразів.

План

1. Основні закони алгебри множин.
2. Доведення рівності множин.

Література:

1. Коцовський В. М. Дискретна математика та теорія алгоритмів. Ужгород : Видавництво УжНУ «Говерла», 2015. 95 с. URL: https://dspace.uzhnu.edu.ua/jspui/handle/lib/16302 (дата звернення: 01.05.2026).

2. Методи прикладної математики в транспортних технологіях. Елементи теорії множин : метод. рек. для студентів спеціальності «Транспортні технології» / уклад. І. І. Філімоніхіна. Кропивницький : ЦНТУ, 2022. 39 с. URL: https://dspace.kntu.kr.ua/server/api/core/bitstreams/e7e7c935-ff0b-492d-a7cc-3cd429cba0cc/content (дата звернення: 01.05.2026).

3. Дискретна математика: Конспект лекцій (Частина 1) : навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані та математичне моделювання» / уклад. О. Л. Темнікова ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2021. 154 с. URL: https://ela.kpi.ua/server/api/core/bitstreams/990893b6-f853-408a-8476-d3dd7c89d2a1/content (дата звернення: 01.05.2026).

Текст лекції

Множина всіх підмножин універсальної множини називається **алгеброю множин.**

Пріоритет операцій: $\overline{\ }$, $\cap$, $\cup$, $\setminus$.

Основні закони алгебри множин:

1) Закони *комутативності:* $A \cap B = B \cap A$ $A \cup B = B \cup A$ $A \Delta B = B \Delta A$ 2) Закони *асоціативності*: $(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$	6) Властивості *порожньої множини:* $A \cap \varnothing = \varnothing$ $A \cup \varnothing = A$ $A \setminus \varnothing = A$ $A \Delta \varnothing = A$ 7) Властивості *доповнення*:

(A∆B)∆C=A∆(B∆C) 3) Закони *ідемпотентності*: A ∪ A=A, A∩A=A 4) Закони *дистрибутивності*: A∩(B ∪ C)=(A∩B) ∪ (A∩C) A∩(B\C)=(A∩B)\(A∩C) A∩(B∆C)=(A∩B)∆(A∩C) A ∪ (B∩C)=(A ∪ B)∩(A ∪ C) 5) Властивості *універсальної множини:* A∩U=A, A ∪ U=U A\U=∅, A∆U=$\overline{A}$	$\overline{\overline{A}} = A$; A ∪ $\overline{A}$ = U A∩$\overline{A}$ =∅; A ∆ $\overline{A}$ = U 8) Закони *де Моргана*: $\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$ 9) Властивості *різниці*: A\B=$A \cap \overline{B}$ A\A=∅ U\A=$\overline{A}$

Приклади розв'язування завдань

Задача 1: Спростити вираз: $(A \cup B) \cap \overline{B}$.

Розв'язання:

1) Застосуємо дистрибутивний закон: $(A \cup B) \cap \overline{B} = (A \cap \overline{B}) \cup (B \cap \overline{B})$

2) За властивостями доповнення (властивість 7): $B \cap \overline{B} = \varnothing$.

3) За властивістю порожньої множини (властивість 6) отримаємо:

$(A \cap \overline{B}) \cup \varnothing = A \cap \overline{B}$

Задача 2*:* Довести, що $A\backslash(B \cup C) = (A\backslash B) \cap (A\backslash C)$.

Розв'язання:

1) За властивістю різниці: $A\backslash(B \cup C) = A \cap \overline{B \cup C}$.

2) Застосуємо ***закон де Моргана*** до виразу під запереченням: $A \cap (\overline{B} \cap \overline{C})$.

3) Застосуємо ***асоціативний закон****:* $(A \cap A) \cap \overline{B} \cap \overline{C}$.

4) За законом ***асоціативності*** та ***ідемпотентності***: $(A \cap \overline{B}) \cap (A \cap \overline{C})$.

5) За визначенням різниці: $(A\backslash B) \cap (A\backslash C)$.

Задача 3*:* Спростити вираз: $X = (A \cup B) \cap (A \cup \overline{B})$.

Розв'язання:

1) Використовуємо закон дистрибутивності відносно об'єднання:

$X = (A \cup B) \cap (A \cup \overline{B}) = A \cup (B \cap \overline{B})$.

2) За законом виключеного третього (для перерізу): $B \cap \overline{B} = \varnothing$.

3) За властивістю порожньої множини: $A \cup \varnothing = A$.

Задача 4*:* Спростити вираз: $X = (A \cup B) \cap (A \cup \overline{B})$.

Розв'язання:

1) Використовуємо закон дистрибутивності відносно об'єднання:

$(A \cup B) \cap (A \cup \overline{B}) = A \cup (B \cap \overline{B})$

2) За законом виключеного третього (для перерізу)*:* $B \cap \overline{B} = \varnothing$

3) За властивістю порожньої множини: $A \cup \varnothing = A$.

Задача 5*:* Довести рівність $A\backslash(B\backslash C) = (A\backslash B) \cup (A \cap C)$.

1) Перетворимо ліву частину виразу, використовуючи властивість різниці:

$A\backslash(B\backslash C) = A\backslash(B \cap \overline{C}) = A \cap \overline{B \cap \overline{C}}$

2) Застосуємо закон де Моргана до виразу під інверсією: $A \cap \overline{(\overline{B} \cup \overline{\overline{C}})}$

3) За законом подвійного доповнення*:* $A \cap (\overline{B} \cup \overline{\overline{C}}) = A \cap (\overline{B} \cup C)$

4) Застосуємо дистрибутивний закон: $A \cap (\overline{B} \cup C) = (A \cap \overline{B}) \cup (A \cap C)$

5) Замінимо $A \cap \overline{B} = A\backslash B$. Отримаємо: $(A\backslash B) \cup (A \cap C)$.

Завдання для самостійного виконання

Спростити вирази:

1) $(A \cap B \cap C) \cup (\overline{A} \cap B \cap C) \cup \overline{B} \cup \overline{C}$

2) $(A \cap B \cap C \cap \overline{D}) \cup (\overline{A} \cap C) \cup (\overline{B} \cap C) \cup (C \cap D)$

3) $\overline{A \cap \overline{\overline{B}} \cup B}$

4) $\overline{\overline{A} \cup B \cup C} \cap (A \cap (B \cup \overline{C})) \cap \overline{B}$.

Питання для самоконтролю

1. У чому полягає відмінність між операціями різниці та симетричної різниці множин?
2. Сформулюйте закони де Моргана та поясніть їхнє значення для мінімізації логічних схем.
3. Яка множина є нейтральним елементом для операції об'єднання; для операції перерізу?
4. Основні закони алгебри множин, їх використання для розв’язування задач.
5. Спрощення формул з використанням основних законів алгебри множин.

Тема 4. Декартовий добуток множин. Бінарні відношення. Способи подання бінарних відношень

Мета вивчення: ознайомлення з поняттям декартового добутку множин; бінарними відношеннями: основними способами подання бінарних відношень; геометричним та матричним поданням бінарних відношень.

План

1. Декартовий добуток множин.
2. Властивості декартового добутку множин.
3. Поняття бінарного відношення.

4. Способи задання бінарних відношень.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

3. Коцовський В. М. Дискретна математика та теорія алгоритмів. Ужгород : Видавництво УжНУ «Говерла», 2015. 95 с. URL: https://dspace.uzhnu.edu.ua/jspui/handle/lib/16302 (дата звернення: 01.05.2026).

Текст лекції

1. Декартовий добуток множин

Декартовим добутком множин A та B називається множина $A \times B$ усіх упорядкованих пар, перша координата (компонента) яких належить множині A, а друга – множині B. Тобто,

$A \times B = \{(a, b) | a \in A,\ b \in B\}$. При цьому *порядок множників є суттєвим.*

Впорядкована пара елементів (x, y) однозначно визначається через x і y. Прийнято називати x першою, а y – другою координатою впорядкованої пари (x, y).

Приклад. Знайти $A \times B$, якщо $A = \{1, 3, 8\}$, $B = \{x, y\}$

Розв'язання. $A \times B = \{(1, x), (1, y), (3, x), (3, y), (8, x), (8, y)\}$.

Поняття декартового добутку поширюється на довільну скінченну кількість множників:

$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, a_2, .., a_n) | \ a_1 \in A_1, a_2 \in A_2, \ldots, a_n \in A_n\}$.

Елементи декартового добутку n множин називаються ***кортежами*** довжини n (впорядкованими n-ками).

Приклад. Знайти $B \times A \times B$, якщо $A = \{1, 3, 8\}$, $B = \{x, y\}$.

Розв'язання. Знайдемо спочатку $B \times A$.

$B \times A = \{(x,1),(x,3),(x,8),(y,1),(y,3),(y,8)\}$.

Тоді $B \times A \times B = \{(x, 1, x), (x, 1, y), (x, 3, x), (x, 3, y), (x, 8, x), (x, 8, y), (y, 1, x), (y, 1, y), (y, 3, x), (y, 3, y), (y, 8, x), (y, 8, y)\}$.

Чим відрізняється операція декартового добутку від операцій, які розглядалися в розділі теорії множин? Суттєва відмінність полягає в тому, що елементи множини-результату звичайної операції, наприклад, перетину, можуть належати початковій множині, а в результаті декартового добутку ми отримуємо принципово інші за конструкцією елементи.

2. Властивості декартового добутку множин

Стосовно ***властивостей*** декартового добутку, можна відмітити, що декартовий добуток за визначенням ***не є комутативним*** (X = {1, 2} і Y = {2, 3}; X × Y = {(1, 2), (1, 3), (2, 2), (2, 3)} та
Y × X = {(2, 1), (3, 1), (2, 2), (3, 2)}, тобто X × Y ≠ Y × X).

1) $(A \times B) \times C = A \times (B \times C)$;
2) $(A \cup B) \times C = (A \times C) \cup (B \times C)$;
3) $(A \cap B) \times C = (A \times C) \cap (B \times C)$;
4) якщо $A \subseteq B$, то $A \times C \subseteq B \times C$;
5) якщо $A \neq B$, то $A \times B \neq B \times A$;
6) якщо множини A та B – скінченні, то $|A \times B| = |A| \cdot |B|$ – комбінаторне правило множення.
7) якщо множини A та B – зліченні, то множина A × B також є зліченною.

Якщо одна з множин дорівнює порожній, то результатом декартового добутку буде теж порожня множина: X = ∅, Y = {1, 2}, то X × Y= Y × X =∅.

Потужність прямого добутку множин A і B визначають співвідношенням:

|A × B| = |A|· |B|.

Приклад.

A {a, b, c, d }, B {4, 5})

A × B = {(a, 4), (a, 5), (b, 4), (b, 5), (c, 4), (c, 5), (d, 4), (d, 5)}

|A × B| = 4 · 2 = 8.

Зрозуміло, що прямий добуток множин також може містити пари з однаковими першим і другим елементами.

3. Поняття відношення. Задання бінарних відношень

Бінарним (або двомісним) відношенням R називається підмножина декартового добутку A×B , тобто R ⊆A×B (тобто множина, кожен елемент якого є впорядкована пара).

Той факт, що елементи a ∈ A та b ∈ B перебувають у бінарному відношенні R позначається як a*R*b або (a, b)∈ R . Якщо відповідні елементи не перебувають у відношенні R, то це записується як $a\overline{R}b$ або (a,b)∉ R .

Множини A та B називаються ***базисними множинами бінарного відношення***.

Розглянемо підмножину A={(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)} декартового добутку X×Y, де X=Y={1, 2, 3, 4}. Це не що інше, як відношення R_1: x < y – «x менше y». На тій же множині впорядкованих пар можна виділити ще одне відношення R_2 : y = x + 1 – це підмножина {(1, 2), (2, 3), (3, 4)}. Інше відношення R_3: x = y – це підмножина {(1, 1), (2, 2), (3, 3), (4, 4)}.

Бінарне відношення називається ***однорідним,*** якщо його базисні множини співпадають.

Приклад. Розглянемо R – бінарне відношення подільності, визначене на множинах A = {3,5,6} та B = {1, 2, 4}. Вважаємо, що x*R*y тоді і тільки тоді, коли x націло ділиться на y.

Тоді R = {(3, 1), (5, 1), (6, 1), (6, 2)}

Оскільки A ≠ B , то відношення R – неоднорідне.

Приклад. Однорідне бінарне відношення R – «навчатися у одній групі» визначене на множині студентів ЛПІ.

x*R*y тоді і тільки тоді, коли студенти x та y – одногрупники.

Областю визначення бінарного відношення R (позначення: D(R)) називають множину перших координат елементів з R, ***областю значень відношення R*** (позначення: E(R)) – множину других координат елементів з R.

Приклад. Знайти області визначення і значень відношення

A = { $(a, 1)$, ($c, 1$), $(c, 2)$, $(c, 5)$, $(e, 7)$ }.

Розв'язання: Область визначення заданого відношення D(A) = $\{a, c, e\}$, а область значень – E(A) = $\{1, 2, 5, 7\}$.

n-арним відношенням, визначеним на множинах $A_1,...,A_n$, називається довільна підмножина декартового добутку цих множин. …

Якщо n = 1, то маємо унарне відношення, якщо n = 2 – бінарне відношення, якщо n = 3 – тернарне відношення.

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

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

В цьому відношенні беруть участь три множини:

Фірми = {АП «Orion«, ТОВ «День«, ПП «Sit«, ТОВ «House«}.

Міста = {Одеса, Харків}.

Продукція = {меблі, світильники, матеріали}.

Це тернарне відношення можна формально зобразити списком елементів:

{(АП «Orion», Одеса, меблі), (ТОВ «День», Харків, світильники), (ЧКП «Sit», Одеса, меблі), (ПП «Sit», Одеса, світильники), (ТОВ «House» Харків, світильники), (ТОВ «House», Харків, матеріали)}.

4. Способи задання відношення множин

Оскільки n-арні відношення є множинами, то для їх задання можна використовувати ті самі способи, що і для множин. Переліком елементів

можливо задавати n-арні відношення при $n \geq 2$. За допомогою матриці і графа зручно задавати тільки бінарні відношення.

Відношення, визначені на скінченних множинах, звичайно задаються:

1. Списком (перерахуванням) упорядкованих пар, для яких це відношення виконується.

Приклад. На множинах чисел $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $B = \{24, 25, 26\}$ побудуємо відношення «дільник», яке складається з упорядкованих пар вигляду

(x, y), де x – дільник y, $x \in A$, $y \in B$:

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

Крім того, якщо бінарні відношення задані на скінченних множинах, то їх можна задавати за допомогою матриць відношень та графів (діаграм) відношень.

2. При ***матричному способі*** задання відношення елементам множини A ставляться у відповідність рядки матриці M (R), елементам множини B – стовпці. Якщо пара (a,b) перебуває у відношенні R, то на перетині відповідного їм рядка та стовпця матриці записується одиниця, інакше – нуль.

Приклад. Відношення R = {(a, 1), (a, 2), (b, 4), (d, 1), (f, 4)} визначене на множинах A = {a, b, c, d, e, f} та B = {1, 2, 3, 4}. Задати його за допомогою матриці.

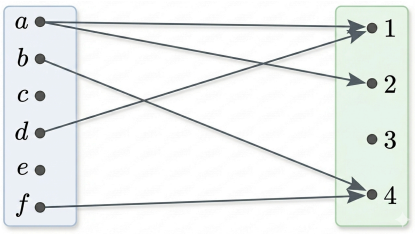
Розв'язання. Матриця відношення R має вигляд

$$M(R) = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

3. При ***графічному способі*** задання відношень елементам множин A та B ставляться у відповідність точки на площині. Якщо пара (a, b) перебуває у відношенні, то точка, яка відповідає елементу a, з'єднується направленим відрізком із точкою, яка відповідає елементу b. 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.

Приклад. Відношення R = {(a, 1), (a, 2), (b, 4), (d, 1), (f, 4)}визначене на множинах

A = {a, b, c, d, e, f } та B = {1, 2, 3, 4}. Задати його за допомогою діаграми (графічно).



Приклад. Наступна матриця W задає відношення R «дільник» для числових множин

A = {1, 2, 3, 4, 5, 6, 7, 8, 9}, B = {24, 25, 26}:

$A\backslash B$	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

Множини pr_1R = {x |(x, y)∈ R} та pr_2R = {y|(x, y)∈ R}відповідно називаються ***першою*** та ***другою проекціями*** відношення R.

Для відношення R = {(a,1),(a,2),(b, 4),(d,1),(f, 4)}, яке визначене на множинах

A = {a,b,c, d,e, f } та B = {1, 2,3, 4}:

$pr_1 R$ = {a,b, d, f }, $pr_2 R$ = {1,2,4}.

Множина R[C] = {y|(c, y)∈ R, c ∈ C} називається ***перерізом бінарного відношення*** R за підмножиною C першої базисної множини A.

Для відношення R із попереднього прикладу R[{a, d}] = {1, 2}.

Переріз R[{a}] називається перерізом бінарного відношення за елементом a. Часто для одноелементних зрізів використовують позначення R[a].

Для відношення R із попереднього прикладу R[b] = {4}, R[c] = ∅.

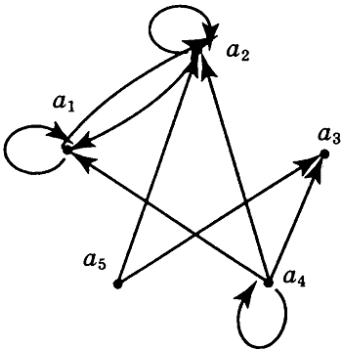
Множина всіх одноелементних перерізів бінарного відношення R, визначеного на множинах A та B, називається ***фактор-множиною*** множини B за відношенням R і позначається B / R. Тобто, B / R = {R[x] x ∈ A}.

Наприклад, для відношення R із діаграмою, наведеною на рис.,

R[a] = {1, 2}, R[b] = R[f] = {4}, R[c] = R[e] = ∅, R[d] = {1}.

Тому B / R = {{1, 2},{4},{1},∅}.

У випадку однорідних бінарних відношень на діаграмі відношення зображується лише одна множина точок.

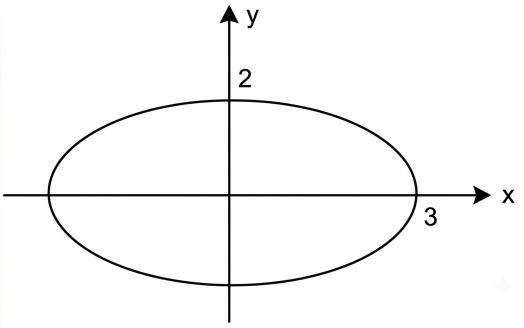


Приклад діаграми однорідного відношення

Діаграмі, зображеній на рис., відповідає однорідне бінарне відношення $R = \{(a_1, a_1),(a_1, a_2),(a_2, a_1),(a_2, a_2),(a_4, a_1),(a_4, a_2),(a_4, a_3),(a_4, a_4),(a_5, a_2),(a_5, a_3)\}$, визначене на множині $\{a_1, a_2, a_3, a_4, a_5\}$.

Розглянемо кілька *прикладів* відношень:

1. Якщо R − множина дійсних чисел, тобто $\left\{(x, y) \in R | \frac{x^2}{9} + \frac{y^2}{4} = 1\right\}$ бінарне відношення на R. Графічно його зобразити можна в такий спосіб:



2. Якщо N − множина натуральних чисел, то відношення $\{(x, y) \in N \times N | x \geq y\}$ виконується для пар (5, 3), (7, 1), (2, 2), але не виконується для пар (1, 7), (9, 11), (2, 5).

3. Якщо X − множина студентів університету, а Y − множина груп університету, то відношення множин X і Y − є множина $\{(x, y) \in X \times Y | x - \text{студент групи } y\}$.

4. Якщо X − множина товарів у магазині, а Y = R^+ − множина дійсних додатних чисел, то відношення множин X і Y − є множина $\{(x, y) \in X \times Y | y - \text{ціна } x\}$.

5. Нехай A = {1, 2, 3}, B = {2, 3, 4}. Знайти декартовий добуток множин A×B та B×A. Записати (A×B) \ (B× A), (A×B)∩(B×A), (A×B)÷(B×A).

Розв'язання:

$A \times B = \{(1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}$

$B \times A = \{(2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (4, 1), (4, 2), (4, 3)\}$

$(A \times B) \backslash (B \times A) = \{(1, 2), (1, 3), (1, 4), (2, 4), (3, 4)\}$

$(A \times B) \cap (B \times A) = \{(2, 2), (2, 3), (3, 2), (3, 3)\}$

$(A \times B) \div (B \times A =$

$\{(1, 2), (1, 3), (1, 4), (2, 4), (3, 4), (2, 1), (3, 1), (4, 1), (4, 2), (4, 3)\}$

6. Нехай A = $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Задати в явному вигляді (списком) і матрицею відношення R ⊆A×A, якщо відношення R означає «бути строго більше».

Розв'язання:

Відношення R містить всі впорядковані пари елементів x, y ∈ A:

R = $\{(x, y) \in A \times A | x > y\}$.

Список відношення R виглядає в такий спосіб:

$R = \{(2, 1), (3, 1), (3, 2), (4, 1), (4, 2), (4, 3), (5, 1), (5, 2), (5, 3),$
$(5, 4), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (7, 1), (7, 2), (7, 3),$
$(7, 4), (7, 5), (7, 6), (8, 1), (8, 2), (8, 3), (8, 4), (8, 5), (8, 6), (8, 7),$
$(9, 1), (9, 2), (9, 3), (9, 4), (9, 5), (9, 6), (9, 7), (9, 8), (10, 1),$
$(10, 2), (10, 3), (10, 4), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9)\}$

Матриця відношення R:

R	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0
5	1	1	1	1	0	0	0	0	0	0
6	1	1	1	1	1	0	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1	0	0	0
9	1	1	1	1	1	1	1	1	0	0
10	1	1	1	1	1	1	1	1	1	0

7. Нехай A = $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Задати в явному вигляді (списком) і матрицею відношення R ⊆A×A, якщо відношення R означає:

а) «мати спільний дільник, відмінний від одиниці»;

б) «їх сума – число, кратне 3».

Розв'язання:

а) відношення R_1 може бути записане в такий спосіб:

$R_1 = \{(x, y) \in A \times A | x$ і y мають спільний дільник, відмінний від $1\}$

Список відношення R_1 :

$R_1 = \{(2, 2), (2, 4), (2, 6), (2, 8), (2, 10), (3, 3), (3, 6), (3, 9), (4, 2),$
$(4, 4), (4, 6), (4, 8), (4, 10), (5, 5), (5, 10), (6, 2), (6, 3), (6, 4), (6, 6),$
$(6, 8), (6, 10), (7, 7), (8, 2), (8, 6), (8, 8), (8, 10), (9, 3), (9, 6), (9, 9),$
$(10, 2), (10, 4), (10, 5), (10, 6), (10, 8), (10, 10)\}$

Матриця відношення R_1:

R_1	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	1	0	1	0	1	0	1	0	1
3	0	0	1	0	0	1	0	0	1	0
4	0	1	0	1	0	1	0	1	0	1
5	0	0	0	0	1	0	0	0	0	1
6	0	1	1	1	0	1	0	1	1	1
7	0	0	0	0	0	0	1	0	0	0
8	0	1	0	1	0	1	0	1	0	1
9	0	0	1	0	0	1	0	0	1	0
10	0	1	0	1	1	1	0	1	0	1

б) відношення R_2 може бути записане в такий спосіб:

$R_2 = \{(x, y) \in A \times A | \, x + y$ кратне 3$\}$.

Список відношення R_2 :

$R_2 = \{(1, 2), (1, 5), (1, 8), (2, 1), (2, 4), (2, 7), (2, 10), (3, 3),$
$(3, 6), (3, 9), (4, 2), (4, 5), (4, 8), (5, 1), (5, 4), (5, 7), (5, 10),$
$(6, 3), (6, 6), (6, 9), (7, 2), (7, 5), (7, 8), (8, 1), (8, 4), (8, 7),$
$(8, 10), (9, 3), (9, 6), (9, 9), (10, 2), (10, 5), (10, 8)\}$.

Матриця відношення R_2:

R_2	1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	1	0	0	1	0	0
2	1	0	0	1	0	0	1	0	0	1
3	0	0	1	0	0	1	0	0	1	0
4	0	1	0	0	1	0	0	1	0	0
5	1	0	0	1	0	0	1	0	0	1
6	0	0	1	0	0	1	0	0	1	0
7	0	1	0	0	1	0	0	1	0	0
8	1	0	0	1	0	0	1	0	0	1
9	0	0	1	0	0	1	0	0	1	0
10	0	1	0	0	1	0	0	1	0	0

Завдання для самостійного виконання

1. A = {1, 2, 3}, B = {2, 3, 4, 5}, C = {−1, 0, 2}. Знайти декартовий добуток (A ÷ B)× (C \ B) і накреслити його в декартовій системі координат.

2. A = {2, 3, 5}, B = {1, 4}, C = {2, 7}. Записати множину (A × B) ÷ (C × B).

3. A = {1, 2, 3, 5, 7}, B = {1, 2, 4}, C = {2, 9, 10}. Записати множину (A \ B)× (C ÷ B).

4. A = {1, 2, 3, 4}, B = {0,1,2,3}, C = {1, 2, 3, 4, 5, 6}. Записати множину ((A ÷ B) \ C)× C.

5. A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Задати в явному вигляді (списком) і матрицею відношення $R \subseteq A \times A$, якщо відношення R означає «їх сума – просте число».

6. A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Задати в явному вигляді (списком) і матрицею відношення $R \subseteq A \times A$, якщо відношення R означає «їх сума – складне число».

7. Множина M – булеан множини A = {1, 2, 3, 4}, (M = P(A)). Скласти матриці відношень R_1, R_2, R_3, заданих на системі множин M , якщо R_1 – «бути строгим включенням», R_2 – «бути доповненням до», R_3 – «перетинатися з». Задати відношення R_1, R_2, R_3 описом його характеристичних властивостей.

8. Відношення R визначене на множині A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Задати списком і матрицею відношення R = {a, b a, b ∈ A, (a + b) – парне}.

9. Відношення R визначене на множині A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Задати списком і матрицею відношення R = {a,b a,b ∈ A, b – дільник (a + b), $b \neq 1$}.

10. Відношення R визначене на множині A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Задати списком і матрицею відношення R = {a,b a,b ∈ A, (a + 2) – дільник (a + b)}.

11. Для визначених нижче відношень навести приклади пар, для яких виконуються відношення і, навпаки, не виконуються:

а) відношення на множині точок дійсної площини:

R_1 – «знаходитися на колі $x^2 + y^2 = 16$»;

R_2 – «знаходитися всередині кола $x^2 + y^2 = 16$»;

R_3 – «знаходитися зовні кола $x^2 + y^2 = 16$»;

R_4 – «бути точкою перетину ліній $x^2 + y^2 = 1$ й $y = x^3$»;

R_5 – «знаходитися на прямій $2x + 5y = 10$ »;

б) відношення на множині студентів вашої групи:

R_6 – «сидіти за однією партою»;

R_7 – «знаходитися за…у списку студентів академгрупи»;

R_8 – «мають однаковий середній бал за результатами сесії»;

в) відношення, задані на множині членів вашої родини:

R_{12} – «бути старше»;

R_{13} – «бути сином (дочкою)»;

R_{14} – «мають спільну кімнату»;

г) відношення, задані на множині міст України:

R_{15} – «знаходитися в одній області»;

R_{16} – «знаходитися в сусідніх областях»;

R_{17} – «знаходитися по одну сторону річки Дніпро».

Питання для самоконтролю

1. Декартовий добуток множин, його властивості.

2. Поняття бінарного відношення.

3. Способи задання бінарних відношень.

Тема 5. Операції над бінарними відношеннями. Властивості бінарних відношень

Мета вивчення: ознайомлення з основними операціями над бінарними відношеннями та властивостями бінарних відношень

План

1. Операції над бінарними відношеннями.
2. Властивості бінарних відношень.
3. Приклади.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Операції над бінарними відношеннями

Оскільки відношення на множині A задаються підмножинами $R \subseteq A \times B$, то для них визначені ті ж операції, що й над множинами, а саме:

1. Об'єднання: $R_1 \cup R_2 = \{(a,b) | (a,b) \in R_1 \text{ або } (a,b) \in R_2\}$

2. Перетин: $R_1 \cap R_2 = \{(a,b) | (a,b) \in R_1 \; i \; (a,b) \in R_2\}$

3. Різниця: $R_1 \setminus R_2 = \{(a,b) | (a,b) \in R_1 \; i \; (a,b) \notin R_2\}$

4. Доповнення: $\overline{R} = U \setminus R$, де $U = A \times B$.

Приклад. Нехай $A = \{1, 2, 3\}$ та $B = \{1, 2, 3, 4\}$. Визначимо відношення R_1 та R_2:

R_1=$\{(1, 1), (2, 2), (3, 3)\}$, R_2=$\{(1, 1), (1, 2), (1, 3), (1, 4)\}$.

Тоді

$R_1 \cup R_2 = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (3, 3)\}$

$R_1 \cap R_2 = \{(1, 1)\}$

$R_1 \setminus R_2 = \{(2,\ 2),\ (3,\ 3)\}$

$R_2 \setminus R_1 = \{(1,\ 2),\ (1,\ 3),\ (1,4)\}$

Крім того, над відношеннями визначені операції знаходження *оберненого відношення* та *композиції відношень*.

Бінарне відношення $R^{-1} = \{(y, x)|(x, y) \in R\}$, визначене на множинах B та A, називається ***оберненим*** до відношення $R \subseteq A \times B$, $(x, y) \in R$.

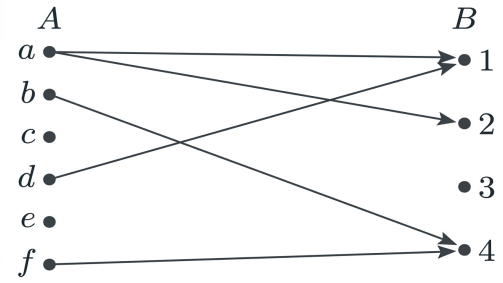
Наприклад, якщо R – «бути старше», то R^{-1} – «бути молодше»; якщо R – «бути дружиною», то R^{-1} – «бути чоловіком».

Нехай R = {(a,b)| b є родич a} або $R = \{(x, y)|x^2 + y^2 = 1\}$. У такому випадку $R = R^{-1}$.

Приклад. Знайти відношення R^{-1}, обернене даному $R = \{(r, 1), (f, 1), (k, 2), (r, 3)\}$.

Розв'язання: $R^{-1} = \{(1, r), (1, f), (2, k), (3, r)\}$

Приклад. Для відношення, діаграма якого наведена на рис., скласти обернене:

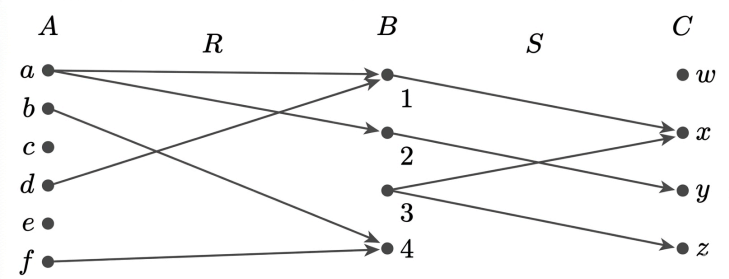


Розв'язання: R^{-1} = {(1, a),(2, a),(4,b),(1, d),(4, f)}

Для бінарного відношення «<», визначеного на довільній числовій множині, оберненим буде відношення «>».

Нехай R – відношення з множини A в множину B, а S – відношення з множини B в множину C. ***Композицією відношень R та S*** називається відношення, яке складається з всеможливих упорядкованих пар (a, c), де $a \in A$, $c \in C$ і для яких існує елемент $b \in B$ такий, що $(a, b) \in R$, $(b, c) \in S$. Композицію відношень R та S позначатимемо $R \circ S$.

Приклад. Знайти добуток бінарних відношень R та S, діаграми яких зображені на рис.



Розв'язання: S◦R = {(a, x),(a, y),(d, x)}.

Приклад. Знайти композицію відношень R та S, де R – відношення з множини A= $\{1, 2, 3\}$ в множину B = $\{1, 2, 3, 4\}$:

R = $\{(1, 1), (1, 4), (2, 3), (3, 1), (3, 4)\}$;

S – відношення з множини B в множину C = $\{1, 2. 3\}$:

S = $\{(1, 0), (2, 0), (3, 1), (3, 2), (4, 1)\}$;

Розв'язання: композицію S ◦ R будуємо, використовуючи всі впорядковані пари з R та S такі, що другий елемент пари з R збігається з першим елементом пари з S.

S ◦ R = $\{(1, 0), (1, 1), (2, 1), (2, 2), (3, 0), (3, 1)\}$

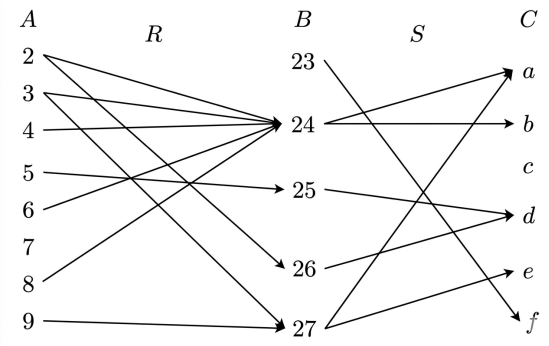
Приклад. Нехай A = {2, 3, 4, 5, 6, 7, 8, 9}, B = {23, 24, 25, 26, 27}, C = {a, b, c, d, e, f}, бінарне відношення R визначене на множинах A та B наступним чином: xRy тоді і тільки тоді, коли y націло ділиться на x, S – бінарне відношення між елементами множин B та C:

S = {(23, f), (24, b), (25, d), (26, d), (27, a), (27,e)}.

Знайти проекції відношення T = S◦R та вказати $T^{-1}\{[a, c, d, f]\}$.

Розв'язання:

R=$\{(2, 24), (2, 26), (3, 24), (3, 27), (4, 24), (5, 25), (6, 24), (8, 24), (9, 27)\}$



T = {(2, b), (2, d), (3, a), (3, b), (3, e), (4, b), (5, d), (6, b), (8, b), (9, a), (9, e)}.

Тому $\mathrm{pr}_1 T$ = {2,3, 4,5,6,8,9}, $\mathrm{pr}_2 T$ = {a,b, d,e}.

Вкажемо перелік елементів відношення T^{-1}:

T^{-1} = {(b, 2), (d, 2), (a, 3), (b, 3), (e, 3), (b, 4), (d, 5), (b, 6), (b, 8), (a, 9), (e, 9)}.

Тоді переріз $T^{-1}[\{a, c, d, f\}]$ = {2,3,5,9}.

Приклад. Нехай на множині A = {1, 2, 3, 4} задано відношення

R = {(1, 1), (2, 1), (3, 2), (4, 3)}. Знайти R^n, n = 1, 2, 3, 4.

Розв'язання:

$R^2 = R \circ R = \{(1, 1), (2, 1), (3, 1), (4, 2)\}$

$R^3 = R^2 \circ R = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$

$R^4 = R^3 \circ R = \{(1, 1), (2, 1), (3, 1), (4, 1)\}$

Приклад. Нехай A = {1, 2, 3, 4}, B = $\{x, y, z\}$, $C = \{\Delta, \odot\}$ і нехай відношення R на $A \times B$ і S на $B \times C$ задані у вигляді:

$R = \{(1, x), (2, x), (3, z), (4, y)\}$,

$S = \{(x, \Delta), (x, \odot), (y, \Delta), (y, \odot), (z, \odot)\}$.

Знайти композицію S◦R.

Розв'язання:

$S \circ R = \{(1, \Delta), (1, \odot), (2, \Delta), (2, \odot), (3, \odot), (4, \Delta), (4, \odot)\}$

Приклад. Нехай на множині натуральних непарних чисел A = {1, 3, 5, 7}, визначене відношення R – «бути більше». Задати характеристичною властивістю і списком відношення R, обернене відношення R^{-1}, доповнення $\overline{R}$, композицію $R \circ R$.

Розв'язання:

$R = \{(a, b) | a > b\}$ – «бути більше»;

$R = \{(3, 1), (5, 1), (5, 3), (7, 1), (7, 3), (7, 5)\}$

$R^{-1} = \{(a, b) | a < b\}$ – «бути менше»;

$R^{-1} = \{(1, 3), (1, 5), (3, 5), (1, 7), (3, 7), (5, 7)\}$

$\overline{R} = (A \times A) \backslash R = \{(a, b) | a \leq b\}$ – «бути не більше»

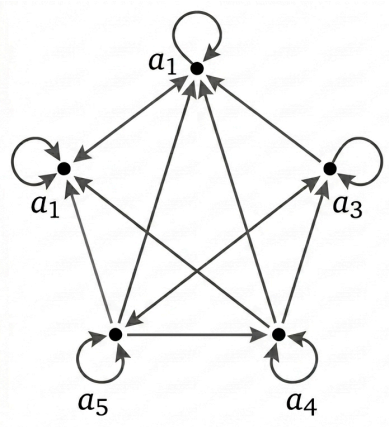
$\overline{R} = \{(1, 1), (1, 3), (1, 5), (1, 7), (3, 3), (3, 5), (3, 7), (5, 1), (5, 3), (5, 5), (5, 7), (7, 7)\}$

$R \circ R = \{(5, 1), (7, 1), (7, 3)\}$

2. Властивості бінарних відношень

Відношення R на A×A називається ***рефлексивним***, якщо має місце *(a,a)*∈ *R* (*aRa*), для кожного *a*∈*A*, тобто кожен елемент a∈A знаходиться у відношенні R до самого себе.

Наприклад, відношення «жити в одній країні» – рефлексивне. Відношення «бути канцелярським приладдям» рефлексивне для всього вмісту пенала. Відношення «≤» на множині дійсних чисел рефлексивне. Відношення рівності є рефлексивним. Відношення x≥y, де x, y∈R є рефлексивним, оскільки x ≥ x. Відношення x > y, де x, y ∈ R – антирефлексивне, оскільки для жодного числа не здійснимо x > x.

