Міністерство освіти і науки України
Львівський національний університет імені Івана Франка

Андрійчук В.І., Комарницький М.Я., Іщук Ю.Б.

Вступ до дискретної математики

Львів
Видавничий центр ЛНУ ім. Івана Франка
2003

УДК [510.22+519.1](075.8)
А – 65

Андрійчук В.І., Комарницький М.Я., Іщук Ю.Б., Вступ до дискретної математики. — Львів: Видавничий центр ЛНУ імені Івана Франка, 2003. — 254с.

Розглянуто множини та відношення, натуральні числа та елементи комбінаторики, бульові алгебри, графи, автомати й алгоритми; деякі методи кодування та шифрування. Підібрано вправи до кожної теми.

Для студентів молодших курсів університетів.

ББК В174.я73–1

Рекомендовано до друку кафедрою
алгебри і топології
Протокол № 2 від 22.10.2002

Рецензенти:
В.М. Усенко д-р фіз.-мат. наук, проф.
(Луганськ, Луганський державний педагогічний університет імені Тараса Шевченка);
О.Д. Артемович д-р фіз.-мат. наук, проф.
(Львів, Львівський національний університет імені Івана Франка).

Зміст

Життя прекрасне у двох випадках:
коли ти відкриваєш математику
і навчаєш математики.

С. Пуассон

Передмова

Дискретну математику часто визначають як частину математики, яка вивчає здебільшого так звані скінченні структури, тобто скінченні множини, на яких задано певні відношення, що задовольняють деякі аксіоми. Типовий приклад структур — алгебричні структури: групи, кільця, поля тощо. Теорія скінченних груп чи скінченних полів входить до розділів алгебри, а також дискретної математики. Скінченні структури виникають не лише в алгебрі, а й в геометрії, топології, математичному аналізі, теорії ймовірності та інших математичних дисциплінах. Вони є математичними моделями багатьох об'єктів і явищ природи та діяльності людини: структура атомів і молекул, розклад занять чи руху поїздів, гра в шахи, програма для комп'ютера та сам комп'ютер тощо.

На противагу дискретній математиці класична (неперервна) математика вивчає властивості неперервного характеру. Варто зауважити, що поділ математики на дискретну та неперервну дуже умовний. Вивчаючи певні задачі, доволі часто використовують дискретні та неперервні методи, що свідчить про взаємопов'язаність дискретної та неперервної математики.

До дискретної математики за традицією належать: комбінаторний аналіз, бульові алгебри, теорія графів, теорія кодування, мови та граматики, функціональні системи, скінченні автомати та деякі інші підрозділи. Дискретна математика пов'язана з усіма розділами математики, але найтісніше з алгеброю і теорією чисел, обчислювальною математикою, теорією ймовірностей, математичною логікою та іншими, в яких об'єкти вивчення мають дискретний характер. Особливу роль у дискретній математиці (як і в усій математиці) відіграє теорія множин.

Теми у посібнику розташовано нерівномірно за складністю викладеного матеріалу. Проте кожен студент може вибрати доступний для себе рівень. До кожної теми підібрано вправи з різних джерел, які можна виконувати на практичних заняттях і вдома.

Математики не мають справи з об'єктами,
а з відношеннями між об'єктами; тому вони
вільно замінюють деякі об'єкти іншими доти,
доки відношення залишаються незмінними.
Їх зміст не має значення, вони цікаві лише формою.

А. Пуанкаре

Розділ 1

Множини та відношення

Наука про множини (теорія множин) — один з найважливіших розділів математики. Як частина математики теорія множин відома з XIX ст. Перші результати одержали математики, які ставили перед собою мету — розробити основи математичного аналізу (Больцано, Дедекінд, Дюбуа-Реймон). Ці результати здебільшого пов'язані з числовими множинами або множинами функцій.

Засновник теорії множин — німецький математик Г. Кантор (1845–1918) почав розглядати довільні множини. «Під множиною розуміють об'єднання в одне ціле об'єктів, що добре розрізняються нашою інтуїцією або нашою думкою» — таке означення множини дав учений. Траплялось і таке означення множини: «це сукупність об'єктів (елементів), що мають ту чи іншу властивість». Подібні означення майже не мали жодних заперечень з боку переважної більшості математиків. Так тривало аж до початку XX ст., коли з'явились перші парадокси (антиномії) теорії множин. Найвідоміший парадокс Рассела.

1.1. Множини

1.1.1. Парадокс Рассела

Парадокс став відомим у 1903 р. Перед тим, як розглянути цей парадокс, нагадаємо, що запис $x \in y$ означає таке: x — елемент множини y. За загальноприйнятими поглядами на множину ніщо не заважало роз-

глянути множину y всіх тих множин x, що не містять себе як елемент. Запишемо цю множину y так:

$$y = \{x \mid x \notin x\}.$$

Тепер можна сформулювати запитання, чи $y \in y$? Якщо так, то за означенням y, ми повинні б мати $y \notin y$. Якщо ж $y \notin y$, то знову за цим же означенням $y \in y$. Одержали суперечність.

Приблизно тоді ж відкрили й інші парадокси. Це означало, що поняття про множину як про довільну сукупність об'єктів треба переглянути так, щоб, з одного боку, зберегти всі ті глибокі та красиві результати теорії множин, які одержали до того часу, з іншого — щоб у ній не виникали парадокси. Для досягнення цієї мети математики вибрали шлях, запропонований Д. Гільбертом (1862–1943) у праці «Основи геометрії», що вийшла у 1898 р. Суть цієї праці сформулював Д. Гільберт дещо раніше у вигляді жартівливого зауваження: «Слід добитися того, щоб з однаковим успіхом можна було говорити замість точок, прямих і площин про столи, стільці і пивні кухлі». Якщо говорити серйозніше, то основні геометричні поняття (точка, пряма і площина) не означаються. Вони виникають тільки у зв'язку з аксіомами, що описують співвідношення між ними.

Д. Гільберт розглядав об'єкти трьох різних сортів. Об'єкти першого сорту він називає точками і позначає їх буквами $A, B, C, \ldots$. Об'єкти інших двох сортів — прямими і площинами. Між цими об'єктами існують деякі відношення, які він називає інцидентністю, паралельністю, конгруентністю тощо. Властивості цих відношень описують аксіоми. Об'єкти та відношення між ними не визначаються звичними уявленнями про них. Наприклад, A і B (точки) можуть означати будь-які об'єкти за умови, що їм відповідає єдиний об'єкт l (пряма) й аналогічно для інших аксіом.

Зауважимо, що Д. Гільберт вимагав, щоб система сформульованих у його праці аксіом задовольняла такі логічні вимоги:

повинна бути *повною*, тобто такою, щоб з неї можна було вивести кожну теорему;

повинна бути *незалежною*, тобто відсутність однієї з аксіом робить неможливим доведення хоч однієї теореми;

повинна бути *несуперечливою*, тобто не дає змоги одержати дві теореми, що суперечать одна одній.

Щоб перетворити науку про множини у несуперечливу математичну теорію, потрібно було відмовитися від означень понять «множина» і «належить», а розглядати ці поняття як первісні. Тепер у теорії множин розглядається клас об'єктів (клас множин) і одне відношення $x \in y$ між парами множин x і y, яке читають «*x належить до y*» або «*x є елементом множини y*». Про ці об'єкти-множини відомо тільки те, що вони мають деякі властивості, сформульовані в аксіомах. Таку аксіоматику теорії множин подамо у п.1.1.3. цього розділу. Для того щоб її записати, у математиці існує спеціальна логічна мова $\mathcal{L}_{\mathrm{Set}}$, за допомогою якої можна записати не тільки аксіоми, а й усі теореми теорії множин. У результаті наука про множини перетворюється в логічну аксіоматичну теорію.

1.1.2. Мова $\mathcal{L}_{\mathrm{Set}}$

Вивчення формальних логічних мов, пристосованих для запису і вивчення різноманітних математичних теорій, є предметом математичної логіки. Тому ми обмежимося коротким описанням мови $\mathcal{L}_{\mathrm{Set}}$. Мова $\mathcal{L}_{\mathrm{Set}}$ дає змогу записувати і доведення теорем (ми не будемо пояснювати цю техніку, оскільки вона належить до математичної логіки).

Будь-яка мова починається з алфавіту. Алфавіт мови $\mathcal{L}_{\mathrm{Set}}$ складається з таких груп символів.

1. *Логічні* символи: $\neg$ — не, $\wedge$ — і, $\vee$ — або, $\to$ — випливає, $\leftrightarrow$ — тоді і тільки тоді, $\forall$ — для всіх, $\exists$ — існує.

Символи $\neg, \wedge, \vee, \to, \leftrightarrow$ називають *логічними зв'язками*, а символи $\forall$ та $\exists$ — *кванторами загальності* та *існування*.

2. Символи *змінних*: $x_1, x_2, \ldots, x_n, \ldots$.

Інколи, якщо можна обмежитися використанням малої кількості змінних, то для їхнього позначення використовують букви $x, y, z, \ldots$ без індексів з метою скорочення записів.

3. *Предикатні* символи: $\in$ — належить (є елементом) та $=$ — дорівнює.

4. *Допоміжні* символи: (,), ,, . (ліва та права дужки, кома, крапка).

Скінченні послідовності букв алфавіту називають *словами*. Для нас цікаві не всі слова, а лише ті, які назвемо *формулами*.

Якщо x та y — символи змінних, то слова $x = y$ та $x \in y$ називають *атомарними формулами*. Формули мови $\mathcal{L}_{\mathrm{Set}}$ одержують з атомарних за допомогою логічних і допоміжних символів.

Означення 1.1. *Формули мови* $\mathcal{L}_{\mathrm{Set}}$ — *це ті і тільки ті слова, які задовольняють такі умови:*

1) всі атомарні формули є формулами;

2) якщо A *і* B — *формули, то слова* $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \to B)$, $(A \leftrightarrow B)$, $\forall\, x A$, $\exists\, x A$ *є формулами.*

Зауваження 1.1. *Формули* $\neg x \in y$ *та* $\neg x = y$ *записують ще у вигляді* $x \notin y$ *та* $x \neq y$.

Наведемо деякі приклади формул

$$\forall z(z \in x \to z \in y), \tag{1.1}$$

$$\forall x \forall y(\forall z(z \in x \leftrightarrow z \in y) \leftrightarrow x = y), \tag{1.2}$$

$$\neg\exists y \forall z(z \in y \leftrightarrow z \notin z). \tag{1.3}$$

Якщо A — формула, x — змінна, то формули $\forall x A$ ($\exists x A$) називають *областю дії квантора* $\forall x$ ($\exists x$). Входження змінної x в формулу називають *зв'язаним*, якщо це входження перебуває в області дії квантора $\forall\, x$ або $\exists\, x$. В іншому випадку входження x називають *вільним*. Змінна x називається *вільною змінною* формули C, якщо в C знайдеться вільне входження x і *зв'язаною змінною* формули C, якщо в C знайдеться зв'язане входження x. Наприклад, змінна z зв'язана у всіх формулах (1.1), (1.2), (1.3), змінні x та y є вільними у формулі (1.1) і зв'язаними у формулах (1.2) і (1.3). Формули (1.2) і (1.3) зовсім не містять вільних входжень змінних. Такі формули у логіці ще називають *реченнями*. Зауважимо, що всі аксіоми теорії множин, які ми сформулюємо в наступному параграфі, будуть реченнями.

Якщо A — формула, яка містить вільні входження змінних $x_1, x_2, \ldots, x_n$, то, щоб підкреслити цей факт, вживатимемо запис $A(x_1, \ldots, x_n)$ замість короткого запису A.

Мова $\mathcal{L}_{\mathrm{Set}}$ створена для того, щоб чітко збудувати теорію множин. Описуючи множини за допомогою цієї мови, вважають, що символи змінних становлять деякий непорожній клас U, об'єкти $x, y, \ldots$ якого є множинами, тобто символи змінних є позначеннями для змінних множин. Крім того, два об'єкти x і y з класу U можуть бути зв'язані (або ні) одним з двох відношень:

$x = y$ (або $x \neq y$) читають: x дорівнює (або не дорівнює) y;

$x \in y$ (або $x \notin y$) читають: x належить до y (або x не належить до y).

Різниця між елементом і множиною нечітка. Кожний об'єкт класу U завжди є множиною, цей об'єкт x є елементом, якщо він записаний на першому місці у формулі $x \in y$ або $x \notin y$. Поняття *множина* та *належить* є первісними поняттями теорії множини. Тому для них не дають означень, а лише пояснюють та формулюють аксіоми, які їх зв'язують.

1.1.3. Аксіоми Цермело-Френкеля

$\mathbf{ZF}_1$. Аксіома об'ємності. Запишемо цю аксіому, використовуючи мову $\mathcal{L}_{\mathrm{Set}}$

$$\forall x \forall y \big(x = y \leftrightarrow \forall z (z \in x \leftrightarrow z \in y)\big).$$

Аксіома ZF_1 стверджує, що кожна множина цілком визначається своїми елементами. Її ще можна прочитати так: дві множини x та y рівні тоді і тільки тоді, коли кожний елемент множини x належить до множини y і навпаки, кожний елемент з y належить і до x. Тепер для зручності введемо ще одне відношення між множинами $x \subset y$ — це скорочення для формули $\forall\, z(z \in x \to z \in y)$.

Якщо $x \subset y$, то кажуть, що множина x є підмножиною множини y. Коли $x \subset y$ і $x \neq y$, то x — власна підмножина множини y. Запис $x \not\subset y$ означає, що існує елемент множини x, який не є елементом множини y.

$\mathbf{ZF}_2$. Аксіома порожньої множини. $\exists\, x \forall\, y(\neg y \in x)$. Ця аксіома гарантує існування хоч однієї множини. Множина x, існування якої стверджує аксіома ZF_2, не має жодного елемента і називається *порожньою множиною*. Вона єдина за аксіомою об'ємності ZF_1 і позначається символом $\emptyset$.

$\mathbf{ZF}_3$. Аксіома пар. Якщо задано дві множини x та y, то існує множина, єдиними елементами якої є x і y

$$\forall x \forall y \exists z \forall t (t \in z \leftrightarrow t = x \vee t = y).$$

За аксіомою об'ємності ZF_1 існує тільки одна така множина z. Її позначають через $\{x, y\}$ і називають *парою* або *невпорядкованою парою* множин x і y. Якщо $x = y$, то бачимо, що існує єдина множина $\{x, x\}$, яка має єдиний елемент x. Множину $\{x, x\}$ позначають через $\{x\}$. Треба розрізняти x і $\{x\}$. Наприклад, множина $\emptyset$ не має елементів, а множина $\{\emptyset\}$ має лише один елемент, а саме $\emptyset$.

Означення 1.2. *Множину $\{x, \{x, y\}\}$ називають впорядкованою парою множин x та y. Впорядковану пару скорочено позначають (x, y).*

$\mathbf{ZF}_4$. Аксіома об'єднання. Якщо x — множина, то існує множина y, елементами якої є елементи елементів множини x і тільки вони

$$\forall x \exists y \forall z \big(z \in y \leftrightarrow \exists t (t \in x \wedge z \in t)\big).$$

Множину y, існування якої стверджує аксіома ZF_4, називають *об'єднанням множин* t і позначають $\bigcup_{t \in x} t$. Якщо $x = \{a, b\}$ — пара, то об'єднання $\bigcup_{t \in x} t$ позначають $a \cup b$.

$\mathbf{ZF}_5$. Аксіома степеня. Для кожної множини x існує множина y, що має своїми елементами тільки підмножини множини x

$$\forall x \exists y \forall z (z \in y \leftrightarrow z \subset x).$$

Множина y єдина за аксіомою ZF_1. Її позначають 2^x і називають *множиною всіх підмножин* множини x або *буліаном* множини x.

$\mathbf{ZF}_6$. Аксіома регулярності. Кожна непорожня множина не має спільних елементів з деяким своїм елементом. Мовою $\mathcal{L}_{\mathrm{Set}}$ аксіому регулярності записуємо так:

$$\forall x \big(\neg x = \emptyset \to \exists y (y \in x \wedge y \cap x = \emptyset)\big),$$

де $y \cap x = \emptyset$ — скорочений запис для $\neg \exists z (z \in y \wedge z \in x)$.