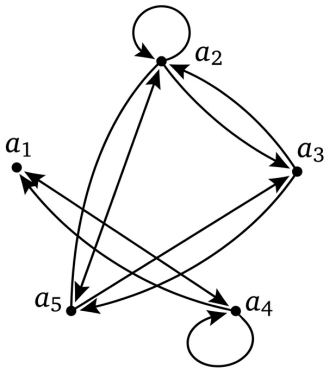


	a_1	a_2	a_3	a_4	a_5
a_1	1	1	0	0	0
a_2	1	1	0	0	0
a_3	0	0	1	0	0
a_4	1	1	1	1	0
a_5	0	1	1	0	1

Відношення R на A×A називається ***антирефлексивним***, якщо ні для якого $a \in A$ не виконується $(a,a) \in R$, тобто із $(a,b) \in R$ слідує, що a ≠ b. Головна діагональ матриці такого відношення містить тільки нулі.

Наприклад, відношення «бути дочкою», «бути молодше» антирефлексивні. Відношення «<» на множині дійсних чисел антирефлексивне.

Відношення R на A×A називається ***симетричним,*** якщо для всіх $a,b \in R$ з умови $a,b \in R$ слідує, що $b, a \in R$. Матриця симетричного відношення симетрична щодо головної діагоналі.



	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

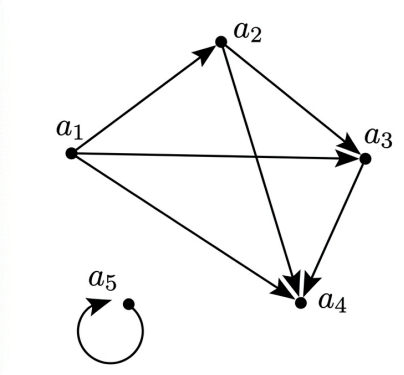
Наприклад, відношення «працювати в одній компанії», «вчитися в одному класі» – симетричні. Відношення «=» на множині дійсних чисел симетричне.

Приклади. З того, що «*x* родич *y*», випливає, що «*y* родич *x*», – відношення симетричне. Відношення «*x* – сестра *y*», визначене на множині всіх людей, несиметричне: можливо, що *y* є братом *x*. Проте те ж відношення, визначене на множині жінок, є симетричним.

Відношення R на A×A називається ***антисиметричним,*** якщо для всіх $a,b \in R$ з умов $(a,b) \in R$ і $(b,a) \in R$ слідує, що $a = b$, тобто ні для яких елементів a і b , що розрізняються (a ≠ b), не виконуються одночасно відношення $(a,b) \in R$ і $(b,a) \in R$. У матриці антисиметричного відношення відсутні одиниці, симетричні щодо головної діагоналі.

Наприклад, відношення «бути дочкою», «бути керівником» – антисиметричні. Відношення «≤» на множині дійсних чисел антисиметричне. Дійсно, якщо a ≤ b й b ≤ a , то a = b.

Відношення R на A×A називається транзитивним, якщо для будь-яких a, b, c з умов *(a,b)* ∈*R* і *(b,c)* ∈*R* слідує *(a,c)* ∈*R*.



	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

Наприклад, відношення «жити в одному місті», «бути сестрою» – транзитивні. Відносини «≤», «=» на множині дійсних чисел транзитивні.

Бінарне відношення називається ***еквівалентним***, якщо воно рефлексивне, симетричне і транзитивне.

Бінарне відношення R називається ***відношенням порядку*** **(**порядком) на множині A, якщо воно є антисиметричним та транзитивним. Якщо порядок є рефлексивним, то він називається частковим (нестрогим) порядком. Прикладом є відношення «≤» на множині дійсних чисел.

Антирефлексивний порядок називається строгим порядком. Прикладом є відношення строгого включення множин. Відношення R є строгим порядком тоді і тільки тоді, коли воно є одночасно асиметричним і транзитивним. Наприклад, відношення «≤» є відношенням часткового порядку, яке відповідає відношенню строгого порядку «<».

Наприклад, відношення «жити в одному місті» – еквівалентне на множині: «велика кількість людей».

Розв'язування задач

Приклад. Нехай $A = \{1, 2, 3, 4, 5, 6\}$ і відношення $R_1 \subseteq A \times A$ є множина $R_1 = \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (1, 2), (1, 4), (2, 1), (2, 4), (3, 5), (5, 3), (4, 1), (4, 2)\}$.

Які властивості має дане відношення?

Розв'язання.

Побудуємо матрицю відношення:

R_1	1	2	3	4	5	6
1	1	1	0	1	0	0
2	1	1	0	1	0	0
3	0	0	1	0	1	0
4	1	1	0	1	0	0
5	0	0	1	0	1	0
6	0	0	0	0	0	1

Відношення R_1 рефлексивне, бо для кожного $a \in A$ $(a, a) \in R_1$. Головна діагональ матриці відношення R_1 містить одиниці. Відношення R_1 не є антирефлексивним, бо з умови $(a, b) \in R$ не випливає $a \neq b$, наприклад, $(2, 2) \in R_1$, але $2 = 2$.

Розглянувши всі можливі випадки методом безпосереднього перебору (табл. а) можна показати, що відношення R_1 симетричне. Крім того, матриця відношення R_1 симетрична відносно головної діагоналі. R_1 не є антисиметричне, бо $(3, 5) \in R_1$ і $(5, 3) \in R_1$, але $3 \neq 5$. Скориставшись методом безпосереднього перебору (табл. б) можна показати, що відношення R_1 транзитивне.

(а)

№	$\langle a, b\rangle \in R$	$\langle b, a\rangle$	$\langle b, a\rangle \in R$?
1	$\langle 1, 2\rangle$	$\langle 2, 1\rangle$	так.
2	$\langle 1, 4\rangle$	$\langle 4, 1\rangle$	так.
3	$\langle 2, 1\rangle$	$\langle 1, 2\rangle$	так.
4	$\langle 2, 4\rangle$	$\langle 4, 2\rangle$	так.
5	$\langle 3, 5\rangle$	$\langle 5, 3\rangle$	так.
6	$\langle 5, 3\rangle$	$\langle 3, 5\rangle$	так.
7	$\langle 4, 1\rangle$	$\langle 1, 4\rangle$	так.
8	$\langle 4, 2\rangle$	$\langle 2, 4\rangle$	так.

(б)

№	$\langle a, b\rangle \in R$	$\langle b, c\rangle \in R$	$\langle a, c\rangle \in R$	$\langle a, c\rangle \in R$?
1	$\langle 1, 2\rangle$	$\langle 2, 1\rangle$	$\langle 1, 1\rangle$	так.
2	$\langle 1, 2\rangle$	$\langle 2, 2\rangle$	$\langle 1, 2\rangle$	так.
3	$\langle 1, 2\rangle$	$\langle 2, 4\rangle$	$\langle 1, 4\rangle$	так.
4	$\langle 1, 4\rangle$	$\langle 4, 1\rangle$	$\langle 1, 1\rangle$	так.
5	$\langle 1, 4\rangle$	$\langle 4, 2\rangle$	$\langle 1, 2\rangle$	так.
6	$\langle 1, 4\rangle$	$\langle 4, 4\rangle$	$\langle 1, 4\rangle$	так.
7	$\langle 2, 1\rangle$	$\langle 1, 1\rangle$	$\langle 2, 1\rangle$	так.
8	$\langle 2, 1\rangle$	$\langle 1, 2\rangle$	$\langle 2, 2\rangle$	так.
9	$\langle 2, 1\rangle$	$\langle 1, 4\rangle$	$\langle 2, 4\rangle$	так.
10	$\langle 2, 4\rangle$	$\langle 4, 1\rangle$	$\langle 2, 1\rangle$	так.
11	$\langle 2, 4\rangle$	$\langle 4, 2\rangle$	$\langle 2, 2\rangle$	так.
12	$\langle 2, 4\rangle$	$\langle 4, 4\rangle$	$\langle 2, 4\rangle$	так.
13	$\langle 3, 5\rangle$	$\langle 5, 3\rangle$	$\langle 3, 3\rangle$	так.
14	$\langle 3, 5\rangle$	$\langle 5, 5\rangle$	$\langle 3, 5\rangle$	так.
15	$\langle 4, 1\rangle$	$\langle 1, 1\rangle$	$\langle 4, 1\rangle$	так.
16	$\langle 4, 1\rangle$	$\langle 1, 2\rangle$	$\langle 4, 2\rangle$	так.
17	$\langle 4, 1\rangle$	$\langle 1, 4\rangle$	$\langle 4, 4\rangle$	так.
18	$\langle 4, 2\rangle$	$\langle 2, 1\rangle$	$\langle 4, 1\rangle$	так.
19	$\langle 4, 2\rangle$	$\langle 2, 2\rangle$	$\langle 4, 2\rangle$	так.
20	$\langle 4, 2\rangle$	$\langle 2, 4\rangle$	$\langle 4, 4\rangle$	так.
21	$\langle 5, 3\rangle$	$\langle 3, 3\rangle$	$\langle 5, 3\rangle$	так.
22	$\langle 5, 3\rangle$	$\langle 3, 5\rangle$	$\langle 5, 5\rangle$	так.

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

a) $\{(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4)\}$;

b) $\{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (4, 4)\}$;

c) $\{(2, 4), (4, 2)\}$;

d) $\{(1, 2), (2, 3), (3, 4)\}$;

e) $\{(1, 1), (2, 2), (3, 3), (4, 4)\}$;

f) $\{(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 4)\}$;

Завдання 2. Які властивості мають відношення:

1. На множині натуральних чисел N:

а) R_1 – «бути не менше»;

б) R_2 – «бути рівним»;

в) R_1 – «їх сума – парне число».

Розв'язання.

На множині натуральних чисел N:

а) $R_1 = \{(a,\ b) |\ a \geq b\}$

– рефлексивне, не антирефлексивне, тому що $a \geq a$ виконується для всіх $a \in N$. Наприклад, $5 \geq 5$;

– не симетричне, тому що $5 \geq 2$, але $2 \geq 5$ невірно;

– антисиметричне, тому що $a \geq b$, і $b \geq a$ виконується тільки тоді, коли $a = b$;

– транзитивне, бо коли $a \geq b$, і $b \geq c$, то $a \geq c$. Наприклад, якщо $8 \geq 5$, і $5 \geq 1$, то $8 \geq 1$;

б) $R_2 = \{(a,\ b) |\ a = b\}$

– рефлексивне, не антирефлексивне, тому що виконується $a = a$ для всіх $a \in N$. Наприклад, $5 = 5$;

– симетричне, бо коли $a = b$, то і $b = a$;

– антисиметричне, бо коли $(a,\ b) \in R_2$ і $(b,\ a) \in R_2$, то $a = b$;

– транзитивне, бо коли $a = b$, і $b = c$, то $a = c$.

в) $R_3 = \{(a,\ b) |\ a + b = 2n,\ n \in N\}$

– рефлексивне, не антирефлексивне, тому що сума двох парних чисел і двох непарних чисел є число парне. Наприклад, $4 + 6 = 10$ – парне і $5 + 7 = 12$ – парне;

– симетричне, не антисиметричне, бо коли $a + b$ – парне, то і $b + a$ – теж парне. Наприклад, $3 + 5 = 8$ – парне і $5 + 3 = 8$ – теж парне;

– транзитивне, бо коли $a + b$ – парне, $b + c$ – парне, то і $a + c$ – теж парне. Наприклад, $3 + 5 = 8$ – парне і $5 + 7 = 12$ – парне, то і $3 + 7 = 10$ – теж парне.

2. На множині «велика кількість людей»:

а) R_1 – «бути дочкою»;

б) R_2 – «бути сестрою»;

в) R_1 – «жити в одній країні».

Розв'язання.

На множині «велика кількість людей»:

а) $R_1 = \{(a,\ b) |\ a -$ дочка $b\}$

– не рефлексивне, антирефлексивне, тому що ні для яких a не може бути виконана умова $a -$ дочка a;

– не симетричне, антисиметричне, бо ні для яких $a \neq b$ не може бути виконана умова $a -$ дочка b і $b -$ дочка a;

– не транзитивне, бо коли $a -$ дочка b і $b -$ дочка c, то неправильно, що $a -$ дочка c.

б) $R_2 = \{(a, b) | \, a -$ сестра $b\}$

– не рефлексивне, антирефлексивне, тому що та сама людина не може бути сама собі сестрою;

– не симетричне, бо у загальному випадку, якщо між братом a і сестрою b має місце $(a, b) \in R_2$, то $(b, a) \notin R_2$;

– не антисиметричне, бо якщо a і b сестри, то $(a, b) \in R_2$ і $(b, a) \in R_2$, але $a \neq b$;

– транзитивне, якщо вважати сестрами людей, що мають загальних батьків.

в) $R_3 = \{(a, b) | \, a$ живе в одній країні з $b\}$

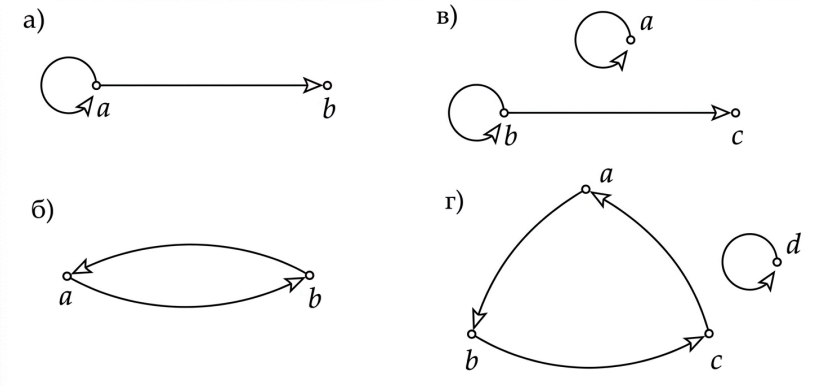
– рефлексивне, не антирефлексивне, тому кожна людина живе в одній і тій же країні із самим собою, тобто $(a, b) \in R_3$;

– симетричне, бо якщо a живе в одній країні з b, то і b живе в одній країні з a, тобто якщо $(a, b) \in R_3$, то і $(b, a) \in R_3$;

– не антисиметричне, тому що з умов $(a, b) \in R_3$ і $(b, a) \in R_3$ не слідує $a = b$;

– транзитивне, бо якщо a живе в одній країні з b, b живе в одній країні з c, то a живе в одній країні з c.

Завдання 3. Визначте, які властивості має кожне з відношень, що зображені такими графами:



Завдання 4. На множині $A = \{a, b, c, d\}$ визначені відношення R_1 та R_2, які задані матрицями (рис.). Які властивості мають відношення R_1 та R_2? Виконати операції над відношеннями R_1 та R_2. Які властивості мають отримані відношення?

R	a	b	c	d
a	1	1	1	1
b	0	0	1	1
c	0	1	1	1
d	0	1	0	0

(а)

R_1	a	b	c	d
a	0	1	0	0
b	1	0	1	0
c	0	1	1	0
d	1	0	0	0

(б)

R_2	a	b	c	d
a	0	1	0	0
b	1	0	0	1
c	0	0	0	0
d	0	1	0	0

(в)

Завдання 5. Відношення R задане матрицею (рис.). Визначити властивості відношення R. Виконати унарні операції над відношенням R, визначити властивості отриманих відношень.

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

(а)

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

(б)

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

(в)

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

(г)

Завдання 6. Відношення R_1 та R_2 задані матрицями (рис.). Визначити властивості цих відношень. Виконати бінарні операції над відношеннями R_1 та R_2 та визначити властивості отриманих відношень.

R_1	a	b	c	d
a	1	1	0	1
b	0	1	0	0
c	1	0	1	0
d	0	1	0	1

R_2	a	b	c	d
a	0	0	0	1
b	1	1	1	1
c	0	0	1	0
d	1	0	0	0

(а)

R_1	a	b	c	d
a	0	1	0	0
b	1	0	1	1
c	0	1	1	0
d	0	0	0	1

R_2	a	b	c	d
a	0	1	0	0
b	0	0	0	1
c	0	1	0	0
d	1	0	1	0

(б)

Питання для самоконтролю

1. Які стандартні теоретико-множинні операції можна виконувати над бінарними відношеннями і як визначаються об'єднання ($R \cup S$) та перетин ($R \cap S$) двох відношень?

2. Що таке обернене відношення (R^{-1}) та композиція (добуток) двох відношень ($R \circ S$), і за яким правилом вони знаходяться?
3. Дайте означення чотирьом фундаментальним властивостям відношень: рефлексивності, антирефлексивності, симетричності та антисиметричності.
4. У чому полягає суть транзитивності бінарного відношення та за якої умови відношення вважається транзитивним?
5. Як властивості відношень (наприклад, рефлексивність чи симетричність) відображаються на їхніх матрицях суміжності або графах?

Тема 6. Функціональні відношення

Мета вивчення: засвоїти поняття функціонального відношення, вивчити властивості та види відображень: ін'єкцію, сюр'єкцію, бієкцію; опанувати метод композиції відношень.

План

1. Функціональні відношення. Основні означення.
2. Види функцій.
3. Приклади

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Функціональні відношення. Основні означення

Бінарне відношення *f,* визначене на множинах A та B, називається ***функціональним***, якщо для довільного $x \in A$ існує не більше ніж один $y \in B$, такий що $(x, y) \in f$.

Інакше кажучи, ***функцією f*** називається таке відношення *R*, що будь-які два різних елементи якого не мають однакових перших координат.

Тобто f є функцією тоді й тільки тоді, коли вона задовольняє наступним умовам:

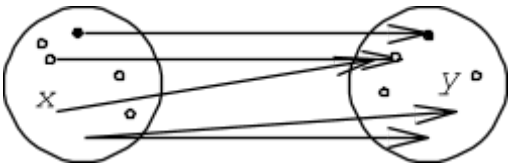
- елементами f є упорядковані пари;
- якщо упорядковані пари (a, b) і (a, c) – елементи функції f, то $b = c$.

Отже, відношення f на A×B називається ***функцією з A в B*** і позначається як $\boldsymbol{f: A \to B}$. Якщо f: A → B функція і $(a,b) \in f$, то говорять, що $b = f(a)$.

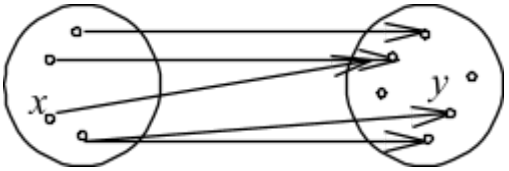
При цьому говорять, що f є ***відображення множини A в множину B*** і записують f: A→B.

Тоді елемент *b* називається *образом* елементу *a*, а *a* – *прообразом* елементу *b*, або *змінною*, або *аргументом*.

На рис. показані відповідність між множинами *A* і *B* та відображення множини *A* в множину B.



Відповідність



Відображення

Множина **A** називається ***областю визначення функції f*** і позначається ***D(f),*** а множина **B** – ***областю значень функції f*** і позначається ***E(f)***.

Приклад. Які із наведених відношень є функціями:

а) $\{(1, 3),\ (2, 7),\ (4, 9),\ (2, 13)\}$;

б) $\{(4, 2),\ (5, 3),\ (1, 11),\ (7, 2)\}$;

в) $\left\{(x,\ x^2) |\ x \in R\right\}$;

г) $\left\{(x^2, x) |\ x \in R\right\}$;

Розв'язання:

а) відношення не є функцією, тому що два елементи (2, 7) і (2, 13) мають однакову першу координату;

б) відношення є функцією, тому що перший елемент кожної впорядкованої пари зустрічається рівно один раз;

в) відношення є функцією, графіком якої буде парабола;

г) відношення не є функцією, тому що його елементами є, наприклад, і (1,1), і (1, −1).

Приклад. Знайти область визначення і область значень функції:

а) $\{(4, 2),\ (5, 3),\ (1, 11),\ (7, 2)\}$;

б) $\left\{(x,\ x^2) |\ x \in R\right\}$;

Розв'язання:

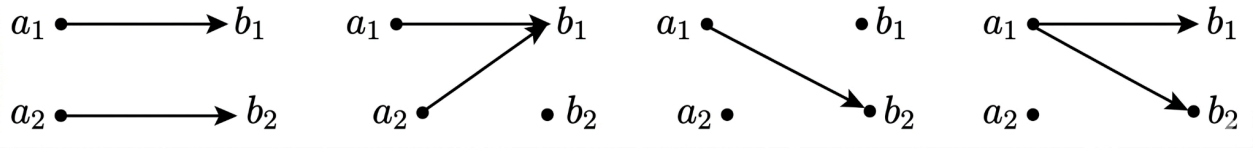
а) область визначення функції A = $\{1, 4, 5, 7\}$, а область значень B = $\{2, 3, 11\}$;

б) область визначення – x∈R, а область значень – y ∈ R^+ .

Якщо базисні множини функціонального відношення скінченні, то кожний рядок матриці M(f) містить не більше однієї одиниці та з кожної вершини на діаграмі відношення виходить не більше однієї дуги.

Приклад. Відношення, діаграми яких зображені на рис. а)-в), –

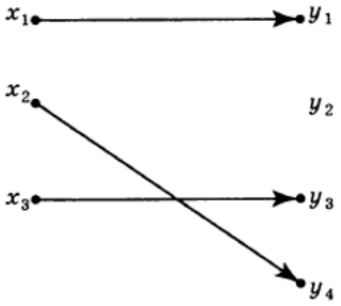
функціональні, а відношення з діаграмою на рис. г) – не є функціональним.

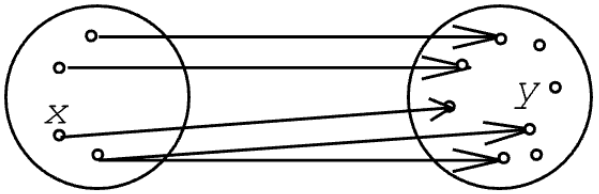


2. Види функцій

Функція f: A → B називається ***ін'єктивною*** (взаємно однозначною), або ***ін'єкцією***, якщо з $x_1 \neq x_2$ слідує, що $f(x_1) \neq f(x_2)$.

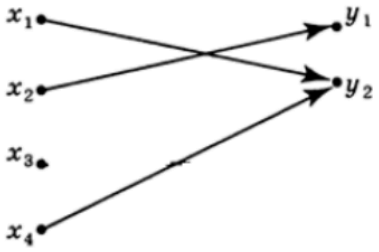
На діаграмі, яка відповідає ін'єктивній функції, у кожну вершину заходить не більше однієї дуги (див. рис)

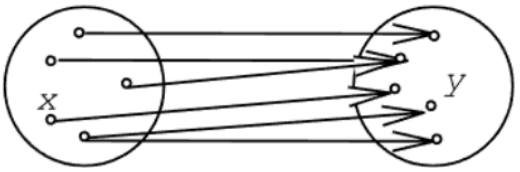




Функція f: A → B називається «відображенням на», ***сюр'єктивною функцією (***покривним відображенням, або відображенням множини A на множину B), або ***сюр'єкцією***, якщо E(f)= B.

Приклад діаграми часткового сюр'єктивного відображення між елементами множин A = $\{x_1, x_2, x_3, x_4\}$ та B = $\{y_1, y_2\}$

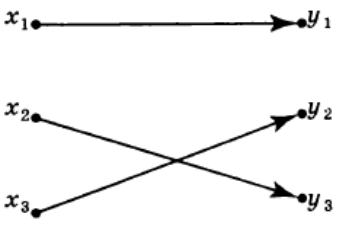


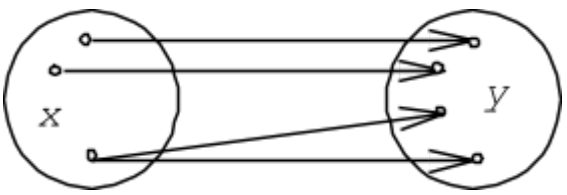


На діаграмі сюр'єктивної функції у кожну вершину, яка відповідає елементам множини B, заходить принаймні одна дуга.

Функція називається ***бієктивною*** (взаємно однозначною або бієкцією), якщо вона є сюр'єктивною та ін'єктивною.

Приклад бієктивного відображення наведено на рис.





Приклад. Чи є функція f (x)= 3x +5 взаємно однозначною?

Розв'язання:

f (x_1)= $3x_1$ +5

f (x_2)= $3x_2$ +5

З умови $f(x_1) = f(x_2)$ слідує: $3x_1$+5= $3x_2$+5. Отже, $x_1 = x_2$ і функція є взаємно однозначною.

Нехай f – функція із множини A в множину B, тобто f: A→B (f⊆A×B). Обернене відношення f^{-1}⊆A×B визначається як $\{(b,a)|\ (a,b) \in f\}$. При цьому f^{-1} називається ***перетворенням функції f*** або її ***оберненою функцією***.

Приклад. Знайти функцію, обернену до даної: y = 5x −1.

Розв'язання: Обертаючи функцію, одержуємо $\{(y,x)|\ y = 5x - 1\}$, але це те ж саме, що і $\{(x,y)|x = 5y - 1,\ x,y \in R\}$. Розв'язуючи рівняння відносно y, одержуємо $\left\{(x,y)|y = \frac{1}{5}(x+1),\ x,y \in R\right\}$. Тобто, якщо

f = $\{(x,y)|y = 5x - 1\}$, то $f^{-1} = \left\{(x,y)|y = \frac{1}{5}(x+1),\ x,y \in R\right\}$

Приклад. Перевірити властивості та вказати обернені функції до функції f: $R \to R$ за умови, що

а) f(x)=3x+6;

б) f(x)=x^2-4;

в) f(x)=x^3-1;

Якщо R^{-1} – обернене відношення до взаємно однозначного функціонального відношення R, то R^{-1} визначає функцію f^{-1}, яку називають оберненою до функції f.

Ін'єктивна функція f має обернену функцію f^{-1} .

Функція f^{-1}, обернена до бієктивної, є відображенням не на множину X, а в множину X . Взаємно однозначність функції зручно доводити виходячи з міркувань:

«з умови $f(x_1) = f(x_2)$ слідує $x_1 = x_2$».

*Приклад***.** З'ясувати, чи є дані відношення функціями? Якщо так, то чи будуть вони взаємно однозначні? У випадку позитивної відповіді, знайти обернені функції:

а) $f(x) = \left\{(x,y)|\frac{x^2}{9} + \frac{y^2}{4} = 1,\ x,y \in R\right\}$,

б) $f(x) = \left\{(x,y)|y^2 = -x,\ x,y \in R\right\}$

в) $f(x) = \{(x,y)|3x - 2y + 6 = 0,\ x,y \in R\}$,

Розв'язання:

а) відношення не є функцією, тому що існує два різних елементи, що мають однакові перші координати (див. рис. а);

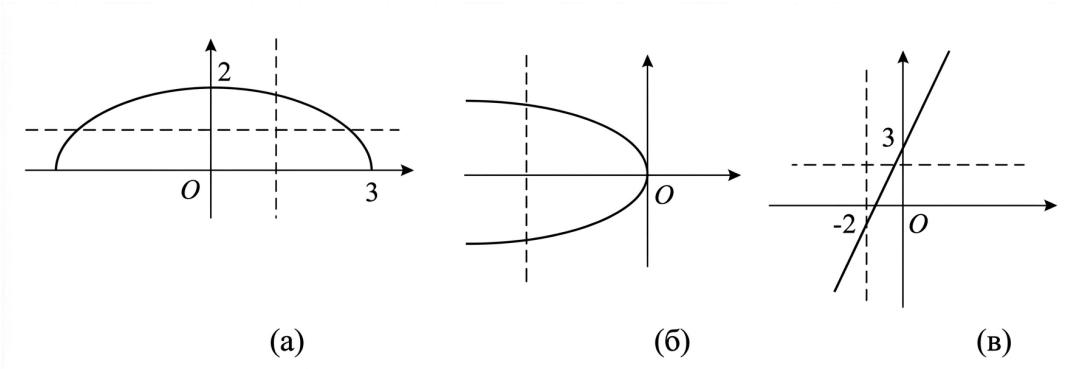
б) відношення є функцією, тому що не існує елементів, що мають однакові перші координати. Дана функція не є взаємно однозначною, тому що існують елементи, що мають однакові другі координати (див. рис. б);

в) відношення є функцією. Дана функція є взаємно однозначною, тому що переводить різні елементи в різні (див. рис. в). Знайдемо функцію, обернену до даної:

$$f = \{(x, y) | 3x - 2y + 6 = 0,\ x, y \in R\},$$

$$f^{-1} = \{(x, y) | 3y - 2x + 6 = 0,\ x, y \in R\},$$

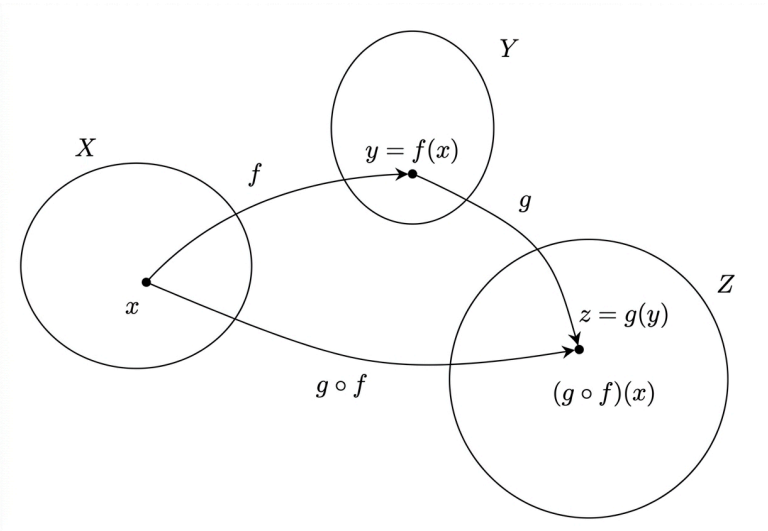
$$f^{-1} = \left\{(x, y) | y = \frac{2}{3}x - 2,\ x, y \in R\right\}$$



(а) (б) (в)

Нехай дані дві функції f : X →Y і g :Y →Z , де X, Y, Z – довільні множини. Функція f визначена на X і приймає значення на Y, а функція g визначена на Y і приймає значення на Z . Якщо для кожного x∈X знайдеться такий елемент z∈Z , що z = g(f (x)), то така відповідність між множинами X і Z називається ***композицією*** або ***суперпозицією*** функцій g і f і позначається ***g ◦ f (***рис.).

Теорема. Нехай f: A → B, g: B → C. Тоді g ◦ f – функція із A у C, причому (g ◦ f)(x)= g(f(x)).



*Приклад***.** Функції $f(x) = x^2$ і $g(x) = x + 5$ задані на множині дійсних чисел. Знайти композицію функцій f ◦ g і g ◦ f .

Розв’язання:

$f \circ g = f(g(x)) = f(x + 5) = (x + 5)^2$

$g \circ f = g(f(x)) = g(x^2) = x^2 + 5$

Розв’язування задач

1. Нехай f ⊆R× R. Знайти області визначення і значень наступних функцій:

$a) f(x) = \frac{1}{x^2+9}$

в) $f(x) = |x + 9|$

2. З'ясуйте, які з наведених нижче відношень є функціями. Визначте властивості функцій. Для взаємно однозначних функцій знайдіть обернені.

а) $f = \{\langle x, y\rangle \mid y = 5x - 1,\ x, y \in R\}$;

б) $f = \left\{\langle x, y\rangle \mid \frac{x^2}{16} - \frac{y^2}{9} = 1,\ x \in R, y \in R^+\right\}$;

в) $f = \{\langle x, y\rangle \mid x^2 + (y-1)^2 = 9, x, y \in R^+\}$.

3. Знайти композицію функцій f ◦ g й g ◦ f , заданих на множині дійсних чисел:

$f(x) = e^{5x}$, $g(x) = \cos 3x$;

$f(x) = \sin 2x$, $g(x) = x - 4$.

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

а) $(a, b) \in R$, якщо $a = b^2$

б) $(a, b) \in R$, якщо $a^2 = b$

в) $(a, b) \in R$, якщо $a \leq b$

г) $(a, b) \in R$, якщо $ab = 6$

Питання для самоконтролю

1. Сформулюйте означення функціонального відношення (відображення).

2. У чому полягає принципова відмінність між довільним бінарним відношенням і функціональним відношенням? Чи кожне відношення є функцією?

3. Дайте означення області визначення та області значень для функціонального відношення.

4. Яке відображення називається ін’єктивним (ін’єкцією)? Наведіть приклад.

5. Сформулюйте означення сюр’єктивного відображення (сюр’єкції). Яка умова має виконуватися для області значень (образу множини) при сюр’єкції?

6. Що таке бієктивне відображення (бієкція)? Які дві властивості воно в собі поєднує? Чому бієкцію інакше називають «взаємно однозначною відповідністю»?

Тема 7. Логіка висловлювань. Алгебра логіки. Основні поняття. Формули логіки висловлювань

Мета вивчення: вивчити основні поняття логіки висловлювань, основні операції над висловленнями, формули логіки висловлень; формувати вміння виконувати рівносильні перетворення формул.

План

1. Основні поняття логіки висловлювань.
2. Логічні операції.
3. Формули логіки висловлювань.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Основні поняття логіки висловлювань

Логіка – це наука про міркування, яка дає змогу визначити істинність або хибність математичного твердження, виходячи з первинних припущень, які називають аксіомами. ***Формальна логіка*** займається аналізом речень, звертаючи основну увагу на їх форму і відволікаючись від змісту.

Логіку також застосовують в інформатиці для побудови комп’ютерних програм і доведення їх коректності. Поняття, методи й засоби логіки покладено в основу сучасних інформаційних технологій.

Розділ логіки, який вивчає висловлювання та їхні властивості, називають ***пропозиційною логікою або логікою висловлювань***.

Під ***висловлюванням*** розуміють розповідне речення, у якому щось стверджується, і про яке у даних умовах місця та часу можна сказати, істинне

воно чи хибне.

Значення «істина» чи «хибність», яких набуває висловлювання, називають його значенням істинності. Значення «істина» часто позначають 1, «хибність» – 0.

Приклад 1. Розглянемо речення:

- Сніг зелений.
- Київ – столиця України.
- Якщо погода хороша, то ми їдемо відпочивати.
- x +1 = 3.
- Котра година?
- Читай уважно!

Три перші речення – висловлювання, решта три – ні, бо четверте речення набуває істинного чи хибного значення залежно від значення змінної x, п'яте та шосте речення – не розповідні. Перше висловлювання є хибним, друге та третє – істинними.

Для позначення змінних висловлювань використовують великі латинські літери. Висловлювання поділяються на *прості (елементарні)* та *складні*.

Атомами (елементарними висловлюваннями) називаються висловлювання, які відповідають простим розповідним реченням, які не мають складових частин.

Складні висловлювання утворюють із простих висловлювань за допомогою *логічних операцій (логічних зв'язок)*.

Перші два висловлювання прикладу 1 є простими, третє – складне.

2. Логічні операції

Запереченням висловлення A називається висловлення $\neg A$ (читається «не А»), істинне тоді і тільки тоді, коли висловлювання А є хибним.

Кон'юнкцією висловлювань A та B називається нове висловлювання, яке позначається $A \wedge B$ (читається «А і В») і є істинним тоді і тільки тоді, коли обидва висловлювання А та В істинні.

Диз'юнкцією висловлювань A та B називається нове висловлювання, яке позначається $A \vee B$ (читається «А або В») і є хибним тоді і тільки тоді, коли обидва висловлювання А та В хибні.

Імплікацією називається висловлювання $A \Rightarrow B$, яке хибне тоді і тільки тоді, коли висловлювання А істинне, а В – хибне. Висловлювання ***A – припущення (засновок, антецедент)*** імплікації, ***B – висновок (консеквент) імплікації.***

Причинно-наслідковий зв'язок між А і В, що виражається імплікацією, на природній мові описується такими зворотами: «якщо А, то В», «з А випливає В», «А лише тоді, коли В», «А є достатнім для В», «В є необхідним для А».

Еквівалентністю висловлювань A та B називається висловлювання ***A*** $\Leftrightarrow$ ***B***, яке є істинним тоді і тільки тоді, коли висловлювання А та В мають однакові значення істинності.

У природній мові цій операції відповідають звороти: «А тоді і тільки тоді, коли В», «А є необхідним і достатнім для В», «А та В є еквівалентними».

Значення операцій логіки висловлювань наведені у таблиці:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \Leftrightarrow B$	$A \oplus B$
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	0	0	1
1	1	0	1	1	1	1	0

3. Формули логіки висловлювань

Означення формули логіки висловлювань:

1) кожний атом – це формула;

2) якщо ϕ та ψ – формули, то $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \Rightarrow \psi)$, $(\phi \Leftrightarrow \psi)$ – формули.

3) формули можуть бути породжені тільки скінченною кількістю застосувань вказаних правил.

Для зменшення кількості дужок вважають, що логічні операції мають наступний пріоритет (від найвищого до найнижчого): $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ і замість $(\neg\phi)$ пишуть $\overline{\phi}$.

Приклад 2. Записати формулу, яка відповідає висловлюванням

1) «Якщо Іван пропустить лекцію з дискретної математики або не повторить матеріал самостійно, то він погано напише модуль».
2) «Оскільки Петро пізно ліг спати, то він проспав і через це не встиг на автобус та запізнився на пару».

Розв'язок.

1) Виділимо елементарні висловлювання, які входять до складу нашого першого складного висловлювання:

   А – «Іван пропустить лекцію з дискретної математики»,

   В – «Іван повторить матеріал самостійно»,

   С – «Іван напише модуль погано».

   Тоді структуру складного висловлювання описує формула $A \vee \overline{B} \Rightarrow C$

2) Елементарні висловлювання:

   А – «Петро пізно ліг спати»;

   В – «Петро проспав»;

C – «Петро встиг на автобус»;

D – «Петро спізнився на пару».

Тоді структуру другого складного висловлювання описує формула $(A \Rightarrow B) \Rightarrow \overline{C} \wedge D$

Під ***формальною інтерпретацією формули логіки висловлювань*** розуміють присвоєння змінним висловлюванням, з яких побудована формула, деяких значень істинності 0 або 1.

Приклад 3. Вказати значення формули $\varphi = \overline{A \wedge \overline{B}} \Leftrightarrow \overline{C} \Rightarrow B \vee A$

Розв'язок. $\phi\,(1,0,1) = \overline{1 \wedge \overline{0}} \Leftrightarrow \overline{1} \Rightarrow 0 \vee 1 = \overline{1 \wedge 1} \Leftrightarrow 0 \Rightarrow \overline{1} = 1 \Leftrightarrow 1 = 0 \Leftrightarrow 1 = 0.$

Таблиця, які містить значення формули логіки висловлювань на всіх її інтерпретаціях змінних, називається ***таблицею істинності формули.***

Приклад 4. Побудувати таблицю істинності формули

$\varphi = \overline{\overline{A} \wedge B} \Rightarrow (C \Leftrightarrow B \wedge \overline{A})$

Розв'язок. Поставимо у відповідність кожній підформулі формули ϕ окремий стовпчик таблиці.

A	B	C	$\overline{A}$	$\overline{A} \vee B$	$\overline{\overline{A} \wedge B}$	$B \wedge \overline{A}$	$C \Leftrightarrow B \wedge \overline{A}$	ϕ
0	0	0	1	1	0	0	1	1
0	0	1	1	1	0	0	0	1
0	1	0	1	1	0	1	0	1
0	1	1	1	1	0	1	1	1
1	0	0	0	0	1	0	1	1
1	0	1	0	0	1	0	0	0
1	1	0	0	1	0	0	1	1
1	1	1	0	1	0	0	0	1

Логічним рівнянням називається рівняння вигляду $\phi = \psi$, де ϕ, ψ – формули логіки висловлювань або логічні константи 0 чи 1.

Розв'язати логічне рівняння означає вказати усі набори значень атомів, які перетворюють його на правильну рівність.

Приклад 5. Розв'язати логічне рівняння $P \wedge \overline{Q} \Rightarrow \overline{R} \vee Q = 0$.

Розв'язок. Оскільки імплікація набуває значення 0 лише для одного набору значень аргументів, то рівняння рівносильне системі

$$\begin{cases} P \wedge \overline{Q} = 1, \\ \overline{R} \vee Q = 0. \end{cases}$$

Розв'язуючи останню систему, отримаємо

$$\begin{cases} P=1, \\ \bar{Q}=1, \\ \bar{R}=0, \\ Q=0. \end{cases} \Leftrightarrow \begin{cases} P=1, \\ Q=0, \\ R=1. \end{cases}$$

Відповідь: {(1,0,1)}.

Приклад 6. Розв'язати логічне рівняння $P \wedge \bar{Q} \vee R = \bar{R} \Leftrightarrow P$.

Розв'язок. Оскільки логіка висловлювань двозначна, то можливими є два випадки:

1-й випадок:

$$\begin{cases} P \wedge \bar{Q} \vee R = 0, \\ \bar{R} \Leftrightarrow P = 0. \end{cases}$$

2-й випадок:

$$\begin{cases} P \wedge \bar{Q} \vee R = 1, \\ \bar{R} \Leftrightarrow P = 1. \end{cases}$$

Розв'яжемо спочатку першу систему:

$$\begin{cases} P \wedge \bar{Q} \vee R = 0, \\ \bar{R} \Leftrightarrow P = 0. \end{cases} \Leftrightarrow \begin{cases} P \wedge \bar{Q} = 0, \\ R = 0, \\ 1 \Leftrightarrow P = 0. \end{cases} \Leftrightarrow \begin{cases} 0 \wedge \bar{Q} = 0, \\ R = 0, \\ P = 0. \end{cases} \Leftrightarrow \begin{cases} P = 0, \\ R = 0. \end{cases}$$

Зауважимо, що перше рівняння передостанньої системи перетворюється у правильну рівність при всіх значеннях невідомої Q. Отже, у першому випадку ми отримали два розв'язки (0,0,0) та (0,1,0).

Розв'яжемо другу систему. Для цього спочатку проаналізуємо її друге рівняння. Еквівалентність приймає значення 0 у двох випадках. Тому

$$\begin{cases} P \wedge \bar{Q} \vee R = 1, \\ \bar{R} \Leftrightarrow P = 1. \end{cases} \Leftrightarrow \left[\begin{array}{l} \begin{cases} \bar{R} = 0, \\ P = 0, \\ 0 \wedge \bar{Q} \vee 1 = 1 \end{cases} \\ \begin{cases} \bar{R} = 1, \\ P = 1, \\ 1 \wedge \bar{Q} \vee 0 = 1. \end{cases} \end{array} \right. \Leftrightarrow \left[\begin{array}{l} \begin{cases} R = 1, \\ P = 0. \end{cases} \\ \begin{cases} R = 0, \\ P = 1, \\ \bar{Q} = 1. \end{cases} \end{array} \right. \Leftrightarrow \left[\begin{array}{l} \begin{cases} P = 0, \\ R = 1. \end{cases} \\ \begin{cases} P = 1, \\ Q = 0, \\ R = 0. \end{cases} \end{array} \right.$$

Отже, ми отримали ще три розв'язки (0, 0, 1), (0, 1, 1) та (1, 0, 0).

Відповідь: {(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0)}.

Формула логіки висловлювань називається ***загальнозначущою*** (***тотожно істинною*** або ***тавтологією***), якщо вона є істинною на всіх своїх інтерпретаціях. Той факт, що формула φ є загальнозначущою позначається $\models \varphi$.

Формула логіки висловлювань називається ***суперечливою*** (***тотожно хибною),*** якщо вона є хибною на всіх своїх інтерпретаціях.

Формула логіки висловлювань називається ***виконуваною***, якщо вона є істинною принаймні на одній своїй інтерпретації.

Приклад 7. Перевірити, чи є суперечливою формула

$\varphi = \overline{(\overline{A} \Leftrightarrow B) \vee (A \Rightarrow B)}$

Розв'язок. Побудуємо таблицю істинності формули φ.

A	B	$\overline{A}$	$\overline{A} \Leftrightarrow B$	$A \Rightarrow B$	$(\overline{A} \Leftrightarrow B) \vee (A \Rightarrow$)	ϕ
0	0	1	0	1	1	0
0	1	1	1	1	1	0
1	0	0	1	0	1	0
1	1	0	0	1	1	0

Оскільки формула φ хибна на усіх інтерпретаціях, то формула φ суперечлива.

Приклад 8. Довести, що формула $\phi = \overline{\overline{B} \Rightarrow \overline{C}} \Rightarrow (C \Leftrightarrow A \vee C)$ є загальнозначущою.

Розв'язок. Використаємо метод від супротивного. Припустимо, що формула ϕ не є загальнозначущою. Тоді рівняння ϕ = 0 повинно мати хоча б один розв'язок. Перевіримо це.

$$\overline{\overline{B} \Rightarrow \overline{C}} \Rightarrow (C \Leftrightarrow A \vee C) = 0 \Rightarrow \begin{cases} \overline{\overline{B} \Rightarrow \overline{C}} = 1, \\ (C \Leftrightarrow A \vee C) = 0. \end{cases} \Rightarrow \begin{cases} \overline{B} \Rightarrow \overline{C} = 0, \\ (C \Leftrightarrow A \vee C) = 0. \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} \overline{B} = 1, \\ \overline{C} = 0, \\ (C \Leftrightarrow A \vee C) = 0. \end{cases} \Rightarrow \begin{cases} B = 0, \\ C = 1, \\ (1 \Leftrightarrow A \vee 1) = 0. \end{cases} \Rightarrow \begin{cases} B = 0, \\ C = 1, \\ 1 \Leftrightarrow 1 = 0. \end{cases}$$

У результаті ми отримали систему, останнє рівняння якої є неправильною рівністю, оскільки $1 \Leftrightarrow 1 = 1 \neq 0$. Отже рівняння ϕ = 0 не має розв'язків. Тому формула ϕ є загальнозначущою (⊨ϕ).

Питання для самоконтролю

1. Сформулюйте означення поняття «висловлювання» в алгебрі логіки.
2. Наведіть 3 приклади речень, що є висловлюваннями, та 3 приклади речень, що не є висловлюваннями (запитання, накази, оклики).
3. Які значення істинності можуть приймати висловлювання? Як ці значення зазвичай позначаються цифровими символами та словесно?
4. Основні логічні операції. Побудова таблиць істинності.
5. Яким є пріоритет (порядок) виконання логічних операцій при обчисленні значень складних висловлювань, якщо немає дужок?
6. Формули логіки висловлень, їх види.
7. Спрощення та доведення формул.

Тема 8. Закони логіки висловлень. Рівносильні перетворення формул. Спеціальні форми подання булевих функцій

Мета вивчення: вивчити основні закони логіки висловлювань, формувати вміння виконувати рівносильні перетворення формул, здійснювати аналіз міркувань з використанням логічного слідування; ознайомити студентів із поняттям спеціальних форм подання булевих функцій

План

1. Рівносильні перетворення формул.
2. Логічне слідування. Аналіз міркувань.
3. Кон'юнктивна нормальна форма (КНФ) та диз'юнктивна нормальна форма (ДНФ).

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Рівносильні перетворення формул

Формули ϕ та ψ називаються ***рівносильним*** (еквівалентними) формулами логіки висловлювань, якщо вони приймають однакові значення істинності на всіх інтерпретаціях змінних, кожна з яких входить принаймні у одну з формул ϕ або ψ. Той факт, що формули ϕ та ψ рівносильні, позначається як $\phi \equiv \psi$.

Відношення рівносильності формул, побудованих із деякої фіксованої множини атомів, задовольняє умови рефлексивності ($\phi \equiv \phi$), симетричності ($\phi \equiv \psi \Rightarrow \psi \equiv \phi$) та транзитивності ($\phi \equiv \psi$, $\psi \equiv \omega \Rightarrow \phi \equiv \omega$), а отже є відношенням еквівалентності. Тому множина усіх формул розбивається на класи, у кожному з яких містяться рівносильні між собою формули.

Приклад 1. Перевірити, чи рівносильні формули

$\phi = \overline{C} \Rightarrow A$ та $\psi = \overline{A \vee C} \Rightarrow (A \vee \overline{B} \Rightarrow \overline{C}) \wedge C$.

Розв'язання. Побудуємо таблицю істинності формул φ та ψ.

A	B	C	$\overline{C}$	φ	$A \vee C$	$\overline{A \vee C}$	$\overline{B}$	$A \vee \overline{B}$	$A \vee \overline{B} \Rightarrow \overline{C}$	$(A \vee \overline{B} \Rightarrow \overline{C}) \wedge C$	ψ
0	0	0	1	0	0	1	1	1	1	0	0
0	0	1	0	1	1	0	1	1	0	0	1
0	1	0	1	0	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	1	1	1
1	0	0	1	1	1	0	1	1	1	0	1

A	B	C	$\overline{C}$	φ	$A \vee C$	$\overline{A \vee C}$	$\overline{B}$	$\frac{A}{B} \vee$	$A \vee \overline{B} \Rightarrow \overline{C}$	$(A \vee \overline{B} \Rightarrow \overline{C}) \wedge C$	ψ
1	0	1	0	1	1	0	1	1	0	0	1
1	1	0	1	1	1	0	0	1	1	0	1
1	1	1	0	1	1	0	0	1	0	0	1

Оскільки значення формул на усіх інтерпретаціях співпадають, то формули φ та ψ рівносильні.

Рівносильні перетворення формул ґрунтуються на законах логіки висловлювань, основні з яких наведені у таблиці:

№	Назва	Формулювання закону
1.	Закон комутативності кон'юнкції	$A \wedge B \equiv B \wedge A$
2.	Закон комутативності диз'юнкції	$A \vee B \equiv B \vee A$
3.	Закон асоціативності кон'юнкції	$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$
4.	Закон асоціативності диз'юнкції	$(A \vee B) \vee C \equiv A \vee (B \vee C)$
5.	Перший закон дистрибутивності	$A \wedge (B \vee C) \equiv A \wedge B \vee A \wedge C$
6.	Другий закон дистрибутивності	$A \vee B \wedge C \equiv (A \vee B) \wedge (A \vee C)$
7.	Закон подвійного заперечення	$\overline{\overline{A}} \equiv A$
8.	Закон суперечності	$A \wedge \overline{A} \equiv 0$
9.	Закон виключення третього	$A \vee \overline{A} \equiv 1$
10.	Перший закон де Моргана	$\overline{A \wedge B} \equiv \overline{A} \vee \overline{B}$
11.	Другий закон де Моргана	$\overline{A \vee B} \equiv \overline{A} \wedge \overline{B}$
12.	Закон ідемпотентності кон'юнкції	$A \wedge A \equiv A$
13.	Закон ідемпотентності диз'юнкції	$A \vee A \equiv A$
14.	Властивості нуля	$A \wedge 0 \equiv 0,\ A \vee 0 \equiv A$
15.	Властивості одиниці	$A \wedge 1 \equiv A,\ A \vee 1 \equiv 1$
16.	Перший закон поглинання	$A \vee A \wedge B \equiv A$
17.	Другий закон поглинання	$A \wedge (A \vee B) \equiv A$
18.	Модифікований закон поглинання	$A \vee \overline{A} \wedge B \equiv A \vee B$
19.	Закон усунення імплікації	$A \Rightarrow B \equiv \overline{A} \vee B$
20.	Перший закон усунення еквівалентності	$A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$
21.	Другий закон усунення еквівалентності	$A \Leftrightarrow B \equiv (\overline{A} \wedge \overline{B}) \vee (A \wedge B)$

Теорема 1. Формули φ та ψ рівносильні тоді і тільки тоді, коли $\models \phi \Leftrightarrow \psi$ (формула $\phi \Leftrightarrow \psi$ є загальнозначущою).

Теорема 2. Якщо ϕ_1 — підформула формули ψ_1, $\phi_1 \equiv \phi_2$ і формула ψ_2 отримується з формули ψ_1 заміною підформули ϕ_1 на ϕ_2, то формули ψ_1 та ψ_2 рівносильні.

Теорема 2 та закони 1-21 використовується для побудови ланцюжків рівносильних формул.

Наприклад, формули $\overline{A \wedge B} \wedge \overline{A} \Rightarrow C$ та $(\overline{A} \vee \overline{B}) \wedge \overline{A} \Rightarrow C$ рівносильні, оскільки за першим законом де Моргана підформула $\phi_1 = A \wedge B$ першої

формули рівносильна формулі $\phi_2 = A \vee B$ другої формули.

Приклад 2. Для формули $\phi = \overline{A} \wedge B \Rightarrow (C \Leftrightarrow \overline{A})$ побудувати рівносильну їй формулу, яка би містила лише атоми, символи операцій заперечення та диз'юнкції.

Розв'язання.

$$\overline{A} \wedge B \Rightarrow \left(C \Leftrightarrow \overline{A}\right) \overset{19}{\equiv} \overline{\overline{A} \wedge B} \vee \left(C \Leftrightarrow \overline{A}\right) \overset{4,10}{\equiv} \overline{\overline{A}} \vee \overline{B} \vee \left(C \Leftrightarrow \overline{A}\right) \overset{4,7,21}{\equiv} A \vee \overline{B} \vee \overline{C} \wedge \overline{\overline{A}} \vee C \wedge \overline{A} \equiv$$

$$\equiv A \vee \overline{B} \vee \overline{C} \wedge A \vee C \wedge \overline{A} \overset{7}{\equiv} A \vee \overline{B} \vee \overline{\overline{C} \wedge A} \vee \overline{C \wedge \overline{A}} \overset{7,10}{\equiv} A \vee \overline{B} \vee \overline{C \vee \overline{A}} \vee \overline{\overline{C} \vee A}$$

2. Логічне слідування. Аналіз міркувань

Формула ϕ називається ***логічним наслідком*** формул $\phi_1,.., \phi_n$ (позначається $\phi_1,...,\phi_m \vDash \phi$), якщо на всіх спільних інтерпретаціях змінних, на яких кожна з формул ϕ_1, є істинною, формула ϕ також є істинною.

Приклад 1.

Перевірити, чи формули $\phi = R \Rightarrow \overline{Q}$; $\psi = Q \Leftrightarrow R$ є логічним наслідком формул $\phi_1 = P \Rightarrow Q$, $\phi_2 = R \Leftrightarrow P$.

Розв'язання. Побудуємо таблиці істинності формул ϕ_1, ϕ_2 , φ та ψ.

P	Q	R	$\overline{Q}$	$\varphi_1 = P \Rightarrow \overline{Q}$	$\varphi_2 = R \Leftrightarrow P$	$\varphi = R \Rightarrow \overline{Q}$	$\psi = Q \Leftrightarrow R$
0	0	0	1	1	1	1	1
0	0	1	1	1	0	1	0
0	1	0	0	1	1	1	0
0	1	1	0	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	1	0
1	1	0	0	0	0	1	0
1	1	1	0	0	1	0	1

На трьох виділених у таблиці інтерпретаціях змінних (0, 0, 0), (0, 1, 0) та (1, 0, 1) обидві формули ϕ_1 та ϕ_2 є істинними. Формула φ також є істинною на цих інтерпретаціях. Тому формула φ є логічним наслідком формул ϕ_1 та ϕ_2 .

Теорема 1. Формула ϕ є логічним наслідком формул ϕ_1, …, ϕ_m тоді і тільки тоді, коли формула $\psi = \phi_1 \wedge \phi_2 \ldots \wedge \phi_m \Rightarrow \phi$ є загальнозначущою.

Теорема 2. Формула ϕ є логічним наслідком формул ϕ_1, …, ϕ_m тоді і тільки тоді, коли формула $\psi = \overline{\phi_1} \vee \overline{\phi_2} \ldots \vee \overline{\phi_m} \vee \phi$ є загальнозначущою.

Теорема 3. Формула ϕ є логічним наслідком формул ϕ_1, …, ϕ_m тоді і тільки тоді, коли формула $\psi = \phi_1 \wedge \phi_2 \ldots \wedge \phi_m \wedge \overline{\phi}$ є суперечливою.

Приклад 2. Проаналізувати міркування

1) сьогодні не сонячний день і холодніше, ніж учора;

2) ми підемо сьогодні купатися тоді і тільки тоді, якщо сьогодні сонячний день;
3) якщо ми не підемо сьогодні купатись, то будемо кататися на човні;
4) якщо ми будемо кататися на човні, то повернемося пізно ввечері.

Отже, ми повернемося пізно ввечері.

Розв'язання. Введемо атоми:

A – сьогодні сонячний день;

B – сьогодні холодніше, ніж учора;

C – ми підемо сьогодні купатися;

D – ми будемо кататися на човні;

E – ми повернемося пізно ввечері.

Тоді міркування можна записати у вигляді:

1) $\overline{A} \wedge B$
2) $A \Leftrightarrow C$
3) $\overline{C} \Rightarrow D$
4) $D \Rightarrow E$

E

Перевіримо, чи є висновок міркування логічним наслідком засновків, тобто чи $\overline{A} \wedge B, A \Leftrightarrow C, \overline{C} \Rightarrow D, D \Rightarrow E \vDash E$

Для цього розглянемо формулу $\psi = \overline{A} \wedge B \wedge (A \Leftrightarrow C) \wedge \overline{(C} \Rightarrow D) \wedge (D \Rightarrow E) \wedge \overline{E}$ і перевіримо її суперечливість.

$$\overline{A} \wedge B \wedge (A \Leftrightarrow C) \wedge (\overline{C} \Rightarrow D) \wedge (D \Rightarrow E) \wedge \overline{E} \equiv \overline{A} \wedge B \wedge (\overline{A} \wedge \overline{C} \vee A \wedge C) \wedge (C \vee D) \wedge$$
$$\wedge (\overline{D} \vee E) \wedge \overline{E} \equiv (\overline{A} \wedge B \wedge \overline{A} \wedge \overline{C} \vee \overline{A} \wedge B \wedge A \wedge C) \wedge (C \vee D) \wedge (\overline{D} \wedge \overline{E} \vee E \wedge \overline{E}) \equiv$$

$$\equiv \overline{A} \wedge B \wedge \overline{C} \wedge (C \vee D) \wedge \overline{D} \wedge \overline{E} \equiv \overline{A} \wedge B \wedge \overline{C} \wedge (C \wedge \overline{D} \vee D \wedge \overline{D}) \wedge \overline{E} \equiv \overline{A} \wedge B \wedge \overline{C} \wedge$$

$$\wedge (C \wedge \overline{D} \vee 0) \wedge \overline{E} \equiv \overline{A} \wedge B \wedge \overline{C} \wedge C \wedge \overline{D} \wedge \overline{E} \equiv \overline{A} \wedge B \wedge 0 \wedge \overline{D} \wedge \overline{E} \equiv 0.$$

Отже, формула є суперечливою. Тому міркування є вірним.

3. Кон'юнктивна нормальна форма (КНФ) та диз'юнктивна нормальна форма (ДНФ)

Літерал – це атом або заперечення атома. Прикладами літералів є $\overline{A}$, B, C. Літерал називають ***позитивним***, якщо він не має знака заперечення і ***негативним***, якщо він має знак заперечення.

Кажуть, що формулу f записано у ***кон'юнктивній нормальній формі (КНФ)***, якщо вона має вигляд $f = f_1 \wedge f_2 \ldots \wedge f_n$, $(n \geq 1)$і всі f_i (i=1,2,..,n) різні. Тут кожна з формул $f_1, f_2, \ldots, f_n$ є диз'юнкцією літералів, у якій всі атоми різні.

Приклад. Нехай p, q, r - атоми. Тоді $f = (p \vee \overline{q} \vee \overline{r}) \wedge (\overline{p} \vee q)$ – кон'юнктивна нормальна форма.

Кажуть, що формулу f записано у ***диз'юнктивній нормальній формі (ДНФ)***, якщо вона має вигляд $f = f_1 \vee f_2 ... \vee f_n$, $(n \geq 1)$і всі f_i (i=1,2,..,n) різні. Тут кожна з формул f_1, f_2,.., f_n є кон'юнкцією літералів, у якій всі атоми різні.

Приклад. Нехай p, q, r - атоми. Тоді $f = (p \wedge \overline{q} \wedge \overline{r}) \vee (\overline{p} \wedge q)$ – диз'юнктивна нормальна форма.

Довільну формулу можна перетворити в одну з нормальних форм застосуванням законів алгебри логіки. Для побудови нормальних форм необхідно виконати таку послідовність кроків еквівалентних перетворень:

1) Використати правила $f \to g = \overline{f} \vee g$ та $f \leftrightarrow g = (f \to g) \wedge (g \to f)$ для усунення $\to$, $\leftrightarrow$.

2) Використати закон подвійного заперечення та закони де Моргана.

3) Використати закони дистрибутивності.

Приклад. Побудувати диз'юнктивну нормальну форму формули $((p \vee \overline{g}) \to r) \wedge (\overline{r} \to s)$

Приклад. Побудувати кон'юнктивну нормальну форму формули $(p \wedge (g \to r)) \to s$

Розв'язування задач

Приклад 1. Виконати рівносильні перетворення наступної логічної функції, використовуючи правила булевої алгебри: $F(A,B,C)=A \cdot \overline{B}+A \cdot B$.

Розв'язання:

F(A, B, C) = A · ($\overline{B}$+B) (закон дистрибутивності)

F(A, B, C) = A · 1 (закон доповнення)

F(A, B, C) = A(закон поглинання)

Приклад 2. Спрощення виразу: $F(A, B, C) = \overline{A} \cdot B \cdot C + A \cdot B \cdot C$.

Розв'язання:

F(A, B, C) = ($\overline{A}$+A) · B · C (закон дистрибутивності)

F(A, B, C) = 1 · B · C (закон доповнення)

F(A, B, C) = 1

F(A, B, C)=B · C

Завдання. Спростити вираз з використанням закону де Моргана:

$F(A, B, C) = \overline{\overline{A} \cdot B}+C$

Завдання. Спростити вираз: F(A, B) = (A+B) · (A+$\overline{B}$)

Питання для самоконтролю

1. Яка мета виконання рівносильних перетворень логічних формул та чому формула, отримана в результаті застосування законів логіки, називається рівносильною початковій?

2. Сформулюйте формальну умову того, що формула B є логічним наслідком із формули A і як це поняття використовується для аналізу правильності міркувань?

3. Назвіть основні закони логіки, які використовуються для розкриття дужок та спрощення заперечень складних виразів (зокрема, закони дистрибутивності та де Моргана).

4. Яку структуру має логічна формула у диз'юнктивній нормальній формі (ДНФ) та з яких елементарних конструкцій вона складається?

5. У чому полягає принципова різниця у побудові кон'юнктивної нормальної форми (КНФ) порівняно з ДНФ?

Тема 9. Числення предикатів

Мета вивчення: ознайомити з поняттями числення предикатів; розглянути основні елементи та операції числення предикатів; навчити формалізовувати висловлювання з використанням предикатів та кванторів; навести приклади розв'язування завдань для закріплення матеріалу.

План

1. Основні поняття логіки предикатів.
2. Використання квантифікації.
3. Основні операції.
4. Закони логіки першого ступеня.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Основні поняття логіки предикатів

Недостатність засобів логіки висловлювань

У природній мові, з потреб формалізації якої виникла математична логіка, часто зустрічаються умовиводи, для опису яких засоби логіки висловлювань є недостатніми. Розглянемо класичну задачу, яка полягає у аналізі наступного міркування:

1) Усі люди смертні;

2) Сократ – людина;

Отже, Сократ смертний.

Уведемо атоми A – «усі люди смертні», B – «Сократ – людина», C – «Сократ – смертний». Тоді для перевірки правильності міркування треба переконатися, що має місце логічне слідування A, B⊨C . Формула C є логічним наслідком засновків A та B тоді і тільки тоді, коли формула $A \wedge B \wedge \overline{C}$ є суперечливою. Однак, остання формула не є суперечливою, оскільки вона приймає значення істинності «істина» на інтерпретації (1, 1, 0). Отже, використовуючи засоби логіки висловлювань, одержуємо, що міркування не є вірним. Отриманий результат суперечить «інтуїтивній» правильності міркування. Отримане протиріччя випливає з того, що при використанні логіки висловлювань жодним чином не враховано наявність у атомі *A* узагальнення «усі», а також відсутній зв'язок між першою гіпотезою та висновком міркування.

У зв'язку з подібними проблемами виникла необхідність удосконалити логіку висловлювань. Розширенням логіки висловлювань є логіка предикатів, у якій введено додаткові поняття «предикат» та «квантор».

Основні поняття логіки предикатів

У програмуванні є оператори у вигляді «Повторювати цикл доти, доки змінні x та y не стануть рівними, або припинити обчислення після 100 повторень». Позначивши через *i* лічильник *n.*

Існують також речення, які не являють собою висловлювання і містять змінні. Наприклад, *x+1=3.* Речення зі змінними не є висловлюваннями, але вони перетворюються на висловлювання, якщо надати їм певних значень. Зокрема, в моваовторень, умову закінчення циклу можна задати виразом: «$(x = y) \vee (i > 100)$». Тоді оператор циклу набере вигляду: Повторювати, якщо $\neg((x = y) \vee (i < 100))$.

Числення предикатів (логіка предикатів) є розширенням логіки висловлень і дозволяє аналізувати складніші висловлювання, що включають змінні та квантори. Воно є основою для формалізації математичних тверджень, програмування та автоматизації доказів.

Речення x > 3, x = y + 3, x + y = z містять змінні. Вони не істинні і не хибні доти, доки змінним не буде задано певних значень.

Першу змінну x називають ***предметом***, а другу – «більше 3» – ***предикатом***. Часто предикатом називають усе речення.

Логіка предикатів (логіка першого ступеня) передбачає, що до основних понять логіки додають нові поняття – предикатів. Для формулювання складних думок використовують атоми як основні неподільні елементи формул.

Для запису атомів першого ступеня використовують такі типи символів:

- сталі: імена об'єктів (2, 5, T, F, Дискретна математика, Марія);
- предметні змінні (x, y, z);

- предикатні символи (імена, які позначають предикати та записують великими буквами (P, Q) або словами (БІЛЬШЕ, ЛЮБИТЬ).

Приклад. x > 3

x – предметна змінна;

P – предикатний символ, який позначає відношення >3.

Твердження x > 3 можна записати P(x) або БІЛЬШЕ(x, 3).

Предикат, який містить n предметних змінних $x_1, x_2,.., x_n$, називають ***n-місним предикатом*** і записують $P(x_1, x_2,.., x_n)$.

Атом логіки першого ступеня має *вигляд* $P(a_1, a_2,.., a_n)$, де P – предикатний символ, а $a_1, a_2,.., a_n$ – предметні змінні або сталі.

Щойно *змінна* x набуває певного значення, предикат P(x) набуває значення T чи F і перетворюється на висловнювання.

Приклад. Нехай *предикат* Q(x,y) означає «x+y=2». Тоді Q(1,2) – хибне висловлювання, а Q(0,2) – істинне. Q(1,2)=F, Q(0,2)=T.

Приклад. Задамо речення» x любить y». Тоді істинне речення «Іван любить Марію» можна подати істинним висловлюванням ЛЮБИТЬ (Іван, Марія).

Приклад. Якщо БІЛЬШЕ(x, y) – предикат «x+y=2», то БІЛЬШЕ(5, 2) – істинне висловлювання, а БІЛЬШЕ(1, 3) - хибне.

2. Використання квантифікації

Нехай P(x) – предикат, D – задана предметна область, $x \in D$. Використовують два спеціальні символи ∀, ∃, які називають відповідко квантором загальності та квантором існування.

∀***x*** читають «для будь-якого x». Вираз $(\forall x)P(x)$ означає, що $P(x)$ істинний для всіх x з предметної області.

Вираз ∃x читають «існує x». Вираз $(\exists x)P(x)$ означає, що в області D існує принаймні одне x, таке, що $P(x)$ істинний.

Перехід від $P(x)$до $\forall xP(x)$ або $\exists xP(x)$ називають ***зв'язуванням предметної змінної x***, а саму змінну x – ***зв'язаною.*** Незв'язану змінну називають ***вільною***. Говорять, що у виразах $\forall xP(x)$, $\exists xP(x)$ предикат $P(x)$ є ***в області дії відповідного квантора.***

Приклад. Нехай предикат $P(x)$ відповідає реченню «$x \geq 1$», а предметна область складається з усіх дійсних чисел. Тоді висловлювання $\forall xP(x)$ фальшиве: $\forall xP(x)$=F. Якщо ж предметна область складається з натуральних чисел, то висловлювання $\forall xP(x)$ істинне: $\forall xP(x)$=T.

Правило побудови формули логіки першого ступеня:

- атом – це формула;

- якщо H і G – формули, то $\neg H,\ H \vee G, H \vee G,$ – формули;
- якщо H – формула, а x – вільна змінна у формулі H, то ∀xH та ∃xH формули.

3. Основні операції:

Заперечення (¬) – якщо предикат істинний, то його заперечення є хибним, і навпаки.

Наприклад: ¬P(x) – "x не має властивості P".

Кон'юнкція (∧) – логічне "і", яке істинне тоді, коли обидва предикати істинні.

Наприклад: P(x)∧Q(x) – "x має властивості P і Q".

Диз'юнкція (∨) – логічне "або", яке істинне, коли хоча б один із предикатів істинний.

Наприклад: P(x)∨Q(x) – "x має властивість P або Q".

Імплікація (→) – логічне "якщо... то".

Наприклад: P(x)→Q(x) – "якщо x має властивість P, то x має властивість Q".

Правила виведення (дозволяють будувати логічні доведення):

1. **Modus Ponens:** Якщо P істинне та P→Q істинне, то Q істинне.
2. **Modus Tollens:** Якщо ¬Q істинне та P→Q істинне, то ¬P істинне.
3. **Узагальнення:** Якщо P(x) істинне для довільного x, то ∀xP(x) істинне.
4. **Конкретизація:** Якщо ∀xP(x) істинне, то P(a) істинне для будь-якого конкретного a.

Приклад. Розглянемо висловлювання: *«Кожен студент групи вивчав дискретну математику».*

x – студент групи. Уведемо предикат C(x): «x вивчав дискретну математику»: $\forall x\ C(x)$

Приклад. Розглянемо висловлювання: *«Усі студенти вивчають дискретну математику».* Формально це можна записати як: ∀x(S(x)→D(x)), де S(x) означає «x є студентом», а D(x) означає «x вивчає дискретну математику». Тут використано квантор загальності та імплікацію.

Приклад. Розглянемо висловлювання: «*Існує студент, який не вивчає дискретну математику*».

Позначимо: *S(x):* «*x є студентом*», *D(x):* «*x вивчає дискретну математику*».

Формалізація*: ∃x(S(x)∧¬D(x)).*

Приклад. Позначимо висловлювання: «*x – просте число»* як *P(x)*, *«x – раціональне число»* як *Q(x)*, *«x – дійсне число»* як *R(x) та «x менше y»* як *МЕНШЕ(x, y).*

Розглянемо такі істинні твердження:

1. Кожне раціональне число дійсне . $\forall x(Q(x) \to R(x)$

2. Існує просте число. $\exists x P(x)$

3. Для кожного числа x існує y, таке, що x<y. $\forall x \exists y$ МЕНШЕ(x, y).

Приклад. Розглянемо висловлювання:

«Деякі студенти групи відвідали Париж».

M(x) – *«x відвідав Париж»*

Якщо предметна область складається з усіх студентів групи, то це речення можна записати: $\exists x\, M(x)$.

Якщо нас цікавлять інші особи, крім студентів групи, то вводимо ще предикат

P(x) – «x студент групи».

$\exists x\, (P(x) \wedge M(x))$

Приклад. Розглянемо висловлювання: *«Кожний студент групи відвідав Париж або Рим».*

P(x) – «x студент групи».

M(x) – «x відвідав Париж».

R(x) – «x відвідав Рим»

$\forall x(P(x) \to (M(x) \vee R(x))$

Приклад. Перекладемо на українську мову формулу $\forall x \forall y(((x > 0) \wedge (y <0))) \to (xy < 0)$.

Добуток додатного числа на від'ємне число є від'ємним числом.

Приклад. Перекладемо на українську мову формулу $\forall x(C(x) \vee \exists y(C(y) \wedge F(x, y)))$.

$C(x)$означає *«x має комп'ютер»*,

$F(x, y)$ – *«x і y друзі»,*

предметна область для x та y – усі студенти першого курсу.

Зміст формули: *«Кожний студент курсу має комп'ютер або має друга, у якого є комп'ютер».*

Приклад. Перекладемо на українську мову формулу $\exists x \forall y \forall z((F(x, y) \wedge F(x, z) \wedge (x \neq z)) \to \overline{F}(y, z))$.

$F(x, y)$ – x і y друзі,

предметна область для x, y, z – усі студенти університету.

Зміст формули: *«Є такий студент x, що для всіх студентів y та z, якщо x і y друзі та x і z друзі, то y і z не друзі».*

Приклад. Подамо речення «Якщо певний чоловік – один з батьків, то він - тато».

Предметна область кожної змінної – усі люди.

$C(x)$ – «x – особа чоловічої статі»

$P(x)$ – «x – один з батьків»

F(x, y) – «батько y»

$\forall x((C(x) \wedge P(x)) \rightarrow \exists y F(x, y)$.

Приклад. Розглянемо висловлювання: «*Сума двох додатних чисел – додатне число»*

$\forall x \forall y(((x > 0) \wedge (y > 0)) \rightarrow (x + y > 0)$

4. Закони логіки першого ступеня

Формула логіки предикатів називається ***загальнозначущою*** (тотожно істинною або тавтологією), якщо вона є істинною на всіх своїх інтерпретаціях. Той факт, що формула φ є загальнозначущою позначається $\vDash \varphi$.

Формула логіки предикатів називається ***суперечливою*** (тотожно хибною), якщо вона є хибною на всіх своїх інтерпретаціях.

Формула логіки предикатів називається ***виконуваною***, якщо вона є істинною принаймні на одній своїй інтерпретації.

Дві формули F_1 та F_2 називають логічно еквівалентними (рівносильними), якщо $F_1 \leftrightarrow F_2$ – тавтологія, тобто $\vDash F_1 \leftrightarrow F_2$.

Формула	**Пояснення**
$\neg(\forall x\, P(x)) \equiv \exists x\, (\neg P(x))$	Заперечення «для всіх» перетворюється на «існує» із запереченням предиката
$\neg(\exists x\, P(x)) \equiv \forall x\, (\neg P(x))$	Заперечення «існує» перетворюється на «для всіх» із запереченням предиката
$\forall x\ \forall y\, P(x,y) \equiv \forall y\ \forall x\, P(x,y)$	Порядок загальних кванторів можна міняти місцями
$\exists x\ \exists y\, P(x,y) \equiv \exists y\ \exists x\, P(x,y)$	Порядок існування також можна міняти
$\forall x\, (P(x) \wedge Q(x)) \equiv (\forall x\, P(x)) \wedge (\forall x\, Q(x))$	Загальний квантор розподіляється через «і»
$\exists x\, (P(x) \vee Q(x)) \equiv (\exists x\, P(x)) \vee (\exists x\, Q(x))$	Існування розподіляється через «або»
$\forall x\, (P(x) \vee Q(x)) \neq (\forall x\, P(x)) \vee (\forall x\, Q(x))$	«Для всіх» не розподіляється через «або»

$\exists x\,(P(x) \wedge Q(x)) \neq (\exists x\,P(x)) \wedge (\exists x\,Q(x))$	«Існує» не розподіляється через «і»
Всі закони логіки висловлень діють всередині предикатів	Наприклад: $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

Приклад. $\overline{\forall x P(x)} \equiv \exists x \overline{P(x)}$**.** Розглянемо заперечення речення «Кожний студент університету вивчає математичний аналіз». Заперечення: «Існує такий студент університету, який не вивчає математичного аналізу».

Питання для самоконтролю

1. Що таке предикат і в чому полягає його принципова відмінність від звичайного логічного висловлювання?
2. Назвіть два основні види кванторів та поясніть, що вони означають.
3. Які змінні у формулах числення предикатів називаються вільними, а які – зв’язаними?
4. Як визначаються та виконуються базові логічні операції (кон’юнкція, диз’юнкція, заперечення) над предикатами?
5. Сформулюйте закони заперечення кванторів (закони де Моргана для кванторів загальності та існування).

Змістовий модуль ІІ. КОМБІНАТОРИКА. ТЕОРІЯ ГРАФІВ, АВТОМАТИ ТА ФОРМАЛЬНІ МОВИ

Тема 10. Основні комбінаторні конфігурації та правила підрахунку

Мета вивчення: ознайомити з базовими правилами комбінаторики (правилами суми та добутку); сформувати поняття про основні комбінаторні сполуки без повторень (розміщення, перестановки, комбінації) та з повтореннями; навчити аналізувати прикладні задачі, обирати відповідну комбінаторну схему та обчислювати кількість можливих варіантів конфігурацій.