$\mathbf{ZF}_7$. Аксіома нескінченності. $\exists x \big(\emptyset \in x \wedge \forall y (y \in x \to \{y\} \in x)\big)$.
Ця аксіома гарантує існування множини, що містить елементи $\emptyset$, $\{\emptyset\}$, $\{\{\emptyset\}\}$, $\{\{\{\emptyset\}\}\}, \ldots$, за допомогою яких у теорії множин вводяться натуральні числа $0, 1, 2, 3, \ldots$.

$\mathbf{ZF}_8$. Аксіома підстановки. Перш ніж сформулювати цю аксіому, введемо деякі позначення. Нехай $P(y)$ — формула мови $\mathcal{L}_{\mathrm{Set}}$, в яку вільно входить y. Надалі запис $\exists! y P(y)$ означатиме скорочення для формули

$$\exists y P(y) \wedge \forall x \forall y (P(x) \wedge P(y) \to x = y).$$

Цю формулу можна прочитати так: існує єдина множина y з властивістю P. Якщо до формули P вільно входять інші змінні, крім y, то $\exists! y P(y)$ треба розуміти як запис факту, що P задає y як «неявну функцію» від цих інших змінних.

Враховуючи ці скорочення, аксіому підстановки можна записати так:

$$\forall z_1 \ldots \forall z_n \forall u \Big(\forall x \big(x \in u \to \exists! y P(x, y, z_1, \ldots, z_n)\big) \to$$

$$\to \exists w \forall y \big(y \in w \leftrightarrow \exists x (x \in u \wedge P(x, y, z_1, \ldots, z_n))\big)\Big).$$

Цю аксіому можна прочитати: «якщо P задає y як функцію від $x \in u$ при довільних значеннях параметрів $z_1, z_2, \ldots, z_n$, то образ множини u щодо цієї функції є деякою множиною w».

Аксіома підстановки — одна з найскладніших і найменш очевидних аксіом теорії множин. За словами П.Дж. Коена,[1] «вона сформульована настільки продумано і ретельно, що (як прийнято вважати!) не призводить до жодної суперечності».

Звернемо увагу на те, що аксіома підстановки не є однією аксіомою, а цілою родиною аксіом (по одній аксіомі для кожної формули P). Щоб підкреслити цю обставину у такому випадку кажуть про «*схему аксіом*».

$\mathbf{ZF}'_8$. Аксіома виділення. Зі схеми аксіом підстановки виводять слабші твердження, які називають *схемами аксіом виділення*. Ми сформулюємо схему аксіом виділення у випадку, коли до формули P входить одна вільна змінна z. Тоді $P(z)$ можна інтерпретувати як твердження, що множина z має властивість P.

Мовою $\mathcal{L}_{\mathrm{Set}}$ аксіому виділення записуємо так:

$$\forall x \exists y \forall z \Big(z \in y \leftrightarrow \big(z \in x \wedge P(z)\big)\Big). \tag{1.4}$$

Множина y, існування якої стверджує аксіома ZF'_8, єдина за аксіомою об'ємності ZF_1. Для неї вводять позначення $y = \{z \in x \mid P(z)\}$.

Ще раз зауважимо, що аксіома виділення, як і аксіома підстановки, не є однією аксіомою. Знову маємо «схему аксіом» по одній аксіомі для кожної властивості P.

Спробуємо взяти на роль «аксіоми» таке дуже схоже на аксіому ZF'_8 твердження: для кожної формули $P(z)$ існує множина y, що має своїми елементами множини, які задовольняють властивість $P(z)$. У термінах мови $\mathcal{L}_{\mathrm{Set}}$ ця «аксіома» виглядала б так:

$$\exists y \forall z \big(z \in y \leftrightarrow P(z)\big). \tag{1.5}$$

Вона негайно ж призведе до суперечностей. Справді, нехай $P(z)$ означає $z \notin z$. Тоді з (1.5) одержимо

$$\exists y \forall z (z \in y \leftrightarrow z \notin z). \tag{1.6}$$

[1] П.Дж.Коен — один з найвідоміших математиків XX ст., який у 1963р. розв'язав проблему континууму в теорії множин, за що отримав філдсівську премію — найвищу міжнародну нагороду в галузі математики.

Приймемо в (1.6) $z = y$. Тоді

$$\exists y(y \in y \leftrightarrow y \notin y), \tag{1.7}$$

що є суперечністю. Ця суперечність є такою самою, як парадокс Рассела.

Аксіома виділення означає, що частина елементів z заданої множини x, які задовольняють властивість P, утворює множину. Інакше кажучи, кожна властивість визначає підмножину заданої множини x.

Аксіома ZF_8 була сформульована настільки чітко, щоб з неї не можна було вивести суперечності типу парадоксу Рассела.

Цікаво перевірити аксіому виділення (1.4) на властивість $z \notin z$. Чи можлива тепер суперечність? Спробувавши підставити в (1.4) $z \notin z$ замість $P(z)$, одержуємо

$$\forall x \exists y \forall z \big(z \in y \leftrightarrow (z \in x \wedge z \notin z)\big). \tag{1.8}$$

Приймемо в (1.8) $z = y$

$$\forall x \exists y \big(y \in y \leftrightarrow (y \in x \wedge y \notin y)\big). \tag{1.9}$$

Можемо довести таку теорему.

Теорема 1.1. $\forall x \exists y\ (y \notin x)$.

Доведення. Міркуємо від супротивного. Якби $\exists x \forall y\ (y \in x)$, то з (1.9) випливало б, що $\exists y(y \in y \leftrightarrow y \notin y)$. Але останій запис виражає суперечність.

Теорема 1.1 стверджує, що для кожної множини x існує множина y, що не належить до x або, інакше кажучи, не існує множини, яка має своїми елементами всі множини. Тому не можна говорити про множину всіх множин.

Повернемось ще раз до формули (1.6). Оскільки з неї ми одержали суперечність (1.7), то правильне заперечення формули (1.6) і, отже, доведена така теорема.

Теорема 1.2. $\neg \exists y \forall z (z \in y \leftrightarrow z \notin z)$.

Теорема 1.2 стверджує, що не існує множини, яка мала б своїми елементами множини, які не містять себе як елемент. Тому в теорії множин

не можна говорити про «множину всіх множин, які не містять себе як елемент» і парадокс Рассела остаточно знімається.

$\mathbf{ZF}_9$. Аксіома вибору. Для того щоб сформулювати цю аксіому, нам потрібні поняття функції та відображення. Нехай a і b множини. *Функцією* f називається така множина впорядкованих пар (див. аксіому ZF_3) (x,y), де $x \in a$, $y \in b$ яка містить не більше ніж одну пару для кожного $x \in a$, тобто

$$\big((x,y) \in f \wedge (x,z) \in f\big) \to y = z.$$

Якщо для кожного $x \in a$ існує такий $y \in b$, що $(x,y) \in f$, то функція f називається *відображенням* з множини a в множину b.

Тепер можна чітко сформулювати аксіому вибору

$$\begin{aligned} \forall x \Big(\neg x = \emptyset \to \exists f \big(&«f \text{ — відображення з } x \text{ в } \bigcup_{u \in x} u» \wedge \\ &\wedge \forall u \big((u \in x \wedge \neg u = \emptyset) \to \exists v (v \in u \wedge «(u,v) \in f») \big) \big) \Big), \end{aligned} \tag{1.10}$$

тобто f вибирає по одному елементу з кожного непорожнього елемента $u \in x$.

Інакше кажучи, аксіома вибору означає таке: якщо задано непорожню множину x непорожних множин u, то завжди можна вибрати з усіх множин u по елементу одноактним прийомом.

Вираз 1.10 не є записом аксіоми вибору мовою $\mathcal{L}_{\mathrm{Set}}$. Цей вираз можна перетворити у вираз мови $\mathcal{L}_{\mathrm{Set}}$, якщо записати мовою $\mathcal{L}_{\mathrm{Set}}$ фрагменти «f — відображення» та використати повний запис скорочень $\neg x = \emptyset$ та $\neg u = \emptyset$.

Аксіома вибору має особливе значення серед інших аксіом теорії множин. Вона схожа на постулат Евкліда про паралельні прямі в геометрії. Нагадаємо, якщо в геометрії замінити аксіому паралельності твердженням "через точку поза заданою прямою можна провести дві прямі, паралельні до заданої прямої", то одержимо геометрію Лобачевського несуперечливу, якщо несуперечлива геометрія Евкліда.

Нехай ZF^- — система аксіом Цермело-Френкеля без аксіоми вибору. Знаменитий австрійський математик К. Гедель у 1939 р. довів таке: якщо ZF^- несуперечлива, то вона залишається несуперечливою і після приєднання до неї аксіоми вибору, тобто і вся система аксіом

Цермело-Френкеля ZF несуперечлива. У 1963 р. американський математик П.Дж. Коен показав, що ZF^- залишається несуперечливою і після приєднання до неї заперечення аксіоми вибору.

Не всі математики беззастережно користуються аксіомою вибору. Деякі ставляться до неї з підозрою, тому що з аксіоми вибору випливають досить дивні і несподівані наслідки, які суперечать нашій інтуїції. Найвідомішим з них є так званий парадокс Банаха-Тарського: використовуючи аксіому вибору, можна розбити кулю на скінченну кількість частин, які можна переставити так, що одержимо дві кулі за розмірами рівні початковій кулі.

Доведення та коментар цього останнього результату можна знайти, наприклад, у [16]. Можна поставити запитання, чи можна обійтись без аксіоми вибору? Відповідь неоднозначна: вона залежить від конкретного розділу математики. Якщо йдеться, наприклад, про дослідження скінченних об'єктів, то можна обійтися без аксіоми вибору. У тій частині математики, яка досліджує нескінченні абстрактні структури, зокрема в алгебрі, топології, аналізі тощо важко обійтися без аксіоми вибору.

У математиці також доводиться розглядати деякі сукупності множин, які не є множинами. Наприклад, можна натрапити на такі терміни, як «клас всіх множин», «клас всіх груп» і т.д. Інтуїтивно клас означає сукупність всіх множин, які мають деяку властивість P. Деякі класи є множинами, але не всі. Наприклад, клас всіх множин, що не містять себе як елемент, не є множиною, тому що в такому випадку ми приходимо до парадоксу Рассела.

Існує ще одна поширена аксіоматика теорії множин, яка називається *аксіоматикою Геделя-Бернайса-Неймана*. Її ідея полягає в тому, що приймається інше первісне поняття — клас. Класи складаються з множин, тобто множини є елементами класів. Ми не будемо детально описувати систему аксіом Геделя-Бернайса-Неймана (її можна знайти, наприклад, у [17]). Зауважимо, що тут парадокс Рассела не виникає тому, що всі множини, які не містять себе як елемент, утворюють не множину, а клас.

Ще одна важлива проблема виникає після того, як ми прийняли ту чи іншу систему аксіом теорії множин. Чи застраховані ми від суперечностей? Усі доведення у математиці, які використовували систему аксіом Цермело-Френкеля або Геделя-Бернайса, поки-що не призвели до

суперечностей. Строге доведення несуперечливості цих аксіоматичних систем поки що ніхто не опублікував.

Якщо ж коли-небудь і виникне суперечність, то це не буде катастрофою математики, а стане нагодою для уточнення та пояснення наших уявлень про світ математики в новій ситуації.

1.1.4. Основні операції над множинами

Трохи модифікуємо наші позначення. Домовимося позначати множини великими буквами латинського алфавіту $A, B, C, \ldots, X, Y, Z$, а елементи — малими латинськими $a, b, c, \ldots, x, y, z$.

Визначимо основні операції над множинами.

Об'єднання $A \bigcup B$ двох множин A і B — це множина, існування якої гарантує аксіома об'єднання ZF_4

$$A \bigcup B = \{x \mid x \in A \vee x \in B\}.$$

Різницю $A \setminus B$ множин A і B означають так:

$$A \setminus B = \{x \in A \mid x \notin B\}.$$

Сукупність $A \setminus B$ є множиною за аксіомою виділення, а її єдиність випливає з аксіоми об'ємності.

Перетин $A \bigcap B = A \setminus (A \setminus B)$. Перетин $A \bigcap B$ можна визначити і по-іншому

$$A \bigcap B = \{x \in A \mid x \in B\}.$$

Симетричну різницю $A \oplus B$ двох множин A і B означають, наприклад, через різницю й об'єднання такою формулою:

$$A \oplus B = (A \setminus B) \bigcup (B \setminus A). \tag{1.11}$$

Симетричну різницю можна також означити через об'єднання, різницю і перетин

$$A \oplus B = (A \bigcup B) \setminus (A \bigcap B).$$

Ці операції ілюструють за допомогою так званих діаграм Ейлера-Венна (див. рис. 1.1).

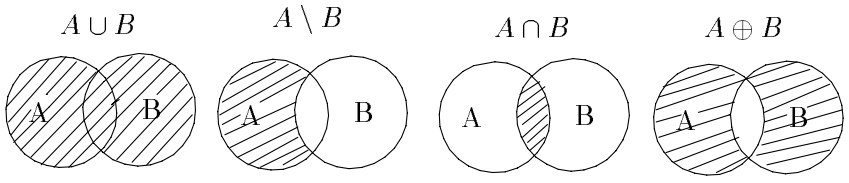


Рис. 1.1.

1.1.5. Властивості основних операцій

Теорема 1.3. *Операції об'єднання, перетину та різниці мають такі властивості:*

1) $A \bigcup A = A$, $A \bigcap A = A$ *(ідемпотентність* $\bigcup$ *та* $\bigcap$*);*

2) $A \bigcup B = B \bigcup A$, $A \bigcap B = B \bigcap A$ *(комутативність* $\bigcup$ *та* $\bigcap$*);*

3) $(A \bigcup B) \bigcup C = A \bigcup (B \bigcup C)$ *(асоціативність* $\bigcup$*),* $(A \bigcap B) \bigcap C = A \bigcap (B \bigcap C)$ *(асоціативність* $\bigcap$*);*

4) $(A \bigcup B) \bigcap C = (A \bigcap C) \bigcup (B \bigcap C)$ *(дистрибутивність* $\bigcap$ *щодо* $\bigcup$*),*
$(A \bigcap B) \bigcup C = (A \bigcup C) \bigcap (B \bigcup C)$ *(дистрибутивність* $\bigcup$ *щодо* $\bigcap$*);*

5) $A \bigcup \emptyset = A$, $A \bigcap \emptyset = \emptyset$*;*

6) $A \setminus (B \bigcup C) = (A \setminus B) \bigcap (A \setminus C)$, $A \setminus (B \bigcap C) = (A \setminus B) \bigcup (A \setminus C)$ *(закони де Моргана).*

Доведення. Обмежимось доведенням дистрибутивності перетину $\bigcap$ щодо об'єднання $\bigcup$: $x \in (A \bigcup B) \bigcap C \leftrightarrow x \in A \bigcup B \wedge x \in C \leftrightarrow (x \in A \vee x \in B) \wedge x \in C \leftrightarrow (x \in A \wedge x \in C) \vee (x \in B \wedge x \in C) \leftrightarrow x \in A \bigcap C \vee x \in B \bigcap C \leftrightarrow x \in (A \bigcap C) \bigcup (B \bigcap C)$. Пропонуємо самостійно довести інші властивості.

Зміст теореми 1.4 відображає основні властивості симетричної різниці.

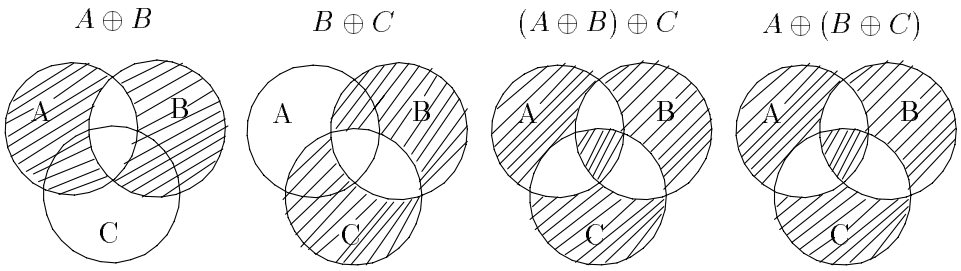


Рис. 1.2.