План

1. Правила суми та добутку в комбінаториці.
2. Комбінаторні сполуки без повторень: перестановки, розміщення, комбінації.
3. Комбінаторні сполуки з повтореннями.
4. Універсальний алгоритм вибору комбінаторної схеми для прикладних задач**.**

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Правила суми та добутку в комбінаториці

Комбінаторика – це розділ дискретної математики, який вивчає способи підрахунку кількості елементів у скінченних множинах, утворених за певними правилами. В основі всіх комбінаторних розрахунків лежать два фундаментальних правила.

Правило суми (принцип АБО)

Якщо об'єкт А можна вибрати m способами, а об'єкт В – n способами, причому вибори є взаємовиключними (не можна вибрати А і В одночасно), то вибрати або об'єкт А, або об'єкт В можна $m + n$ способами.

Приклад: на полиці стоять 5 підручників з програмування та 4 з баз даних. Скількома способами студент може вибрати ОДИН підручник для підготовки?

Розв'язання: За правилом суми: 5 + 4 = 9 способів.

У програмуванні правилу суми відповідає умовна конструкція розгалуження ***if-else***, де виконується або одна гілка коду, або інша.

Правило добутку (принцип І)

Якщо об'єкт А можна вибрати m способами, і після кожного такого вибору об'єкт В можна вибрати n способами, то пару об'єктів (А, В) у вказаному порядку можна вибрати $m \cdot n$ способами.

Приклад: студенту потрібно обрати один підручник з програмування (із 5 наявних) І один підручник з баз даних (із 4 наявних). Скільки існує варіантів такого вибору?

Розв'язання: За правилом добутку: $5 \cdot 4 = 20$ способів.

У програмуванні правило добутку асоціюється із вкладеними циклами (наприклад, два вкладених цикли ***for***), де загальна кількість ітерацій дорівнює добутку ітерацій зовнішнього та внутрішнього циклів.

2. Комбінаторні сполуки без повторень

Часто задачі вимагають вибору кількох елементів із загальної множини, де всі елементи унікальні, і повторювати їх не можна. Залежно від того, чи важливий для нас порядок елементів, виділяють три типи сполук.

Перестановки

Перестановками з n елементів називають такі сполуки, які складаються з усіх n елементів і відрізняються одна від одної лише порядком їхнього

розташування.

Формула для обчислення кількості перестановок:

$P_n = n!$

(де $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$, а за означенням 0! = 1)

Приклад: скількома способами можна розсадити 5 студентів на 5 місць у комп'ютерному класі?

Розв'язання: $P_5 = 5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$ способів.

Розміщення

Розміщеннями з n елементів по k ($k \leq n$) називають такі сполуки, що складаються з k елементів, вибраних із даних n елементів, і відрізняються між собою як складом елементів, так і їхнім порядком.

Формула для обчислення:

$A_n^k = \frac{n!}{(n-k)!}$

Приклад: у групі навчається 15 студентів. Скількома способами можна обрати серед них старосту та його заступника?

Розв'язання: порядок важливий (староста і заступник – різні посади), тому це розміщення:

$A_{12}^2 = \frac{12!}{(12-2)!} = n \cdot (n-1) \cdots (n-k+1)$ способів.

Комбінації

Комбінаціями з n елементів по k ($k \leq n$) називають такі сполуки, що складаються з k елементів, вибраних із даних n елементів, і відрізняються між собою лише складом (порядок розташування елементів значення не має).

Формула для обчислення:

$C_n^k = \frac{n!}{k!(n-k)!}$

Приклад: з 15 студентів групи потрібно вибрати трьох делегатів на студентську конференцію. Скільки варіантів вибору існує?

Розв'язання: порядок вибору не має значення (всі делегати рівноправні), тому це комбінації:

$C_{15}^3 = \frac{15!}{3!(15-3)!} = \frac{15 \cdot 14 \cdot 13}{3 \cdot 2 \cdot 1} = 455$ способів

3. Комбінаторні сполуки з повтореннями

У багатьох практичних задачах (особливо в ІТ: генерація паролів, кодування даних) елементи у вибірці можуть повторюватися.

Розміщення з повтореннями

Якщо вибираються k елементів з n типів, причому елементи можуть повторюватись і їхній порядок важливий, то кількість таких сполук обчислюється як:

$\overline{A}_n^k = n^k$

Приклад: скільки різних 4-значних PIN-кодів можна створити, використовуючи цифри від 0 до 9?

Розв'язання: оскільки цифри можуть повторюватись і порядок важливий, то n = 10 (всього цифр), k = 4 (довжина коду).

$\overline{A}_{10}^4 = 10^4 = 10000$ варіантів.

Перестановки з повтореннями

Якщо задано *n* елементів, серед яких перший елемент повторюється n_1 разів, другий – n_2 разів і т.д. (так що $n_1 + n_2 + ... + n_m = n$), то кількість різних перестановок цих елементів дорівнює:

$P_n(n_1, n_2, ..., n_m) = \frac{n!}{n_1! \cdot n_2! \cdot \cdots n_m!}$

Приклад: скільки різних "слів" (навіть без змісту) можна отримати, переставляючи літери у слові "АЛГОРИТМ"? А у слові "МАСИВ"?

Розв'язання:

1) У слові "АЛГОРИТМ" усі 8 літер унікальні: $P_8 = 8! = 40320$.

2) У слові "МАСИВ" літера 'М' (1), 'А' (1), 'С' (1), 'И' (1), 'В' (1) – теж унікальні, але якщо взяти слово "БАНАН" (Б-1, А-2, Н-2):

$P_5(1, 2, 2) = \frac{5!}{1! \cdot 2! \cdot 2!} = \frac{120}{4} = 30$.

Комбінації з повтореннями

Вибір *k* елементів з *n* типів, де порядок елементів не має значення, але один і той самий тип елементів може бути обраний кілька разів.

$\overline{C}_n^k = C_{n+k-1}^k = \frac{(n+k-1)!}{k!(n-1)!}$

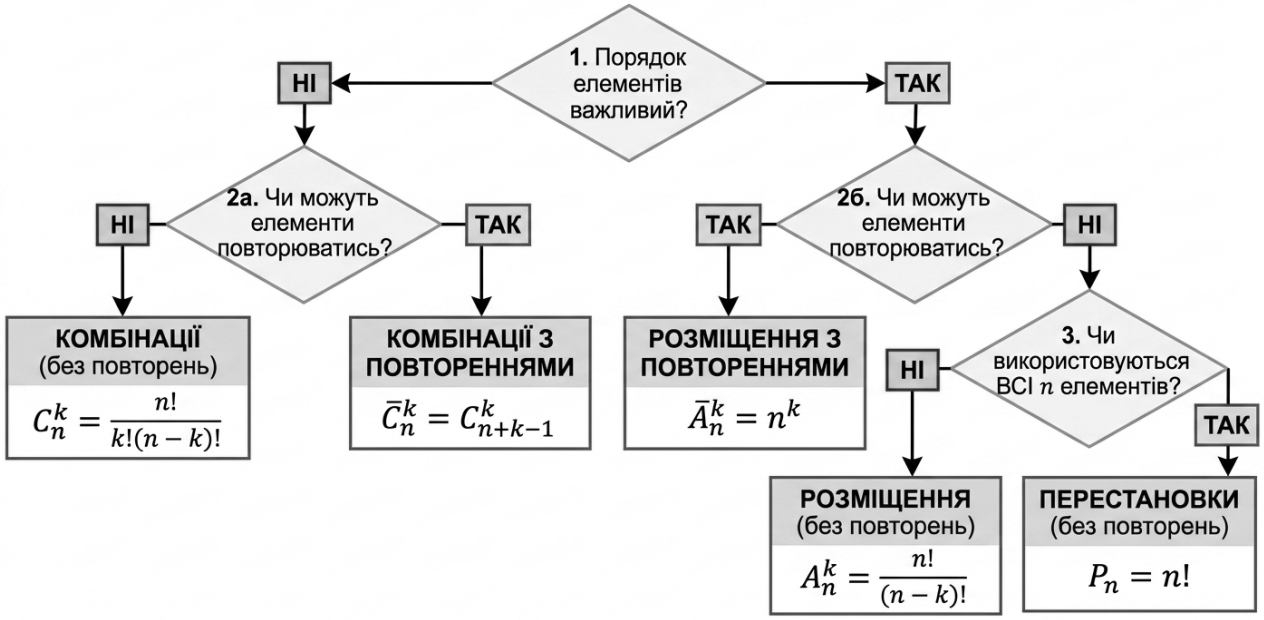
Приклад: в автоматі продаються тістечка 3 видів. Скількома способами покупець може придбати 5 тістечок?

Розв'язання: порядок неважливий, типи можуть повторюватись. n=3, k=5.

$\overline{C}_3^5 = C_{3+5-1}^5 = \frac{7!}{5!2!} = 21$ спосіб.

4. Універсальний алгоритм вибору комбінаторної схеми

Для успішного розв'язання будь-якої комбінаторної задачі достатньо відповісти на три послідовних запитання:



Такий структурний підхід дозволяє мінімізувати логічні помилки під час написання алгоритмів перебору та оптимізації програмного забезпечення.

Питання для самоконтролю

1. Сформулюйте правило суми та правило добутку. Наведіть приклади їх реалізації в мовах програмування.
2. Чим принципово відрізняються розміщення від комбінацій?
3. Що таке факторіал числа і чому дорівнює 0!?
4. Яку комбінаторну схему слід використати для підрахунку кількості можливих IP-адрес IPv4?
5. Скільки існує способів обрати старосту, профорга та секретаря в групі з 20 студентів? Яку формулу ви використаєте?

Тема 11. Спеціальні методи комбінаторики та біноміальна теорема

Мета вивчення: ознайомити з методами комбінаторного аналізу; вивчити формулу бінома Ньютона та властивості біноміальних коефіцієнтів, розглянути їхній зв’язок із трикутником Паскаля; сформувати поняття про принцип включень-виключень та принцип Діріхле (шухляд); розвинути вміння застосовувати ці методи для розв'язування складних комбінаторних задач.

План

1. Біном Ньютона. Формула та властивості біноміальних коефіцієнтів.
2. Трикутник Паскаля та його застосування.
3. Принцип включень-виключень для двох та більше множин.
4. Принцип Діріхле: проста та узагальнена форми.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Біном Ньютона. Формула та властивості біноміальних коефіцієнтів

Нам відомі формули скороченого множення $(a + b)^2 = a^2 + 2ab + b^2$ та $(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$. Біноміальна теорема узагальнює ці вирази для довільного натурального показника ступеня n.

Теорема (Біном Ньютона). Для будь-якого натурального числа *n* справедлива формула:

$$(a + b)^n = \sum_{k=0}^{n} C_n^k a^{n-k} b^k = C_n^0 a^n + C_n^1 a^{n-1} b + C_n^2 a^{n-2} b^2 + \ldots + C_n^n b^n$$

Числа $C_n^k = \frac{n!}{k!(n-k)!}$ називаються ***біноміальними коефіцієнтами.***

Приклад: Розкласти $(x - 2y)^4$.

Основні властивості біноміальних коефіцієнтів:

1. Симетрія: $C_n^k = C_n^{n-k}$ (кількість способів вибрати *k* предметів дорівнює кількості способів відкинути *n-k* предметів).

2. Рекурентне співвідношення: $C_{n+1}^k = C_n^{k-1} + C_n^k$ (лягає в основу побудови трикутника Паскаля).

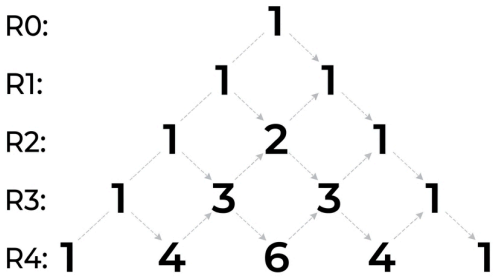
3. Сума коефіцієнтів: $\sum_{k=0}^{n} C_n^k = 2^n$(доводиться підстановкою *a=1, b=1*). Комбінаторний зміст: кількість усіх підмножин множини з *n* елементів.

4. Знакозмінна сума: $\sum_{k=0}^{n} (-1)^k C_n^k = 0$ (доводиться підстановкою *a=1, b=-1*).

2. Трикутник Паскаля

Для зручного знаходження біноміальних коефіцієнтів використовують таблицю, що має форму рівнобедреного трикутника, яку називають трикутником Паскаля.

Кожне число в ньому, починаючи з другого рядка, дорівнює сумі двох чисел, що стоять над ним у попередньому рядку (властивість 2).



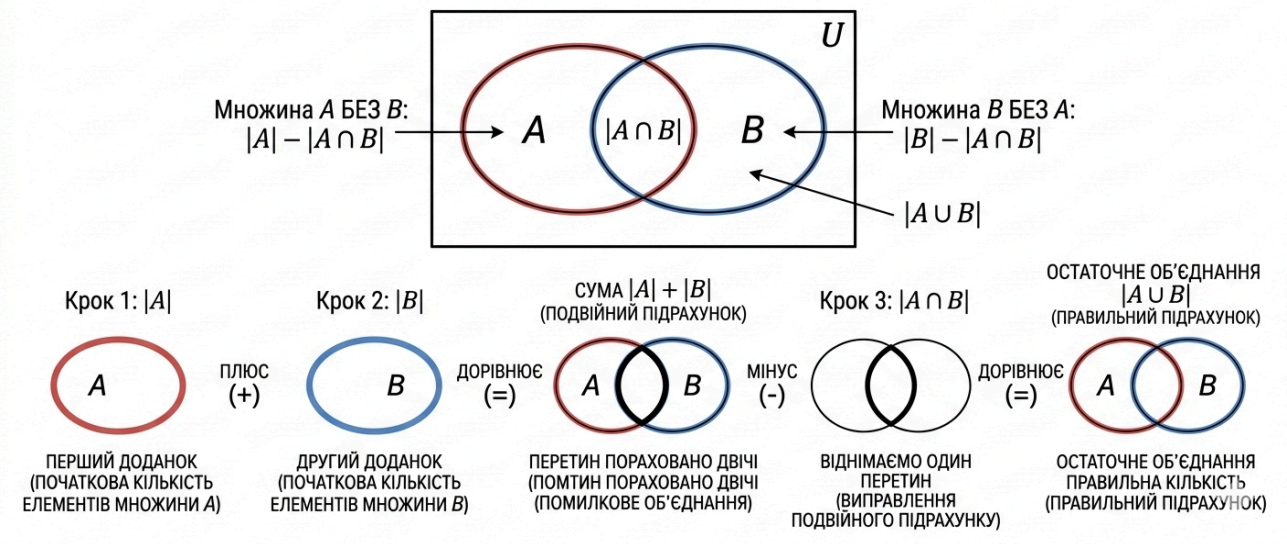
Приклад: Використовуючи трикутник Паскаля, записати розклад $(2x + 1)^5$.

3. Принцип включень-виключень

У комбінаториці ми часто рахуємо кількість елементів, які мають певну властивість. Якщо об'єкти можуть володіти кількома властивостями одночасно, просте додавання кількостей призводить до помилки – ми рахуємо деякі об'єкти двічі (або більше разів).

Для двох множин: Нехай A та B – скінченні множини. Тоді потужність (кількість елементів) їхнього об'єднання дорівнює:

$$|A \cup B| = |A| + |B| - |A \cap B|$$



Узагальнена форма (для n множин): Нехай A_1, A_2, ..., A_n – скінченні множини. Кількість елементів, які належать хоча б одній із цих множин:

$$|A_1, A_2, \ldots, A_n| = \sum_i |A_i| - \sum_{i<k} |A_i \cap A_j| + \sum_{i<j<k} |A_i \cap A_j \cap A_k|$$

Ми спочатку включаємо всі елементи множин, потім виключаємо ті, що були пораховані двічі (перетини по дві множини), включаємо ті, що були виключені занадто багато разів (перетини по три) і так далі.

Типовий приклад використання: задача про безлад (кількість перестановок без нерухомих точок) або підрахунок кількості цілих чисел в діапазоні, що не діляться на заданий набір чисел.

Приклад: Знайти кількість цілих чисел від 1 до 100, які не діляться ні на 2, ні на 3, ні на 5.

4. Принцип Діріхле (шухляд)

Цей метод часто використовується для доведення існування певних конфігурацій, навіть якщо ми не можемо вказати їх явно.

Проста форма: нехай n голубів розміщено у m голуб'ятнях (шухлядах). Якщо $n > m$, то знайдеться хоча б одна голуб'ятня, в якій знаходиться не менше ніж два голуби.

Узагальнена форма: якщо n предметів розміщено у m шухлядах, то знайдеться принаймні одна шухляда, що містить $[m/n]$ предметів (де $[x]$ – стеля числа x, найменше ціле, не менше за x.

Для застосування цього принципу головне – правильно визначити, що в даній задачі є «предметами» (голубами), а що – «шухлядами» (голуб'ятнями).

Приклад 1: у кімнаті перебуває 13 людей. Довести, що принаймні двоє з них народилися в один і той самий місяць.

Приклад 2: дано 12 цілих чисел. Довести, що серед них можна вибрати два числа, різниця яких ділиться на 11.

Питання для самоконтролю

1. Запишіть формулу бінома Ньютона для $(a - b)^n$.

2. Який комбінаторний зміст має сума біноміальних коефіцієнтів k-го рядка трикутника Паскаля?

3. Поясніть рекурентну властивість $C_{n+1}^k = C_n^{k-1} + C_n^k$.

4. Сформулюйте принцип включень-виключень для трьох множин.

5. У школі навчається 400 учнів. На основі принципу Діріхле, яке твердження про їхні дні народження можна гарантувати? (Вважати, що в році 366 днів).

6. В групі з 30 осіб 20 володіють англійською мовою, 15 – німецькою. Скільки осіб володіють принаймні обома мовами?

Тема 12. Основні поняття теорії графів

Мета вивчення: ознайомити з основними поняттями теорії графів, видами графів та прикладами зображення графів.

План

1. Основні поняття теорії графів.
2. Степінь вершини графа.
3. Деякі спеціальні класи простих графів.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Основні поняття теорії графів

Теорія графів – одна з істотних частин математичного апарату інформатики та кібернетики. У термінах теорії графів можна сформулювати багато задач, пов’язаних із дискретними об’єктами. Такі задачі виникають у проектуванні інтегральних схем і схем управління, у дослідженні автоматів, в економіці й статистиці, теорії розкладів і дискретній оптимізації.

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

Сім мостів Кеніґсберґа – видатна історична задача з математики. Доведення неможливості її розв'язання Леонардом Ейлером в 1735 призвело до створення теорії графів і передувало ідеї топології.

Місто Кеніґсберґ в Пруссії було на берегах річки Преголя, рукави якої ділили місто на чотири частини, в тому числі й два острови що поєднувалися сімома мостами.

Необхідно було знайти такий маршрут через місто, щоб пройти всі сім мостів і кожним мостом пройти рівно один раз. На острів не можна було потрапити інакше як через міст, і кожен з мостів мав бути пройденим за один раз (тобто не можна було пройти на середину мосту і повернутися назад, а потім з іншого берега пройти другу половину). Ейлер довів, що розв'язку не існує.

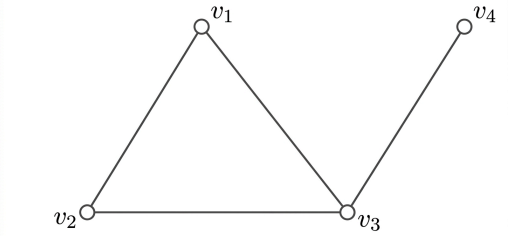
Граф – це сукупність вузлів (вершин), що з’єднані ребрами (дугами).

Графом G = (V, E) називають об’єкт, який заданий парою множин (V, E).

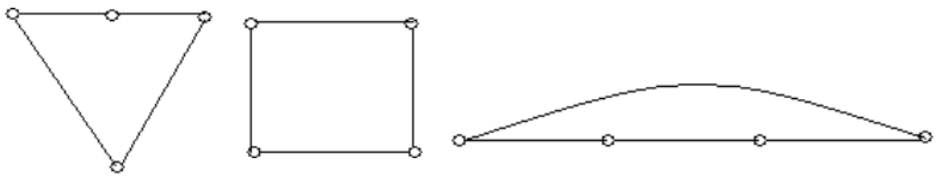
Множину вершин позначають V(G), а множину ребер – E(G).

Приклад. На рис. зображено *простий граф С* з множиною вершин V={v_1, v_2, v_3, v_4} і множиною ребер E= {{v_1, v_2}, {v_1, v_3}, {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.



Необхідно відзначити, що при зображенні графа не всі деталі рисунка мають значення. Так, наприклад, несуттєвою є довжина і кривизна ребер, взаємне розташування вершин на площині. Принциповим є тільки відношення інцидентності.

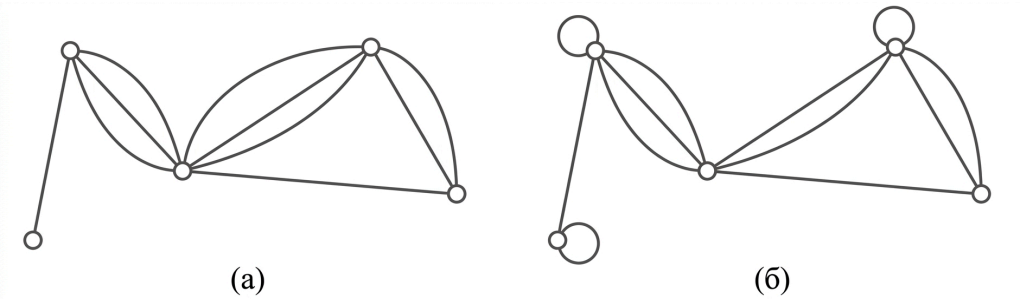


Моделі, зображені на рис. а, б, в, з погляду теорії графів однакові.

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, а E – сім'я невпорядкованих пар різних елементів множини V. Тут застосовано термін «сім'я» замість поняття «множина», бо елементи в E (ребра) можуть повторюватись. Ребра, що з'єднують одну й ту саму пару вершин, називають ***кратними*** (або паралельними). Окрім кратних ребер розглядають також ***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 рис. а зображено мультиграф і псевдограф (рис.б).



Розглянуті три типи графів називають неорієнтованими. Псевдограф – це найзагальніший тип неорієнтованого графа, бо він може містити петлі та 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.

Граф є ***зв'язним***, якщо існує шлях між будь-якими двома його вершинами.

Порядок графа – це кількість його вершин. Якщо граф G=(V,E), де V – множина вершин, а E – множина ребер, то порядок графа – це число $|V|$ (кардинальність множини вершин).

Приклади:

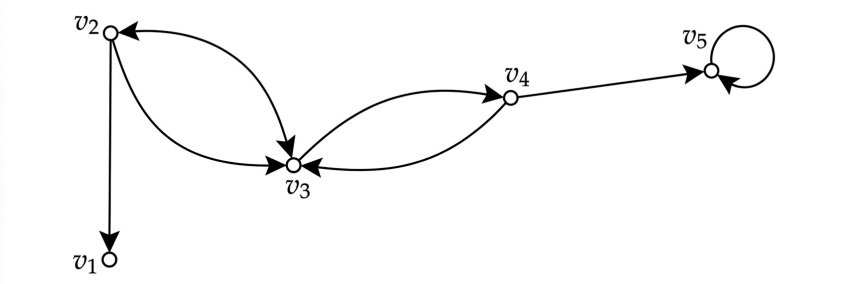
- Якщо в графі є 5 вершин, то його порядок дорівнює 5.
- У повному графі *Kn* порядок дорівнює *n*, оскільки він має *n* вершин.

Розмір графа – це кількість ребер (число $|E|$). Не слід плутати його з порядком графа.

У орієнтованих графах порядок також визначається як кількість вершин, незалежно від напрямків ребер.

Розглядають також орієнтовані графи. Якщо дуги мають напрямок, то такий граф називають ***орієнтованим*** (ортографом). ***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*** (чи орієнтованими ребрами). Дугу (v, v) називають ***петлею***.

Приклад. На рис. 3.3 зображено орієнтований граф із множиною вершин V={v_1, v_2, v_3, v_4, v_5} і множиною ребер 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)}.

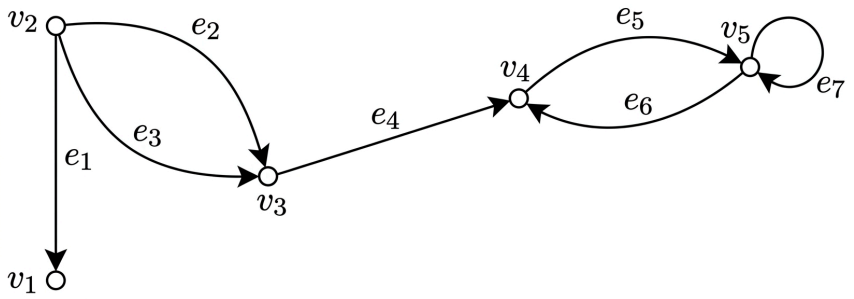


Зазначимо, що дута – це впорядкована пара вершин (записують у круглих дужках), тому в графі на рис. дуги (v_2, v_3) та (v_3, v_2) різні. На рисунках дуги позначають стрілками.

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

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

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



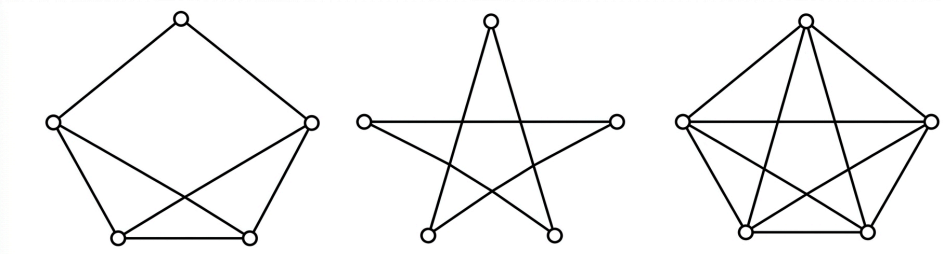
Надалі ми будемо використовувати термін «граф» для опису довільних графів – орієнтованих і неорієнтованих, із петлями та кратними ребрами чи без них, а термін «неорієнтований граф» або «псевдограф» – для довільного неорієнтованого графа, який може мати кратні ребра й петлі. Означення різних типів графів зведено в табл.:

Тип графа	***Ребра***	***Чи дозволені кратні ребра?***	***Чи дозволені петлі***
Простий граф	Неорієнтовані	Ні	Ні
Мультиграф	Неорієнтовані	Так	Ні
Псевдограф	Неорієнтовані	Так	Так
Орієнтований граф	Орієнтовані (дуги)	Ні	Так
Орієнтований мультиграф	Орієнтовані (дуга)	Так	Так

Дві вершини u та v в неорієнтованому графі G називають ***суміжними***, якщо існує ребро $\{u, v\} \in E$. 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.

Доповненням графа G називається граф G , що має ті ж вершини, що і граф G і тільки ті ребра, які необхідно додати до графа G , щоб він став повним.

Приклад. Доповненням графа G, зображеного на рис.а є граф G , зображений на рис. б. Для порівняння, повний граф зображений на рис. в.

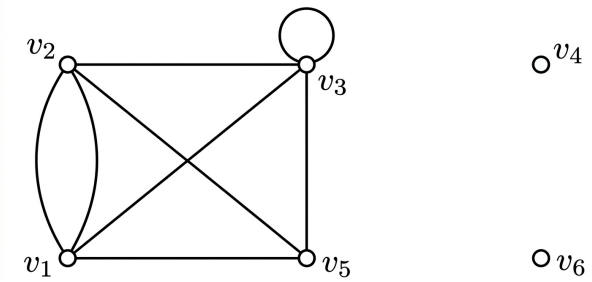


(а) (б) (в)

2. Степінь вершин графа

Степінь вершини в неорієнтованому графі – це кількість інцидентних їй ребер, 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. степені вершин такі: $deg(v_1)$ = 4, $deg(v_2)$ = 4, $deg(v_3)$=6, $deg(v_4)$ = l, $deg(v_5)$=3, $deg(v_6)$ = O. Отже, вершина v_6 – ізольована, а v_4 – висяча.



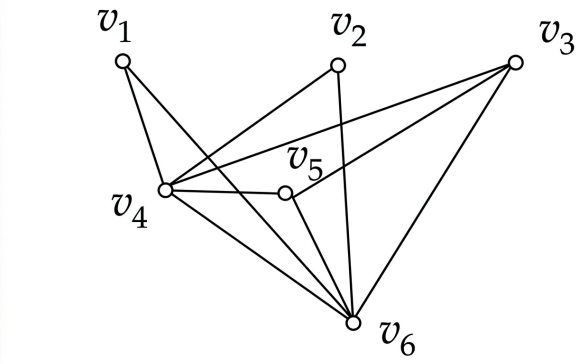
ТЕОРЕМА. Нехай *G=(V, E)* – неорієнтований граф з *m* ребрами.

Тоді $\sum_{v \in V} deg(v) = 2m.$

Це твердження стосується будь-якого неорієнтованого графа, зокрема з петлями та кратними ребрами. З цієї теореми випливає, що сума степенів усіх вершин неорієнтованого графа - парне число.

ТЕОРЕМА. Неорієнтований граф має парну кількість вершин непарного степеня.

Приклад. Визначити степені вершин графа, зображеного на рис.



deg v_1 = 2; degv_2 = 2; deg v_3 = 3; degv_4 = 4; deg v_5 = 3; degv_6 = 4 .

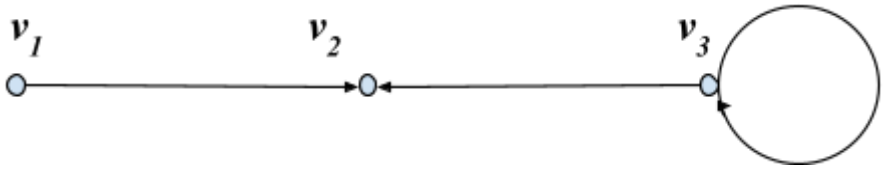
$\sum$deg v = 2 + 2 + 3 + 4 + 3 + 4 = 18 = 2 · 9 = 2m . У розглянутому графі дев'ять ребер, а вершин непарного степеня дві: v_3 ; v_5 .

Розглянемо орієнтований мультиграф *G=(V, E)*. Якщо *(u,v) є 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. Дуги орієнтованого графа називають ***суміжними***, якщо вони

мають спільну вершину. Вершину u називають інцидентною дузі v, якщо u – початкова чи кінцева вершина цієї дуги.

Для 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 $\deg^+(v)$.

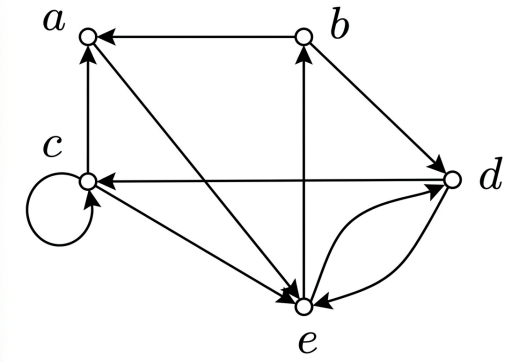
*Приклад***.** Для графа, зображеного на рис., напівстепені вершин такі: $\deg^-(v_1)=0$, $\deg^+(v_1)=1$. $\deg^-(v_2)=2$, $\deg^+(v_2)=0$, $\deg^-(v_3)=1$, $\deg^+(v_3)=2$.



ТЕОРЕМА. Нехай $G=(V, E)$ - орієнтований граф, який має m дуг. Тоді

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

Приклад. Визначити степені вершин орграфа, зображеного на рис.



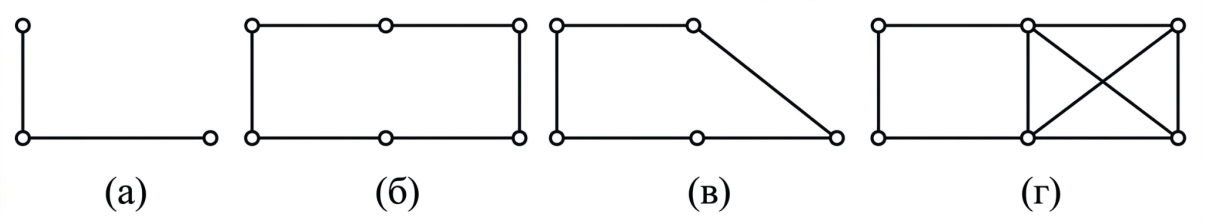
Розв'язання: on(a) = 1, on(b) = 2, on(c) = 3. on(d) = 2, on(e) = 2, in(a) = 2, in(b) = 1, in(c)= 2, in(d) = 2, in(e) = 3.

$$\sum_{v\in V} on(v) = 1 + 2 + 3 + 2 + 2 = 10 = \sum_{v\in V} in(v) = 2 + 1 + 2 + 2 + 3 = 10 = m.$$

Неорієнтований граф H називають ***підграфом*** неорієнтованого графа $G=(V, E)$, якщо всі вершини графа H належать V, а всі його ребра належать E.

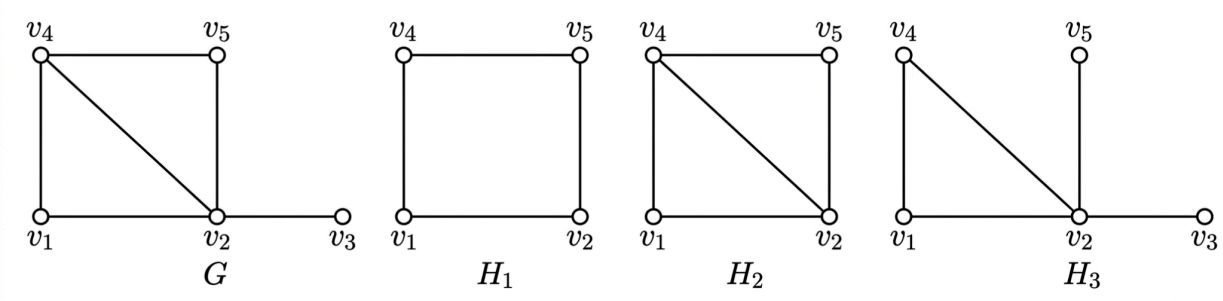
Граф G' називається ***підграфом*** графа G , якщо кожна вершина і кожне ребро графа G' є відповідно вершиною і ребром графа G .

Граф G' називається ***остовом (каркасом)*** графа G , якщо містить всі його вершини. Він також є ***підграфом*** графа G .

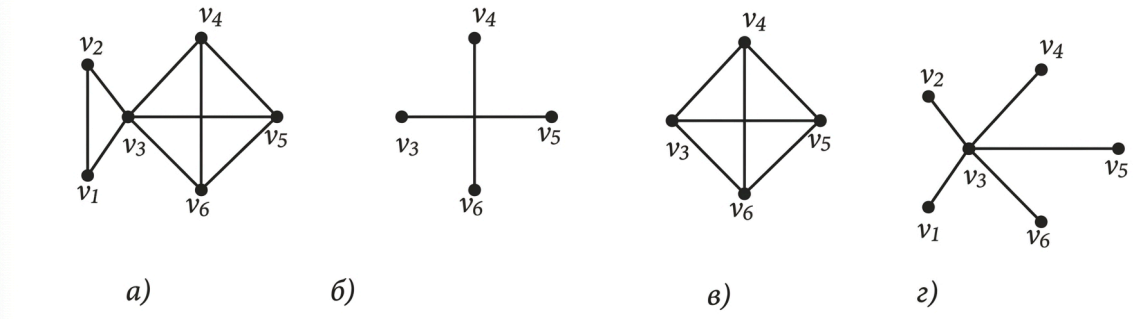


На рис. а, б, в зображені підграфи графа, зображеного на рис. г, причому підграф на рис. б є його каркас.

*Приклад***.** На рис зображено граф G та 3 його підграфа H_1, H_2, H_3,серед яких H_3 – каркасний.



Частина графа – це підграф, з якого, можливо, видалені деякі ребра. Наприклад, частина графа G на рис. б) містить вершини v_3, v_4, v_5, v_6, але не містить ребер (v_3, v_4), (v_4, v_5), (v_5, v_6), (v_3, v_6), в той час, як його підграф на рис. в) містить всі ребра, що з'єднують ці вершини.

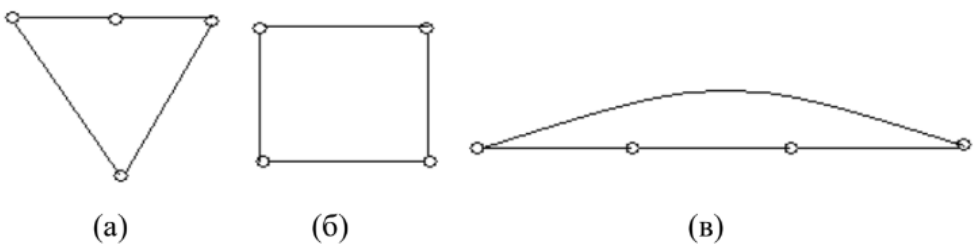


Частина графа, утворена вершиною v_i і всіма вершинами, суміжними з нею, називається ***зіркою*** вершини v_i.

Один і той же граф можна зображувати по-різному. Різним чином можна розташовувати вершини на площині і ребра можна зображувати не тільки відрізками прямих (різної довжини), але і дугами. Тому порівнюючи графи, будемо спиратися на наступні визначення.

Графи G_1 і G_2 ***рівні***, якщо множини їх вершин і ребер, визначених через пари інцидентних їм вершин, збігаються.

Наприклад, графи, зображені на рис.а – в рівні.

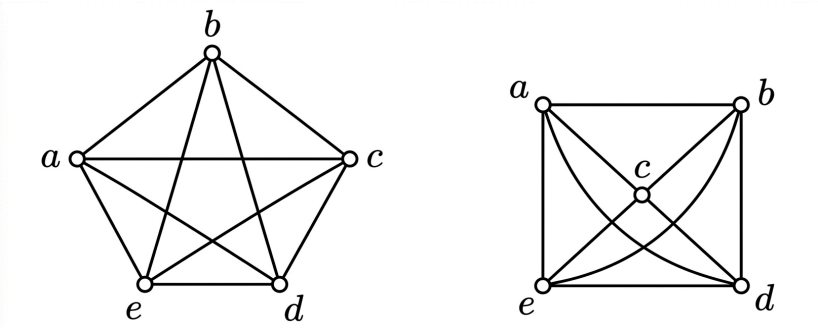


Задати граф означає описати множини його вершин і ребер, а також відношення інцидентності. Коли граф G – скінченний, для його опису досить пронумерувати вершини і ребра.

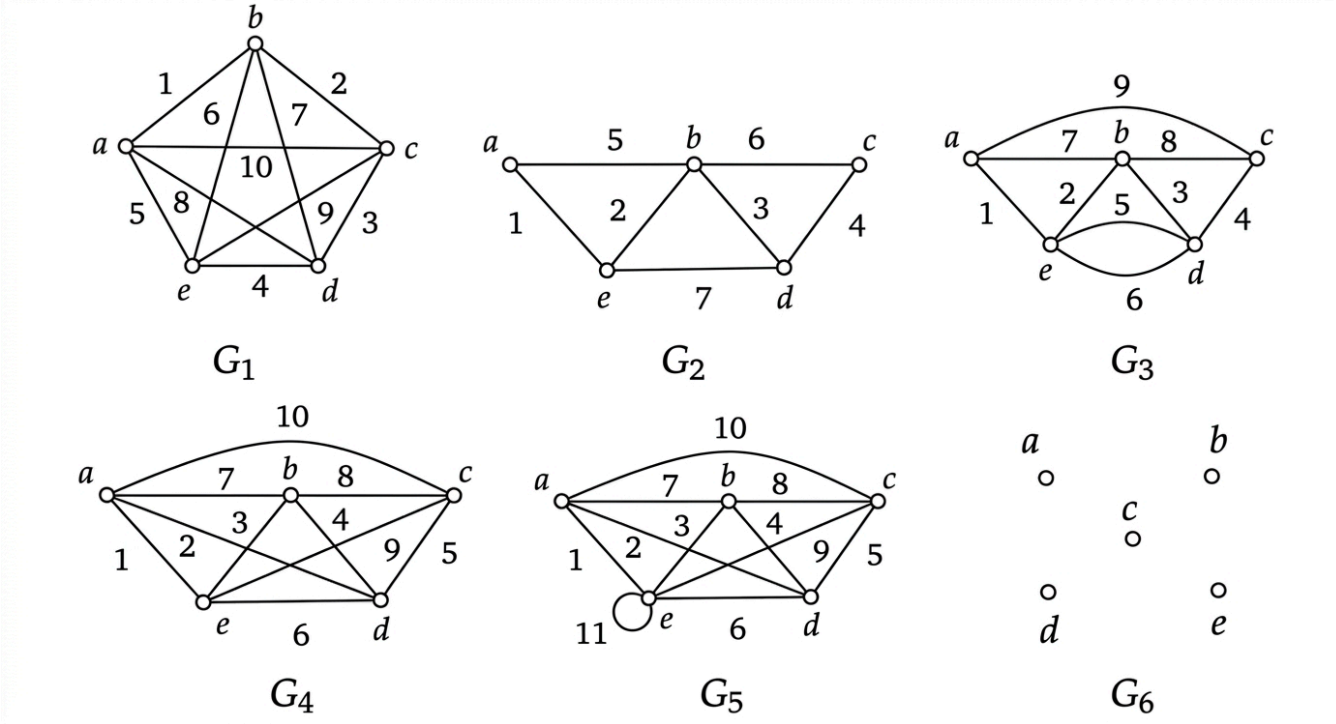
Граф G називається ***повністю заданим*** у точному значенні, якщо нумерація його вершин і ребер зафіксована. Графи, що відрізняються тільки нумерацією, називаються ***ізоморфними***.

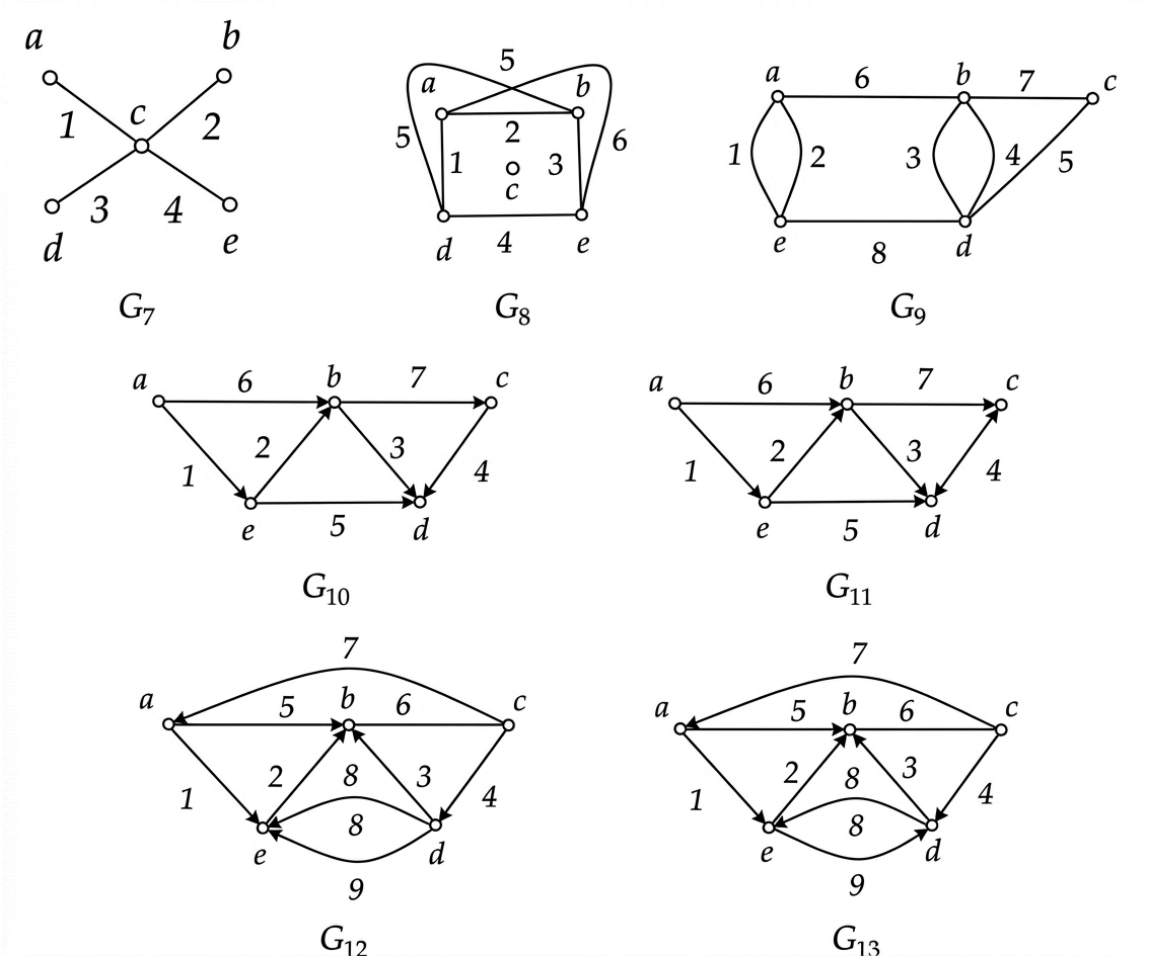
Таким чином, ізоморфні графи відрізняються фактично лише ідентифікаторами (іменами) своїх вершин. З точки зору теорії графів ця відмінність не є суттєвою, тому звичайно ізоморфні графи ототожнюють і, зображаючи графи у вигляді діаграм, або зовсім не ідентифікують їх вершини, або нумерують вершини натуральними числами.

Приклад. Графи, зображені на рис. (а), (б) – ізоморфні.



Приклад. На рис. зображені графи з 5 вершинами в кожному. Порівняти графи.





Розв'язання:

Графи $G_1 - G_9$ – неорієнтовані графи, а $G_{10} - G_{13}$ – орієнтовані.

Графи G_1 і G_4 повні, причому $G_1 = G_4$. Граф G_5 не є повним, бо незважаючи на те, що кожна пара вершин з'єднана ребром, є петля.

Графи G_3 і G_9 є мультиграфами, тому що містять кратні ребра.

Граф G_6 – має порожню множину ребер, всі вершини графа є ізольованими.

Графи G_7 і G_8 є доповненням один до одного.

Графи G_{10} і G_{11} не є рівними, тому що ребра 5 мають різний напрямок.

Граф G_{12} – орієнтовний мультиграф, тому що має кратні ребра, у той час як граф G_{13} не є мультиграфом, тому що ребра 8 і 9 по-різному орієнтовані.

Графи G_1 та G_2 ізоморфні тоді і тільки тоді, коли матрицю суміжності (матрицю інцидентності) одного з цих графів можна одержати з матриці суміжності (матриці інцидентності) іншого графа за допомогою відповідних перестановок рядків та стовпчиків.

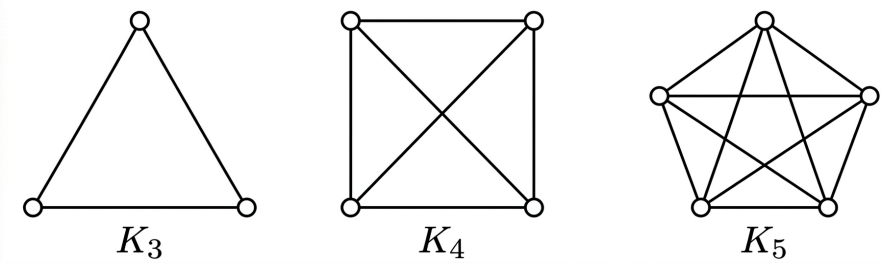
3. Деякі спеціальні класи простих графів

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

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

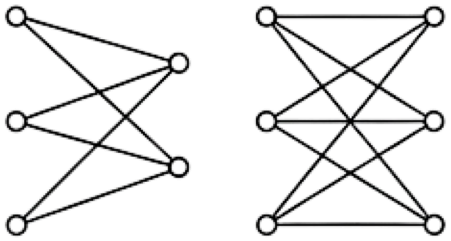
Кількість ребер у графі K_n дорівнює $C_n^2 = n(n-1)/2$.

*Приклад***.** На рис.зображено графи K_n для яких n = 3, 4, 5

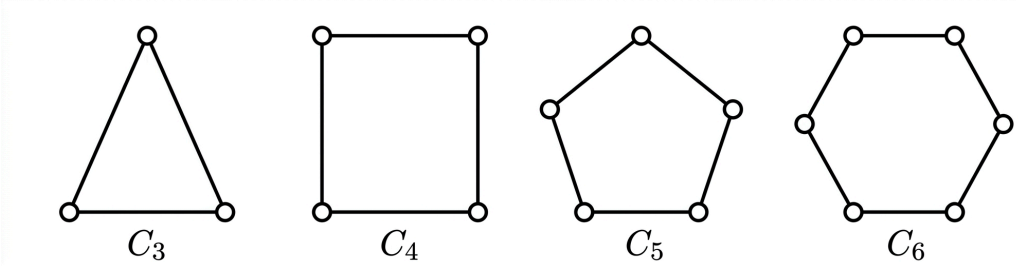


Граф називають ***порожнім***, якщо E=∅, тобто такий граф не мас ребер. Порожній граф з n вершинами позначають як O_n.

Простий граф *G=(V, E)* називають ***двочастковим***, якщо множину його V можна розбити на дві підмножини V_1, і V_2, що не перетинаються, так, що кожне ребро з'єднує вершину з V_1, і вершину з V_2. Двочастковий граф називають ***повним двочастковим графом***, якщо кожну вершину з V_1 з'єднано ребром із кожною вершиною з V_2.



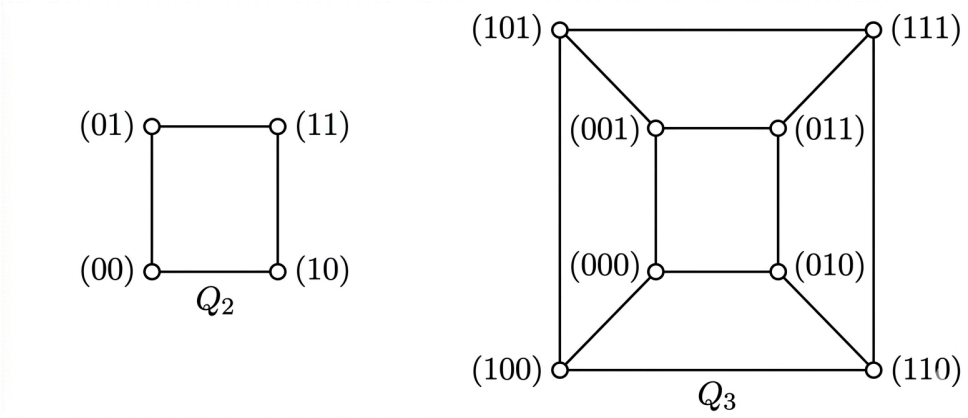
Приклад циклів



Приклади колеса (утворюється з циклу додаванням ще однієї вершини, яку з'єднують з усіма вершинами новими ребрами)

Простий граф, вершини якого відповідають усім 2^n бітовим рядкам довжиною n, називають ***n-вимірним кубом*** і позначають Q_n. Дві вершини в Q_n з'єднано ребром тоді дише тоді, коли бітові рядки, які їх подають, відрізняються точно в одному біті.

Приклад. На рис. зображено графи Q_1, Q_2 та Q_3



Питання для самоконтролю

1. Що таке простий граф і чим він відрізняється від мультиграфа та псевдографа?

2. Дайте означення степеня вершини графа. Яку вершину називають ізольованою, а яку – висячою?

3. Сформулюйте лему про рукостискання. Який важливий наслідок стосовно кількості вершин із непарними степенями з неї випливає?

4. Які графи називаються дводольними? Наведіть приклад застосування або зображення такого графа.

5. Опишіть структуру n-вимірного куба Q_n як спеціального класу простих графів. Чому дорівнює кількість його вершин та степінь кожної з них?

Тема 13. Способи визначення та типи графів

Мета вивчення: ознайомити з основними способами подання графів: рисунком, матрицею інцидентності, матрицею суміжності, списком пар (списком ребер); формувати вміння подавати графи з використанням різних способів.

План

1. Способи подання графів. Подання графа рисунком.
2. Матриця інцидентності.
3. Матриця суміжності.
4. Подання графа списком пар (списком ребер).

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

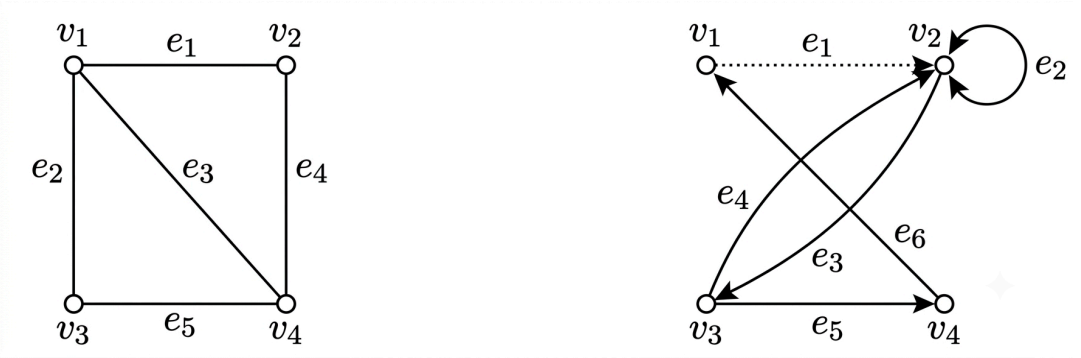
2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. 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 двох найважливіших видів графів: простого й орієнтованого:



Матрицю, кожний елемент якої дорівнює 0 чи 1, називають ***булевою***.

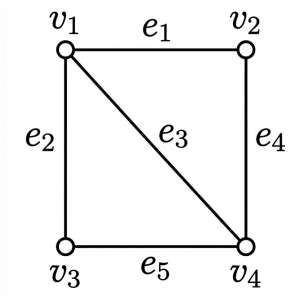
2. Матриця інцидентності

Нехай $G=(V, E)$ – простий граф із множиною вершин $V=\{v_1,..., v_n\}$ і множиною ребер $E= \{e_1, e_2, \ldots, e_m\}$.

Матрицею інцидентності графа G, яка відповідає заданій нумерації вершин і ребер, називають булеву $n \times m$ матрицю M з елементами $m_{ij}(i = 1, 2,..., n,\ j = 1, 2,..., m)$, де

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

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



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

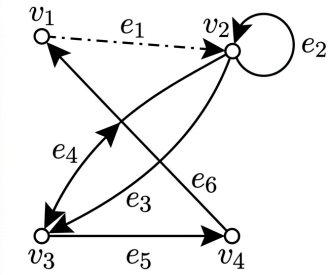
Отже, для простого графа в матриці інцидентності в кожному стовпці точно дві одиниці, і немає однакових стовпців. Матрицю інцидентності можна використовувати й для подання мультиграфа. Тоді з'являться однакові стовпці (вони відповідають кратним ребрам). Для подання псевдографа, якщо в ньому є петлі, у відповідних позиціях матриці ставимо 2 (у цьому разі матриця інцидентності не булева).

За допомогою матриці інцидентності можна подавати й орієнтовані графи. Для таких графів вона також не булева. Нехай *G=(V, E)* – орієнтований граф із множиною вершин V = {v_1,..., v_n} і множиною дуг E = {e_1, e_2, …, e_m}.

Матрицею інцидентності орієнтованого графа *G,* яка відповідає заданій нумерації вершин і дуг, називають *n*×*m* матрицю *M* з елементами $m_{ij}(i = 1, 2,..., n,\ j = 1, 2,..., m)$. Де

$$\varepsilon_{ij} = \begin{cases} -1, \text{ якщо } v_i - \text{початок ребра } e_j; \\ 1, \text{ якщо } v_i - \text{кінець ребра } e_j; \\ 0, \text{ якщо вони не інцидентні;} \\ \alpha, \text{ якщо } e_j - \text{петля, а } v_i - \text{інцидентна їй вершина.} \end{cases}$$

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



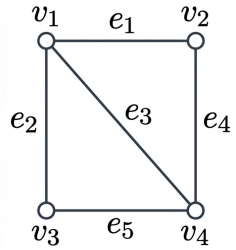
$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 2 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array}$$

3. Матриця суміжності

Нехай *G=(V, E)* – простий граф, |*V*| = n. Припустимо, що вершини графа *G* занумеровані: v_1,..., v_n. ***Матрицею суміжності графа G*** (яка відповідає даній нумерації вершин) називають булеву *n*×*n* матрицю *M* з елементами $a_{ij}(i, j = 1, 2,..., n)$, де

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

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



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

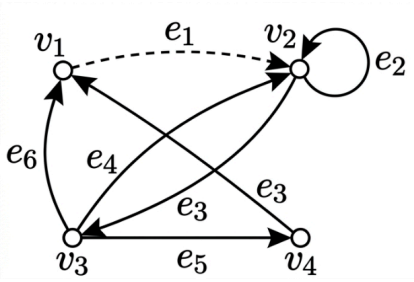
Цілком очевидно, що для неорієнтованого графа $a_{ij} = a_{ji}$, тобто матриця суміжності симетрична. Більше того, позаяк у простому графі немає петель, то для нього в матриці суміжності a_{ii} = 0 (i= 1, ..., n).

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.

Для подання орієнтованих графів також використовують матрицю суміжності. Це булева $n \times m$ матриця A з елементами a_{ij} = 0 (i= 1, ..., n), де

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

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



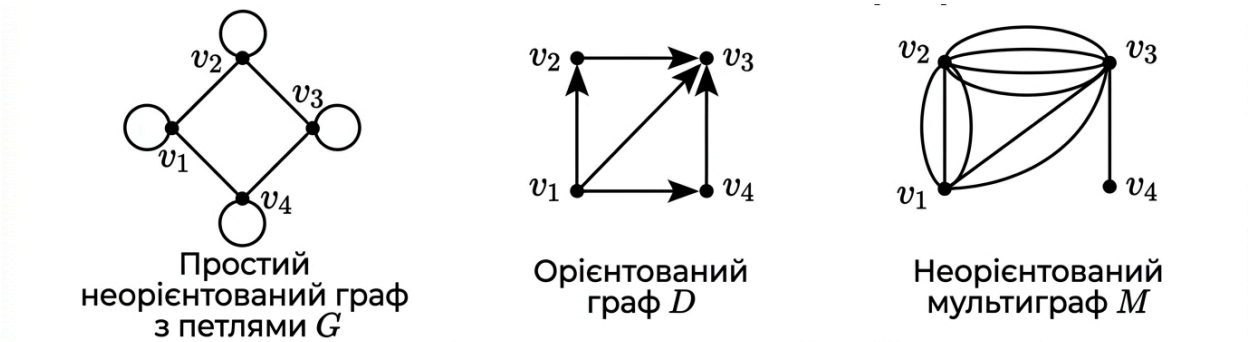
$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

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

Матрицю суміжності можна використовувати також 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.

Велика перевага матриці суміжності як способу подання графа – швидкий доступ до інформації: за один крок можна одержати відповідь на питання, чи існує ребро (дуга) з v_i у v_j. Вада полягає в тому, що незалежно від кількості ребер обсяг пам'яті становить n^2 комірок.

Приклад матриці суміжності для графів:



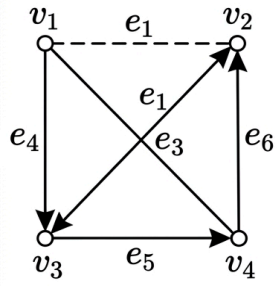
$$A(G) = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}, A(D) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, A(M) = \begin{pmatrix} 0 & 3 & 2 & 0 \\ 3 & 0 & 4 & 0 \\ 2 & 4 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Розглянемо ще спосіб подання графів у пам'яті комп'ютера. Будь-яку скінченну послідовність довільних елементів будемо називати списком, а кількість елементів списку – його ***довжиною***.

4. Подання графа списком пар (списком ребер)

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

Списки пар для графів:



Список ребер

Вершина 1	Вершина 2
v_1	v_2
v_1	v_3
v_1	v_4
v_2	v_4
v_3	v_4

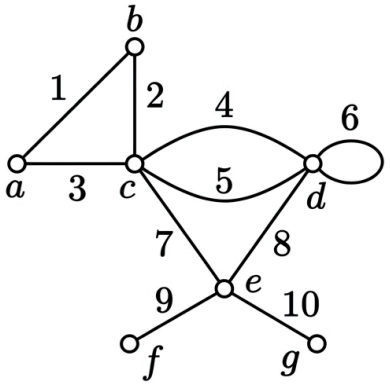
Матриця суміжності

	v_1	v_2	v_3	v_4
v_1	0	1	0	0
v_2	0	1	1	0
v_3	0	1	0	1
v_4	1	0	0	0

Список орієнтованих ребер

Від	До
v_1	v_2
v_2	v_2
v_2	v_3
v_3	v_2
v_3	v_4
v_4	v_1

Приклад. Задати матрицями інцидентності і суміжності, а також списком ребер неорієнтований граф, зображений на рис.



Розв'язання:

Матриця інцидентності

	1	2	3	4	5	6	7	8	9	10
a	1	0	1	0	0	0	0	0	0	0
b	1	1	0	0	0	0	0	0	0	0
c	0	1	1	1	1	0	1	0	0	0
d	0	0	0	1	1	2	0	1	0	0
e	0	0	0	0	0	0	1	1	1	1
f	0	0	0	0	0	0	0	0	1	0
g	0	0	0	0	0	0	0	0	0	1

Матриця суміжності

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	1	1	0	0	0	0
b	1	0	1	0	0	0	0
c	1	1	0	2	1	0	0
d	0	0	2	1	1	0	0
e	0	0	1	1	0	1	1
f	0	0	0	0	1	0	0
g	0	0	0	0	1	0	0

Список ребер

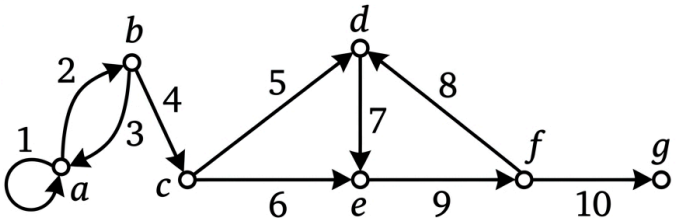
Ребро	1	2	3	4	5	6	7	8	9	10
початок	*a*	*b*	*a*	*c*	*c*	*d*	*c*	*d*	*e*	*e*
Вершини кінець	*b*	*c*	*c*	*d*	*d*	*d*	*e*	*e*	*f*	*g*

Як бачимо, у кожному стовпці матриці інцидентності є тільки два елементи, відмінних від нуля (або один, якщо ребро – петля).

Матриця суміжності симетрична щодо головної діагоналі. Список ребер є найбільш компактним способом задання графів.

Кожний із наведених способів однозначно описує граф, зображений на рис.

Приклад. Задати матрицями інцидентності, суміжності, списком ребер орієнтовний граф, зображений на рис.



Розв'язання:

Матриця інцидентності

	1	2	3	4	5	6	7	8	9	10
a	2	−1	1	0	0	0	0	0	0	0
b	0	1	−1	−1	0	0	0	0	0	0
c	0	0	0	1	−1	−1	0	0	0	0
d	0	0	0	0	1	0	−1	1	0	0
e	0	0	0	0	0	1	1	0	1	0
f	0	0	0	0	0	0	0	−1	−1	−1
g	0	0	0	0	0	0	0	0	0	1

Матриця суміжності

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	1	1	0	0	0	0	0
b	1	0	1	0	0	0	0
c	0	0	0	1	1	0	0
d	0	0	0	0	1	0	0
e	0	0	0	0	0	0	0

f	0	0	0	1	1	0	1
g	0	0	0	0	0	0	0

Ребро	1	2	3	4	5	6	7	8	9	10
початок	a	a	b	b	c	c	d	f	f	f
Вершини кінець	a	b	a	c	d	e	e	d	e	g

Питання для самоконтролю

1. Проаналізуйте фундаментальну відмінність між візуальним поданням графа (рисунком) та його математичними формами (матрицями, списками).

2. Детально опишіть структуру матриці суміжності для неорієнтованих, орієнтованих графів та графів з петлями.

3. Визначте та поясніть ключові відмінності в принципах опису структури графа матрицею суміжності та матрицею інцидентності.

4. Порівняйте ефективність використання списку ребер порівняно з матрицями для різних типів графів.

Тема 14. Операції над графами. Зв’язність графа. Маршрути, шляхи, ланцюги, цикли

Мета вивчення: ознайомити з основними операціями над графами, поняттям зв’язності графа, маршрута, шляху, ланцюга та циклу; формувати вміння знаходити об’єднання, перетин, доповнення і різницю графів; приклади: маршруту; ланцюга; простого ланцюга; циклічного маршруту; циклу; простого циклу.

План

1. Операції над графами.
2. Зв’язність графа. Маршрути, шляхи, ланцюги, цикли

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Операції над графами

Операція вилучення вершини v з графа $G=(V, E)$ полягає у вилученні з множини V елемента v, а з множини E − всіх ребер, інцидентних v.

Операція вилучення ребра e з графа $G=(V, E)$ − це вилучення елемента e з множини E. При цьому всі вершини зберігаються.

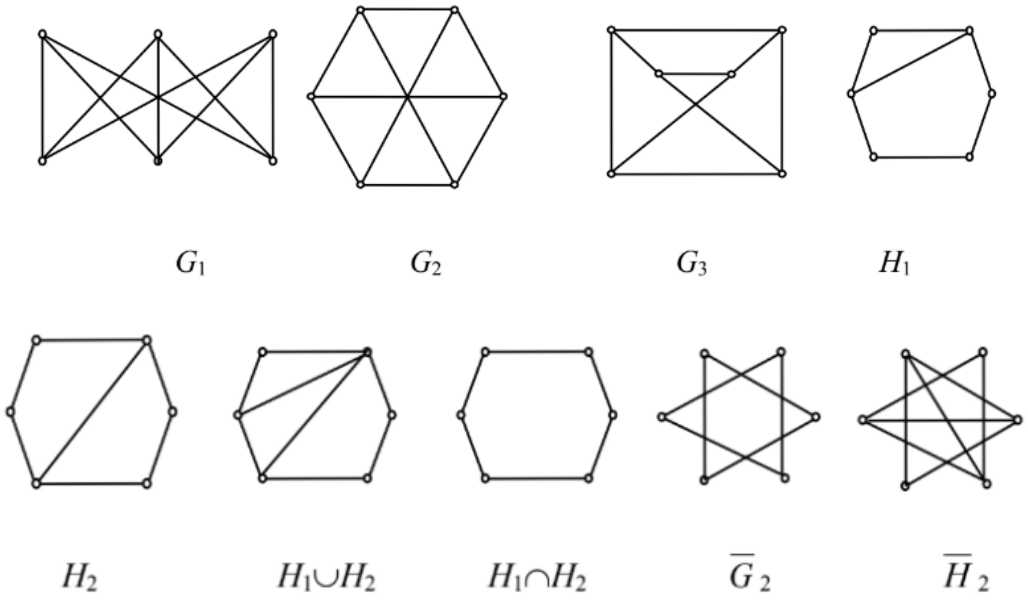
Для графів можна визначити операції об'єднання, *перетину* і різниці.

Об'єднанням графів $G_1=(V_1, E_1)$ і $G_2=(V_2, E_2)$ називається граф $G=(V_1 \cup V_2, E_1 \cup E_2)$; позначається $G = G_1 \cup G_2$. *Об'єднання* $G = G_1 \cup G_2$ називається ***прямою сумою*** графів G_1 і G_2, якщо $V_1 \cap V_2 = \varnothing$.

Перетином і різницею графів $G_1 = (V, E_1)$ і $G_2=(V, E_2)$ з однаковими множинами вершин називаються *графи* $G'= (V, E_1 \cap E_2)$ і $G''=(V, E_1 \setminus E_2)$ відповідно; позначаються $G'=G_1 \cap G_2$ *і* $G''= G_1 \setminus G_2$.

Графи G_1 і G_2 ізоморфні *тоді* і тільки тоді, коли ізоморфні їх доповнення $\overline{G}_1$ і $\overline{G}_2$.

Об'єднання і *перетин* графів H_1 і H_2 та доповнення графів G_2 і H_2 зображені на рис.



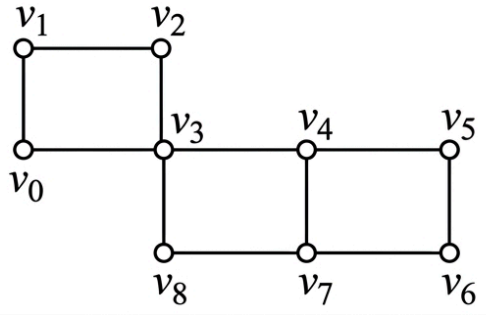
2. Зв'язність графа. Маршрути, шляхи, ланцюги, цикли

Маршрутом (шляхом) у графі G називається така послідовність ребер $M(e_1,e_2,...,e_n)$, у якій кожні два сусідніх ребра e_{i-1} і e_i мають спільну вершину. У маршруті те саме ребро може зустрічатися кілька разів. Іншими словами ***маршрут*** – це сукупність ребер, які об'єднані вершинами так, що можна рухатися по них уздовж графа. ***Початок маршруту*** – це вершина v_0 , інцидентна ребру e_1 і не інцидентна ребру e_2 . ***Кінець маршруту*** – це вершина v_n , інцидентна ребру e_n і не інцидентна e_{n-1}. Якщо ребра (e_1,e_2), ... (e_{n-1}, e_n) - кратні, то необхідно додатково вказувати, яку із двох інцидентних вершин вважати початком (кінцем) маршруту.

Ми будемо позначати маршрут як $v_0 v_1 v_2 .... v_{n-1} v_n$.

Маршрут довжини k – послідовність, що містить k ребер. Іншими словами, ***довжиною маршруту*** називається кількість ребер у ньому; при цьому кожне ребро враховується стільки разів, скільки разів воно зустрічається в маршруті. Маршрут, всі ребра якого різні, називається ***ланцюгом***. Ланцюг, який не має вершин, що повторюються, називається ***простим***.

Приклад. Визначити можливі маршрути (і їх довжину) з вершини v_0 в v_8 у графі, зображеному на рис.



Розв'язання: З вершини v_0 у v_8 ведуть, наприклад, шляхи:

1) $v_0v_3v_8$ – довжини *2;*

2) $v_0v_1v_2v_3v_8$ – довжини 4;

3) $v_0v_3v_4v_7v_8$ – довжини 4;

4) $v_0v_3v_4v_5v_6v_7v_8$ – довжини 6;

5) $v_0v_3v_4v_5v_4v_7v_8$ – довжини 6;

6) $v_0v_1v_2v_3v_4v_7v_8$ – довжини 6;

7) $v_0v_1v_2v_3v_4v_5v_6v_7v_8$ – довжини 8;

8) $v_0v_1v_2v_3v_4v_7v_6v_5v_4v_3v_8$ – довжини 10.

Шляхи 5), 8) не є простими.

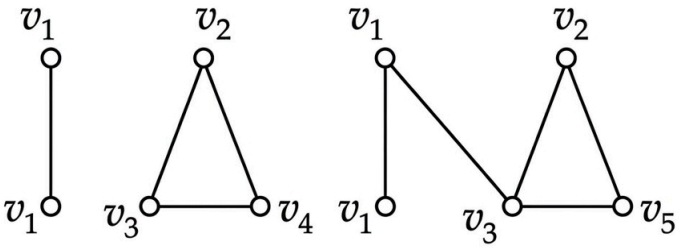
Маршрут, у якому збігаються початок і кінець, називається ***циклічним***. Циклічний маршрут називається ***циклом***, якщо він є ланцюг, і ***простим циклом*** – якщо це простий ланцюг.

Наприклад, маршрут $v_0v_1v_2v_3v_0$ для графа, зображеного на рис., є простим циклом; а маршрут $v_3v_4v_5v_6v_7v_4v_3$ є циклом, але не буде простим, тому що містить вершини, які повторюються.

Вершини v' і v'' графа G називаються ***зв'язаними***, якщо існує маршрут з початком у v' і кінцем у v'' . Маршрут між зв'язаними вершинами може бути поданий простим ланцюгом.

Граф G ***називається зв'язним***, якщо будь-які пари його вершин зв'язані між собою.

Наприклад, граф, зображений на рис. а – не зв'язний, а граф на рис. б – зв'язний.



(а) (б)

Твердження. Якщо існує маршрут з вершини v_0 в v_n графа G , то існує простий ланцюг, що з'єднує вершини v_0 і v_n .

Наслідок. Граф G є зв'язним тоді і тільки тоді, коли між будь-якими двома його вершинами існує простий ланцюг.

Максимальний непустий зв'язний підграф G' графа G називається ***компонентом*** графа G .

Отже, кожен граф є об'єднанням своїх компонент, які попарно не перетинаються. Незв’язний граф має, як мінімум, два компоненти. Наприклад, граф, зображений на рис. а, має два компоненти: v_1v_2 і $v_3v_4v_5$.

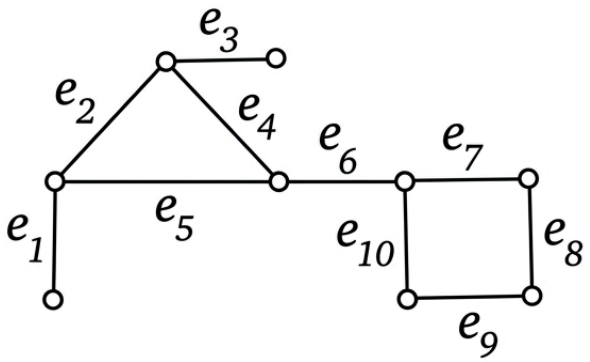
Вершина v називається ***точкою зчленування***, якщо видалення її із графа приводить до збільшення числа компонент зв’язності.

Ребро e називається ***мостом***, якщо видалення його із графа приводить до збільшення числа компонент зв’язності.

Наприклад, у графі, зображеному на рис. б вершина v_3 є точкою зчленування, а ребро, що з'єднує вершини v_2v_3 – міст.

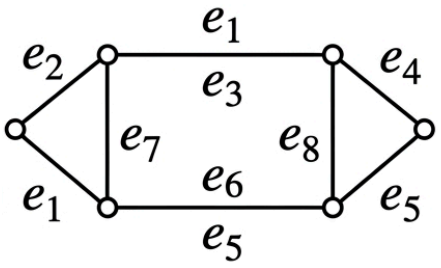
Множина ребер C зв'язного графа G називається ***множиною розрізу***, якщо видалення ребер із множини C порушує зв’язність графа, а видалення власної підмножини множини C залишає граф зв'язним. Якщо множина C складається з одного ребра, то це ребро називається ***ребром розрізу.***

Приклад. Визначити ребра розрізу графа, зображеного на рис.



Розв’язання: Ребра e_1, e_3 і e_6 – є ребрами розрізу. Їх видалення порушує зв’язність графа.

Приклад. Визначити множини розрізу для графа, зображеного на рис.



Розв'язання. Множинами розрізу для даного графа можуть бути, наприклад: $\{e_1,e_2\}$, $\{e_4,e_5\}$, $\{e_3,e_6\}$, $\{e_2,e_4,e_7,e_8\}$, $\{e_1,e_4,e_7,e_5\}$ і т.д.

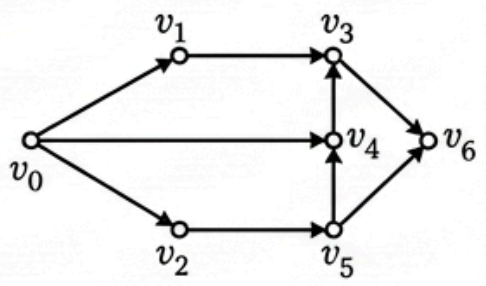
G – орієнтований граф

Послідовність ребер, у якому кінець кожного попереднього ребра e_{i-1} збігається з початком наступного e_i, називається ***шляхом*** або ***орієнтованим маршрутом***. У шляху те саме ребро може зустрічатися кілька разів.