Теорема 1.4. *Нехай A, B, C — множини, $\emptyset$ — порожня множина.*

1. $A \oplus B = B \oplus A$.
2. $(A \oplus B) \oplus C = A \oplus (B \oplus C)$.
3. $A \oplus \emptyset = A$.
4. $A \oplus A = \emptyset$.
5. $(A \oplus B) \bigcap C = (A \bigcap C) \oplus (B \bigcap C)$.

Властивості 1, 3, 4 теореми 1.4 безпосередньо випливають з означень. Доведення властивостей 2 і 5 теж прості, проте дещо громіздкіші. Тому ми обмежимося ілюстрацією цих властивостей діаграмами Ейлера-Венна (див. рис. 1.2), запропонувавши читачеві довести їх самостійно.

1.1.6. Множина 2^M

З аксіоми степеня випливає, що для кожної множини M можна розглянути множину всіх підмножин множини M, яку прийнято позначати 2^M. Зауважимо таке: якщо множина M скінченна і має n елементів, то множина 2^M має 2^n елементів (доведіть це!).

На множині 2^M існує важлива операція доповнення $\overline{A} = M \setminus A$. Зрозуміло, що $\overline{A} \subset M$, якщо $A \subset M$. Крім того, операції об'єднання $\bigcup$, перетину $\bigcap$, різниці та симетричної різниці підмножин множини M знову приводять до підмножин множини M. Ці операції мають властивості, перераховані у теоремах 1.3 і 1.4. Додатково, легко переконатися, що правильні такі рівності: $\overline{A \bigcup B} = \overline{A} \bigcap \overline{B}$, $\overline{A \bigcap B} = \overline{A} \bigcup \overline{B}$, $A \bigcup M = M$, $A \bigcap M = A$, $\overline{\overline{A}} = A$.

Означення 1.3. *Нехай U — підмножина множини M. Відображення $f_U : M \to \{0,1\}$ для якого*

$$f_U(x) = \begin{cases} 1, & \text{якщо } x \in U, \\ 0, & \text{якщо } x \notin U \end{cases}$$

називається характеристичною функцією підмножини U.

1.1.7. Декартовий добуток

Означення 1.4. *Нехай A і B — дві множини. Декартовий добуток $A \times B$ множин A і B означають як множину всіх впорядкованих пар (a,b), де $a \in A$ і $b \in B$*

$$A \times B = \{(a,b) \mid a \in A \wedge b \in B\}.$$

Наприклад, якщо $A = \{0,1,2\}$ і $B = \{0,7\}$, тобто A складається з чисел $0,1,2$, а B — з чисел 0 і 7, то $A \times B$ складається з таких пар чисел:

$$A \times B = \{(0,0),(0,7),(1,0),(1,7),(2,0),(2,7)\}.$$

Декартовий добуток множин зручно ілюструвати за допомогою прямокутників, точки сторін яких ототожнюються з елементами множин A і B (див. рис. 1.3). Тоді елементи декартових добутків ототожнюються з точками прямокутників.

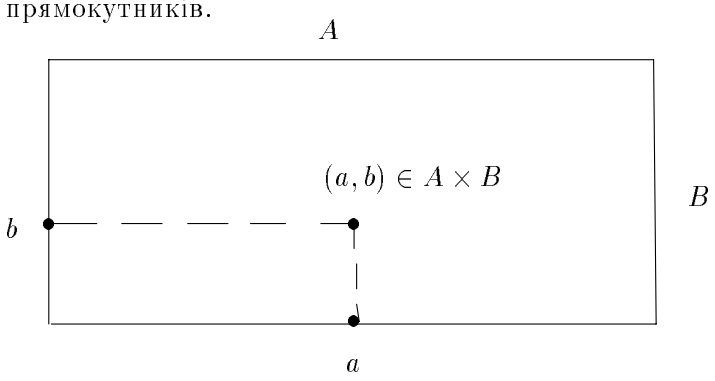


Рис. 1.3.

У наступній теоремі наведено деякі найпростіші властивості декартових добутків.

Теорема 1.5. *1.* $(A \bigcap B) \times C = (A \times C) \bigcap (B \times C)$,

$C \times (A \bigcap B) = (C \times A) \bigcap (C \times B)$.

2. $(A \bigcup B) \times C = (A \times C) \bigcup (B \times C), \quad C \times (A \bigcup B) = (C \times A) \bigcup (C \times B)$.

3. $(A_1 \bigcap A_2) \times (B_1 \bigcap B_2) = (A_1 \times B_1) \bigcap (A_2 \times B_2)$.

4. $(A \setminus B) \times C = (A \times C) \setminus (B \times C), \quad C \times (A \setminus B) = (C \times A) \setminus (C \times B)$.

5. $A \times B = \emptyset \Leftrightarrow A = \emptyset \vee B = \emptyset$.

Пропонуємо читачеві довести ці властивості самостійно.

1.1.8. Узагальнення на випадок родини множин

Нехай маємо множину B, елементи A якої є множинами. У такому випадку говорять про *родину множин*. Аксіома вибору дозволяє вибрати у кожній множині по елементу $i \in A$. Множину всіх вибраних елементів позначимо буквою $\mathcal{I}$. Якщо у множині A вибраний елемент $i \in \mathcal{I}$, то цю множину позначають A_i, родину множин B позначають $\{A_i\}_{i \in \mathcal{I}}$. Для родини множин можна визначити операції об'єднання та перетину

$$\bigcup_{i \in \mathcal{I}} A_i = \{x \mid \exists i \in \mathcal{I} \wedge x \in A_i\},$$

$$\bigcap_{i \in \mathcal{I}} A_i = \{x \mid \forall i \in \mathcal{I}\ x \in A_i\}.$$

Нехай $\{A_i\}_{i \in \mathcal{I}}$ — родина множин. *Декартовим добутком* $\prod_{i \in \mathcal{I}} A_i$ цієї родини називається множина всіх відображень (див. означення відображення в п. 1.1.3. перед формулюванням аксіоми ZF_8) $f\colon \mathcal{I} \to \bigcup_{i \in \mathcal{I}} A_i$ таких, що $f(i) \in A_i$ для всіх $i \in \mathcal{I}$.

Аксіома вибору стверджує, що декартовий добуток непорожньої родини непорожніх множин є непорожнім.

Якщо $A_i = A$ для всіх $i \in \mathcal{I}$, то скорочено позначимо декартовий добуток $\prod_{i \in \mathcal{I}} A_i$ через $A^{\mathcal{I}}$ і назвемо цю множину декартовим степенем множини A. Декартовий добуток $\underbrace{A \times \cdots \times A}_{n \text{ разів}} = \prod_{i=1}^{n} A_n$ позначається через A^n. Це множина всіх впорядкованих послідовностей з n елементів множини A (впорядкованих n-ок множини A).

1.2. Відношення

1.2.1. Означення та приклади відношень

Означення 1.5. *Бінарним відношенням R між множинами A і B називається підмножина R декартового добутку $A \times B$.*

n-місним відношенням між множинами $A_1, \ldots, A_n$ називають підмножину декартового добутку $A_1 \times \cdots \times A_n = \prod_{i=1}^{n} A_i$.

У випадку, коли $A_1 = \cdots = A_n = A$, то кажуть про n-місне відношення на множині A.

Здебільшого ми розглядатимемо бінарні відношення ($n = 2$). Наведемо декілька прикладів бінарних відношень:

1) $\{(0,1),(0,3),(1,2)\} \subset \{0,1,5\} \times \{1,2,3\}$.

$\{(0,1),(0,3),(1,2)\}$ — бінарне відношення між множинами $\{0,1,5\}$ і $\{1,2,3\}$;

2) будь-яка підмножина множини $\mathbb{R}^2$ є бінарним відношенням на множині дійсних чисел. На рис. 1.4 зображено декілька таких відношень. Спробуйте впізнати серед них відомі відношення між числами;

3) відношення на скінченній множині $A = \{a_1, \ldots, a_n\}$ можна описувати матрицями або графами. Пояснимо на прикладі як це робити. Нехай $A = \{0,1,2,3\}$, $R = \{(0,1), (0,3), (1,2), (1,3), (2,2)\}$. Тоді відношенню R можна поставити у відповідність таблицю (матрицю цього відношення)

	0	1	2	3
0	0	1	0	1
1	0	0	1	1
2	0	0	1	0
3	0	0	0	0

Загалом відношенню на множині $A = \{a_1, \ldots, a_n\}$ ставлять у відповідність квадратну таблицю з n рядків і n стовпців, причому на перетині i-го рядка і j-го стовпця стоїть 1, якщо $(a_i, a_j) \in R$ і 0 в іншому випадку. Отже, елементами матриці відношення є значення характеристичної функції F_R підможини $R \subset A^2$.

Це ж відношення можна зобразити у вигляді графа, тобто множини точок на площині, деякі з яких з'єднані стрілками. Щоб бінарне відношення зобразити графом, ми ставимо у відповідність елементам множини $A = \{a_1, \ldots, a_n\}$ деякі точки площини. Якщо $(a_i, a_j) \in R$, то з'єднуємо точки, відповідні елементам a_i і a_j, стрілкою з кінцем у точці, відповідній a_j і з початком у точці, відповідній a_i.

Об'єднання і перетин двох відношень між множинами $A_1, \ldots, A_n$ — це відповідно об'єднання та перетин відповідних підмножин декартового добутку.

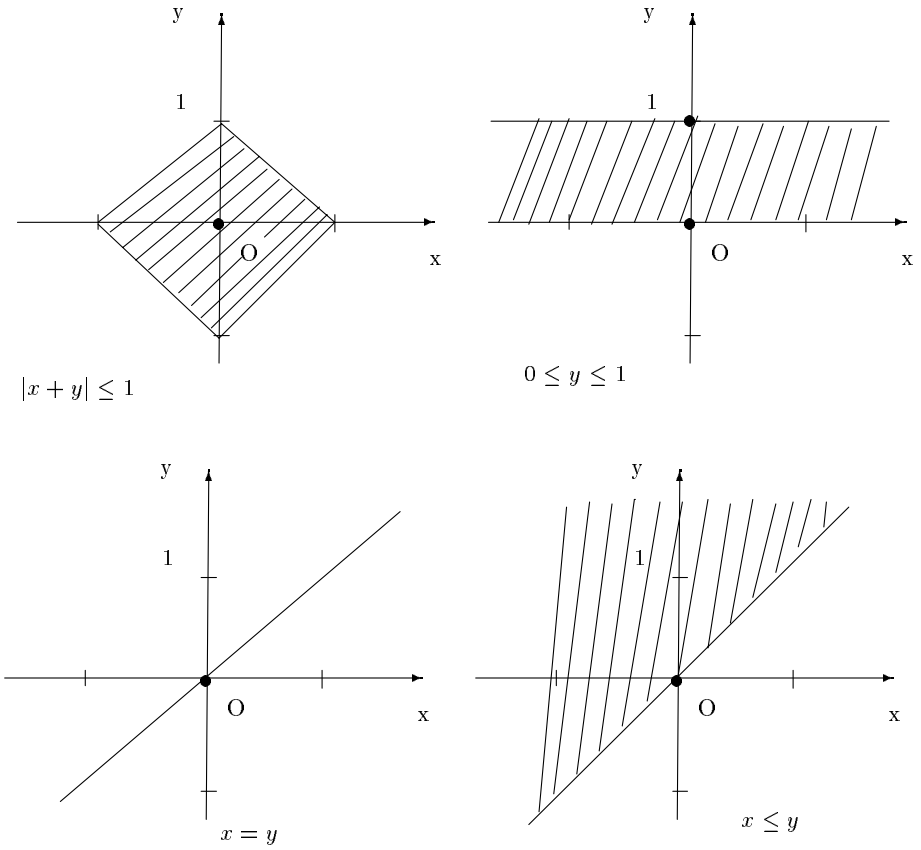


Рис. 1.4.

Якщо $R_1 \subset A \times B$ і $R_2 \subset B \times C$ — два бінарні відношення, то можна визначити *добуток відношень* $R_2 \circ R_1$. Добуток $R_2 \circ R_1$ — це така підмножина декартового добутку $A \times C$:

$$R_2 \circ R_1 = \{(a,c) \in A \times C \mid \exists b \ \ (a,b) \in R_1 \wedge (b,c) \in R_2\}.$$

Нехай $R \subset A \times B$ — бінарне відношення. Тоді обернене відношення R^{-1} є такою підмножиною декартового добутку $B \times A$:

$$R^{-1} = \{(b,a) \in B \times A \mid (a,b) \in R\}.$$

Для бінарного відношення R зручніше писати aRb замість $(a,b) \in R$.

1.2.2. Відношення еквівалентності

Означення 1.6. *Відношення R на множині A називається відношенням еквівалентності, якщо воно має такі властивості:*
1) $\forall a \in A \quad (a,a) \in R$ *(рефлексивність);*
2) $\forall a,b \in A \quad (a,b) \in R \to (b,a) \in R$ *(симетричність);*
3) $\forall a,b,c \in A \quad (a,b) \in R \wedge (b,c) \in R \to (a,c) \in R$ *(транзитивність).*

Відношення еквівалентності дуже важливе відношення у математиці. Наведемо декілька прикладів відношень еквівалентності.

1. Нехай A — довільна множина. Приймемо $(x,y) \in R$, якщо $x = y$. Легко переконатися у тому, що відношення $x = y$ — рефлексивне, симетричне та транзитивне. Отже, рівність є відношенням еквівалентності. Тому еквівалентність можна вважати узагальненням рівності.

2. Нехай M — множина всіх опуклих многокутників на площині. Для $x, y \in M$ розглянемо такі 5 відношень:
   а) xR_1y тоді і тільки тоді, коли многокутники x та y конгруентні;
   б) xR_2y тоді і тільки тоді, коли многокутники x та y мають однакову площу;
   в) xR_3y тоді і тільки тоді, коли x та y мають однакові периметри;
   г) xR_4y тоді і тільки тоді, коли вони мають однакові кількості сторін;
   д) xR_5y тоді і тільки тоді, коли x та y подібні.

Всі відношення $R_1, \ldots, R_5$ є відношеннями еквівалентності. Зауважимо, що $R_5 \subset R_4$.

3. $\mathbb{N}$ — множина натуральних чисел, $R \subset \mathbb{N}^2$

$$R = \{(m,n) \in \mathbb{N}^2 \mid m \text{ ділиться на } n\}.$$

Відношення R несиметричне, тому воно не є відношенням еквівалентності.

Якщо R — відношення еквівалентності на множині A, то замість $(a,b) \in R$ прийнято писати $a \underset{R}{\sim} b$ або, ще коротше, $a \sim b$.

1.2.3. Розбиття та відношення еквівалентності

Означення 1.7. *Якщо множина A є об'єднанням скінченної або нескінченної сім'ї множин $\{A_i\}_{i \in \mathcal{I}}$, причому $A_i \bigcap A_j = \emptyset$ для $i \neq j$, то кажуть, що задане розбиття множини A.*

Наведемо декілька прикладів розбиттів.

1. Нехай $\mathbb{Z}$ — множина цілих чисел, $2\mathbb{Z} = \{0, \pm 2, \pm 4, \ldots\}$ — множина парних чисел. Тоді $2\mathbb{Z} \bigcup (\mathbb{Z} \setminus 2\mathbb{Z})$ — розбиття множини $\mathbb{Z}$. Об'єднання

$$\{0\} \bigcup \{1,-1\} \bigcup \{2,-2\} \bigcup \cdots \bigcup \{n,-n\} \bigcup \ldots$$

є також розбиттям множини $\mathbb{Z}$.