Початком шляху є вершина v_0 ребра e_1 , **а *кінцем шляху*** є вершина v_n ребра e_n .

Довжиною орієнтованого маршруту називається кількість орієнтованих ребер, що входять у цей шлях.

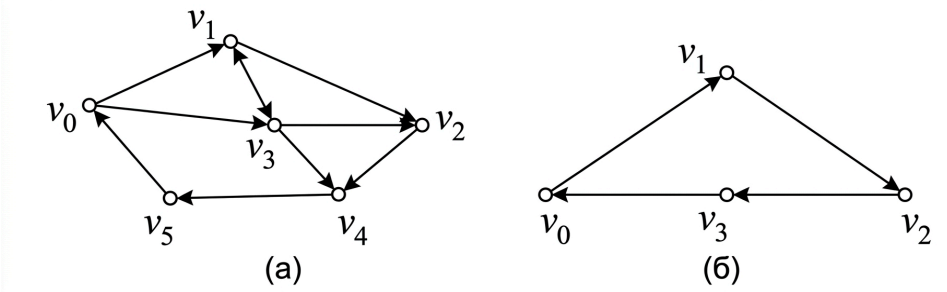
Приклад. Для графа, зображеного на рис., наведемо приклади орієнтованих маршрутів з вершини v_0 до вершини v_6 : $v_0\, v_1v_3v_6$; $v_0v_2v_5v_6$; $v_0v_4v_3v_6$ – довжини 3; $v_0v_2v_5v_4v_3v_6$ – довжини 5.



Орієнтованим ланцюгом називається шлях, кожне ребро якого зустрічається не більше одного разу, і ***простим ланцюгом***, якщо будь-яка вершина орграфа G інцидентна не більше, ніж двом його ребрам.

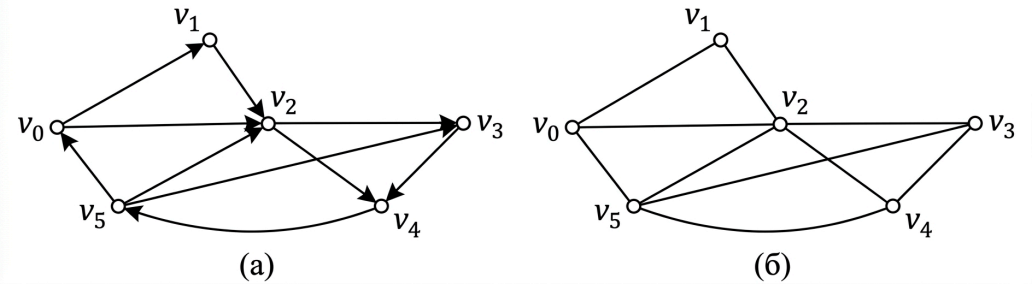
Контуром називається шлях, початок v_0 і кінець v_n якого збігаються. Контур називається *циклом*, якщо він є ланцюгом і *простим циклом* – якщо це простий ланцюг.

Для графа, зображеного на рис. а, приклади: орієнтованого ланцюга – $v_0v_1v_3v_2v_4v_5$; контуру – $v_0v_1v_3v_2v_1v_3v_4v_5v_0$; циклу – $v_0v_3v_4v_5v_0$. Для графа, зображеного на рис. б, приклади: простого орієнтованого ланцюга – $v_0\, v_1v_2$; простого циклу – $v_0v_1v_2v_3v_0$. При цьому зауважимо, що при записі циклу як для орієнтованого, так і для неорієнтованого графа, як початком, так і кінцем може бути обрана будь-яка вершина. Наприклад: $v_1v_2v_3v_0v_1$; $v_2v_3v_0v_1v_2$ і т.п.



Для кожного орієнтованого графа G може бути побудований неорієнтований граф G^s , такий, що всі вершини цих графів збігаються, а кожне ребро G (крім петель) стане неорієнтованим ребром графа G^s . У такому випадку граф G^s називається ***співвіднесеним графом*** орієнтованого графа G .

Приклад. Для графа G, зображеного на рис. а, співвіднесений граф G^s буде мати вигляд (рис.б):



Вершина $v'' \in G$ називається ***досяжною*** з вершини $v' \in G$, якщо існує шлях з початком у v' і кінцем у v'' .

Орієнтований граф G називається ***зв'язним***, якщо його співвіднесений граф G^s є зв'язним. Орієнтований граф називається ***сильно зв'язним***, якщо для будь-якої пари вершин v', v'' існує орієнтований шлях з v' у v'' .

Питання для самоконтролю

1. Які основні операції над графами дозволяють знаходити їхнє об'єднання, перетин, доповнення та різницю?
2. Що означає поняття зв’язності графа і як воно характеризує взаємозв'язок між його вершинами?
3. У чому полягає принципова відмінність між маршрутом, ланцюгом та простим ланцюгом у структурі графа?
4. Як розрізнити між собою циклічний маршрут, звичайний цикл та простий цикл під час аналізу графа?
5. Які ключові властивості відрізняють шлях у графі від інших видів послідовностей вершин і ребер?

Тема 15. Алгоритми з графами

Мета вивчення: ознайомити з поняттям ейлерового циклу та ейлерового графа, розглянути поняття циклу і шляху Гамільтона; сформувати поняття дерева, вміння будувати дерева та використовувати алгоритми пошуку остовного дерева вглиб і вшир в графі.

План

1. Ейлерів цикл. Ейлерів граф.
2. Шляхи і цикли Гамільтона.
3. Дерева і ліс.
4. Алгоритми пошуку остовного дерева вглиб і вшир в графі.

Література:

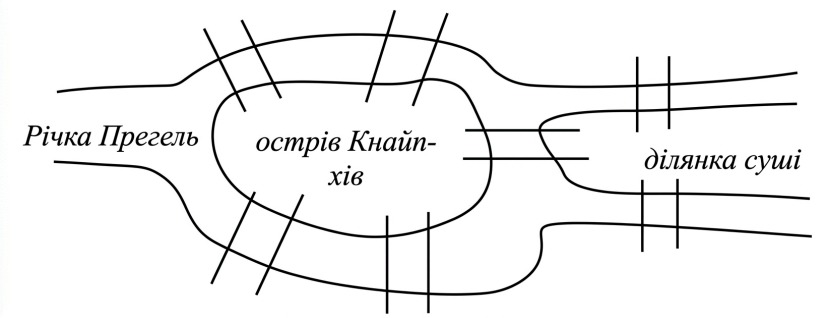
1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

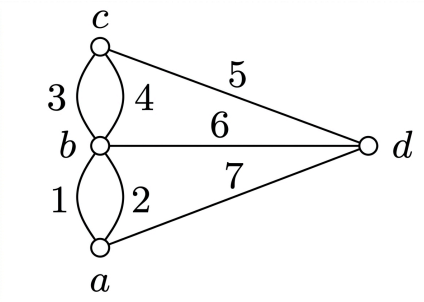
Текст лекції

1. Ейлерів цикл. Ейлерів граф

Теорія графів бере свій початок з розв’язання у 1736 році видатним математиком Ейлером задачі про кенігсбергські мости. Вона виникла в пруським містечку Кенігсберг на річці Прегель. Жителі Кенігсберга полюбляли гуляти по доріжках, які включають сім мостів через річку. Людей цікавило питання, чи можуть вони, почавши шлях з однієї ділянки суші, обійти всі мости по черзі за одну ходу і повернутися в початок шляху, не перепливаючи річку. План розташування семи мостів у Кенігсберзі наведений на рис.



Ейлер замінив план міста графом (рис.), у якому ділянки суші зобразив як вершини, а мости, що їх з'єднують – як ребра.



Для того, щоб відповісти на поставлене запитання, дамо ряд визначень.

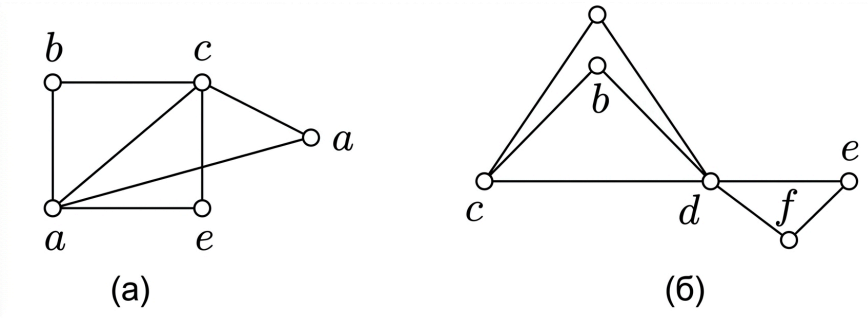
Ейлеровим циклом називається цикл, що містить всі ребра графа.

Ейлерів цикл можна вважати як слід олівця, що вимальовує граф, не відриваючись від паперу. Відшукання ейлерових циклів пов'язане з рядом прикладних задач. Наприклад, при перевірці дорожньої мережі необхідно по кожній дорозі досліджуваної мережі проїхати тільки один раз і повернутися у вихідний пункт. Очевидно, що такі цикли існують не в будь-якому графі.

Ейлеровим графом називається граф, що містить ейлерів цикл.

Теорема Ейлера. Неорієнтований граф ейлерів тоді і тільки тоді, коли він зв'язний і степінь всіх його вершин парний (при підрахунку степеня вершини, будь-яку інцидентну їй петлю вважати двічі).

Приклад. Чи мають графи, зображені на рис. (а, б), ейлерів цикл.



Розв’язання: Щоб відповісти на поставлене запитання, порахуємо степені вершин графа:

а) deg(a) = 4; deg(b) = 2 ; deg(c) = 4; deg(d) = 2; deg(e) = 2

Степені всіх вершин графа, зображеного на рис. а, парні, отже, граф має ейлерів цикл;

б) deg(a) = 2; deg(b) = 2 ; deg(c) = 3; deg(d) = 5; deg(e) = 2; deg(f) = 2.

Степені вершин c і d графа, зображеного на рис. б, непарні, отже, граф не має ейлерового циклу.

Що стосується кенігсбергських мостів, можна задати інше питання: “чи можливо пройти кожен міст по одному разі і не обов'язково повертатися у вихідну точку маршруту?” Відповідь на це питання потребує знання наступного визначення і теореми.

Шлях, що включає кожне ребро графа *G* тільки один раз, називається **е*йлеровим шляхом***. У тому випадку, якщо ейлерів шлях не є ейлеровим

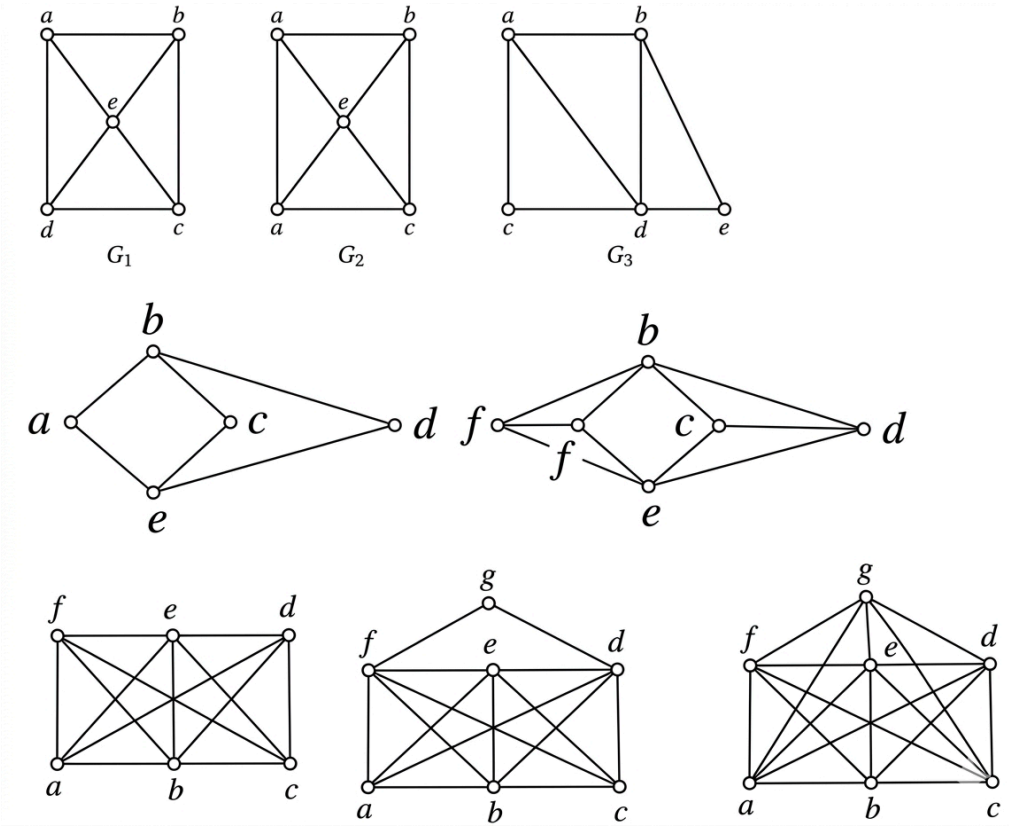
циклом, він називається ***власним ейлеровим шляхом,*** а граф G – ***напівейлеровим***.

Теорема. Граф G має власний ейлерів шлях тоді і тільки тоді, коли він зв'язний і рівно дві його вершини мають непарний степінь.

Оскільки граф для задачі про кенігсбергські мости має чотири вершини з непарними степенями, то можна зробити висновок про те, що даний граф не має власного ейлерового шляху, а отже, неможливо пройти кожен міст один раз, навіть якщо не потрібно повертатися у вихідну точку маршруту.

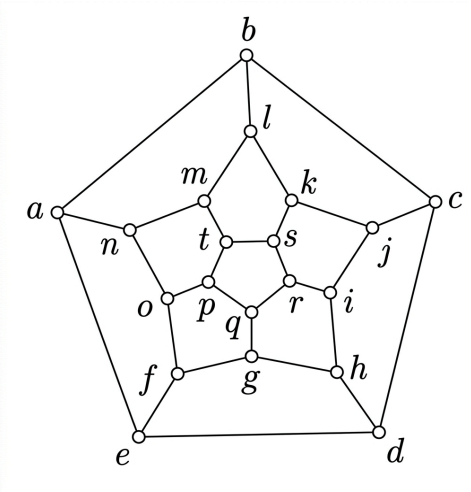
Так, наприклад, граф, зображений на рис. б, не має ейлерового циклу, але має власний ейлерів шлях, тому що дві його вершини мають непарний степінь.

Приклад. Серед наведених графів знайти ті, які мають ейлерів цикл, власний ейлерів шлях.



2. Шляхи і цикли Гамільтона

В 1859 році ірландський математик У. Гамільтон придумав іграшку-головоломку. Ця іграшка являла собою додекаедр – правильний многогранник, 12 граней якого - це правильні п'ятикутники. У кожному з 20 кутів просвердлювалась дірка, у яку вставляли кілочок, що зображував місто. За допомогою мотузки було потрібно знайти шлях через міста, відвідати кожне місто один раз і повернутися у вихідне місто. Додекаедр на площині зображується так, як показано на рис.

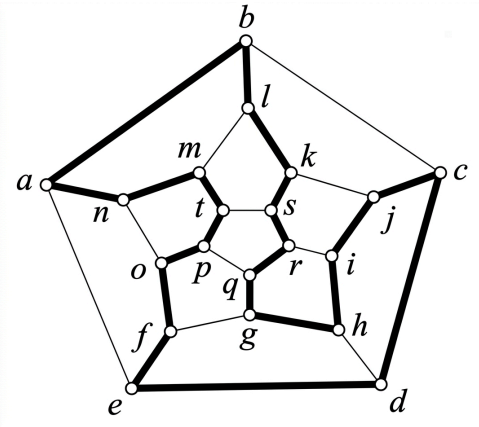


Задача головоломки зводиться до знаходження циклу в графі, що проходить через кожну вершину, крім початкової, тільки один раз.

Шляхом Гамільтона (або гамільтоновим ланцюгом) називається простий ланцюг, що проходить через всі вершини графа, з початком і кінцем у різних вершинах v', $v'' \in G$.

Цикл Гамільтона – це простий цикл, що проходить через всі вершини графа G .

Гамільтонів цикл протилежний ейлеровому циклу, що проходить через всі ребра один раз, хоча до певного моменту обидва цикли можуть здаватися схожими. Цикл Гамільтона виявляється набагато складніше і для його знаходження поки немає ефективних алгоритмів, що вимагають істотно меншого часу, ніж пряме перебирання варіантів. Проте приведемо ряд теорем, що дозволяють нам судити про можливість відшукати гамільтонів цикл у досліджуваному графі. А для початку покажемо один з варіантів розв'язання головоломки, запропонованої Гамільтоном (рис.).



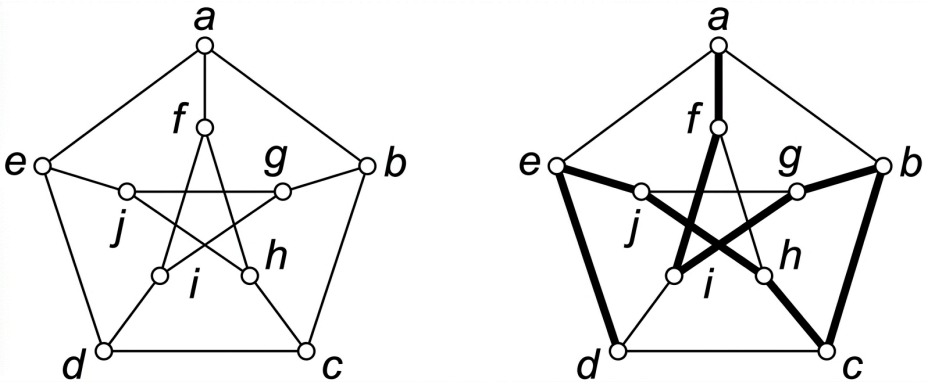
Теорема. Для будь-якої вершини із циклу Гамільтона існує рівно два ребра із цього циклу, інцидентні даній вершині.

Граф, що має цикл Гамільтона, називається ***гамільтонів.***

Виходячи з наведеного визначення, як наслідок теореми, робимо висновок про те, що будь-який граф, що має вершину степені 1 не є гамільтонів.

Помітимо також, що для того, щоб граф мав цикл Гамільтона, необхідно, щоб він був зв'язним.

Приклад. Граф Петерсена, зображений на рис., а, має шлях Гамільтона (рис., б), але не має циклу Гамільтона.

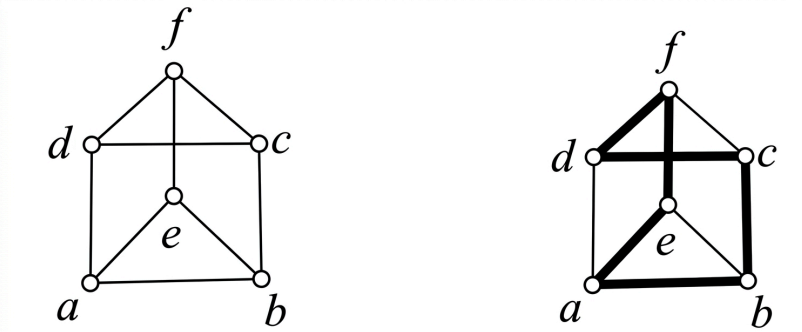


Теорема. Якщо граф G має ребро розрізу, то він не може мати цикл Гамільтона.

Теорема. Якщо G - зв'язний граф з n вершинами (n > 3) і для кожної пари несуміжних вершин v', v" є V сума степенів вершин задовольняє умові degv' + degv" > n, тоді граф G має цикл Гамільтона.

Наслідок (теорема Дірака). Якщо G – зв'язний граф з n вершинами (n > 3) і для кожної вершини v є V виконується умова deg v > n/2, то граф G має цикл Гамільтона.

Приклад. Знайдіть цикл Гамільтона, якщо він існує, для графа G, зображеного на рис.а.



(а) (б)

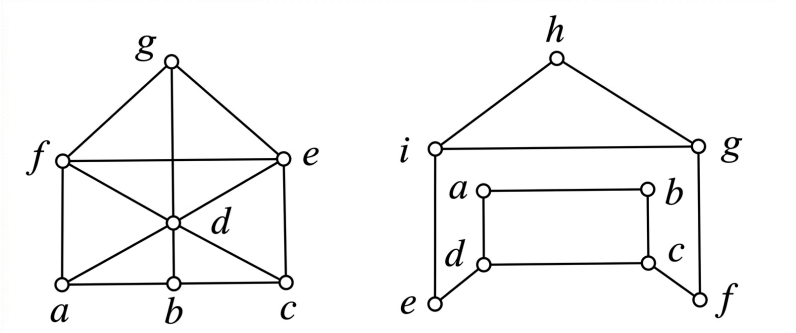
Розв'язання: Граф *G* - зв'язний, кількість вершин графа n = 6. Степінь кожної з вершин дорівнює 3, тобто кожна з вершин графа задовольняє умові deg $v \geq \frac{n}{2}$. Отже, даний граф є гамільтонів, тобто існує цикл Гамільтона. Знайти його можемо тільки методом перебору. Результат прямого перебору – цикл *baefdcb* (рис. б).

Практичне застосування циклів Гамільтона знаходимо в розв'язанні класичної задачі комівояжера, яка цікава для людей, у чиє коло обов'язків входить знаходження оптимальних шляхів, наприклад, об'їзду філій фірми, транспортування вантажів, доставки товарів і інше.

Задача комівояжера. Комівояжер повинен відвідати кілька міст і повернутися у вихідний пункт. Відстані між містами відомі. Потрібно знайти дорогу найкоротшої довжини. При такій постановці задачі схема доріг - це граф, у якому кожному ребру v запропонована певна довжина d(v). Задача комівояжера зводиться до знаходження в отриманому графі із заданими довжинами ребер циклу Гамільтона C мінімальної довжини.

Існує ряд алгоритмів досить громіздких, що дозволяють знаходити найкоротший шлях від вершини v' до вершини v'', таких як алгоритми Дейкстри, Флойда-Уоршолла, метод гілок і меж, метод генетичних алгоритмів, а так само алгоритм мурашиної колонії і т.п. Але ці алгоритми не є ефективними для графів великих об'ємів і тому часто використовують наближені алгоритми, що використовують евристики.

*Приклад.*Знайдіть цикл (шлях) Гамільтона, якщо він існує, для кожного з наведених на графів рис.

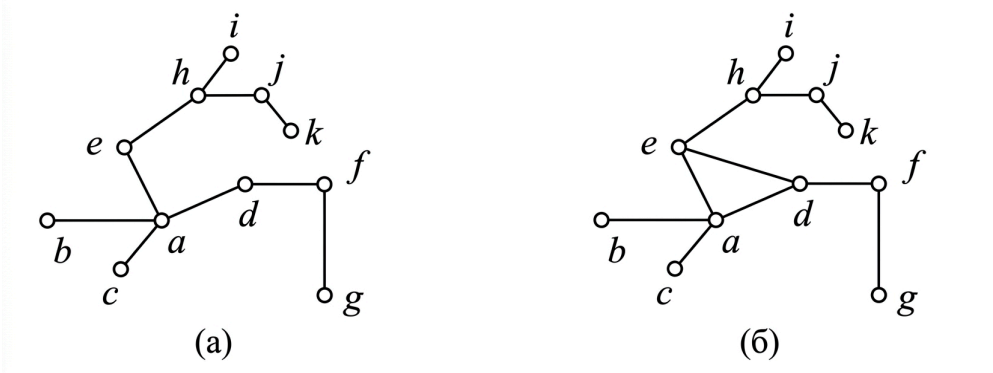


3. Дерева і ліс

Неорієнтованим деревом (або просто ***деревом***) називається неорієнтований зв'язний граф без циклів, отже без петель і кратних ребер, що має не менше двох вершин.

Дерево є мінімальним зв'язним графом у тому розумінні, що видалення хоча б одного ребра приводить до того, що граф виявляється незв'язним.

Приклад. Чи є графи, зображені на рис. 4.50, (а, б) деревами



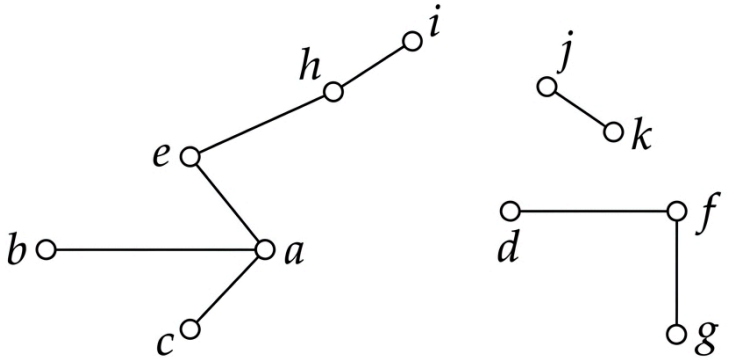
Розв'язання: Граф на рис. а є деревом, тому що він зв'язний і не містить циклів. Граф на рис. б не є деревом, тому що містить цикл *adea*.

Теорема. Будь-яке дерево G з n вершинами містить n -1 ребро.

Теорема. Нехай G – граф, що містить більше однієї вершини. Видаляючи його ребра можна одержати дерево тоді і тільки тоді, коли він зв'язний.

Лісом називається незв'язний неорієнтований граф без циклів. Зв'язні компоненти лісу є деревами.

Приклад. На рис. .наведений приклад лісу, що містить три дерева.



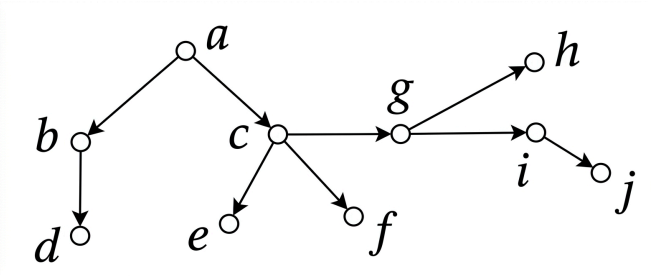
..

Добре зрозумілим прикладом дерева є: будь-яке генеалогічне дерево; організаційна структура будь-якого підприємства, організації.

Орієнтованим деревом називається вільний від петель орієнтований граф, співвіднесений граф якого є деревом; тому якщо існує шлях від вершини v' до вершини v'', то він єдиний.

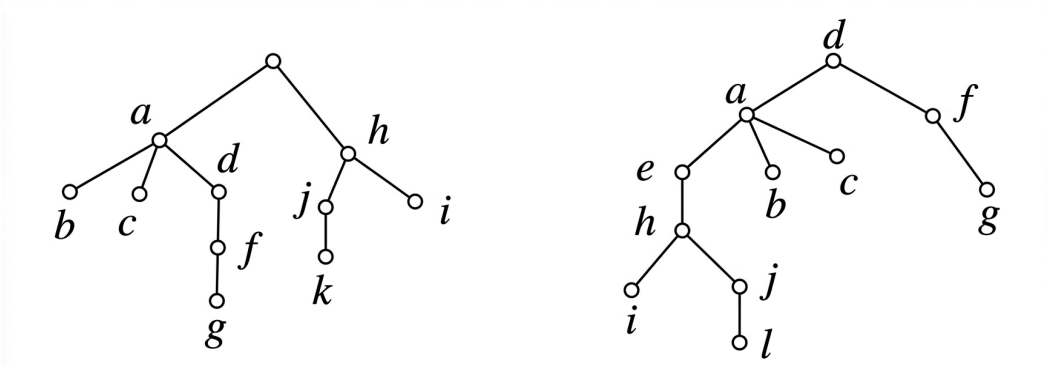
Зауважимо, що якщо в орієнтованому дереві є ребро (v', v''), то немає ребра (v'', v'), інакше шлях $v'v''v'$ був би циклом.

Приклад. Приклад орієнтованого дерева:



Вершини дерева степеня 1 називаються ***листками***, інші вершини – ***внутрішніми вершинами***.

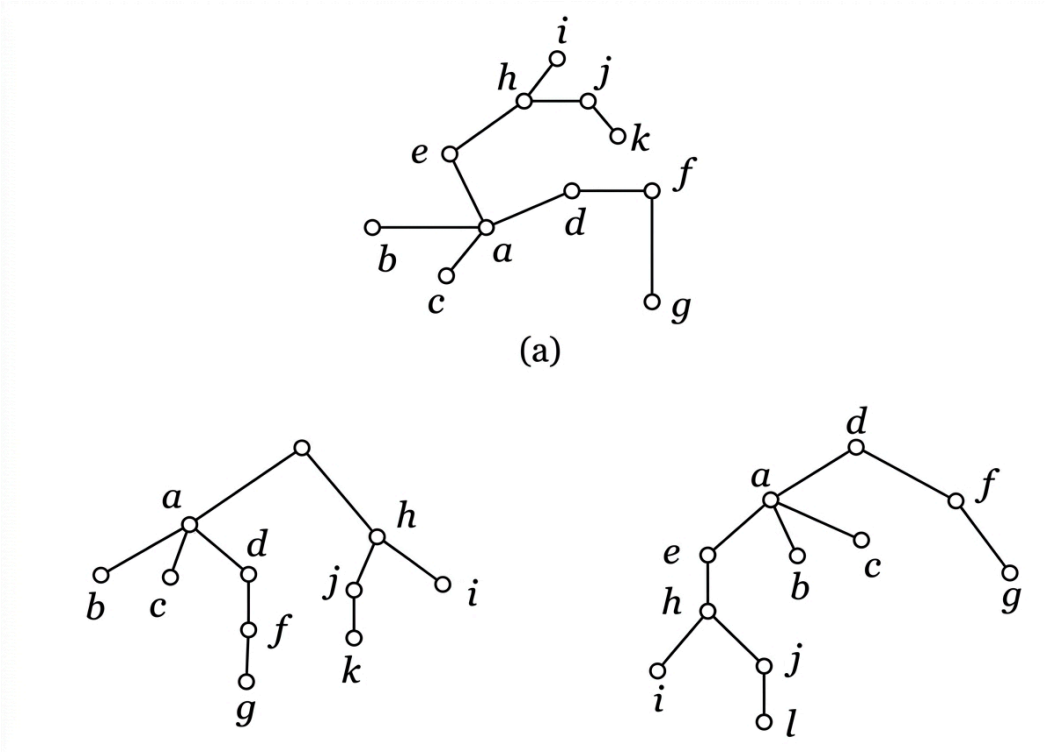
Наприклад, у дерева, зображеного на рис. а, листя - це вершини c, b, g , i, k . Інші вершини є внутрішніми.



Дерева визначаються такою властивістю: для будь-яких двох вершин v', v'' дерева T шлях з вершини v' до вершини v'' єдиний. І, навпаки, якщо для

будь-яких двох вершин v', v'' графа G існує єдиний шлях з вершини v' до вершини v'' , тоді граф G – дерево.

Припустимо, що дерево – це якийсь фізичний об'єкт, рухливий у вершинах. Його можна підвісити за кожну з вершин. Так, наприклад, якщо дерево, зображене на рис.а, а підвісити за вершину e або d , то воно буде виглядати, як показано на рис. нижче.

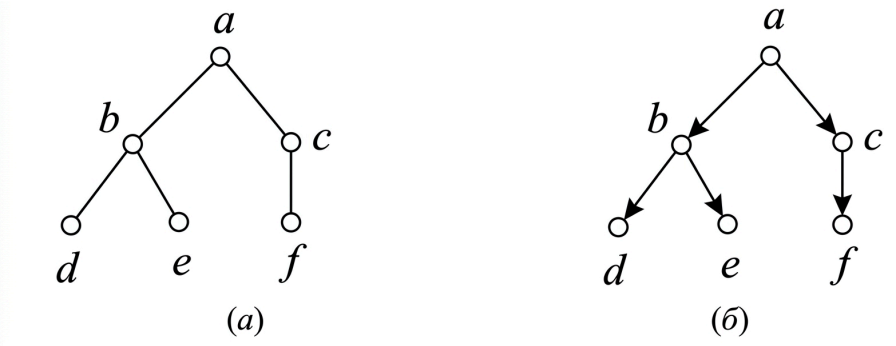


Обрана нами вершина називається *коренем* дерева і розташовується в самій верхній його частині. Для дерев на рисунках коренем є вершина e і вершина d .

Дерево, корінь якого визначений, називається ***кореневим деревом.***

Аналогічно визначається і ***орієнтоване кореневе дерево.*** При цьому варто пам'ятати, що всі його ребра орієнтуються від кореня. При заміні кореневого неорієнтованого дерева T на кореневе орієнтоване дерево T' говорять, що T' є ***породженим кореневим деревом*** T .

Приклад На рис. а зображене неорієнтоване кореневе дерево, а на рис. б – породжене ним орієнтоване кореневе дерево.

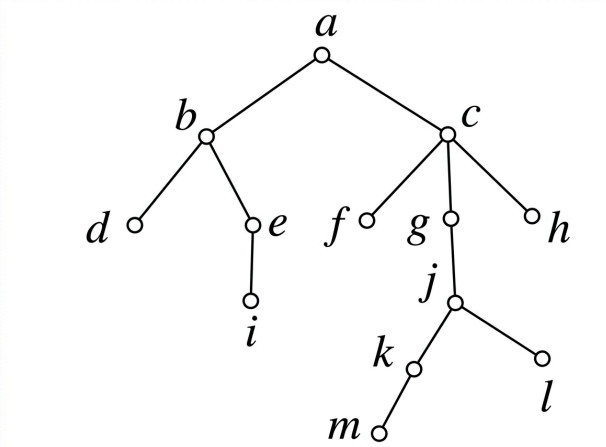


Якщо корінь дерева обраний, то можна ввести ще ряд визначень.

Рівнем вершини v називається довжина єдиного шляху від кореня дерева до вершини v.

Висотою дерева називається довжина найдовшого шляху від кореня дерева до листка.

Приклад. Для кореневого дерева, зображеного на рис., визначити рівень вершини d, висоту дерева.

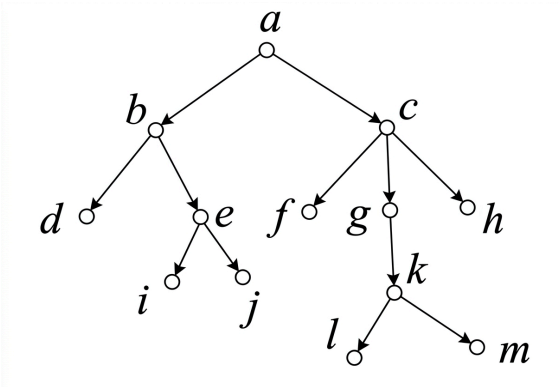


Розв'язання: Рівень вершини *d* дорівнює двом; а висота дерева з коренем у вершині *a* дорівнює максимальному шляху від кореня до вершини *m* – *acgjkm* і дорівнює п'яти.

Для генеалогічних дерев дамо ряд визначень. Їх можна поширити на будь-які орієнтовані кореневі дерева.

Якщо існує орієнтоване ребро з *u* в *v*, то вершина *u* називається ***безпосереднім попередником (предком)*** вершини *v*, а вершина *v* – ***безпосередньо підлеглою*** вершині *u* ***(нащадком)*** або ***безпосередньо досяжною з вершини u***.

Приклад. Визначити предків і нащадків кореневого орієнтованого дерева, зображеного на рис.



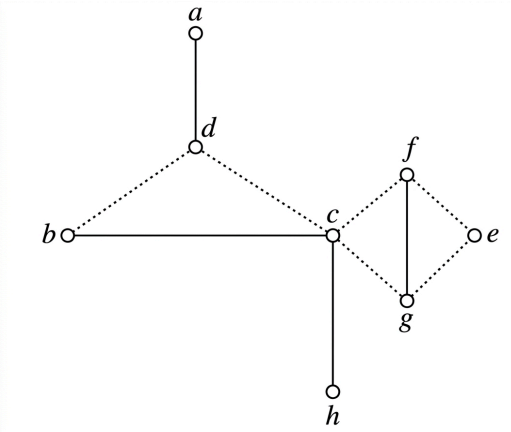
Розв'язання:

Груп нащадків і предків можна виділити дуже багато, наведемо лише деякі приклади: *d*, *e*, *i*, *j* – нащадки *b*, отже, *b* – предок для *d*, *e*, *i*, *j* ; *l* є нащадком *c* отже, *c* – предок *l* .

Ациклічна множина ребер, що поєднують усі вершини графа і загальна вага яких мінімальна, називається ***мінімальним каркасним (або остовним) деревом***. Для побудови остовних дерев існують різні методи. Найбільш поширеними є два методи: метод пошуку в ширину і метод пошуку в глибину.

Згідно ***методу пошуку остовного дерева в ширину*** довільну вершину v_0 графа G вибираємо як корінь дерева T. Для кожної вершини v графа G, суміжної з вершиною v_0, в дерево T додається вершина v і ребро (v_0, v). Це вершини рівня 1 (DFS-номер). Потім беремо кожну вершину v_1 рівня 1 і для кожної вершини графа G v_j, суміжною з вершиною v_i з тих, що ще не обрані, додаємо в дерево T вершину v_j і ребро (v_i , v_j). Вершини, додані на цьому етапі, – це вершини рівня 2. Продовжуємо процес, поки в графі G не залишиться вершин, які можна було б додати в дерево. За побудовою T є деревом. Якщо відстань від v_0 до вершини v графа G рівно n, то ця вершина буде додана в дерево на рівні n. Отже, T безсумнівно, є остовне дерево. При пошуку в ширину насамперед відшукуються всі вершини, суміжні з заданою вершиною, а потім здійснюється перехід на наступний рівень. При пошуку в глибину зусилля спрямовані на побудову для дерева якомога довшого шляху.

Метод пошуку в глибину починається з завдання вершини графа, яку будемо вважати коренем. Вибираємо вершину v_i, суміжну з коренем, і формуємо ребро дерева. Потім вибираємо вершину v_j, суміжну з раніше вибраною вершиною v_i, і формуємо нове ребро. По ходу необхідно позначати використані вершини, щоб кожна вершина використовувалася один раз. Якщо, перебуваючи у вершині v, ми вибираємо іншу вершину w і виявляємо, що вершина w вже була додана в дерево, то ребро (v, w) між цими вершинами не може бути додане в дерево.



Питання для самоконтролю

1. Що визначає поняття ейлерового циклу та за якими критеріями довільний граф можна класифікувати як ейлеровий?

2. У чому полягає відмінність гамільтонових шляхів і циклів від ейлерових та які умови їхнього існування в графі?

3. Які специфічні структурні властивості та ознаки відрізняють поняття дерева й лісу від інших видів графів?

4. Як саме функціонує алгоритм пошуку вглиб (DFS) для побудови та виділення остовного дерева з початкового графа?

5. Яким чином алгоритм пошуку вшир (BFS) забезпечує покрокове формування остовного дерева у довільному графі?

Тема 16. Формальні мови

Мета вивчення: ознайомити з поняттям формального алфавіту, формальної мови, регулярного виразу; формувати вміння створювати граф регулярного виразу.

План

1. Формальні мови, основні поняття. Синтаксис, семантика.
2. Регулярні вирази.
3. Граф регулярного виразу.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

1. Формальні мови

Для розгляду формальних мов дамо, насамперед, визначення низки необхідних понять. Формальні мови, як і природні, передбачають можливість подання текстів у формі послідовності символів певного алфавіту. Формальні мови використовуються для опису штучних мов, наприклад, мов програмування, мов команд операційної системи. Будь-яке програмне забезпечення створюється і експлуатується з допомогою тієї чи іншої формальної мови.

Алфавітом називають непорожню множину неподільних символів (букв, літер) і позначають великими літерами грецького алфавіту чи латиниці (можливо з індексами), як-то Σ, A, *B*, V.

Наприклад, алфавіти $\Sigma_1 = \{a, b, c\}$, $A_1 = \{00, 01,10,11\}$ і $V_1 = \{0,1\}$ містять 3, 4 і 2 символи відповідно. В алфавіті A_1 кожен з чотирьох символів вважається неподільним за визначенням.

Ланцюжок (слово) над алфавітом Σ – це послідовність літер цього алфавіту. Для позначення ланцюжків часто використовують літери грецького

алфавіту, наприклад, α, β чи ω, або літери другої половини латинського алфавіту, наприклад *p, s, t, x, y* тощо.

Довжина слова α визначається як кількість літер у цьому слові. Позначається як $|\alpha|$, або як $l(\alpha)$. Наприклад, |bc| = 2 над алфавітом Σ_1, |010011| = 3 над алфавітом A_1 і |010011| = 6 над алфавітом V_1.

Порожній ланцюжок позначають через ε або λ. Очевидно, що довжина порожнього слова |ε| = 0.

Конкатенацією слів α та β є слово γ = αβ, тобто до слова α справа дописують β. Так, якщо α = *ab* та β = *cd*, то їх конкатенація αβ = *abcd.* Конкатенація – некомутативна операція, оскільки, αβ ≠ βα*,* адже *abcd* ≠ *cdab.*

Очевидним є результат конкатенації довільного слова *x* з порожнім рядком: ε*x* = *x*ε = *x*.

Якщо слово ω = *x*y, то кажуть, що *x* – префікс (голова), а *y* – суфікс (хвіст) слова *w*. Префікс *x* називають ***власним (правильним) префіксом (правильною головою)*** ω, якщо *x* ≠ ε і *x* ≠ ω. Відповідно *y* називають ***власним (правильним) суфіксом (хвостом)*** слова w, якщо *y* ≠ ε і *y* ≠ ω. І взагалі, довільний підрядок s рядка (слова, ланцюжка) *w* = *asв* є його власним підрядком, якщо s ≠ ε і *s* ≠ ω.

Для спрощення та формалізації роботи з довгими словами застосовують степінь ланцюжка, де показник степеня вказує на кількість повторень підрядка у певному місці слова. Так слово *aaaaa* зручно позначити як a^5, *abababab* – як $(ab)^4$, а *abababbbaaaabab* – як $(ab)^3b^2a^3(ab)^2$. Взагалі, степені довільного слова α означають:

$\alpha^0 = \varepsilon$

$\alpha^1 = \alpha$

$\alpha^2 = \alpha\alpha$

$\alpha^3 = \alpha\alpha\alpha$ ...

$\alpha^n = \alpha^{n-1}\alpha$

Над довільними множинами ланцюжків можна визначити операцію ***конкатенації*** (добутку над операцією конкатенації): $AB = \{xy \mid x \in A, y \in B\}$.

Наприклад, для $A = \{m,n\}$, $B = \{0,1\}$ добуток $AB = \{m0, ml, n0, nl\}$.

Оскільки алфавіт – це множина, то всі ланцюжки довжини *n* над алфавітом можна представити як його n-ий степінь. Нехай алфавіт Σ= {a,b}. Тоді всі слова довжини 3 над Σ можна представити у формі

$\Sigma^3 = \Sigma\Sigma\Sigma = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$.

Загалом:

$\Sigma^0 = \{\epsilon\}$ - єдине порожнє слово,

$\Sigma^1 = \Sigma = \{a, b\}$ - усі можливі слова довжини 1,

$\Sigma^2 = \Sigma\Sigma = \{aa, ab, ba, bb\}$ - усі можливі слова довжини 2,

$\Sigma^3 = \Sigma\Sigma\Sigma = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$ - усі слова довжини 3,

...

$\Sigma^n = \Sigma^{n-1}\Sigma$ – усі можливі слова довжини *n*.

Множину всіх ланцюжків над алфавітом Σ називають ***замиканням Кліні*** (***зірка Кліні, ітерація Кліні***, або просто замикання) та позначають

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots \cup \Sigma^n \ldots$$

Замикання Кліні алфавіту Σ = *{a, b}* – множина всіх можливих слів, довжиною від нуля до нескінченності:

Σ^* = *{ε, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb..., aaaa,...* }.

Позитивним замиканням алфавіту (плюс Кліні) називають множину

$$\Sigma^+ = \bigcup_{i=0}^{\infty} \Sigma^i = \Sigma^1 \cup \Sigma^2 \ldots \cup \Sigma^n \ldots$$

Так, для Σ = *{a, b}* позитивне замикання Σ^+ – множина всіх можливих над Σ слів, довжиною від одного до нескінченності:

Σ^+ = *{a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb..., aaaa,...* }.

З означень безпосередньо випливає $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$.

Мовою *L* над алфавітом Σ називають деяку підмножину замикання Кліні алфавіту

$L \subset \Sigma^*$.

Так, замикання алфавіту Σ = {*a*} – це нескінченна множина слів довільної довжини Σ^* = *{ε, a, aa, aaa, aaaa,...* }. Визначимо мову *L* як множину слів, довжина кожного з яких – непарна, тобто *L = {a, aaa, aaaaa,...* }.

До формальних мов широко застосовують як звичайні множинні операції: об'єднання, перетин та доповнення, так і специфічні: конкатенація – L_1L_2 = *{xy | x ∈ L1,y ∈ L2*}; ітерація або Кліні зірочка (*****) (оператор замикання); Кліні плюс (+); заміна (підстановка); морфізми та зворотні морфізми; обернення (реверс); перемішування тощо.

Для означення мови використовують запис у формі *L = {*ω | – опис властивостей ω*},* які читаються так: "мова *L* - це множина слів ω*,* таких, що мають вказані після вертикальної риски властивості".

Для наведеного вище прикладу мова *L* може бути визначена, наприклад, так:

*L = {*ω | *l(*ω*)* – непарне число}, або, з більш формальним визначенням довжини слова: *L = {*ω | ω∈ Σ^*, |ω| = *2n* + 1, n≥ 0}. В останньому записі кома означає кон'юнкцію, тобто визначення може бути прочитане так: "мова *L* – це множина слів ω із множини Σ^* з довжиною, що визначається виразом *2n* + 1, де *n* – невід'ємне ціле число, або число нуль".

Отже, для визначення формальної мови *L* необхідно:

1) обрати алфавіт та назвати його, наприклад, Σ;

2) побудувати замикання алфавіту Σ^*;

3) вказати вимоги до властивостей слів з Σ^*, які входять до означуваної мови.

Такий підхід дозволяє перевірити, чи справді кожне слово даного тексту (наприклад, комп'ютерної програми) написане означеною формальною мовою (вказаною мовою програмування). Таке завдання виконується з використанням розпізнавачів – автоматів (автоматів з магазинною пам'яттю, або, у найпростіших випадках – скінченних автоматів чи, еквівалентних їм, регулярних виразів).

Альтернативний підхід полягає у використанні формальних граматик, які містять правила, що дозволяють згенерувати тільки такі ланцюжки символів, які гарантовано належать означеній мові.

Тож для реалізації вказаних підходів до визначення мов, розглянемо такі абстрактні математичні системи, як:

1) регулярні вирази;

2) формальні (генеративні) граматики;

3) розпізнавачі (автомати).

2. Регулярні вирази

Регулярні вирази використовують для визначення регулярних мов. Тому спочатку дамо визначення регулярої мови (регулярної множини), а потім – регулярного виразу.

Рекурсивно регулярна мова над алфавітом Σ може бути визначена так:

1) порожня мова $\varnothing$ є регулярною мовою;

2) $\{\varepsilon\}$ є регулярною мовою над алфавітом Σ (вона містить єдине слово – порожнє);

3) для кожного $a \in \Sigma$ множина {a} є регулярною мовою (вона містить єдине слово – a);

4) якщо *R* та *S* - регулярні мови над алфавітом Σ, то регулярними також є мови:

- $R \cup S$ - об'єднання регулярних мов;
- *RS* - конкатенація регулярних мов;
- R^* - замикання Кліні регулярної мови (довільної над алфавітом Σ);

5) інших регулярних мов над алфавітом Σ не існує.

Регулярні вирази – засіб позначення регулярних мов, прийнятий для представлення та формальних маніпуляцій над регулярними мовами, визначаються так:

1) $\varnothing$ – регулярний вираз, що позначає порожню мову $\varnothing$;

2) $\{\varepsilon\}$ – регулярний вираз, що позначає регулярну мову $\{\varepsilon\}$;

3) {*a*} –регулярний вираз, що позначає регулярну мову *{a};*

4) якщо **r** та **s** – регулярні вирази, що позначають, відповідно, регулярні мови *R* та *S* над алфавітом Σ, то:

- **r** + **s** – регулярний вираз, що позначає об'єднання регулярних мов $R \cup S$;
- **rs** – регулярний вираз, що позначає конкатенацію регулярних мов *RS*;
- **r*** – регулярний вираз, що позначає замикання Кліні регулярної мови R^*;
- (**r**) – регулярний вираз, що позначає ту ж регулярну мову R, що й регулярний вираз **r** (тобто зайві дужки можуть бути опущені);

5) інших регулярних виразів над алфавітом Σ не існує.

Наведемо приклади регулярних виразів, що визначають відповідні регулярні мови над певним алфавітом: *ab* позначає мову над алфавітом Σ = *{a, b}*, що містить одне слово {*ab*};

Замість оператора **+** також використовується символ **|** (вертикальна риска).

Оператор конкатенації, зазвичай, опускають, але інколи застосовують явне позначення, наприклад символ . (крапка).

- *a + b* позначає мову над алфавітом Σ = *{a, b},* що містить два слова *{a, b};*
- *a* + (b^*) позначає мову над алфавітом Σ = *{a, b}: {a}∪{b}* = {a,ε,b,bb,bbb,... };*
- *aa(a + b)** позначає мову над алфавітом Σ = *{a, b},* кожне слово якої має префікс *aa*, а решта слова – нуль або більше символів, кожен з яких – *це a чи b;*
- *a(ba)* + b(ab)** позначає мову без порожнього слова над алфавітом Σ = *{a, b},* кожне слово якої починається і закінчується одним і тим же символом, і в якому символи *a* і *b* почергово змінюють один одного;
- *0 + 1)(0 + 1 + de)** позначає мову над алфавітом Σ = *{0,1, d, e},* кожне слово якої починається з 0 або 1, а решта слова містить нуль чи більше ланцюжків, кожний з яких — це *0, 1* або *de.*

Одна й та ж регулярна мова може бути представлена різними регулярними виразами. Справді, наприклад, *aεb, εab,* та *abε* позначають одну й ту ж мову *{ab}* над алфавітом Σ= *{a,b}*. Два регулярні вирази α та β, що позначають одну й ту ж регулярну мову, називають ***еквівалентними*** та пишуть α = β. Еквівалентність регулярних виразів з'ясовується (доводиться) через доведення рівності відповідних їх мов.

Для регулярних виразів *α, β* та *γ* справедливі, зокрема, такі еквівалентності:

- комутативність та асоціативність оператора альтернативи +:

  $\alpha + \beta = \beta + \alpha,$

  $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma;$
- дистрибутивність (зліва та справа) конкатенації відносно альтернативи +*:*

  $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma,$

$(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma;$

- порожнє слово ε є одиницею відносно конкатенації: $\alpha\varepsilon = \varepsilon\alpha = \alpha;$
- порожнє слово ε завжди міститься в ітерації: $\alpha^* = (\alpha + \varepsilon)^*;$
- порожня мова $\varnothing$ є нулем відносно конкатенації та нулем відносно альтернативи +: $\alpha\varnothing = \varnothing\alpha = \varnothing$, $a + \varnothing = a;$
- ідемпотентність ітерації *: $(\alpha^*)^* = \alpha^*;$

Метамова представлення регулярних виразів

Правила запису регулярних виразів фактично є метамовою – мовою для опису іншої мови. Для усунення неоднозначності при опусканні чи відсутності дужок, необхідно враховувати пріоритет операторів (всі вони – лівоасоціативні), поданих далі у порядку його зменшення:

• унарний постфіксний оператор ∗ – ітерація (замикання Кліні), має найвищий пріоритет;

• бінарний оператор конкатенації, як правило не має явного позначення і має проміжне значення пріоритету щодо * та +;

• бінарний оператор + – альтернативи (об'єднання мов), має найнижчий пріоритет.

Часте застосування регулярних виразів, зокрема і в практиці програмування, призвели до значного розширення мінімальної метамови регулярних виразів та виникнення численних діалектів. Типовими, зокрема, є використання таких операторів та метасимволів:

• унарний постфіксний оператор позитивного замикання + (Тоді для позначення оператора альтернативи використовується символ | (вертикальна риска): $\alpha^+ = \alpha\alpha^*$ – один або більше екземплярів α . Пріоритет дорівнює пріоритету оператора ∗;

• унарний постфіксний оператор ?:

α? = ε +α – нуль або один екземпляр α. Пріоритет дорівнює пріоритету оператора ∗;

• квадратні дужки (метасимволи) [ ] – позначають клас символів (один з перелічених):

$a\boldsymbol{o} + a\boldsymbol{i} + ... + a_n$, де $a_i \in X$ представляють у формі [$a\boldsymbol{o}a\boldsymbol{i}... a_n$], наприклад 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + *A* + *B* + *C* + *D* + *E* + *F* можна представити як [0123456789ABCDEF], або, якщо визначені стандартні послідовності літер та цифр, у формі [0 – 9A – *F*].

3. Граф регулярного виразу

Довільний регулярний вираз може бути представлений у формі орієнтованого поміченого графа, дуги якого помічені символами з Σ ∪ ε, де Σ – алфавіт регулярної мови.

Побудову поміченого орієнтованого графа регулярного виразу r представимо у вигляді набору (пронумерованих) правил з графічними ілюстраціями до них, див. рис. 1–4. Для побудови графа регулярного виразу r:

1) подають його у формі двох вершин – стартової та заключної (подвійне коло), з дугою, поміченою символом r:

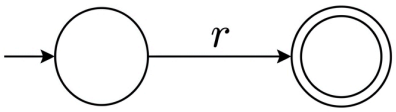


Рис. 1. До правила 1 побудови графа

2) виконують заміну кожної дуги з альтернативою $r_1 + r_2$ на дві дуги між тими ж вузлами з позначками r_1 та r_2:



Рис. 2. До правила 2 побудови графа

3) виконують заміну кожної дуги з конкатенацією $r_1 r_2$ на дві дуги r_1 та r_2 з вершиною між ними:

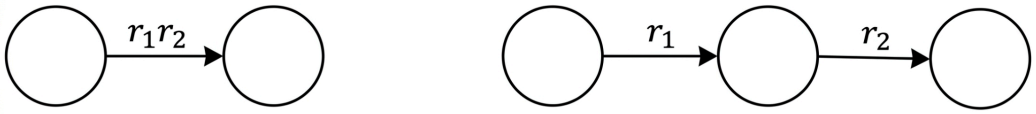


Рис. 3. До правила 3 побудови графа

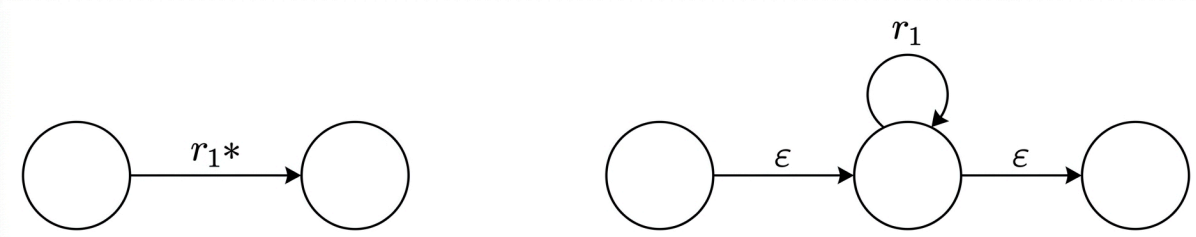


Рис. 4. До правила 4 побудови графа

4) виконують стягування кожної дуги ε, якщо вона є єдиною вихідною дугою кінцевої вершини, або єдиною вхідною дугою нестартової вершини, в одну вершину.

Скористаємось наведеними правилами для побудови графа регулярного виразу $r_1 = a(ba)^*(b + \varepsilon) + b(ab)^*(a + \varepsilon)$, подаючи цей процес покроково, див. рис. 5–11:

1) стартова – заключна вершини:

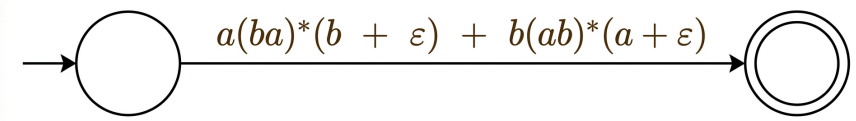


Рис. 5. За правилом 1

2) альтернатива:

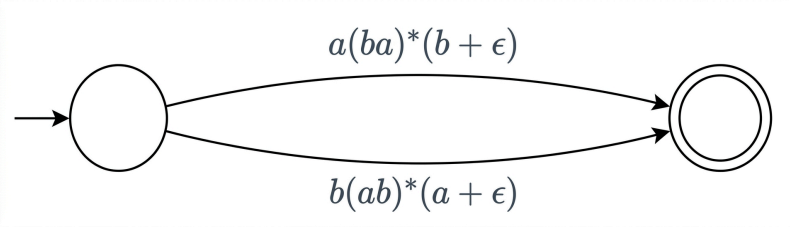


Рис. 6. За правилом 2

3) конкатенація:

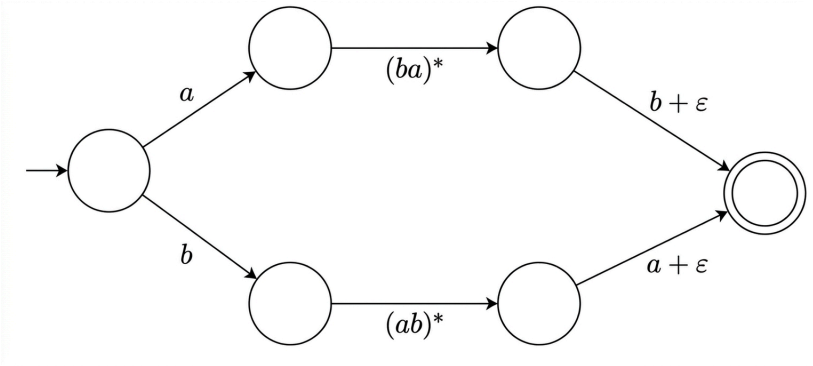


Рис. 7. За правилом 3

4) альтернатива:

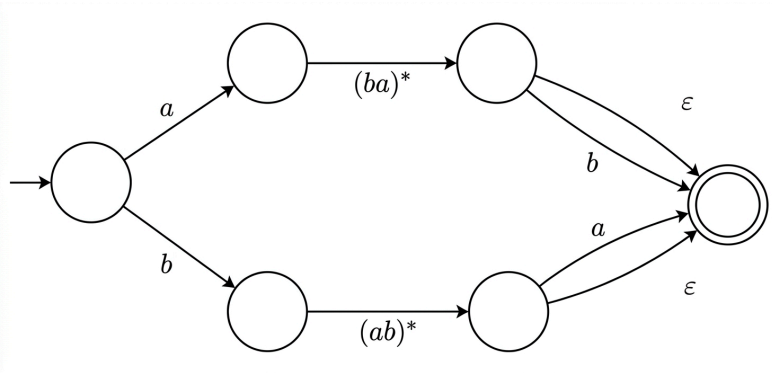


Рис. 8. За правилом 2

5) замикання Кліні у кожній гілці:

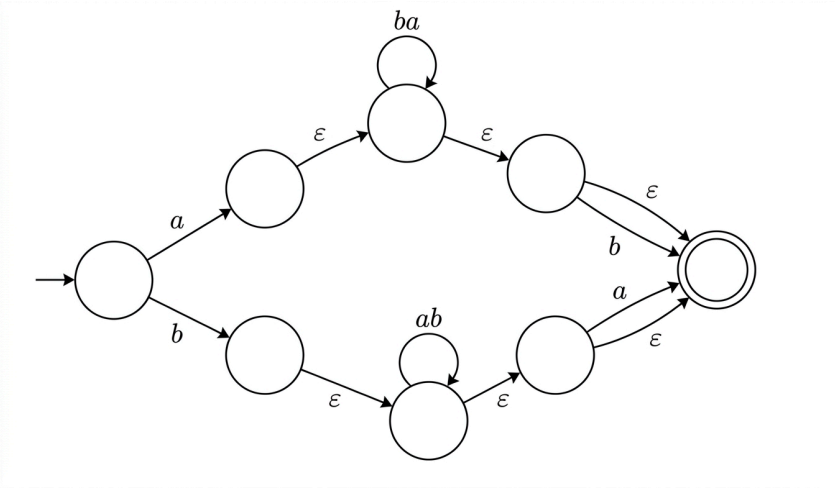


Рис. 9. За правилом 4

6) конкатенація:

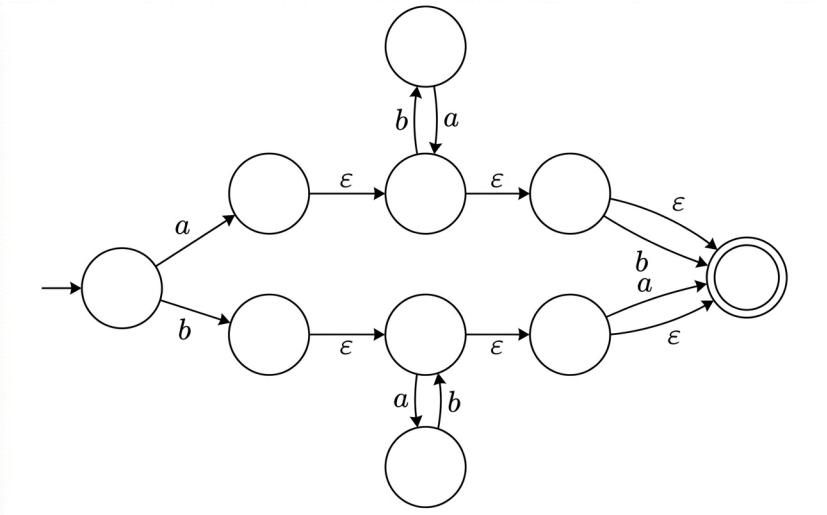


Рис. 10. За правилом 3

7) стягування ε – дуг:

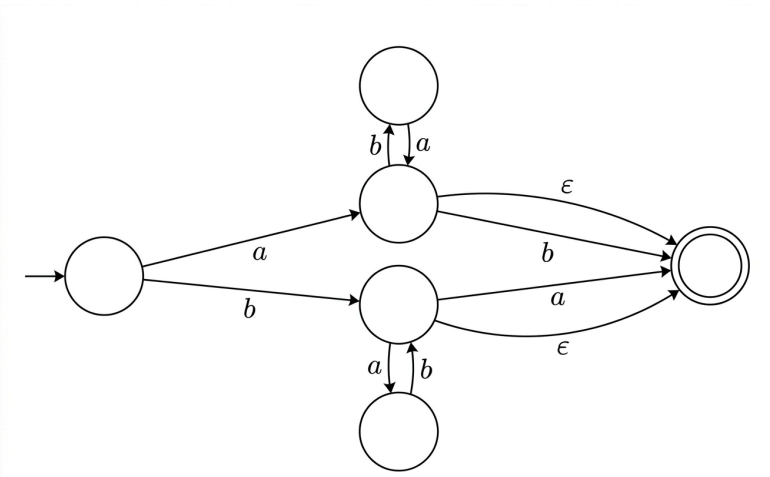
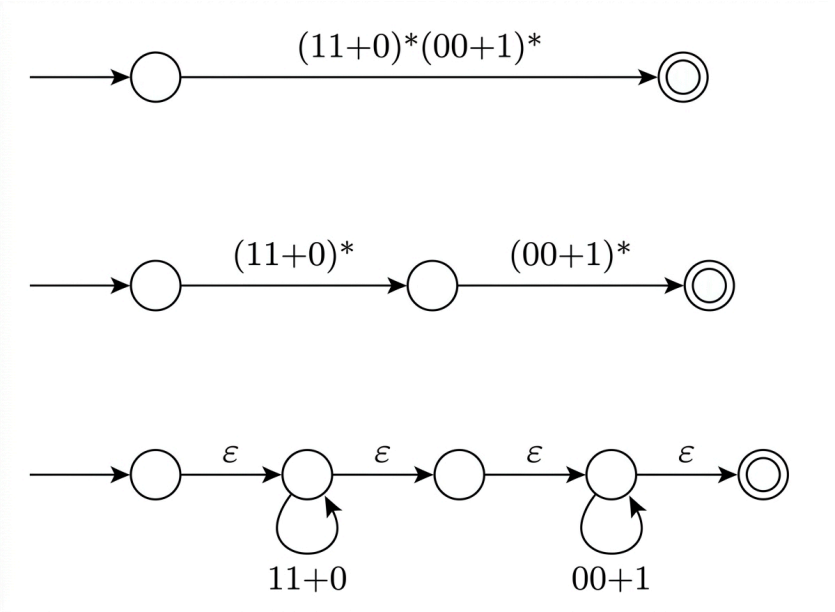
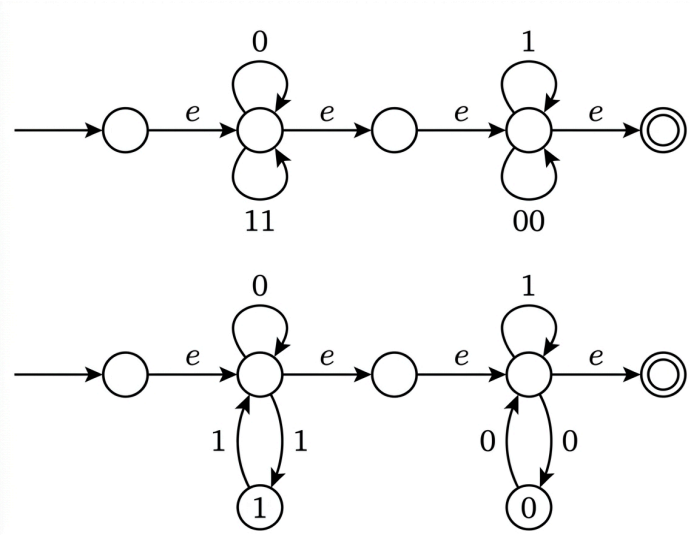


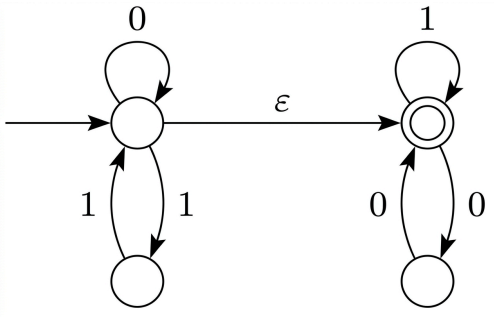
Рис. 11. За правилом 5

Приклад. Побудуємо граф G(r) регулярного виразу r = $(11 + 0)^*$ $(00 + 1)^*$. Зобразимо крок за кроком побудову даного графу:





Стягнувши, де це можливо, e-стрілки, остаточно одержуємо помічений граф G(r) регулярного виразу $r = (11 + 0)^*(00 + 1)^*$:



Розв’язування задач

1. Для наведених регулярних виразів надайте словесний опис множини рядків і визначте, чи належить рядок 1011 цій множині:

а) 10*1*;

б) 1(01)*1*;

в) (10)*(11)*:

г) (10)*1011;

д) 0*(10∪11)*

е) 1*01(0∪1);

є) 1(00)*(11)*;

ж) (1∪00)(01∪0)1*

2. Для алфавітів *A = {a, b, c}, B = {0, 1}* складіть

а) A*, B*;

б) A^+, B^+

3. Нехай задано рядок α = «послідовність». Знайдіть довжину рядка |α|.

4. Побудуйте рядки $0^5 1^2$, $a^3 b^2 c$, $0^n A 1^n$ для n = 3, 5, 0.

5. α = «alpha», β = «bet». Що є конкатенацією рядків αβ?

6. Які з наведених рядків є підрядками рядка α = «індустріалізація»: β_1= «індус», β_2= «індустрія», β_3= «три», β_4= «акція».

7. Знайти слово найменшої довжини в кожній з наступних регулярних мов, заданій регулярним виразом. Які непорожні слова є найкоротшими в даних мовах?

а) $(00 + 11 + (01 + 10)(00 + 11)^* (01 + 10))^*$;

б) $((00 + 11)^* + (001 + 110)^*)^*$.

8. Визначити, чи належить рядок 1011 регулярним мовам, заданим регулярними виразами:

а) (10)+1011;

б) 1(00)*(11)* ;

в) (1 + 00)(01 + 0)1* .

9. Записати регулярний вираз для множини всіх слів у двійковому алфавіті, які містять підслово 001. Кожне таке слово запишеться у вигляді α001β, де α і β – довільні слова в двійковому алфавіті. Таким чином, одержуємо шуканий регулярний вираз: $(0 + 1)^* 001(0 + 1)^*$.

10. Визначити алфавіт та побудувати регулярний вираз для наступної мови: множина всіх непарних натуральних чисел, записаних в четвірковій системі числення.

11. Побудувати помічений граф для наступних регулярних виразів:

а) (00+11)* +(01+11)*;

б) a* b(c + da* b)* .

12. Побудуйте регулярний вираз для множини всіх цілих чисел відрізка [10, 2012].

Питання для самоконтролю

1. Що таке формальний алфавіт та формальна мова? У чому полягає принципова відмінність між синтаксисом і семантикою формальної мови.

2. Дайте визначення регулярного виразу. Які базові операції (наприклад, об'єднання, конкатенація, ітерація) використовуються для їхнього опису?

3. Яке призначення має граф регулярного виразу? Що у такому графі зазвичай відображають вершини (кола), а що – спрямовані дуги (стрілки)?

4. Опишіть правила побудови фрагментів графа для таких конструкцій

регулярного виразу:

- альтернатива або додавання ($r_1 + r_2$);
- послідовне виконання або множення ($r_1 r_2$);
- ітерація Кліні (r^*).

5. Яку функцію виконують порожні переходи (ε-переходи) під час побудови графа регулярного виразу? Як вони допомагають зв'язати окремі блоки автомата в єдину структуру?

Тема 17. Формальні граматики

Мета вивчення: ознайомити з поняттям формальної граматики, алфавітом граматики та алфавітом мови; з поняттям мови, породженої формальною граматикою, типами формальних мов і граматик.

План

1. Формальна граматика. Алфавіт граматики і алфавіт мови.
2. Мова, породжена граматикою. Правило виведення.
3. Неоднозначні та еквівалентні граматики.
4. Типи формальних мов і граматик.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

Ми визначили формальну мову L як множину слів скінченної довжини в алфавіті X. Якщо L містить скінченну кількість слів, то найбільш очевидний спосіб описати мову – скласти перелік всіх її слів. Однак для багатьох мов не можна (або, можливо, не бажано) обмежувати довжини слів. Отже, доводиться розглядати мови, які містять як завгодно багато слів. Відразу виникає важливе питання: як описати мову L в тому випадку, коли вона нескінченна. Очевидно, що такі формальні мови неможливо визначити вичерпним переліком слів, які їм належать, і тому потрібно шукати інші способи їх опису. Як і раніше, ми хочемо, щоб опис мови був скінченним, хоча сама мова може бути й нескінченною. Розглянемо метод, який полягає в використанні структури, яка називається граматикою.

Формальною породжувальною граматикою називають кортеж

Г = (N, T, P, S), де

N – множина нетермінальних символів (нетерміналів, змінних);

T – множина термінальних символів (терміналів, основних символів);

$N \cap T = \varnothing$;

P – множина правил виведення (продукцій, правил підстановки) $\{(\alpha, \beta)\}$;

S – початковий символ (стартовий символ, аксіома).

Продукції $\{(\alpha, \beta)\}$ множини P подаються у формі $\alpha \to \beta$, причому $\alpha \in (N \cup T)^*N(N \cup T)^*$ і $\beta \in (N \cup T)^*$, звідки $P \in (N \cup T)^*N(N \cup T)^* \times (N \cup T)^*$

Зауважимо, що алфавіт граматики і алфавіт мови завжди розрізняються. Алфавітом граматики $\Gamma \in N \cup T$, оскільки в граматиці використовуються як нетермінальні, так і термінальні символи. У мові, яку визначає граматика – тільки термінали, отже алфавітом мови є множина T.

Далі, якщо не вказано інше, використовуються такі символи для позначень:

• великі латинські літери A, B, C,... – для нетерміналів. Літера S, якщо вона є, вважається стартовим символом;

• великі латинські літери X, Y,... – для довільних окремих символів (терміналів чи нетерміналів);

• малі латинські літери з початку алфавіту a, b, c,... та цифри 0, 1,... – для терміналів;

• грецькі α, β, γ,... – для ланцюжків, до складу яких можуть входити термінали та нетермінали;

• літери u, v, w, x, y – для ланцюжків термінальних символів;

• усі вказані літери можуть використовуватись з нижніми індексами, або зі штрихом, наприклад, $A^{/}$, B_{12} , c_1 , γ_3 , u_2 , $v^{//}$ тощо.

Приклад граматики: Γ = ({A, S}, {a, b}, {S → aAb, A→aAb, A→ε}, S). Як бачимо тут: множина нетерміналів N = {A, S}, множина терміналів T = {a, b}, множина правил P = {S → aAb, A →aAb, A →ε}, стартовий символ - S.

Часто, через значну кількість правил, граматику зручніше подати у такій формі:

Γ = ({A, S}, *{a, b}, P, S*), де P містить продукції:

S → aAb

A→aAb

A→ε

Наведені продукції читають, наприклад, так ”стартовий нетермінал S визначається як (слово) aAb, нетермінал A визначається як (слово) aAb або як порожній рядок ε”. Замість слів ”визначається як (слово)” кажуть також ”є за означенням”, ”породжує (слово)” тощо.

Множину продукцій P **можна** записати компактніше, використовуючи оператор альтернативи | (вертикальна риска):

S → aAb

A→aAb | ε

Метамова представлення граматик

Наведені правила запису граматик фактично є метамовою – мовою для опису іншої мови (мови представлення граматик). Крім наведених правил позначення, вона містить такі оператори:

• стрілочка (→), розділяє праву та ліву частину кожного правила;

• альтернативи, позначається як (|), або розуміється між правими частинами продукцій з однаковими лівими частинами;

• конкатенації, його символ опускається, у продукції S → aAb він передбачається між символами a, A та b;

Пріоритети операторів зростають від найменшого (стрілочка) до найвищого (конкатенація) через проміжне значення (альтернатива).

Виведення

Нехай граматика Г містить продукцію $\alpha \to \beta$ і слово $\xi_1 = \gamma_1\alpha\gamma_2$, де слова $\gamma_1, \gamma_2 \in (N \cup T)^*$. Застосування продукції $\alpha \to \beta$ до слова ξ_1 породжує слово $\xi_2 = \gamma_1 \beta\gamma_2$ (застосування зводиться до заміни α на β в ξ_1). Тоді кажуть, що слово ξ_2 безпосередньо виводиться зі слова ξ_1 , і позначають як $\xi_1 \Rightarrow \xi_2$.

Якщо $\xi_o, \xi_1, \ldots, \xi_n$ - слова над $(N \cup T)^*$ та $\xi_o \Rightarrow \xi_i \Rightarrow \bullet\bullet\bullet \Rightarrow \xi_{n-i} \Rightarrow \xi_n$, то кажуть, що ξ_n виводиться з $\xi 0$, та записують як $\xi_0 \Rightarrow^* \xi_n$.

Послідовність кроків для отримання ξ_n з ξ_0 називають виведенням, а n – довжиною (кількістю кроків) виведення.

Виведення $\xi_0 \Rightarrow \xi_1 \Rightarrow \cdots \Rightarrow \xi_{n-1} \Rightarrow \xi_n$ називають повним, якщо слово ξ_n не містить нетерміналів. Саме слово ξ_n при цьому називають реченням. Довільне слово ξ_i виведення називають його сентенційною формою.

Тепер можна сформулювати ще одне, еквівалентне вже наведеним, визначення мови: мовою L(Г), породжену граматикою Г = *(N,T,P,S*), називають множину всіх термінальних ланцюжків (речень), які виводяться з аксіоми S:

$$L(\Gamma) = \{\xi \mid \xi \in T^*, S \Rightarrow^* \xi\}.$$

Мовою, породженою граматикою G = (N, T, P, S), називають множину всіх слів, які виводяться з початкового символу S і містять тільки термінальні символи. Її позначають L(Г). Таким чином, $L(\Gamma) = \{\omega \in T^* \mid S \Rightarrow^* \omega\}$

Використання граматик

Оскільки формальні граматики здатні моделювати більшу частину довільної мови програмування, то виникає кілька типових завдань практичного характеру, зокрема:

1. Визначити, чи належить ланцюжок певній мові.