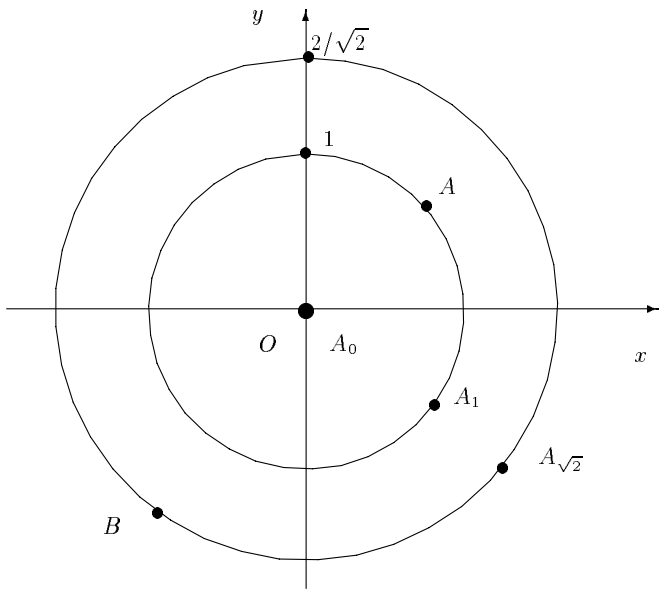


Рис. 1.5.

2. $\mathbb{R}$ — множина дійсних чисел. $A_i = \{(x,y) \in \mathbb{R}^2 \mid x^2 + y^2 = i^2\}$, де $i \in \mathbb{R}, i \geq 0$. Тоді $\mathbb{R}^2 = \bigcup_{i \in \mathbb{R}, i \geq 0} A_i$ — розбиття множини $\mathbb{R}^2$ (див. рис. 1.5).

Означення 1.8. *Нехай R — відношення еквівалентності на множині A і $a \in A$. Множина $\overline{a} = \{b \in A \mid b \underset{R}{\sim} a\}$ називається суміжним класом з представником a.*

Приклади

1. Для $x, y \in \mathbb{Z}$ $x \sim y \leftrightarrow x - y$ ділиться на 5. Суміжний клас з представником -12 це множина цілих чисел

$$-\overline{12} = \{\ldots, -12, -7, -2, 3, 8, 13, 18, \ldots\} = \{3 + 5k \mid k \in \mathbb{Z}\}.$$

2. Нехай M — множина точок площини. Для $P, Q \in M$ скажемо, що $P \sim Q$, якщо відрізки OP та OQ рівні, де O — фіксована точка площини. Легко переконатися, що так ми одержуємо відношення еквівалентності на множині всіх точок площини. На рис. 1.5 зображено три суміжні класи A_0, A_1 і $A_{\sqrt{2}}$ з представниками відповідно O, A і B.

Теорема 1.6 (критерій рівності суміжних класів).

$$\overline{a} = \overline{a_1} \leftrightarrow a \sim a_1.$$

Доведення. ($\Rightarrow$)Імплікація $\overline{a} = \overline{a}_1 \to a \sim a_1$ очевидно випливає з означень.
($\Leftarrow$)Доведемо обернену імплікацію. Нехай $b \in \overline{a}$. Тоді $b \sim a$. Але $a \sim a_1$. Тому за транзитивністю відношення $\sim$ маємо $b \sim a_1$. Отже, $b \in \overline{a}_1$ і ми довели, що $\overline{a} \subset \overline{a}_1$. Так само доводиться і протилежне включення $\overline{a}_1 \subset \overline{a}$. Отже, $\overline{a} = \overline{a}_1$.

Теорема 1.7. *Кожне розбиття множини A однозначно визначає деяке відношення еквівалентності на цій множині. Навпаки, якщо на множині A задано відношення еквівалентності, то воно визначає розбиття множини A, елементами якого є суміжні класи. Зокрема, різні суміжні класи попарно не перетинаються.*

Доведення. Якщо $A = \bigcup_{i \in \mathcal{I}} A_i$ — розбиття і $a, b \in A$, то скажемо, що $a \sim b$, якщо існує $i \in \mathcal{I}$ таке, що $a \in A_i$ і $b \in A_i$. Легко перевірити, що так означене відношення є відношенням еквівалентності.

Навпаки, нехай на множині A задано відношення еквівалентності $\sim$. Розглянемо множину всіх суміжних класів. Зрозуміло, що кожний елемент a множини A міститься у суміжному класі $\overline{a}$. Тому $A = \bigcup_{a \in A} \overline{a}$. Об'єднання $\bigcup_{a \in A} \overline{a}$ загалом не є розбиттям, тому що для різних $a, a' \in A$ ми можемо мати $\overline{a} = \overline{a'}$. Щоб одержати розбиття, розглянемо множину всіх різних суміжних класів і в кожному з них виберемо по представнику (використовуємо аксіому вибору). Нехай C — множина представників всіх різних суміжних класів, тобто така підмножина множини A, що для різних $a, b \in C$ маємо $\overline{a} \neq \overline{b}$ і для кожного суміжного класу $\overline{a}$ знайдеться $c \in C$, для якого $\overline{c} = \overline{a}$.

Переконаємося у тому, що $\bigcup_{a \in C} \overline{a}$ — розбиття множини A. Для цього достатньо показати таке: якщо $a, b \in C$ і $a \neq b$, то $\overline{a} \cap \overline{b} = \emptyset$. Справді, якби існував елемент $d \in \overline{a} \cap \overline{b}$, то $d \sim a$ і $d \sim b$, тому $a \sim b$ і за критерієм рівності суміжних класів $\overline{a} = \overline{b}$. Одержали суперечність з вибором множини C. Тому $\bigcup\limits_{a \in C} \overline{a}$ є розбиттям множини A.

Наслідок 1.1. *Якщо два суміжні класи мають спільний елемент, то вони збігаються, тобто $\overline{a} \cap \overline{b} \neq \emptyset \Rightarrow \overline{a} = \overline{b}$.*

Означення 1.9. *Якщо на множині A задано відношення еквівалентності E, то множина всіх суміжних класів щодо E називається фактор-множиною множини A за відношенням E і позначається A/E.*

1.2.4. Функціональні відношення та відображення

Означення 1.10. *Відношення $R_f \in A \times B$ називається функціональним відношенням між множинами A і B, якщо R_f задовольняє таку умову:*

$$(x, y_1) \in R_f \wedge (x, y_2) \in R_f \to y_1 = y_2, \quad \text{де } x \in A,\ y_1, y_2 \in B. \qquad (1.12)$$

Якщо $R_f \subset A \times B$ — функціональне відношення, то пишуть $y = f(x)$ замість $(x, y) \in R_f$ і кажуть, що задана функція f з множини A у множину B. Отже, за означенням поняття функціонального відношення R_f та поняття функції це просто різні назви тієї самої множини $R_f \subset A \times B$, що задовольняє умову (1.12).

Означення 1.11. *Множину $\mathcal{D}(f) = \{x \in A \mid \exists y \in B,\ y = f(x)\}$ називають областю визначення функції f, множину $\mathcal{I}mf = \{y \in B \mid \exists x \in A,\ y = f(x)\}$ називають областю значень цієї функції.*

Якщо $y = f(x)$, то y називають образом елемента x, а x — прообразом елемента y.

Множина $f^{-1}(y) = \{x \in A \mid f(x) = y\}$ називається повним прообразом елемента y.

Наприклад, множина $\{(x, \sin x) \mid x \in \mathbb{R}\}$ є функціональним відношенням на множині $\mathbb{R}$. Йому відповідає функція $f(x) = \sin x$. Повний прообраз дійсного числа 0 для цієї функції — це множина $\{\pi k \mid k \in \mathbb{Z}\}$, а повним прообразом числа -3 є порожня множина.

Множина $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$ не є функціональним відношенням, тому що, наприклад, обидві пари $(0, 1)$ і $(0, -1)$ належать цій множині, отже, умова (1.12) не виконується.

Означення 1.12. *Функція f з множини A у множину B називається відображенням з A в B, якщо $\mathcal{D}(f) = A$.*

Відображення f з множини A в множину B позначають

$$f\colon A \to B.$$

Означення 1.13. *Відображення $f\colon A \to B$ називається:*

1) *ін'єктивним, якщо для всіх $x_1, x_2 \in A$ з $f(x_1) = f(x_2)$ випливає $x_1 = x_2$;*
2) *сюр'єктивним, якщо $\mathcal{I}mf = B$;*
3) *біективним, якщо воно ін'єктивне і сюр'єктивне.*

Приклади

1. Функція f з множини $\mathbb{R}$ в $\mathbb{R}$, для якої $f(x) = x^{-1}$ не є відображенням, оскільки $\mathcal{D}(f) \neq \mathbb{R}$. Ця функція є відображенням з множини $\mathbb{R} \setminus \{0\}$ у множину $\mathbb{R} \setminus \{0\}$. Це відображення бієктивне.
2. Відображення $f\colon \mathbb{N} \to \mathbb{N}$, для якого $f(n) = n^3 + 1$ є ін'єктивним, але не сюр'єктивним.
3. Відображення $f\colon \mathbb{R} \to \mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$, $f(x) = x^2$ сюр'єктивне, проте не ін'єктивне.

1.2.5. Добуток відображень

Означення 1.14. *Добутком відображень $f\colon A \to B$ і $g\colon B \to C$ називається відображення $g \circ f\colon A \to C$, для якого $(g \circ f)(x) = g\big(f(x)\big)$ для будь-якого $x \in A$.*

Теорема 1.8. *Нехай $f\colon A \to B$, $g\colon B \to C$, $h\colon C \to D$ — три відображення. Тоді*

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

Інакше кажучи, добуток відображень асоціативний.

Доведення. Потрібно довести, що $\forall\, x \in A\ \big(h \circ (g \circ f)\big)(x) = \big((h \circ g) \circ f\big)(x)$. Маємо

$$\big(h \circ (g \circ f)\big)(x) = h\big((g \circ f)(x)\big) = h\big(g\big(f(x)\big)\big),$$

$$\big((h \circ g) \circ f\big)(x) = (h \circ g)\big(f(x)\big) = h\big(g\big(f(x)\big)\big).$$

Порівнюючи ці дві низки рівностей, бачимо, що теорему доведено.

Зауваження 1.2. *У випадку, коли визначені обидва добутки $g \circ f$ і $f \circ g$ відображень f і g, загалом не можна стверджувати, що $g \circ f = f \circ g$, тобто добуток відображень некомутативний. Щоб переконатися у цьому, розглянемо відображення $f\colon \mathbb{R} \to \mathbb{R}$, $f(x) = x+1$ і $g\colon \mathbb{R} \to \mathbb{R}$, $g(x) = 2x$. Тоді*

$$(g \circ f)(x) = g(x+1) = 2x + 2,$$

$$(f \circ g)(x) = f(2x) = 2x + 1.$$

Тому $g \circ f \neq f \circ g$.

Теорема 1.9. *1. Добуток двох ін'єктивних відображень є ін'єктивним відображенням.*

2. Добуток двох сюр'єктивних відображень є сюр'єктивним відображенням.

3. Добуток двох бієктивних відображень — бієктивне відображення.

Доведення. 1. Нехай $f\colon A \to B$, $g\colon B \to C$ — ін'єктивні відображення. Тоді, якщо $(g \circ f)(x_1) = (g \circ f)(x_2)$ для $x_1, x_2 \in A$, то $g\big(f(x_1)\big) = g\big(f(x_2)\big)$. Звідси, за ін'єктивністю відображення g, одержуємо $f(x_1) = f(x_2)$, і, отже, $x_1 = x_2$, тому що f ін'єктивне. Це означає, що відображення $g \circ f$ ін'єктивне.

2. Нехай $f\colon A \to B$, і $g\colon B \to C$ — сюр'єктивні. Покажемо, що для кожного $z \in C$ знайдеться $x \in A$, що $(g \circ f)(x) = z$. Передусім знайдеться такий елемент $y \in B$, що $g(y) = z$. Це випливає з сюр'єктивності відображення g. Тоді за сюр'єктивністю відображення f для y знайдеться такий $x \in A$, що $f(x) = y$. В результаті $g\big(f(x)\big) = z$, тобто $(g \circ f)(x) = z$ і добуток $g \circ f$ сюр'єктивний.

3. Твердження про бієктивність випливає з щойно доведених двох частин теореми.

1.2.6. Одиничне та обернене відображення

Означення 1.15. *Відображення $i\colon A \to A$ називається одиничним відображенням множини A, якщо $i(x) = x$ для кожного елемента $x \in A$. Одиничне відображення множини A позначають 1_A і часто називають тотожним відображенням.*

Означення 1.16. *Відображення $g\colon B \to A$ називається оберненим до відображення $f\colon A \to B$, якщо*

$$g \circ f = 1_A \quad і \quad f \circ g = 1_B.$$

Якщо g — відображення обернене до f, то пишуть f^{-1} замість g.

Приклади

1. Нехай $f\colon \mathbb{R} \setminus \{0\} \to \mathbb{R} \setminus \{0\}$, $f(x) = \frac{1}{x}$. Тоді $f^{-1} = f$, тому що $(f \circ f)(x) = f\big(\frac{1}{x}\big) = (x^{-1})^{-1}$. Це означає, що $f \circ f = 1_{\mathbb{R}\setminus\{o\}}$ і обидві умови з другого означення виконуються.
2. Нехай $A = \big(\frac{\pi}{2}, \frac{\pi}{2}\big)$, $B = \mathbb{R}$. Відображення $f\colon \big(\frac{\pi}{2}, \frac{\pi}{2}\big) \to \mathbb{R}$, $f(x) = \operatorname{tg} x$ і $g\colon \mathbb{R} \to \big(\frac{\pi}{2}, \frac{\pi}{2}\big)$, $g(x) = \operatorname{arctg} x$ є взаємно обернені.

Теорема 1.10. *Для відображення $f\colon A \to B$ існує обернене тоді і лише тоді, коли f бієктивне.*

Доведення. ($\Rightarrow$)Нехай $g\colon B \to A$ обернене відображення до f. Покажемо, що f — бієктивне.

Якщо $f(x_1) = f(x_2)$, то і $g\big(f(x_1)\big) = g\big(f(x_2)\big)$, тобто $(g \circ f)(x_1) = (g \circ f)(x_2)$ або $x_1 = x_2$, оскільки $g \circ f = 1_A$. Це означає, що f — ін'єктивне.

Далі ми маємо $f \circ g = 1_B$. Це означає, що для кожного $y \in B$ $(f \circ g)(y) = y$ або $f\big(g(y)\big) = y$. Елемент $x = g(y)$ є прообразом елемента y щодо відображення f, тобто f — сюр'єктивне.

($\Leftarrow$)Нехай f — бієктивне. За заданим відображенням f побудуємо відображення $g\colon B \to A$, означивши

$$g(y) = x \Leftrightarrow f(x) = y. \tag{1.13}$$

Перевіримо коректність цього правила. Нехай $R_g = \{(y, x) \in B \times A \mid g(y) = x\} = \{(y, x) \in B \times A \mid f(x) = y\}$. Якщо $(y, x_1) \in R_g$ і $(y, x_2) \in R_g$, то $f(x_1) = f(x_2) = y$, отже, $x_1 = x_2$, тому що f ін'єктивне відображення. Це означає, що R_g — функціональне відношення. Знайдемо область визначення $\mathcal{D}(g)$ функції g. $\mathcal{D}(g) = \{y \in B \mid \exists x \in A, g(y) = x\} = \{y \in B \mid \exists x \in A, f(x) = y\} = B$ завдяки тому, що f сюр'єктивне відображення.

Ми довели, що відповідність g, визначена за правилом (1.13), є відображенням. Покажемо, що $g = f^{-1}$. Враховуючи (1.13), маємо для $x \in A$, $y \in B$

$$(g \circ f)(x) = g\big(f(x)\big) = g(y) = x,$$
$$(f \circ g)(y) = f\big(g(y)\big) = f(x) = y.$$

Це означає, що $g \circ f = 1_A$ і $f \circ g = 1_B$, що і треба було довести.

1.2.7. Група $AutM$

Нехай M — довільна непорожня множина. Позначимо через $AutM$ множину всіх бієктивних відображень множини M у себе. За теоремою 1.9 добуток двох відображень з $AutM$ знову належить до $AutM$. Добуток довільних трьох відображень з $AutM$ асоціативний за теоремою 1.8. Далі $1_M \in AutM$ і для відображення $f \in AutM$ існує обернене відображення за теоремою 1.10. Разом це все означає, що множина $AutM$ є групою щодо добутку відображень.

У випадку, коли множина M має n елементів, група $AutM$ позначається через S_n. Група S_n відіграє важливу роль у різних розділах алгебри і далі матимемо нагоду вивчати її детальніше. Ця група часто виступає як інструмент для вимірювання рівня симетричності того чи іншого об'єкта, тому її називають симетричною групою.

1.2.8. Факторизація відображень

Покажемо, що кожне відображення $f\colon A \to B$ можна розкласти в добуток двох відображень — сюр'єктивного та ін'єктивного.

Визначимо за допомогою відображення f відношення E_f на множині A

$$(x_1, x_2) \in E_f \Leftrightarrow f(x_1) = f(x_2).$$

Легко переконатися в тому, що відношення E_f є відношенням еквівалентності на множині A. Ми вже знаємо (див. п. 1.2.3.), що у такому випадку можна розглянути фактор-множину A/E_f множини A, що відповідає відношенню E_f. Позначимо цю фактор-множину через $\widetilde{A}$. Елементами множини $\widetilde{A}$ є суміжні класи $\overline{x} = \{x' \in A \mid f(x') = f(x)\}$.

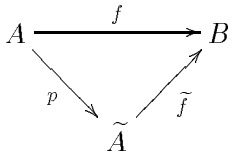
$$\widetilde{A} = \{\overline{x} \mid x \in A\}.$$

Приймемо для $x \in A$ $p(x) = \overline{x}$. Одержимо сюр'єктивне відображення $p\colon A \to \widetilde{A}$. Визначимо тепер відображення $\overline{f}\colon \widetilde{A} \to B$ так, що $\overline{f}(\overline{x}) = f(x)$. Переконаємось у тому, що $\overline{f}$ відображення. Для цього ми повинні перевірити таке: якщо $\overline{x_1} = \overline{x_2}$, то $\overline{f}(\overline{x_1}) = \overline{f}(\overline{x_2})$. Справді, якщо $\overline{x_1} = \overline{x_2}$, то за критерієм рівності суміжних класів (див. п. 1.2.3.) і за нашим означенням еквівалентності на множині A маємо $\overline{f}(\overline{x_1}) = f(x_1) = f(x_2) = \overline{f}(\overline{x_2})$. Отже, $\overline{f}$ є відображенням.

Покажемо, що $\overline{f}$ ін'єктивне. Нехай $\overline{f}(\overline{x_1}) = \overline{f}(\overline{x_2})$. Тоді $f(x_1) = f(x_2)$ і тому $\overline{x_1} = \overline{x_2}$ за критерієм рівності суміжних класів.

Залишається перевірити, що $f = \overline{f} \circ p$. Справді, $(\overline{f} \circ p)(x) = \overline{f}\big(p(x)\big) = \overline{f}(\overline{x}) = f(x)$.

Рівність відображень $f = \overline{f} \circ p$ зображають у вигляді *комутативної діаграми*.



Комутативність цієї діаграми означає, що $f = \tilde{f} \circ p$.

1.2.9. Відношення порядку

Означення 1.17. *Бінарне відношення R на множині A називається відношенням порядку, якщо воно задовольняє такі властивості:*

1) $\forall x \in A \quad (x,x) \in R$ *(рефлексивність);*

2) $\forall x,y \in A \quad \big((x,y) \in R \wedge (y,x) \in R\big) \to x = y$ *(антисиметричність);*

3) $\forall x,y,z \in A \quad \big((x,y) \in R \wedge (y,z) \in R\big) \to (x,z) \in R$ *(транзитивність).*

Якщо R — відношення порядку на множині A, то пишуть $x \le y$ замість $(x,y) \in R$. $x < y$ означає, що $x \le y$ і $x \ne y$.

Множина A з заданим на ній відношенням порядку називається частково впорядкованою.

Приклади

1. Нехай 2^M — множина всіх підмножин множини A. Для $A, B \in 2^M$ приймемо $A \le B$, якщо $A \subset B$, тобто A є підмножиною множини B. Легко переконатися в тому, що одержується відношення порядку.
2. Звичайне впорядкування (тобто $a \le b$ тоді і тільки тоді, коли $b - a$ невід'ємне) кожної з числових множин $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ є відношенням порядку.
3. На множині $\mathbb{N} \setminus \{0\}$ ненульових натуральних чисел визначимо відношення $a \le b$, якщо a є дільником b. Так визначене відношення рефлексивне, антисиметричне і транзитивне. Тому воно є відношенням порядку.

Означення 1.18. *Частково впорядкована $(A, \le)$ множина називається лінійно впорядкованою (або ланцюгом), якщо для довільних $x, y \in A$ виконується одна з двох умов $x \le y$ або $y \le x$.*

Впорядковані множини з прикладу 2 є лінійно впорядкованими, а множини з прикладів 1 і 3 не є лінійно впорядкованими.

Означення 1.19. *Елемент a частково впорядкованої множини A називається максимальним, якщо з $a \le b$ випливає $a = b$. Елемент $b \in A$ називається найбільшим, якщо $a \le b$ для кожного $a \in A$.*

Аналогічно означаємо мінімальний та найменший елементи: a — мінімальний, якщо з $b \le a$ випливає, що $b = a$ і a — найменший, якщо $a \le b$ для кожного $b \in A$.

Частково впорядкована множина A може мати кілька максимальних або кілька мінімальних елементів. Це легко зрозуміти з попередніх прикладів. Водночас, якщо в A існує найбільший (найменший) елемент, то

він єдиний. Справді, якщо, наприклад, a_1 і a_2 два найбільших елементи множини A, то маємо $a_1 \leq a_2$ і $a_2 \leq a_1$, тому $a_1 = a_2$ за антисиметричністю відношення порядку.

1.3. Навколо леми Цорна

1.3.1. Цілком впорядковані множини й ординальні числа

Означення 1.20. *Цілком впорядкованою множиною називається лінійно впорядкована множина, кожна непорожня підмножина якої має найменший елемент.*

Приклади

1. Множина натуральних чисел $\mathbb{N}$ є цілком впорядкованою множиною щодо звичайного впорядкування.
2. Кожна скінченна множина A з n елементів може бути цілком впорядкованою $n!$ способами. Справді, будь-яке бієктивне відображення $f\colon \{1, 2, 3, \ldots, n\} \to A$ з цілком впорядкованої щодо звичайного порядку множини $\{1, 2, 3, \ldots, n\}$ у множину A визначає порядок на множині A такий, що для $a, b \in A$ $a \leq b$ тоді і тільки тоді, коли $f^{-1}(a) \leq f^{-1}(b)$.
3. Множина дійсних чисел $\mathbb{R}$ не є цілком впорядкованою щодо звичайного порядку. Наприклад, інтервал $(0, 1)$ не має найменшого елемента.
4. Відношення порядку на множині A визначає відношення порядку на кожній підмножині множини A. Тому кожна підмножина цілком впорядкованої множини є цілком впорядкованою множиною.

Далі, якщо $f\colon B \to A$ ін'єктивне відображення і A — цілком впорядкована множина, то і B цілком впорядковується: $b_1 \leq b_2$ тоді і тільки тоді, коли $f(b_1) \leq f(b_2)$. Зокрема, звідси випливає, що кожна не більш, ніж зліченна множина A (тобто така множина A, для якої існує ін'єктивне відображення з A у множину $\mathbb{N}$ натуральних чисел) може бути цілком впорядкованою.

Пізніше ми побачимо, що з аксіоми вибору випливає, що на кожній множині можна задати відношення порядку, щодо якого вона стає цілком впорядкованою.

Розглянемо відображення цілком впорядкованих множин, які зберігають відношення порядку.

Означення 1.21. *Нехай $(A_1, \leq_1)$ і $(A_2, \leq_2)$ — частково впорядковані множини. Відображення $f\colon A_1 \to A_2$ називається морфізмом впоряд-*

кованих множин A_1 і A_2, якщо з того, що $x_1 < x_2$, випливає $f(x_1) < f(x_2)$.

Зауваження 1.3. *Морфізм цілком впорядкованих множин є ін'єктивним відображенням.*

Означення 1.22. *Дві цілком впорядковані множини A_1 і A_2 називаються подібними, якщо існує бієктивний морфізм $f\colon A_1 \to A_2$.*

Зрозуміло, що кожна цілком впорядкована множина A подібна собі: 1_A — бієктивний морфізм A в себе. Якщо A_1 і A_2 та A_2 і A_3 подібні і $f\colon A_1 \to A_2$, $g\colon A_2 \to A_3$ бієктивні морфізми, то і $g \circ f\colon A_1 \to A_3$ бієктивний морфізм.

Означення 1.23. *Клас всіх множин, подібних цілком впорядкованій множині A, називають порядковим типом множини A. Порядкові типи ще називають ординальними (або трансфінітними) числами. Щоб задати яке-небудь ординальне число, достатньо зазначити яку-небудь цілком впорядковану множину.*

Порядковий тип порожньої множини за означенням дорівнює 0. Порядковий тип цілком впорядкованої щодо звичайного порядку множини перших n додатних натуральних чисел $\{1, 2, 3 \ldots, n\}$ ототожнюють з натуральним числом n. Порядковий тип множини $\mathbb{N} = \{0, 1, 2, \ldots, n, \ldots\}$ позначають ω. Порядкові типи цілком впорядкованих множин $A, B, C, \ldots$ прийнято позначати малими буквами грецького алфавіту $\alpha, \beta, \gamma, \ldots$. Інколи зручно позначати порядковий тип цілком впорядкованої множини A через $\widetilde{A}$.

У таких позначеннях твердження про подібність цілком впорядкованих множин A і B записується так: $\widetilde{A} = \widetilde{B}$.

Лема 1.1. *Нехай f — морфізм цілком впорядкованої множини A в себе. Тоді $f(x) \geq x$ для всіх $x \in A$.*

Доведення. Якщо в A існують елементи x, для яких $f(x) < x$, то серед них існує найменший елемент x_1, $f(x_1) < x_1$. Для $x_0 = f(x_1)$ остання нерівність перепишеться у вигляді $x_0 < x_1$. Звідси маємо $f(x_0) < f(x_1)$, тобто $f(x_0) < x_0$. Одержана суперечність з вибором x_1 завершує доведення леми.

Означення 1.24. *Нехай A — цілком впорядкована множина і $x \in A$. Множина $A(x) = \{x' \in A \mid x' < x\}$ називається відрізком множини A, відрізаним елементом x.*

Теорема 1.11. *Не існує морфізму цілком впорядкованої множини A у відрізок якої-небудь підмножини $A' \subset A$.*

Доведення. Доводимо від супротивного. Якщо $f\colon A \to A'(x)$ який-небудь морфізм, то $f(x) \in A'(x)$, отже, $f(x) < x$, що суперечить лемі 1.1.

Наслідок 1.2. *Два різних відрізки цілком впорядкованої множини не можуть бути подібними.*

Доведення. Нехай $A(x)$ і $A(x')$ — два різні відрізки цілком впорядкованої множини A. Тоді, наприклад, $x < x'$ і $A(x)$ є відрізком $A(x')$. Залишається застосувати теорему 1.11.

Теорема 1.12. *Існує не більше, ніж один морфізм з однієї цілком впорядкованої множини в іншу.*

Доведення. Від супротивного. Нехай $f, g\colon A \to A'$ — два морфізми. Існує елемент $a \in A$ такий, що $b = f(a) \neq g(a) = b'$. Нехай, наприклад, $b < b'$. Оскільки при кожному морфізмі відрізок $A(x)$ множини A переходить у відрізок $B(y)$ множини B, де $y = f(x)$, то відрізок $A(a)$ множини A подібний до відрізків $B(b)$ і $B(b')$. Звідси випливає, всупереч наслідку з теореми 1.11, що відрізки $B(b)$ і $B(b')$ подібні. Одержана суперечність доводить теорему 1.12.

Наслідок 1.3. *Єдиним подібним відображенням цілком впорядкованої множини в себе є одиничне відображення.*

Теорема 1.13. *Якщо множини A і B цілком впорядковані, то вони або подібні, або A подібна до відрізку множини B, або B подібна до відрізку множини A.*

Доведення. Розіб'ємо доведення цієї теореми на кілька кроків.

1. Назвемо елемент a множини A *нормальним*, якщо існує елемент $b \in B$ такий, що відрізки $A(a)$ і $B(b)$ подібні. Прикладом нормального елемента множини A може бути найменший елемент $a_0 \in A$. Справді, якщо b_0 найменший елемент множини B, то $A(a_0) = B(b_0) = \emptyset$, а порожня множина подібна собі за означенням.

Нехай M — множина всіх нормальних елементів множини A. Покажемо, що $M = A$ або M збігається з деяким відрізком $A(m)$ множини A.

Передусім, якщо $a_1 < a_2$ і a_2 — нормальний елемент, то і a_1 — нормальний. Справді, для a_2 існує $b_2 \in B$, для якого відрізки $A(a_2)$ і $B(b_2)$ подібні. Тоді відрізок $A(a_2)(a_1) = A(a_1)$ подібний деякому відрізку $B(b_2)(b_1) = B(b_1)$. Отже, a_1 — нормальний елемент.

Нехай $M \neq A$ і m — найменший елемент множини $A \setminus M$. Покажемо, що $M = A(m)$. Якщо $a \in M$, то $a < m$, бо випадок $m \leq a$ неможливий за вибором m і за означенням M. Тому $a \in A(m)$ і $M \subset A(m)$.

Якщо ж $a \in A(m)$, то знову $a < m$, тому $a \notin A \setminus M$, отже, $a \in M$ і $A(m) \subset M$. В результаті ми показали, що $M = A(m)$, якщо $M \neq A$.

2. Розглянемо множину B. Аналогічно розглянемо множину N нормальних елементів множини B (тобто таких $b \in B$, для яких існує $a \in A$, що відрізки множин $A(a)$ і $B(b)$ подібні). Так само як і для множини M показуємо, що $N = B$ або $N = B(n)$, де $B(n)$ — деякий відрізок множини B.

3. Покажемо, що множини M і N подібні. Нехай $a \in M$. Це означає, що існує $b \in N$ такий, що відрізки $A(a)$ і $B(b)$ подібні. Назвемо елементи a і b *відповідними*. Якби існував ще і $b' \in N$ такий, що $b' \neq b$ і відрізки $A(a)$ і $B(b')$ були б подібними, то два різних відрізки $B(b)$ і $B(b')$ множини B були б подібними, що суперечило б наслідку з теореми 1.11. Отже, для кожного $a \in M$ існує єдиний такий елемент $b \in N$, що $A(a)$ і $B(b)$ подібні. Аналогічно для кожного $b \in B$ існує єдиний такий елемент $a \in A$, що $A(a)$ і $B(b)$ подібні. Це означає, що, зіставивши кожному елементу $a \in M$ відповідний йому елемент $b \in N$, ми одержуємо бієктивне відображення $h\colon M \to N$.

Переконаємось у тому, що це відображення h зберігає порядок. Нехай $a_1, a_2 \in M$, $a_1 < a_2$, $b_1 = h(a_1)$, $b_2 = h(a_2)$. Потрібно довести, що $b_1 < b_2$. Відрізок $A(a_2)$ подібний до відрізка $B(b_2)$, при цьому відрізок $A(a_2)(a_1) = A(a_1)$ подібний до деякого відрізка $B(b_2)(b') = B(b')$. У множині B є лише один відрізок подібний до відрізка $A(a_1)$, а саме $B(b_1)$. Отже, $b_1 = b'$ і $b_1 < b_2$, тому що $b_1 = b' \in B(b_2)$. Це завершує доведення подібності множин M і N.

4. Згідно з доведеним логічно можливі чотири випадки:

1) $M = A$, $N = B$;

2) $M = A(m)$, $N = B$;

3) $M = A$, $N = B(n)$;

4) $M = A(m)$, $N = B(n)$.

Останній випадок неможливий, бо, за доведеною подібністю множин M і N, він означав би, що m нормальний елемент, отже, $m \in A(m)$, що неможливо.

Залишаються перші три випадки. Враховуючи третій крок доведення теореми, вони якраз і означають те, що сформульовано в теоремі:

1) A і B подібні;

2) B подібна до відрізка A;

3) A подібна до відрізка B.

1.3.2. Порівняння ординальних чисел

У п. 1.3.1. вже згадувалося означення ординального числа. Нагадаємо ще раз, що задати ординальне число (порядковий тип) α означає задати цілком впорядковану множину A. Цю множину називатимемо *множиною типу* α. Натуральні числа $0, 1, \ldots, n, \ldots$ ми ототожнюємо з ординальними числами, а саме 0 задається порожньою множиною, натуральне число n задається цілком впорядкованою множиною $\{1, 2, \ldots, n\}$ зі звичайним порядком.