2. Знайти опис мови, заданої граматикою.

3. Побудувати граматику за описом мови. Визначити, чи належить ланцюжок певній мові. Очевидно, треба за описом мови перевірити, чи має

даний ланцюжок (слово) властивості, що висуваються до усіх слів мови. Слово ξ належить мові з граматикою Г, якщо вдається побудувати його виведення у граматиці даної мови Г, тобто якщо $S \Rightarrow^{*} \xi$, де S – стартовий символ граматики. Або, що те ж саме, якщо вдається побудувати дерево виведення даного слова.

Розглянемо приклад, який ілюструє, ***як знайти опис мови, заданої граматикою.***

Приклад. Виведемо кілька слів, заданих граматикою

Г = ({A, S}, {0, 1}, {S → 0A1, A → 0A1, A → ε}, S).

Для зручності перенумеруємо продукції множини Р:

(1) S → 0A1

(2) A → 0A1 | ε

Виконаємо наступні кроки для виведення одного зі слів:

1. $S \Rightarrow^{1} 0A1$ – за правилом (1);

2. $0A1 \Rightarrow^{2.1} 00A11$ – за правилом (2.1), тобто першою альтернативою правила (2);

3. $00A11 \Rightarrow^{2.2} 00\varepsilon11$ – за правилом (2.2), тобто другою альтернативою правила (2);

4. ε – порожній рядок, тож після підстановки його значення в ланцюжок 00ε11 (тобто викреслювання) маємо 0011.

Отже, слово 0011 належить мові L(Г), оскільки $S \Rightarrow^{*} 0011$. Надалі цю послідовність кроків представлятимемо більш компактно: $S \Rightarrow^{1} 0A1 \Rightarrow^{2.1} 00A11 \Rightarrow^{2.2} 00\varepsilon11 = 0011$, або, опускаючи крок заміни ε його значенням, у формі: $S \Rightarrow^{1} 0A1 \Rightarrow^{2.1} 00A11 \Rightarrow^{2.2} 0011$.

Виведемо ще кілька слів:

$S \Rightarrow^{1} 0A1 \Rightarrow^{2.2} 0\varepsilon1 = 01$

$S \Rightarrow^{1} 0A1 \Rightarrow^{2.1} 00A11 \Rightarrow^{2.2} 00\varepsilon11 = 0011$

$S \Rightarrow^{1} 0A1 \Rightarrow^{2.1} 00A11 \Rightarrow^{2.1} 000A111 \Rightarrow^{2.2} 000\varepsilon111 = 000111,$

$S \Rightarrow^{1} 0A1 \Rightarrow^{2.1} 00A11 \Rightarrow^{2.1} 000A111 \Rightarrow^{2.1} 0000A1111 \Rightarrow^{2.2} 0000\varepsilon1111 = 00001111.$

Легко помітити, що при кожному застосуванні продукції (2.1) префікс поповнюється одним нулем, суфікс – одиницею, між якими знову з'являється нетермінал А. Застосування ж продукції (2.2) приводить до слова, яке є реченням, тобто у його складі не лишається нетермінального символу і процес виведення слова завершується. Отже, заданій граматикою (1.1) мові L(Г) можна дати опис:

1) ***природною мовою*** (українською): мова L(Г) містить непорожні слова, кожне з яких має префікс, що складається з нулів (0) та суфікс – з одиниць (1), і довжина префікса дорівнює довжині суфікса;

2) ***формальний (мовою теорії множин)***:

$L(\Gamma) = \{\omega_1\omega_2 \mid \omega_1 \in \{0\}^+, \omega_2 \in \{1\}^+, |\omega_1| = |\omega_2|\}$;

3) ***формальний (з використанням алгебраїчних формул)***:

$L(\Gamma) = \{0^n1^n \mid n \geq 1\}$.

Можемо впевнитись, що за правилами граматики можуть бути виведені й інші ланцюжки, що відповідають знайденому опису. Продукція 1.1 породжує слово 0A1, з якого можна отримати найкоротше речення за допомогою продукції 2.2, а саме 01. Отже L(Г) – ε-вільна мова.

Продукція 2.1 завжди збільшує на один символ (0) префікс та на один символ (1) суфікс з нетерміналом A між ними, що передбачає подальші кроки виведення. Отже, n застосувань продукції 2.1 збільшить префікс на n нулів та суфікс на n одиниць, а загальна довжина термінальних символів слова – (n + 1 + n + 1). Подальше застосування продукції 2.2 приводить до термінального слова (речення) з префіксом та суфіксом довжини n + 1 кожний. Отже, L(Г) – мова, у якій довжина префікса кожного речення дорівнює довжині його суфікса.

Приклад. Нехай G – граматика з нетермінальним алфавітом N = {S, A}, множиною терміналів T = {a, b}, початковим символом S і множиною продукцій P = {S → bA, S → a, A → bb}. Знайдемо мову L(G), що породжується цією граматикою. Із початкового символу S можна вивести слово bA за допомогою продукції S → bA чи застосувати продукцію S → a, щоб вивести a. З bA, скориставшись продукцією A → bb, можна вивести слово bbb. Оскільки інших виведень не існує, то L(G) = {a, bbb}.

*Приклад***.** Розглянемо граматику G = (N = {S}, T = {0, 1}, S, P = {S → *ε*, S → 0S1}) і знайдемо мову, що породжується нею. Граматика G містить єдиний нетермінальний символ і єдине правило S → 0S1, права частина якого містить цей символ. Таким чином, єдиними виведеннями граматики G є виведення форми: $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow \ldots \Rightarrow 0^nS1^n \Rightarrow 0^n e1^n = 0^n1^n$.

Отже, $L(G) = \{0^n 1^n \mid n \geq 0\}$.

Приклад. Розглянемо граматику G = (N = {S, A, B}, T = {a, b}, S, P), де множина продукцій P містить наступні правила: S → ABA, A → a | bb, B → bS | e, і знайдемо мову L(G). Оскільки нетермінальний символ A може тільки продукувати термінальні символи, то його заміщення слід відкласти наостанок. Таким чином, виведення граматики G мають наступну загальну форму:

$S \Rightarrow ABA \Rightarrow AbSA \Rightarrow AbABAA \Rightarrow AbAbSAA \Rightarrow A(bA)^2BA^3 \Rightarrow \ldots \Rightarrow A(bA)^nBA^{n+1} \Rightarrow A(bA)^nA^{n+1}$.

Насамкінець замінюємо нетермінал A терміналами a та bb, і одержуємо слова вигляду:

$(a + bb)(ba + bbb)^n (a + bb)^{n+1}$, $n \geq 0$.

Таким чином, L(G) складається з усіх побудованих вище слів.

Розглянемо загальну послідовність кроків, яка показує, як ***побудувати граматику за описом мови.***

Будувати граматику можна у такій послідовності :

1. Визначити множину терміналів (алфавіт мови).

2. Побудувати якусь кількість речень у відповідності до опису.

3. Записати правила граматики для виведення слів мови.

4. Впевнитись, що за правилами побудованої граматики виводяться й інші ланцюжки, що відповідають даному опису мови, (бажано – ”ті і тільки ті” замість ”й інші”, що, як вже зазначалось, не завжди можливо);

5. Записати граматику.

Приклад виконання такого завдання для мови, кожне слово якої починається з двох літер ”a” та закінчується літерою ”b”, між якими знаходиться нуль чи більше літер ”c” .

Множина терміналів T = {a, b, c}.

2. Приклади речень мови: aab, aacb, aaccb, aaccccccccb.

3. Записати правила граматики для виведення слів мови: {S → aaAb, A → cA | c | ε}.

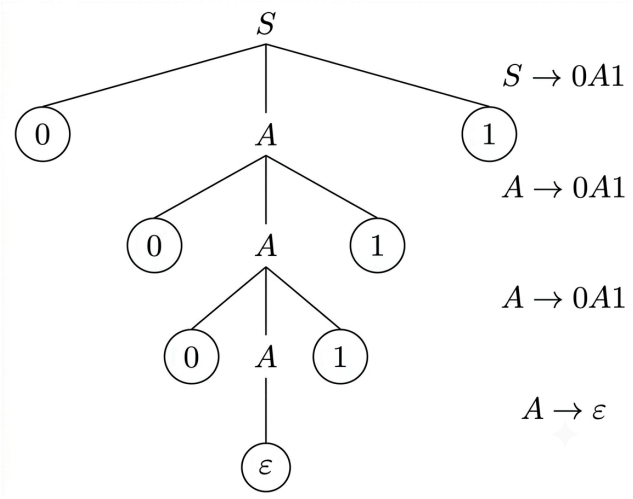
4. Впевнитись, що побудована граматика відповідає опису мови:

Стартовий символ породжує єдине слово aaAb. Наступне застосування правила A → ε породжує речення aab, – нуль літер c. Застосування до слова aaAb продукції A → cA породжує слово aacAb з – однією літерою c та нетерміналом A. Наступні кроки виведення приводять до слова aacb при застосуванні продукції A → ε або до $aac^{n+1}b$ при наступному n-кратному застосуванні продукції A → cA та завершальною A → ε. Отже, граматика породжує слова з префіксом aa, суфіксом b та літерою c між ними у кількості нуль, або більше.

5. Граматика даної мови: Г = ({A, S}, {a, b, c}, {S → aaAb, A → cA | ε}, S)

Дерево виведення

Повертаючись до виведення, зауважимо, що покрокове представлення процесу виведення у формі дерева, називають деревом виведення. На рис. наведено виведення слова 000111 у граматиці 1.1, де, для зручності, показано також продукції, застосовані на кожному кроці виведення. Конкатенація термінальних вузлів (листків) дерева виведення дає слово, що належить мові. Для внутрішніх та термінальних вузлів дерева використовують різні графічні позначення, тут термінальні вузли (листки) позначені символом у колі, нетермінальні – просто символом. Порожній рядок, оскільки нічого не породжує, як і термінальні символи, позначено колом.

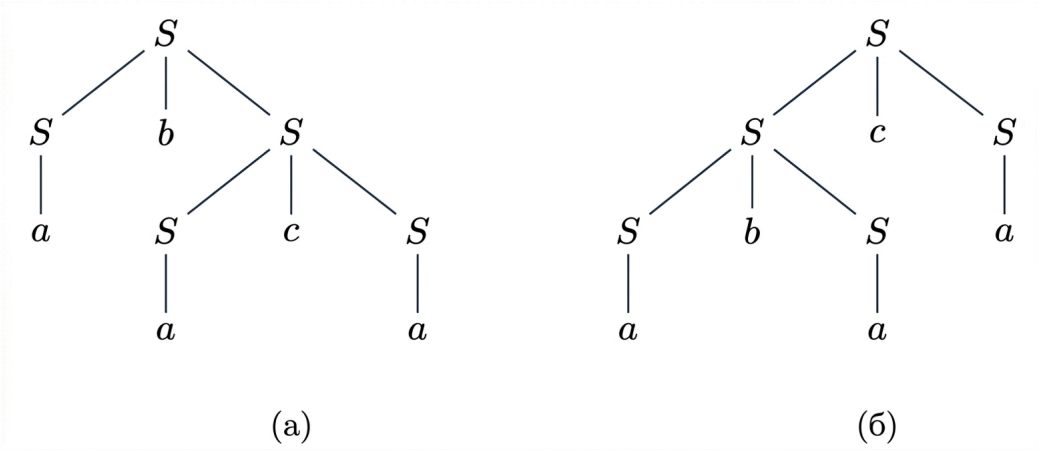


Одне й те ж дерево виведення може відповідати декільком виведенням, які відрізняються тільки порядком, в якому обирають нетермінали слова для застосування продукцій граматики. Щоб забезпечити одноманітність виведення при побудові одного й того ж дерева, застосовують, наприклад, ліве виведення – це коли спершу обирають найлівіший нетермінал. Аналогічно визначається праве виведення. При виведенні обов'язково зазначають, яке саме виведення використовується. Якщо слово ω ∈ L(Г) має два або більше різних дерев виведення, то граматику Г називають неоднозначною.

Наприклад, розглянемо контекстно вільну граматику

G = ({S}, {a, b, c}, S, P), де P = {S → SbS | ScS | a} і слово abaca ∈ L(G).

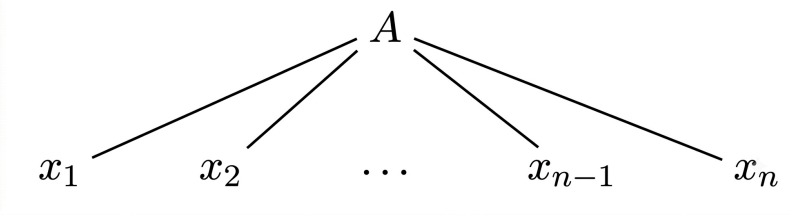
Виведення S ⇒ SbS ⇒ SbScS ⇒ abScS ⇒ abSca ⇒ abaca показано на рисунку (а), а виведення S ⇒ ScS ⇒ SbScS ⇒ abScS ⇒ abSca ⇒ abaca – на рисунку (б).



В загальному випадку для контекстно вільної граматики G = (N, T, S, P) дерево виведення будується наступним чином:

1. Вершиною дерева є початковий нетермінальний символ S.

2. Повторюємо наступні кроки доти, доки кожен листок дерева буде або порожнім символом e, або термінальним символом: для кожного листка A ∈ N дерева обираємо продукцію A → $x_1x_2 \ldots x_n$ граматики G, де $x_i \in N \cup T$, і замінюємо вершину A ∈ N деревом:



Якщо конкатенація листків дерева виведення є словом ω ∈ T^*, то кажемо, що дане дерево є деревом виведення для слова ω.

Дерево виведення слова ω відповідає виведенню ω в граматиці G. Часто дерево виведення відповідає більш ніж одному виведенню слова ω. Наприклад, для граматики G, визначеної вище, є 16 різних виведень слова ω = abaca в граматиці G. Серед них 8 виведень, які починаються з S ⇒ SbS, зображуються одним і тим самим деревом виведення, показаним на рисунку (а). Решту 8 виведень, що починаються з S ⇒ ScS, зображені деревом виведення на рисунку (б). Щоб зробити відповідність між деревами виведення і виведеннями в граматиці G взаємно однозначною, назвемо виведення слова ω ∈ L(G) лівим виведенням, якщо завжди застосовується відповідна продукція граматики G для заміни найлівішого нетермінального символу в проміжному слові при виведенні слова ω. Очевидно, що кожне дерево виведення слова ω відповідає єдиному лівому виведенню слова ω. Наприклад, серед восьми виведень для ω = abaca, які починаються з S ⇒ SbS, єдиним лівим виведенням є S ⇒ SbS ⇒ abS ⇒ abScS ⇒ abacS ⇒ abaca.

Типи формальних мов і граматик

У теорії формальних мов виділяють 4 типи граматик, яким відповідають 4 типи мов. Такий розподіл називається «граматична структура Хомського». Ці граматики виділяються шляхом накладення обмежень на правила граматики.

У граматиці типу 0 немає жодних обмежень на продукції. Граматика типу 1 може мати лише продукції вигляду ξ → η, де довжина слова η не менша, ніж довжина слова ξ, або у вигляді ξ → e. У граматиці типу 2 можуть бути лише продукції A → ω, де A – нетермінальний символ, ω ∈ (N ∪ T) ∗ . Граматика типу 3 може мати такі продукції: A → ωB, A → ω і A → e, де A, B – нетермінали, ω – слово з терміналів.

Граматики типу 0. Граматики типу 0, що називаються граматиками загального вигляду, не мають ніяких обмежень на правила породження. Будь-яке правило P = {χ →ψ} може бути побудоване з використанням довільних ланцюжків η, ψ ∈ $(N \cup T)$. Наприклад, RLW→FWTxAb→ xrtHVD.

Граматики типу 1. Граматики типу 1, що називаються також контекстно-залежними граматиками, не допускають використання будь-яких правил. Правила виведення в таких граматиках повинні мати вигляд: $\chi_1 A \chi_2 \rightarrow \chi_1 \omega \chi_2$, де χ_1, χ_2 – ланцюжки, можливо порожні, з множини (VT∪Va), символ A∈Va і ланцюжок ω∈(VT∪Va). Ланцюжки χ_1 і χ_2 залишаються

незмінними при застосуванні правила, тому їх називають контекстом (відповідно лівим і правим), а граматику – контекстно залежною. Граматики типу 1 значно зручніші на практиці, ніж граматики типу 0, оскільки в лівій частині правила замінюється завжди один нетермінальний символ, який можна зв'язати з деяким синтаксичним поняттям, у той час як у граматиці типу 0 можна заміняти відразу кілька символів, у тому числі й термінальних.

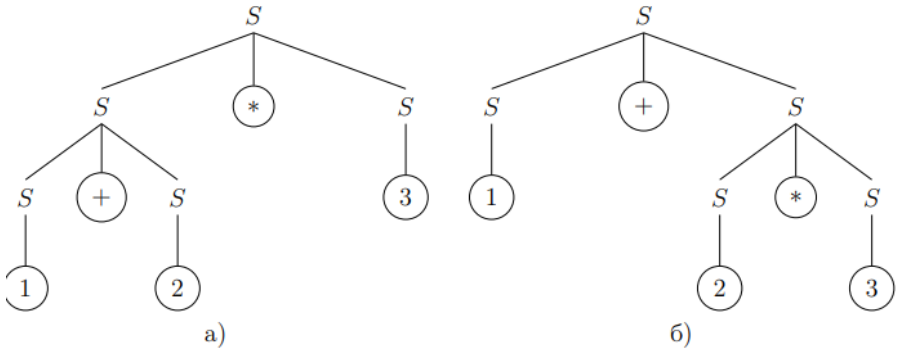


Рис. Дерева виведення слова 1 + 2 ∗ 3 у граматиці Γ_1.

На рис. представлено два різних виведення слова 1 + 2 ∗ 3 у граматиці Γ_1 = ({S}, {1, 2, 3, +, ∗}, {S → S + S | S ∗ S | 1 | 2 | 3}, S). Вона містить єдиний нетермінал S і тому вибір дерева визначається тільки порядком застосування продукцій: дерево а) на рис., якщо першою застосовується продукція S → S ∗ S, або дерево б), рис., якщо спочатку – S → S + S.

Граматики типу 2. Контекстно-вільні граматики. Це найважливіший клас граматик для програмістів. Саме вони описують синтаксичну структуру більшості мов програмування (блоки if-else, математичні вирази з дужками, структуру функцій).

Обмеження на правила: ліва частина правила має складатися строго з одного нетермінала, а права частина – бути довільним рядком: $A \to \gamma$, де $A \in V_N$, $\gamma \in (V_N \cup V_T)^*$.

Чому «контекстно-вільні»? Тому що нетермінал A можна замінити на γ у будь-якому місці ланцюжка, незалежно від того, які символи стоять ліворуч чи праворуч від нього. Контекст не має значення.

Приклад мови: мова паліндромів типу $L = \{\omega\omega^R | \omega \in \{a, b\}^*\}$.

Граматики типу 3. Регулярні граматики. Найбільш обмежений, але водночас найпростіший та найшвидший в аналізі клас граматик.

Обмеження на правила: дозволяються правила лише одного з двох видів (для праволінійних граматик):

1. $A \to aB$

2. $A \to a$, де A, B – нетермінали, а a — термінал). Праворуч може стояти або один термінал, або термінал із одним нетерміналом в кінці.

Приклад мови: $L = \{a^n b^m \mid n,\ m \geq 1\}$.

Питання для самоконтролю

1. Що таке формальна граматика як математичний об'єкт? Опишіть її четвірку основних компонентів та поясніть відмінність між термінальним алфавітом (алфавітом мови) і нетермінальним алфавітом (алфавітом граматики).

2. Як визначається поняття «мова, породжена граматикою»? Опишіть роль правил заміни (продукцій) у процесі виведення та вкажіть відмінність між безпосереднім виведенням і повним виведенням.

3. Яку граматику називають неоднозначною (абстрактною)? Наведіть приклад.

4. Які граматики вважаються еквівалентними? Чи означає еквівалентність граматик те, що вони мають однакові набори правил виведення та нетермінальних символів?

5. Опишіть класифікацію граматик і мов за ієрархією Хомського.

Тема 18. Автомати та способи їх задання

Мета вивчення: ознайомити студентів з поняттям перетворення інформації за допомогою автоматів; сформувати розуміння поняття і принципу дії автомата; розглянути автомати Мілі та Мура; формувати вміння виконувати створення таблиці переходів і виходів та графи переходів.

План

1. Автомати: основні поняття
2. Автомати Мілі та Мура.
3. Створення таблиці переходів і виходів та графи переходів.

Література:

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

Текст лекції

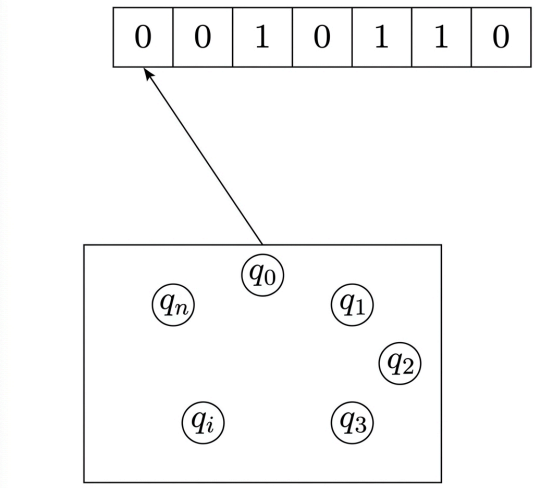
1. Автомати: основні поняття

Існує багато різноманітних методів перетворення інформації. Алгоритм роботи перетворювачів інформації (автоматів) можна інтуїтивно визначити як деяку сукупність правил, за якими даний пристрій перетворює вхідну інформацію у вихідну. Щоб перейти до формальних методів опису процесу

перетворення інформації з допомогою автоматів, розглянемо деякий пристрій, який має n входів і m виходів. Будемо вважати, що на входи інформація подається в дискретні моменти часу t_0, t_1, . . . , t_n. Інтервал часу $t_{i+1} - t_i$ називається ***тактом.*** Аналогічно і з виходів інформація одержується дискретно в часі. На кожному із входів (виходів) сигнал може приймати дискретні значення з деякого набору u_0, u_1, . . . , u_q. Сукупність значень вхідних сигналів у момент часу t_i називатимемо ***вхідною буквою***, а послідовність вхідних букв – ***вхідним словом***. Множину всіх вхідних букв назвемо ***вхідним алфавітом***. Аналогічні означення відносяться і до вихідних сигналів.

Можна уявляти будь-який ***перетворювач інформації*** як пристрій з одним входом, на який надходять слова, складені з букв вхідного алфавіту, і з одним виходом, з якого отримуються слова в вихідному алфавіті. Процес перетворення інформації в такому пристрої зводиться до встановлення деякої відповідності між словами вхідного та вихідного алфавітів. ***Автоматом*** називають перетворювач алфавітної інформації, який має один вхід і один вихід. Для задання умов функціонування автомата фіксуються три множини, елементами яких є букви трьох алфавітів: вхідного $X = \{x_1, . . . , x_m\}$, вихідного $Y = \{y_1, . . . , y_l\}$ і множини внутрішніх станів автомата $Q = \{q_0, . . . , q_n\}$.

Автомат складається з трьох частин: стрічки, головки і керуючого пристрою. Стрічка використовується для запису вхідних даних (вхідного слова). Вона поділена на скінченну кількість комірок.



Головка переглядає стрічку, читає букву з неї і передає одержану інформацію керуючому пристрою. Кожного разу головка переглядає тільки одну комірку стрічки, читає записану в ній букву і переходить до наступної комірки праворуч. Керуючий пристрій складається з елементів множини станів Q. Автомат функціонує в дискретні моменти часу, які прийнято позначати цілими невід'ємними числами t = 0, 1, . . . , k. В початковий момент часу t = 0 керуючий пристрій завжди знаходиться в початковому стані $q_0 = q(0)$, в якому автомат починає роботу над вхідним словом. Потім за активним станом і зчитаною головкою буквою він визначає, в який стан потрібно перейти, і яку

букву треба написати на місці прочитаної вхідної букви. На згаданих вище множинах задаються дві функції. Функція переходів δ, яка визначає стан автомата $q(t + 1) \in Q$ в момент часу $t + 1$ в залежності від його стану $q(t) \in Q$ і значення вхідного сигналу $x(t) \in X$ в момент часу t: $q(t + 1) = \delta(q(t), x(t))$. Тобто зміна станів визначається за допомогою функції двох змінних $\delta : Q \times X \rightarrow Q$. Функція виходів $f : Q \times X \rightarrow Y$ визначає значення вихідного сигналу $y(t) \in Y$ в залежності від стану автомата $q(t) \in Q$ і значення вхідного сигналу $x(t) \in X$ в момент часу t. Таким чином, якщо керуючий пристрій перебуває в стані $q \in Q$ і головка зчитує з стрічки букву $a \in X$, то керуючий пристрій переходить в новий стан $p = \delta(q, a)$ і на місці букви a записується буква $b = f(q, a)$. Коли на вхід автомата подавати слово (послідовність вхідних букв), на виході також з'являється слово (послідовність вихідних букв). У цьому випадку автомат A можна розглядати як алфавітний перетворювач інформації, який перетворює слова вільної напівгрупи F(X) в слова вільної напівгрупи F(Y). Визначена таким чином словесна функція $gA: F(X) \rightarrow F(Y)$ називається функцією, визначеною автоматом A, або автоматною A-функцією. Для повного опису автомата потрібно виділити множину $F \subset Q$ кінцевих станів, в яких він підтверджує, що вхідне слово належить шуканій мові. Таким чином, ми приходимо до такого визначення автомата.

2. Автомати Мілі та Мура

Сімка ***A = (Q, X, Y, δ, f, q0, F) називається автоматом Мілі***, якщо вона складається із множини станів автомата Q, множини вхідних букв X, множини вихідних букв Y, функції переходів $\delta : Q \times X \rightarrow Q$, функції виходів $f : Q \times X \rightarrow Y$, початкового стану q_0 та множини кінцевих станів. Множини X і Y називають відповідно ***вхідним і вихідним алфавітами автомата***. Якщо $\delta(a, x) = a_0$, то кажуть, що автомат Q під дією вхідного сигналу $x \in X$ переходить в стан a_0, або сигнал x переводить автомат Q із стану a в стан a_0 . Якщо $f(a, x) = y$, то кажуть, що автомат Q перетворює в стані a вхідний сигнал $x \in X$ у вихідний сигнал $y \in Y$. Як саме працює автомат на вхідному слові? На початку роботи керуючий пристрій перебуває в початковому стані q_0, на стрічці записане вхідне слово і головка зчитує букву з першої зліва комірки. Потім автомат працює крок за кроком відповідно до функцій переходів δ та виходів f. Він зупиняє роботу тоді, коли головка прочитає букву, записану в останній справа комірці стрічки. Коли обробка автоматом вхідного слова завершилася, то кажуть, що автомат розпізнає вхідне слово, якщо він зупиняється в одному з кінцевих станів множини F. В протилежному випадку, кажуть, що вхідне слово не розпізнається автоматом. Якщо ж множина F кінцевих станів не задана, то вважається, що автомат розпізнає всі слова вхідного алфавіту. Формальна мова всіх слів вхідного алфавіту X, які розпізнаються автоматом A, позначається L(A). В цьому випадку кажуть, що мова L(A) розпізнається автоматом A.

Множина всіх вихідних слів автомата A, які є результатом роботи автомата на мові L(A), називається ***мовою, що породжується автоматом A***. Автомат A = (Q, X, Y, δ, f, q0, F) називається ***скінченним***, якщо всі три множини: Q, X і Y – скінченні, і ***нескінченним***, якщо хоч одна з них нескінченна. Автомат називається ***повним*** або ***повністю визначеним***, якщо функції переходів і виходів всюди визначені, і частковим, якщо хоч одна з них є частково визначеною функцією. Іноді трапляються автомати, в яких компонент δ – не функція, а деяке відношення, тобто в таких автоматах не виконується умова однозначності переходу. Автомати такого типу називаються ***недетермінованими***. Якщо ж відношення δ є функцією, то автомат називається ***детермінованим***.

Автомат без пам'яті – це трійка (X, Y, f), де $f : X \to Y$. Цей автомат виконує побуквену трансформацію букв вхідного алфавіту X у букви вихідного алфавіту Y, при якій на одну і ту ж вхідну букву $x \in X$ він реагує одною і тою ж вихідною буквою $y \in Y$. Поняття автомат без пам'яті – основне в теорії перемикальних схем, яка вивчає способи подання таких автоматів мережами із логічних елементів.

Автономний автомат – це четвірка (Q, Y, δ, f), де $\delta : Q \to Q$, $f : Q \to Y$. Для такого автомата поданням початкового стану визначається весь подальший процес його функціонування.

Автомати Мура є окремим випадком автоматів Мілі. Автомат A = (Q, X, Y, δ, f, q_0, F) називається автоматом Мура, якщо f(q, x1) = f(q, x2) для кожних $q \in Q$ та x1, x2 $\in$ X. В автоматі Мура вихідний сигнал у момент часу t однозначно визначається станом автомата в той же момент часу, тобто y(t) = h(q(t)), де h: Q → Y. Функція h називається функцією відміток автомата, а її значення h(a) на стані a – відміткою цього стану.

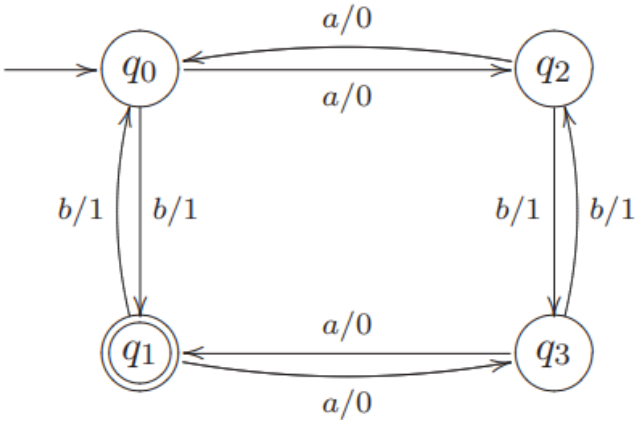
3. Створення таблиці переходів і виходів та графи переходів

У загальному випадку для задання скінченних автоматів користуються двома стандартними способами, які називаються універсальними – це таблиці переходів і виходів та графи переходів і виходів. Таблиці переходів і виходів автомата – це дві матриці однакової розмірності, стовпчики яких відмічені різними буквами вхідного алфавіту, а рядки – різними символами станів автомата. Для функції переходів (перша матриця) на перетині i-го рядка, відміченого станом $a \in Q$, і j-го стовпчика, відміченого буквою $x \in X$, знаходиться значення δ(a, x) = b. Аналогічно для функції виходів (друга матриця) на перетині i-го рядка і j-го стовпчика знаходиться значення f(a, x) = $y \in Y$. Вважають, що початковий стан автомата відмічає перший рядок як в таблиці переходів, так і в таблиці виходів.

Приклад. Автомат A=({q0, q1, q2, q3}, {a, b}, {0, 1}, δ, f, q0, {q1}) задається такими таблицями переходів і виходів:

δ	a	b
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

f	a	b
q_0	0	1
q_1	0	1
q_2	0	1
q_3	0	1



В якому б стані даний автомат не перебував, він замінює букви a і b на 0 і 1 відповідно. Таким чином, скінченний автомат A переписує будь-яке слово в алфавіті {a, b} в слово в алфавіті {0, 1}. Автомат A не є автоматом Мура, оскільки перебуваючи в стані q_0 він замінює букву a на 0, а букву b – на 1.

Граф переходів і виходів скінченного автомата A є альтернативним шляхом для зображення A. Він являє собою помічений орієнтований граф, вершини якого взаємно однозначно відповідають станам автомата. Стрілки графа відмічені парами букв: перша – буква вхідного алфавіту, друга – буква вихідного алфавіту. У даному випадку відповідність між графом переходів і виходів та функціями переходів δ і виходів f така, що коли $\delta(a, x) = a_0$ і $f(a, x) = y$, то в графі з вершини a у вершину a0 веде стрілка, відмічена парою (x/y), і навпаки, коли в графі переходів і виходів автомата існує стрілка (a, a_0), відмічена парою (x/y), то для функцій переходу δ і виходу f виконуються рівності $\delta(a, x) = a\ 0$ і $f(a, x) = y$. Крім того, щоб підкреслити початковий стан q0, проводиться стрілка без початкової вершини, кінцевою вершиною якої є стан q_0. Кінцевий стан позначається двома концентричними колами.

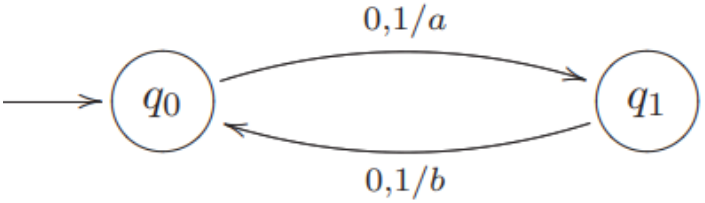
Для того, щоб відмічений граф був графом детермінованого автомата, необхідно і достатньо, щоб виконувались дві умови, які називаються умовами автоматності графа:

1) не існує двох стрілок з однаковими вхідними відмітками, що виходять з однієї і тієї ж вершини (умова однозначності);

2) для будь-якої вершини і вхідного символу існує стрілка, що виходить із цієї вершини і відмічена цим вхідним символом (умова повноти). Отже, для недетермінованих автоматів не виконується умова 1), а для часткових – умова 2). Незважаючи на це, як недетерміновані, так і часткові автомати теж зображують графами.

Приклад. Побудувати детермінований скінченний автомат Мура, який всі букви бінарного слова $\omega \in \{0, 1\}^*$, що стоять на непарних позиціях, замінює на букву a, а на парних – на букву b.

Граф переходів і виходів шуканого автомата A має вигляд:

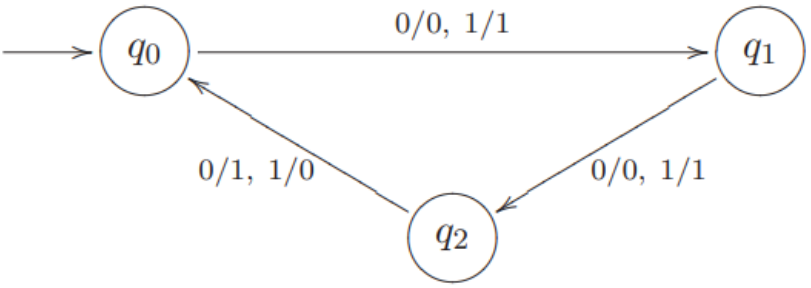


Перебуваючи в стані q_0, даний автомат замінює кожну букву вхідного слова на букву a, а в стані q1 – на букву b. Таким чином, шуканий автомат A задається як шістка A = (Q, X, Y, δ, f, q0), де Q = {q0, q1}, X = {0, 1}, Y = {a, b}, а функції переходів δ і виходів f задані таблицями:

δ	0	1
q_0	q_1	q_1
q_1	q_0	q_0

f	0	1
q_0	a	a
q_1	b	b

Приклад. Нехай X = Y = {0, 1}. Побудуємо детермінований скінченний автомат, який ”шифрує” всі бітові рядки з 0 та 1, замінюючи всі біти, які стоять на позиціях кратних 3, на протилежні. Граф переходів і виходів шуканого автомата A має вигляд:



Зауважимо, що якщо ”зашифровану” послідовність (вихідне слово) з 0 та 1 подати на вхід цього автомата, то одержимо початкову послідовність (вхідне слово), тобто цей автомат одночасно буде й ” дешифруючим пристроєм”.

Питання для самоконтролю:

1. Дайте означення скінченного автомата.

2. Які типи автоматів вам відомі? В чому полягає різниця між автоматом Мура та автоматом Мілі?

3. Які способи задання автоматів ви знаєте?

4. Коли ми кажемо, що стан q скінченного автомата A є досяжним із стану p?

5. Побудувати граф скінченного автомата A = (Q, X, Y, δ, f, q0), де Q = {q_0, q1, q2}, X = {0, 1, 2}, Y = {a, b, c}, а функції переходів δ і виходів f задані таблицями:

δ	0	1	2
q_0	q_1	q_1	q_2
q_1	q_1	q_2	q_0
q_2	q_1	q_0	q_2

f	0	1	2
q_0	a	b	a
q_1	b	c	b
q_2	c	a	b

Визначте, в яке слово перетворює автомат A кожне з наступних вхідних слів:

а) 012; б) 001022; в) 100000; г) 000111222.

СПИСОК ЛІТЕРАТУРИ

1. Висоцька В. А., Литвин В. В., Лозинська О. В. Дискретна математика : практикум (Збірник задач з дискретної математики) : навч. посіб. Львів : Новий Світ-2000, 2022. 575 с.

2. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. 6-те вид., випр. та допов. Львів : Магнолія 2006, 2022. 432 с.

3. Коцовський В. М. Дискретна математика та теорія алгоритмів. Ужгород : Видавництво УжНУ «Говерла», 2015. 95 с. URL: https://dspace.uzhnu.edu.ua/jspui/handle/lib/16302 (дата звернення: 01.05.2026).

4. Методи прикладної математики в транспортних технологіях. Елементи теорії множин : метод. рек. для студентів спеціальності «Транспортні технології» / уклад. І. І. Філімоніхіна. Кропивницький : ЦНТУ, 2022. 39 с. URL: https://dspace.kntu.kr.ua/server/api/core/bitstreams/e7e7c935-ff0b-492d-a7cc-3cd429cba0cc/content (дата звернення: 01.05.2026).

5. Дискретна математика: Конспект лекцій (Частина 1) : навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані та математичне моделювання» / уклад. О. Л. Темнікова ; КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2021. 154 с. URL: https://ela.kpi.ua/server/api/core/bitstreams/990893b6-f853-408a-8476-d3dd7c89d2a1/content (дата звернення: 01.05.2026).