Теорема 1.13 з п. 1.3.1. дає змогу порівнювати два ординальні числа α і β.

Означення 1.25. *Нехай α і β — ординальні числа. Кажуть, що $\alpha < \beta$, якщо яка-небудь (і, отже, кожна) цілком впорядкована множина типу α подібна до деякого відрізка якої-небудь (і, отже, кожної) цілком впорядкованої множини типу β.*

$\alpha \leq \beta$ означає $\alpha < \beta$ або $\alpha = \beta$.

Теорема 1.14. *Для довільних ординальних чисел α і β правильна одна і тільки одна з трьох можливостей: $\alpha < \beta$, $\alpha = \beta$, $\alpha > \beta$.*

Доведення. Це безпосередній наслідок з теореми 1.13.

Теорема 1.15. *Якщо α, β і γ такі ординальні числа, що $\alpha < \beta$ і $\beta < \gamma$, то $\alpha < \gamma$.*

Доведення. Нехай A, B і C — множини типів α, β і γ. Тоді A подібна до відрізка множини B, B подібна до відрізка множини C. Тому A подібна до відрізка множини C, що й доводить теорему 1.15.

Теорема 1.16. *Множина $W(\alpha)$ всіх ординальних чисел, строго менших від α, є цілком впорядкованою множиною.*

Доведення. Нехай A — цілком впорядкована множина типу α. Зіставимо кожному елементові $a \in A$ порядковий тип відрізка $A(a)$. Одержимо множину $W(\alpha)$ всіх порядкових типів (ординальних чисел) менших від α і відображення множин $f\colon A \to W(\alpha)$. Відображення f бієктивне і зберігає порядок.

Теорема 1.17. *Будь-яка множина A, що складається з ординальних чисел, є цілком впорядкованою.*

Доведення. За теоремою 1.14 множина A є лінійно впорядкованою. Нехай $A' \subset A$ і $a' \in A'$. Якщо a' — найменший елемент в A', то доводити нічого. В іншому випадку, $W(a') \cap A' \neq \emptyset$. Оскільки за попередньою теоремою $W(a')$ цілком впорядкована множина, то $W(a') \cap A'$ містить найменший елемент a. Цей елемент a і є, очевидно, найменшим елементом множини A'.

1.3.3. Сума і добуток ординальних чисел

Нехай маємо два ординальних числа α і β. Це означає, що маємо дві цілком впорядковані множини A і B типів відповідно α і β. Вважатимемо, що $A \cap B = \emptyset$. Розглянемо множину $A \cup B$ і введемо на ній порядок так: елементи множин A і B зберігають свій порядок; крім того, для кожного $a \in A$ і для кожного $b \in B$ приймемо $a < b$. Легко переконатися в тому, що множина $A \cup B$ цілком впорядкована щодо тільки що визначеного порядку. Справді, нехай $\emptyset \neq C \subset A \cup B$. Покажемо, що C містить найменший елемент. Якщо $C \cap A \neq \emptyset$, то найменший елемент в $C \cap A$ буде найменшим в C, якщо ж $C \cap A = \emptyset$, то $C \cap B \neq \emptyset$ і найменший елемент множини $C \cap B$ буде найменшим елементом множини C. Позначимо порядковий тип множини $A \cup B$ через $\alpha + \beta$ і назвемо його *сумою ординальних чисел α і β*.

Розглянемо множину $A \times B$ і введемо на ній порядок так: $(a, b) < (a', b')$ якщо $b < b'$ або $b = b'$ і $a < a'$. Виявляється, що множина $A \times B$ є цілком впорядкованою щодо цього порядку. Справді, нехай C непорожня підмножина множини $A \times B$. Доведемо, що в множині C існує найменший елемент. Множина B' других компонент елементів множини C є підмножиною множини B. Оскільки B' — цілком впорядкована,

то в ній існує найменший елемент b_0. Тепер розглянемо підмножину $C' \subset C$: $C' = \{(a, b_0) \mid (a, b_0) \in C\}$. Нехай A' — множина перших компонент множини C'. Множина A' має найменший елемент, бо A цілком впорядкована множина. Очевидно, що елемент (a_0, b_0) є найменшим елементом множини C' і множини C. Отже, $A \times B$ цілком впорядкована множина.

Позначимо порядковий тип множини $A \times B$ через $\alpha \cdot \beta$ і назвемо його *добутком ординальних чисел α і β.*

Приклади

1. Нехай A цілком впорядкована множина типу α. $\{b\}$ — одноелементна множина, $b \in A$. Тоді ординальне число $\alpha + 1$ — це порядковий тип множини $A \cup \{b\}$, де $a < b$ для кожного $a \in A$.
2. Конкретніше. Нехай $\mathbb{N}$ — множина натуральних чисел. Тоді $\mathbb{N}$ цілком впорядкована щодо звичайного порядку множина. Нагадаємо, що її тип позначають через ω. Ординальне число $\omega + 1$ задається множиною $\mathbb{N} \cup \{a\} = \{0, 1, \dots, n, \dots, a\}$, де $\{a\}$ — довільна одноелементна множина і за означенням $n < a$ для будь-якого $n \in \mathbb{N}$. Ординальне число $1 + \omega$ задається, наприклад, множиною цілих чисел із звичайним порядком $\{-1, 0, 1, 2, \dots, n, \dots\} = \{-1\} \cup \mathbb{N}$. Множини $\{-1\} \cup \mathbb{N}$ та $\mathbb{N}$ подібні (відображення $x \mapsto x + 1$ бієктивне і зберігає порядок), а множини $\mathbb{N}$ і $\mathbb{N} \cup \{a\}$ не подібні (в $\mathbb{N} \cup \{a\}$ існує найбільший елемент, а в $\mathbb{N}$ такого елемента немає). Це означає, що $1 + \omega = \omega \neq \omega + 1$. Отже, додавання ординальних чисел некомутативне.
3. Нехай $\mathbb{N}$ — множина натуральних чисел і $A = \{0, 1\}$. Тоді

$$\mathbb{N} \times A = \{(0,0), (1,0), \dots, (n,0), \dots, (0,1), (1,1), \dots, (n,1), \dots\},$$

де порядок у фігурних дужках відповідає впорядкуванню множини $\mathbb{N} \times A$. Ця множина подібна цілком впорядкованій множині $\{1, 3, \dots, 2n-1, \dots, 2, 4, \dots, 2n, \dots\}$, а ординальне число $\omega \cdot 2$ є порядковим типом заданої множини.

$$A \times \mathbb{N} = \{(0,0), (1,0), (0,1), (1,1), (0,2), (1,2), \dots, (0,n), (1,n), \dots\}$$

також є цілком впорядкованою множиною, яка має порядковий тип $2 \cdot \omega$. Множини $A \times \mathbb{N}$ та $\mathbb{N}$ подібні, а множини $\mathbb{N}$ і $\mathbb{N} \times A$ не подібні (в $\mathbb{N} \times A$ існує елемент, який не є найменшим і не має попереднього, а в $A \times \mathbb{N}$ кожен (не найменший) елемент має попередній. Це означає, що $2 \cdot \omega \neq \omega \cdot 2$. Отже, множення ординальних чисел також некомутативне.

Теорема 1.18. *Нехай α — ординальне число. Тоді $\alpha + 1 > \alpha$ і не існує жодного ординального числа α' такого, що $\alpha < \alpha' < \alpha + 1$.*

Доведення. Якщо A — цілком впорядкована множина типу α, то задати число $\alpha + 1$ означає задати цілком впорядковану, як у прикладі 1, множину $A \cup \{b\}$. Позначимо цю множину A'. Тоді $A = A'(b)$ — відрізок в A', тому $\alpha < \alpha + 1$. Кожне ординальне число $\beta < \alpha + 1$ є порядковим

типом деякого відрізка $A'(x)$ множини A'. Якщо $x = b$, то $\beta = \alpha$, якщо ж $x < b$, то $A'(x) = A(x)$ і $\beta < \alpha$.

Означення 1.26. *Ординальне число α називається негранічним, якщо існує таке ординальне число β, що $\alpha = \beta + 1$. В іншому випадку число α називається граничним.*

Наприклад, ординальні числа $1, 2, \ldots, n, \omega + 1$ є негранічними, а ординальні числа 0, ω, $\omega \cdot 2, \ldots$ — граничні.

1.3.4. Трансфінітна індукція

1. *Доведення методом індукції.* Нехай задано деяку цілком впорядковану множину A і деяке твердження $P(x)$, що залежить від змінного елемента x множини A.

Теорема 1.19. *Якщо твердження $P(x)$ істинне для найменшого елемента a_0 множини A і, якщо з того, що $P(a)$ істинне для всіх елементів $a \in A$ таких, що $a < a' \in A$, випливає, що істинне і $P(a')$, то $P(x)$ істинне для всіх елементів $x \in A$.*

Доведення. Якщо існують елементи $x \in A$, для яких $P(x)$ хибне, то серед цих елементів існує найменший x_0, тому що множина A цілком впорядкована. Оскільки $P(x)$ істинне для всіх елементів $x \in A$, які менші від x_0, то істинне і $P(x_0)$. Одержана суперечність доводить теорему.

Якщо у цій теоремі множину A замінити множиною $\mathbb{N}$ (натуральних чисел зі звичайним порядком), то одержимо звичайний метод математичної індукції. Тому теорема 1.19 є узагальненням методу математичної індукції на випадок довільних цілком впорядкованих множин.

Приклад

Розглянемо *послідовність Фібоначчі* $\{u_n\}_{n \in \mathbb{N}}$

$$u_0 = 1, u_1 = 1, u_2 = u_0 + u_1, \ldots, u_n = u_{n-2} + u_{n-1}, \ldots. \tag{1.14}$$

Ось декілька перших членів цієї послідовності $1, 1, 2, 3, 5, 8, 13, 21, \ldots$.

Доведемо методом математичної індукції таку рівність:

$$u_0 + u_2 + u_4 + \cdots + u_{2n} = u_{2n+1}. \tag{1.15}$$

Для $n = 0$ ця рівність набирає вигляду $u_0 = u_1$. Нехай рівність вірна для всіх натуральних чисел, менших від n. Тоді, зокрема, вірна рівність

$$u_0 + u_2 + u_4 + \cdots + u_{2n-2} = u_{2n-1}. \tag{1.16}$$

Покажемо, що тоді рівність (1.15) вірна і для натурального числа n. Маємо

$$u_0 + u_2 + \cdots + u_{2n-2} + u_{2n} = u_{2n-1} + u_{2n} = u_{2n+1}.$$

Тут перша рівність випливає з (1.16), друга з означення (1.14) послідовності Фібоначчі. Рівність (1.15) доведено.

Зауваження 1.4. *У цьому прикладі ми не тільки маємо доведення методом математичної індукції, а й те, що називають «побудовою за індукцією». Щоб задати послідовність Фібоначчі* (1.14)*, ми задали два її перші члени u_0 і u_1, а всі інші члени одержали за допомогою «рекурентного співвідношення» $u_n = u_{n-2} + u_{n-1}$.*

Метод «побудови за індукцією» теж узагальнюється на довільні цілком впорядковані множини.

2. *Побудова методом індукції.* Нехай A — деяка цілком впорядкована множина і нехай нам треба поставити у відповідність кожному елементові $x \in A$ деякий об'єкт $f(x)$. Вважатимемо, що маємо для цього «рекурентне співвідношення», яке однозначно визначає об'єкт $f(b)$, як тільки визначені об'єкти $f(a)$ для всіх $a < b$. Прикладом такого рекурентного співвідношення може слугувати рівність $u_n = u_{n-2}+u_{n-1}$, за допомогою якої одержуємо послідовність Фібоначчі.

Теорема 1.20. *Нехай A — цілком впорядкована множина, M — довільна множина, a_0 — найменший елемент множини A, $m_0 \in M$. Існує лише одна функція $f\colon A \to M$, що має такі властивості:*

а) $f(a_0) = m_0$;

б) образ $f(b)$ елемента $b \in A$ однозначно визначається образами $f(a)$ елементів a, для яких $a < b$.

Доведення. Доведемо спочатку єдиність функції f. Якщо існують дві функції f і g, що задовольняють умови теореми, то існують елементи x множини A, для яких $f(x) \neq g(x)$. Нехай b — найменший з таких елементів. Тоді для всіх $a < b$ маємо $f(a) = g(a)$, звідси ми повинні мати, що і $f(b) = g(b)$. Одержана суперечність доводить єдиність функції f.

Доведемо, що функція f існує. Для цього розглянемо відрізки $A(a) = \{x \in A \mid x < a\}$. Покажемо, що на кожному відрізку $A(a)$ існує функція f, яка задовольняє умови теореми. Нехай це вже доведено для всіх $a' < a$. Розглянемо два випадки.

1. Відрізок $A(a)$ має найбільший елемент c. Відкидаючи його, одержимо відрізок $A(c)$, на якому f існує за припущенням індукції. Тоді значення $f(c)$ однозначно визначається значеннями f на відрізку $A(c)$.

2. Відрізок $A(a)$ не має найбільшого елемента. Тоді кожний елемент $c \in A(a)$ належить деякому меншому відрізку $A(a_1)$, де $c < a_1 < a$. f існує на цьому відрізку $A(a_1)$, тому значення $f(c)$ теж визначене.

Отже, функція f визначена на всіх відрізках множини A. Якщо A не має найбільшого елемента, то кожний елемент з A належить деякому відрізку і в цьому випадку все доведено. Якщо ж A має найбільший елемент m, то f визначено на відрізку $A(m)$, тому $f(x)$ визначено і для $x = m$.

1.3.5. Умова індуктивності

За деяких умов можна узагальнити результати п. 1.3.4. на ширший клас впорядкованих множин. Сформулюємо для частково впорядкованих множин таке твердження.

1. **Умова індуктивності.** Якщо всі мінімальні елементи частково впорядкованої множини A мають деяку властивість P і якщо з правильності властивості P для всіх елементів, менших від деякого елемента $a \in A$, випливає, що P правильна і для a, то всі елементи множини A мають властивість P.

Сформулюємо ще дві властивості, які можуть задовольняти частково впорядковані множини.

2. **Умова мінімальності.** Кожна непорожня підмножина B частково впорядкованої множини A містить хоч один мінімальний елемент.

3. **Умова обриву спадних ланцюгів.** Кожний спадний ланцюг елементів частково впорядкованої множини стабілізується. Це означає таке: якщо $a_1 \geq a_2 \geq \cdots \geq a_n \geq \ldots$ спадний ланцюг елементів множини A, то існує такий індекс n, що $a_n = a_{n+1} = a_{n+2} = \ldots$.

Теорема 1.21. *Умови індуктивності, мінімальності та обриву спадних ланцюгів еквівалентні, тобто з кожної умови випливають дві інші.*

Доведення. Доводимо теорему за схемою $2. \Rightarrow 1. \Rightarrow 3. \Rightarrow 2.$

$2. \Rightarrow 1.$ Нехай B — підмножина частково впорядкованої множини A, елементи якої не мають властивості P. Нехай a — мінімальний елемент

множини B. Всі елементи множини A, що менші ніж a, мають властивість P, тому і елемент a повинен мати цю властивість. Одержана суперечність засвідчує, що всі елементи множини A мають властивість P.

1. $\Rightarrow$ 3. Застосуємо умову індуктивності до такої властивості P: елемент $a \in A$ має властивість P, якщо кожний спадний ланцюг елементів, що починається з a, стабілізується. Цю властивість мають, очевидно, всі мінімальні елементи множини A. Далі нехай всі елементи, які менші від елемента a, мають властивість P. У цьому випадку кожний менший, ніж a, член кожного спадного ланцюга, що починається з a, має властивість P, тому елемент a має властивість P. З умови індуктивності одержуємо, що властивість P задовольняють всі елементи множини A, тобто у множині A стабілізується кожний спадний ланцюг елементів.

3. $\Rightarrow$ 2. Якби яка-небудь підмножина $B \subset A$ не мала б мінімального елемента, то ми могли б побудувати спадний ланцюг $a_1 > a_2 > \cdots > a_n > \ldots$, який не стабілізується.

1.3.6. Аксіома вибору та теорема Цермело

Нагадаємо, що аксіома вибору стверджує, що для кожної непорожньої родини непорожніх множин існує функція, яка кожній множині A цієї родини ставить у відповідність деякий елемент $f(A) \in A$.

Ми вже бачили, якщо множина A є цілком впорядкованою, то, зокрема, можна доводити методом трансфінітної індукції твердження, які залежать від елементів множини A, а також будувати за допомогою індукції функції, визначені на A. Тому зручно мати такий порядок на множині A, щодо якого вона є цілком впорядкованою.

Теорема Цермело гарантує, що кожну множину можна цілком впорядкувати. На жаль, жодне доведення цієї теореми не містить конкретної побудови такого порядку; доводиться лише існування потрібного впорядкування.

Теорема 1.22 (Цермело). *Кожну непорожню множину можна цілком впорядкувати.*

Доведення. Нехай A — непорожня множина. Виберемо, використовуючи аксіому вибору, по елементу $f(B)$ у кожній непорожній підмножині $B \subset A$. Назвемо підмножину $C \subset A$ *хорошою*, якщо C можна цілком впорядкувати так, що для кожного $a \in C \quad a = f\big(A \setminus C(a)\big)$, де $C(a)$

— відрізок цілком впорядкованої множини C, $C(a) = \{x \in C \mid x < a\}$. Хороші множини існують. Такою є одноелементна множина $f(A)$.

Нехай далі C_1 і C_2 — дві хороші цілком впорядковані підмножини. C_1 і C_2 містять спільний найменший елемент $f(A)$. Тоді вони мають і спільний найбільший відрізок C. Відрізок C збігається з C_1 або з C_2. Якби це було не так, то за означенням хорошої підмножини відрізок C визначався б і в C_1 і в C_2 елементом $f(A \setminus C)$, тоді, додавши до C елемент $f(A \setminus C)$, ми одержали б більший спільний відрізок.

Отже, з двох хороших цілком впорядкованих підмножин одна є відрізком іншої. Звідси випливає, що об'єднання E всіх хороших підмножин є хорошою підмножиною.

Залишається перевірити, що об'єднання E збігається з A. Якщо $A \neq E$, то ми могли б одержати більшу ніж E, хорошу підмножину, долучивши до E елемент $f(A \setminus E)$, вважаючи цей елемент більшим за всі елементи з E. Це суперечить означенню множини E і завершує доведення.

Зауваження 1.5. *З теореми Цермело випливає аксіома вибору.*

Справді, нехай $\{X_i\}_{i \in \mathcal{I}}$ — непорожня родина непорожніх множин. За теоремою Цермело кожну множину X_i можна цілком впорядкувати. Тому в кожній множині X_i існує найменший елемент x_i. Функція вибору f ставить у відповідність кожній множині X_i елемент $f(X_i) = x_i$.

1.3.7. Аксіома вибору та лема Цорна

Важливим і зручним фактом, який використовують в алгебрі, є лема Цорна. Як побачимо, вона еквівалентна аксіомі вибору. Для того щоб сформулювати лему Цорна, введемо ще одне означення.

Означення 1.27. *Родина B підмножин множини A називається індуктивною, якщо об'єднання елементів довільного ланцюга (щодо включення) підмножин множини B належить до B (пригадаємо, що ланцюгом називають лінійно впорядковану множину).*

Лема 1.2. *Якщо родина B непорожніх підмножин множини A індуктивна, то вона містить максимальний елемент, тобто підмножину M, яка не міститься в жодній множині з B, відмінній від M.*

Доведення. Зауважимо таке: якщо підмножина A_1 множини B не є максимальним елементом, то можна знайти підмножину $A_2 \in B$ таку, що $A_1 \subset A_2$. Якщо A_2 не є максимальним, то існує $A_3 \in B$, що $A_1 \subset A_2 \subset A_3$. І так далі. Тобто, ми маємо справу з ланцюгами підмножин.

1. Спочатку покажемо, що з теореми Цермело випливає, що кожний ланцюг підмножин множини A міститься в деякому максимальному ланцюзі. Нехай C деякий ланцюг в A. Якщо $C = 2^A$, то доводити нічого. В іншому випадку, розглянемо множину $D = 2^A \setminus C$. За теоремою Цермело множину D можна цілком впорядкувати. Далі використаємо побудову за індукцією. Беремо найменший елемент d_0 множини D. Скажемо, що d_0 належить до першого класу, якщо для кожного $x \in C$ маємо $x \subset d_0$ або $d_0 \subset x$, і до другого класу, якщо існує $x \in C$, для якого $x \not\subset d_0$ і $d_0 \not\subset x$. Всі елементи ланцюга C теж належать до першого класу.

Нехай $d \in D$ і всі елементи $d' \in D$, $d' < d$ вже належать до першого або другого класу. Скажемо, що елемент d належить до першого класу, якщо для кожного елемента x, який вже належить до першого класу, маємо $x \subset d$ або $d \subset x$. В іншому випадку, скажемо, що d належить до другого класу.

Використовуючи теорему про побудову за індукцією, можемо вважати, що кожний елемент множини 2^A належить до першого або другого класів.

Елементи першого класу утворюють ланцюг, який за побудовою максимальний.

2. Розглянемо будь-яку підмножину X індуктивної множини B. Ланцюг, що складається з одного елемента X, за доведеною частиною леми, міститься в максимальному ланцюзі C. Розглянемо множину M, що об'єднує всі елементи ланцюга C. Очевидно, M є максимальним елементом множини B. Справді, якби існував елемент $M_1 \in B$, $M_1 \supsetneq M$, то M_1 можна було б долучити до ланцюга C і одержати більший ланцюг.

Зауваження 1.6. *Можна показати, і це не дуже важко, що з леми Цорна виводиться аксіома вибору, тобто лема Цорна є еквівалентною до аксіоми вибору. Зацікавленому читачеві пропонуємо зазирнути до книг [15], [17].*

Бог створив цілі числа,
все решта — робота людини.

Л. Кронекер

Розділ 2

Натуральні числа, індукція та потужність

2.1. Натуральні числа

Так само як і у випадку теорії множин, строге введення натуральних чисел можна виконати лише аксіоматично.

2.1.1. Аксіоми Пеано

Означення 2.1. *Множина натуральних чисел — це множина* $\mathbb{N}$*, для якої існує відображення* $d : \mathbb{N} \to \mathbb{N}$*, яке має такі властивості (аксіоми Пеано):*

- P_1) *відображення* d *ін'єктивне;*
- P_2) *існує елемент* $0 \in \mathbb{N}$*, такий що* $d(a) \neq 0$ *для кожного елемента* $a \in \mathbb{N}$*;*
- P_3) *якщо* $S \subset \mathbb{N}$ *– підмножина множини* $\mathbb{N}$*, причому* $0 \in S$ *і для кожного* $a \in S$ *його образ* $d(a)$ *теж є елементом множини* S*, то* $S = \mathbb{N}$.

Аксіому P_3 називають аксіомою математичної індукції. Відображення d називають функцією наступності. Зручно писати a' замість $d(a)$. $0'$ далі позначатимемо 1.

З аксіом Пеано випливає, що $\mathbb{N} = \{0, 0' = 1, 0'', \ldots, 0^{''\ldots'}, \ldots\}$, де $0^{''\ldots'} = d(d(\ldots d(0)\ldots)$ (ліворуч n штрихів, праворуч n разів застосована функція d).

Звідси легко вивести таке: коли маємо дві множини $\mathbb{N}_1$ і $\mathbb{N}_2$, які задовольняють аксіоми Пеано, то існує бієктивне відображення з множини $\mathbb{N}_1$ на множину $\mathbb{N}_2$. Елементи довільної множини, яка задовольняє аксіоми Пеано, називатимемо натуральними числами.

2.1.2. Асоціативність додавання натуральних чисел

Додавання натуральних чисел визначають «за індукцією».

Означення 2.2. *Нехай* $a, b \in \mathbb{N}$. *Тоді i)* $a + 0 = a$, *ii)* $a + b' = (a + b)'$.

Твердження 2.1. *Додавання натуральних чисел асоціативне, тобто*

$$(a + b) + c = a + (b + c) \quad \forall a, b, c \in \mathbb{N}.$$

Доведення. Нехай $S = \{c \in \mathbb{N} \mid (a + b) + c = a + (b + c) \quad \forall a, b \in \mathbb{N}\}$. Тоді $0 \in S$, бо $(a + b) + 0 = a + (b + 0) = a + b$.

Доведемо таке: коли $c \in S$, то й $c' \in S$. Якщо $c \in S$, то $(a + b) + c = a+(b+c)$. Звідси одержуємо $((a+b)+c)' = (a+(b+c))'$, і, використовуючи умову ii) з означення додавання натуральних чисел, маємо

$$(a + b) + c' = ((a + b) + c)' = (a + (b + c))' = a + (b + c)' = a + (b + c').$$

Тому $(a+b)+c' = a+(b+c')$ і $c \in S$. За аксіомою P_3 множина S збігається з цілою множиною $\mathbb{N}$, і це завершує доведення.

2.1.3. Комутативність додавання натуральних чисел

Твердження 2.2. $a' = a + 1 = 1 + a \quad \forall a \in \mathbb{N}$.

Доведення. Нехай $S = \{a \in \mathbb{N} \mid a' = a + 1 = 1 + a\}$. Спочатку перевіримо, що $0 \in S$. Справді, $0' = (0+0)' = 0+0' = 0+1, 0' = 1 = 1+0$. Якщо $a \in S$, то $(a')' = (a + 1)' = ((a + 1) + 0)' = a' + 0' = a' + 1$, $(a')' = (1 + a)' = 1 + a'$. Отже, $a' \in S$, а тому, за аксіомою P_3 твердження доведене.

Твердження 2.3. *Множення натуральних чисел комутативне, тобто*

$$a + b = b + a \quad \forall a, b \in \mathbb{N}.$$

Доведення. Розіб'ємо доведення на дві частини. 1. Спочатку доведемо, що $a + 0 = 0 + a$ для кожного $a \in \mathbb{N}$. Для цього розглянемо множину $S = \{a \in \mathbb{N} \mid a + 0 = 0 + a\}$. З означення додавання випливає, що $0 \in S$. Досить перевірити, що й $a' \in S$. Справді, $0 + a' = (0 + a)' = (a + 0)' = a + 0' = a + 1 = a' = a' + 0$.

2. Нехай $T = \{b \in \mathbb{N} \mid a + b = b + a\}$. За першою частиною доведення $0 \in T$. Перевіримо, що з $b \in T$ випливає $b' \in T$. Маємо

$$a + b' = (a + b)' = (b + a)' = b + a' = b + (1 + a) = (b + 1) + a = b' + 1.$$

За аксіомою P_3 $T = \mathbb{N}$ і твердження доведене.

2.1.4. Множення натуральних чисел

Множення натуральних чисел, як і додавання, визначається «за індукцією».

Означення 2.3. *Якщо $a, b \in \mathbb{N}$, то за означенням*

i) $a0 = 0$;

ii) $ab' = ab + a$.

Твердження 2.4 пропонуємо довести самостійно.

Твердження 2.4. *1)* $a(b + c) = ab + ac \quad \forall a, b, c \in \mathbb{N}$;

2) $a(bc) = (ab)c \quad \forall a, b, c \in \mathbb{N}$;

3) $ab = ba \quad \forall a, b \in \mathbb{N}$.

2.1.5. Біном Ньютона

Як ілюстрацію методу математичної індукції наведемо доведення формули бінома Ньютона.

Для $k, n \in \mathbb{N}$ приймемо за означенням

$$n! = 1 \cdot 2 \cdot \cdots \cdot n, \quad C_n^k = \frac{n!}{k!(n-k)!}, \quad 0! = 1.$$

Твердження 2.5.

$$C_n^k + C_n^{k-1} = C_{n+1}^k.$$

Доведення.

$$\frac{n!}{k!(n-k)!} + \frac{n!}{(k-1)!(n-k+1)!} =$$

$$= \frac{n!}{(k-1)!(n-k)!}\Big(\frac{1}{k} + \frac{1}{n-k+1}\Big) = \frac{(n+1)!}{k!(n+1-k)!}.$$

Твердження 2.6. *Нехай a, b — елементи довільного комутативного кільця R (зокрема, дійсні числа), n — натуральне число. Тоді*

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

Доведення. Нехай S — множина тих натуральних чисел n, для яких формула правильна. Тоді $0 \in S$, бо $(a+b)^0 = 1 = C_0^0 a^0 b^0$. Припустимо, що $n \in S$ і доведемо, що й $n+1 \in S$. Маємо

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

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

$$C_{n+1}^0 + \sum_{k=0}^{n} C_{n+1}^k a^{n+1-k} b^k + C_{n+1}^{n+1} b^{n+1} = \sum_{k=0}^{n+1} C_{n+1}^k a^{n+1-k} b^k.$$

Твердження доведене.

2.1.6. Трикутник Паскаля

Використовуючи формулу $C_{n+1}^k = C_n^k + C_n^{k-1}$, бачимо, що коефіцієнти C_n^k у формулі бінома Ньютона є елементами n-го рядка трикутника, який називають трикутником Паскаля.

$$
\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 \\
1 & & 7 & & 21 & & 35 & & 25 & & 21 & & 7 & & 1 \\
\multicolumn{11}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots}
\end{array}
$$

2.1.7. Рекурентні послідовності та рекурентні означення

З означення множини натуральних чисел ми бачимо, що це послідовність, яка починається з нуля, кожний наступний її член дорівнює попередньому, збільшеному на одиницю. Інакше кажучи, ця послідовність цілком визначається умовами

$$a_0 = 0,\ a_n = a_{n-1} + 1 \ \text{ для } \ n > 1. \tag{2.1}$$

Загальні члени послідовностей $\{2^n, n \in \mathbb{N}\}$ та $\{n!, n \in \mathbb{N}\}$ теж обчислюють за правилом, схожим на (2.1). Для першої з цих послідовностей

$$a_0 = 1, \quad a_n = 2a_{n-1},$$

для другої

$$a_0 = 1, a_n = na_{n-1}.$$

У цих випадках маємо прості формули для обчислення a_n. Розглянемо ще один приклад.

$$a_0 = a_1 = 1,\ a_n = a_{n-1} + a_{n-2},\ \text{для } \ n \geq 2. \tag{2.2}$$

Послідовність (2.2) — це послідовність *чисел Фібоначчі*, про яку вже йшлося раніше.

Означення 2.4. *Кажуть, що послідовність* $\{a_0, a_1, \ldots, a_n, \ldots\}$ *елементів деякої множини означена рекурентно, якщо:*

(1) зафіксована деяка скінченна підмножина членів цієї послідовності (здебільшого, декілька перших членів);

(2) інші члени послідовності однозначно визначаються за певним правилом з попередніх членів.

Щойно наведене означення є дуже загальним і не дуже чітким. Розглянемо один частковий випадок цього означення.

Означення 2.5. *Нехай* A *– деяке кільце. Кажуть, що послідовність* $\{a_0, a_1, \ldots, a_n, \ldots\}$ *елементів кільця* A *є лінійною рекурентною послідовністю, якщо існує натуральне число* m *і елементи* $c_0, c_1, \ldots, c_{m-1} \in A$ *такі, що для* $n \geq m$

$$a_n = c_0 a_{n-m} + c_1 a_{n-m+1} + \cdots + c_{m-1} a_{n-1} = \sum_{i=0}^{m-1} c_i a_{n-m+i}. \qquad (2.3)$$

Твердження 2.7. *Якщо кільце* A *є полем, то множина всіх послідовностей, які задовольняють умови (2.3) (при фіксованих* c_i*) є* m*-вимірним лінійним простором. Якщо* $A = \mathbb{F}_q$ *– скінченне поле з* q *елементів, то кількість всіх послідовностей (2.3) дорівнює* q^m*.*

Доведення. Якщо не фіксувати перші m членів лінійної рекурентної послідовності і розглядати всі послідовності, члени яких задовольняють рівняння

$$X_n = \sum_{i=0}^{m-1} c_i X_{n-m+i}, \qquad (2.4)$$

то зрозуміло, що сума двох розв'язків рівняння (2.3) є знову розв'язком цього рівняння. Аналогічне твердження правильне і для добутку розв'язку на скаляр. Легко переконатися в тому, що множина розв'язків рівняння (2.3) є лінійним простором щодо цих операцій. Кожний розв'язок $\{a_0, a_1, \ldots, a_n, \ldots\}$ однозначно визначається вектором $(a_0, a_1, \ldots, a_{m-1}) \in A^m$. Звідси випливає, що вимірність простору розв'язків рівняння (2.3) дорівнює m. Для завершення доведення залишається зауважити, що існує q^m векторів $(a_0, a_1, \ldots, a_{m-1}) \in \mathbb{F}_q^m$.

Вправа. В умовах попереднього твердження довести таке: коли кільце $A = \mathbb{F}_q$ є полем з q елементів, то кожна лінійна рекурентна послідовність є періодичною.

Означення 2.6. *Поліном $f(X) = X^m - c_{m-1}X^{m-1} - \ldots - c_1X - c_0$ називається характеристичним поліномом лінійної рекурентної послідовності (2.3).*

Важливість поняття характеристичного полінома полягає в тому, що іноді корисно знати ті розв'язки рівняння (2.4), які мають вигляд

$$(1, \alpha, \alpha^2, \ldots, \alpha^n, \ldots).$$

Твердження 2.8. *Послідовність $(1, \alpha, \alpha^2, \ldots, \alpha^n, \ldots)$ тоді й лише тоді є розв'язком рівняння (2.4), коли α є коренем відповідного характеристичного полінома.*

Доведення. Достатньо підставити в (2.4) $X_n = \alpha^n$. Одержимо

$$\alpha^n = \sum_{i=0}^{m-1} c_i \alpha^{n-m+i}.$$

Розділивши обидві частини цієї рівності на α^{n-m}, маємо

$$\alpha^m = \sum_{i=0}^{m-1} c_i \alpha^i,$$

тобто α є коренем характеристичного полінома. Обернене твердження одержуємо домноженням останньої рівності на α^{n-m}.

Приклади

1. Для послідовності Фібоначчі характеристичний поліном має вигляд $X^2 - X - 1$. Його коренями є $\alpha_{1,2} = (1 \pm \sqrt{5})/2$. Вектори $(1, \alpha_1)$ і $(1, \alpha_2)$ лінійно незалежні, бо $\det \begin{pmatrix} 1 & \alpha_2 \\ 1 & \alpha_1 \end{pmatrix} = 1 \neq 0$. Тому

$$a_n = c\Big(\frac{1+\sqrt{5}}{2}\Big)^n + d\Big(\frac{1-\sqrt{5}}{2}\Big)^n,$$

де $c, d \in \mathbb{C}$ — загальний вигляд всіх послідовностей комплексних чисел, які задовольняють рівняння Фібоначчі $a_n = a_{n-1} + a_{n-2}$. Для послідовності Фібоначчі $(1, 1, 2, 3, 5, 8, \ldots)$ одержимо $c + d = 1, c\alpha_1 + d\alpha_2 = 1$. Розв'язавши цю систему рівнянь, бачимо, що $c = \alpha_1/\sqrt{5}, d = -\alpha_2/\sqrt{5}$. Отже, загальний член послідовності Фібоначчі має вигляд

$$a_n = \frac{1}{\sqrt{5}}\Big(\Big(\frac{1+\sqrt{5}}{2}\Big)^{n+1} - \Big(\frac{1-\sqrt{5}}{2}\Big)^{n+1}\Big).$$

2.2. Потужність множин

2.2.1. Рівнопотужні множини

Означення 2.7. *Дві множини A і B називають рівнопотужними, якщо існує бієктивне відображення $f : A \to B$.*

Якщо множини A і B рівнопотужні, то це позначають так: $|A| = |B|$.

Твердження 2.9. *1)* $|A| = |A|$; *2)* $|A| = |B| \Longrightarrow |B| = |A|$; *3)* $|A| = |B|$, $|B| = |C| \Longrightarrow |A| = |C|$.

Доведення. Перше твердження одержуємо з бієктивності одиничного відображення. Правильність другого твердження випливає з того факту, що для бієктивного відображення існує обернене відображення, яке теж є бієктивним. Нарешті, добуток двох бієктивних відображень є бієктивним, тобто маємо твердження 3.

Приклади

1. Нехай $A = [0, 1]$, $B = [a, b]$ два інтервали в $\mathbb{R}$. Розглянемо відображення $f : A \to B$, де $f(x) = a + (b - a)x$. Для цього відображення існує обернене $f^{-1}(x) = \frac{x-a}{b-a}$. Звідси випливає, що кожні два інтервали в $\mathbb{R}$ рівнопотужні.
2. Нехай $A = \big(-\pi/1, \pi/2\big), B = \mathbb{R}$. Відображення $f(x) = tg(x)$ є бієктивним відображенням цих множин. Тому вони рівнопотужні.
3. $A = \{0, 1, 4, 9, \dots\} = \{n^2 \mid n \in \mathbb{N}\}$. Відображення $f : \mathbb{N} \to A$, $f(n) = n^2$ бієктивне, тому $|A| = |\mathbb{N}|$.

2.2.2. Зліченні множини

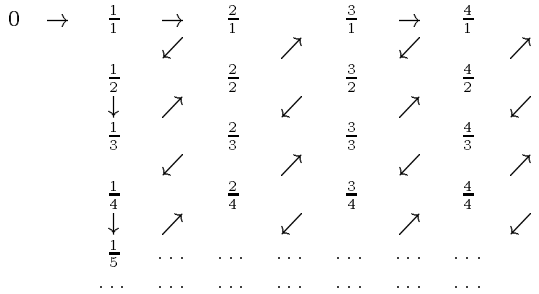
Означення 2.8. *Множину A називають зліченною, якщо $|A| = |\mathbb{N}|$.*

Якщо множина A зліченна, то її можна записати у вигляді послідовності $A = \{a_0, a_1, \dots, a_n, \dots\}$, де $a_i = f(i)$, $f : \mathbb{N} \to A$ – бієктивне відображення.

Приклади

1. Множина $\mathbb{Z}$ зліченна. Щоб довести це, розглянемо відображення $f : \mathbb{N} \to \mathbb{Z}$, де $f(0) = 0$ і $f(2k - 1) = k$, $f(2k) = -k$ для $k \in \mathbb{N} \setminus \{0\}$. Легко переконатися, що відображення f бієктивне: особливо добре видно це, якщо записати f у вигляді таблиці
$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \dots \\ 0 & 1 & -1 & 2 & -2 & 3 & -3 & 4 & \dots \end{pmatrix},$$
в якій елементи нижнього рядка є образами при відображенні f елементів верхнього рядка.

2. Множина $\mathbb{Q}_+ = \left\{\frac{a}{b} \mid a \in \mathbb{N},\ b \in \mathbb{N} \setminus \{0\}\right\}$ зліченна. Щоб переконатися в цьому, розглянемо відображення, яке будують так: множину $\mathbb{Q}_+$ перелічують за правилом, яке зображене нижче за допомогою стрілок



Так перебирають усі раціональні числа, пропускаючи на кожному кроці ті, які вже траплялися раніше. В результаті одержуємо, що множину $\mathbb{Q}_+$ можна записати у вигляді послідовності $\{a_0, a_1, \ldots, a_n, \ldots\}$.

3. Множина $\mathbb{Q}$ всіх раціональних чисел зліченна. Справді, ми щойно бачили, що існує бієктивне відображення $f : \mathbb{N} \to \mathbb{Q}_+ = \{a_o, a_1, \ldots, a_n, \ldots\}$. Відображення $g : \mathbb{Q}_+ \to \mathbb{Q}$, для якого $g(a_{2k}) = a_k$ і $g(a_{2k+1}) = -a_k$, бієктивне. Тому відображення $g \circ f : \mathbb{N} \to \mathbb{Q}$ бієктивне і $|\mathbb{N}| = |\mathbb{Q}|$.

2.2.3. Зліченні об'єднання зліченних множин

Твердження 2.10. *Об'єднання $A = \bigcup_{i\in I} A_i$ скінченної або зліченної родини скінченних або зліченних множин $\{A_i\}_{i\in I}$ є скінченною або зліченною множиною.*

Доведення. Нехай P_1 – множина всіх простих чисел. Відомо, що ця множина нескінченна, тому вона зліченна (бо кожна нескінченна підмножина зліченної множини є зліченною множиною). Для кожного $n \geq 1$ розглянемо підмножини $P_n = \{x^n \mid x \in P_1\}$. Множини P_n попарно не перетинаються і є зліченними.

Якщо множини A_n попарно не перетинаються (тобто $A_n \cap A_m = \emptyset$, $\forall n, m \in I,\ n \neq m$), то існує бієктивне відображення з множини A_n на підмножину множини P_n, тому існує бієктивне відображення з множини A на підмножину множини $\mathbb{N}$. Зрозуміло, що ця остання підмножина є скінченною або зліченною множиною.

Якщо серед A_n існують множини з непорожніми перетинами, то розглянемо множини $A'_1 = A_1, \quad A'_2 = A_2 \setminus A'_1, \ldots, \quad A'_n = A_n \setminus (A'_1 \cup \ldots \cup A'_{n-1}), \ldots$.

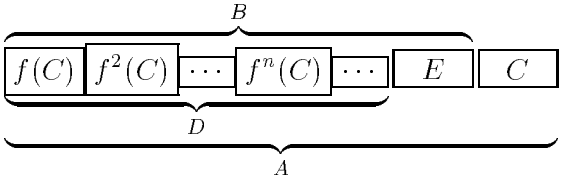
Множини A'_n – скінченні або зліченні, попарно не перетинаються і $A = \bigcup_{n\in I} A'_n$. Отже, і в цьому випадку множина A є скінченною або зліченною.

2.2.4. Теорема Кантора-Бернштейна

Теорема 2.1. *Нехай A B – множини для яких існують ін'єктивні відображення $f: A \to B$ і $g: B \to A$. Тоді $|A| = |B|$.*

Доведення. Припустимо, спочатку, що $B \subset A$. Розглянемо множину $C = A \setminus B$ і множини $f(C)$, $f(f(C)) = f^2(C), \ldots, f(f(\ldots(f(C)\ldots) = f^n(C), \ldots$; оскільки f ін'єктивне, то ці множини не перетинаються.

Нехай $D = \cup_{n=1}^{\infty} f^n(C)$ і $E = B \setminus D$. Тоді $A = C \cup E \cup D$ і множини C, E і D попарно не перетинаються. Зобразимо схематично всі введені нами множини.



Розглянемо відображення $h: A \to B$

$$h(a) = \begin{cases} f(a), & \text{якщо } a \in C \cup D, \\ a, & \text{якщо } a \in E. \end{cases}$$

Ін'єктивність відображення h випливає з ін'єктивності одиничного відображення та відображення f. З іншого боку, з означень множин C, D і E та відображення h випливає сюр'єктивність, отже, і бієктивність відображення h. Отож, теорема Бернштейна-Кантора доведена у випадку $B \subset A$.

Якщо B не є підмножиною множини A, то розглянемо множину $B' = g(B) \subset A$. Відображення $\epsilon = g \circ f: A \to B'$ є добутком ін'єктивних відображень, тому воно ін'єктивне, і за доведеним вище, $|A| = |B'|$. Але $|B'| = |B|$, тому й $|A| = |B|$. Теорема доведена.

2.2.5. Порівняння потужностей. Потужність $\mathfrak{c}$

Означення 2.9. *Нехай A і B — дві множини. Кажуть, що потужність множини A є меншою від потужності множини B, і записують*

це $|A| < |B|$, *якщо* $|A| \neq |B|$ *і існує підмножина* $B_1 \subset B$, *для якої* $|A| = |B_1|$. *Множину* A *називають незліченною, якщо* $|\mathbb{N}| < |A|$.

Теорема 2.2. *Множина всіх дійсних чисел з інтервалу* $[0,1]$ *незліченна.*

Довєдення. Якщо $\alpha \in [0,1]$, то α можна записати у вигляді нескінченного десяткового дробу $\alpha = 0, a_0 a_1 \dots a_k \dots$, $a_k \in \{0,1,\dots,9\}$. Домовимося не використовувати у записах дійсних чисел у вигляді нескінченного десяткового дробу нескінченної кількості дев'яток підряд. Цього можна домогтися, замінивши, коли потрібно, нескінченну кількість дев'яток нескінченною кількістю нулів (наприклад, $0,87546999.. = 0,87547000\dots$). Лише число $1 = 0,999\dots$ записуватимемо з використанням нескінченної кількості дев'яток.

Множина дійсних чисел з інтервалу $[0,1]$ рівнопотужна множині десяткових дробів $0, a_0 a_1 \dots a_k \dots$, які записані з врахуванням щойно зробленого застереження щодо дев'яток.

Тепер міркуємо від супротивного. Якщо $|\mathbb{N}| = |[0,1]|$, то всі без винятку числа з відрізка $[0,1]$ можна розмістити у вигляді послідовності $\alpha_o, \alpha_1, \dots, \alpha_n, \dots$, де

$$\begin{aligned}
\alpha_0 &= 0, a_{00} a_{01} a_{02} \dots a_{0n} \dots, \\
\alpha_1 &= 0, a_{10} a_{11} a_{12} \dots a_{1n} \dots, \\
\alpha_2 &= 0, a_{20} a_{21} a_{22} \dots a_{2n} \dots, \\
&\dots\dots\dots\dots\dots\dots\dots, \\
\alpha_n &= 0, a_{n0} a_{n1} a_{n2} \dots a_{nn} \dots, \\
&\dots\dots\dots\dots\dots\dots
\end{aligned}$$

Розглянемо число $\alpha = 0, a_0 a_1 a_2 \dots a_n \dots$, де $a_i \neq a_{ii}$. Це число не трапляється серед $\alpha_o, \alpha_1, \dots, \alpha_n, \dots$, бо n-ий десятковий знак числа α_n не дорівнює a_n. Одержана суперечність засвідчує, що $|[0,1]| \neq |\mathbb{N}|$.

Для завершення доведення теореми залишилося зазначити підмножину множини $[0,1]$, яка рівнопотужна множині $\mathbb{N}$. Такою підмножиною є, наприклад, множина $B = \{2^{-n} \mid n \in \mathbb{N}\} \subset [0,1]$. Отже, $|\mathbb{N}| < |[0,1]|$ і теорема доведена.

Твердження 2.11. $|[0,1]| = |\mathbb{R}|$.

Доведення. Ми вже знаємо, що $|(-\pi/2, \pi/2)| = |\mathbb{R}|$. З іншого боку, з теореми Бернштейна-Кантора легко випливає, що $|(-\pi/2, \pi/2)| = |[0,1]|$. Справді, множина $[0,1]$ є підмножиною множини $(-\pi/2, \pi/2)$ й існує ін'єктивне відображення $f : (-\pi/2, \pi/2) \to [0,1]$ (наприклад, $f(x) = x/\pi + 1/2$). Тому множини $[0,1]$ і $\mathbb{R}$ рівнопотужні.

Означення 2.10. *Множини, які рівнопотужні множині дійсних чисел* $\mathbb{R}$*, називають множинами потужності континуум. Потужність континууму позначають буквою* $\mathfrak{c}$.

Зауваження 2.1. *Г. Кантор у 1878 р. сформулював гіпотезу про те, що кожна нескінченна підмножина множини дійсних чисел рівнопотужна одній з множин* $\mathbb{R}$ *або* $\mathbb{N}$ *(її називають гіпотезою континууму). Цю гіпотезу протягом довгого часу не вдавалося ні довести, ні спростувати. Лише у 1963 р. американський математик П.Дж.Коен довів, що гіпотезу континууму (як і її заперечення) не можна вивести з аксіом теорії множин. Це надзвичайно цікава тема, якій ми, на жаль, не можемо приділити достатньої уваги.*

2.2.6. Існування як завгодно великих потужностей

Пригадаємо, що 2^A означає множину всіх підмножин множини A.

Теорема 2.3. $|A| < |2^A|$.

Доведення. Спочатку доведемо від супротивного, що $|A| \neq |2^A|$. Нехай $|A| = |2^A|$. Тоді існує бієктивне відображення $f : A \to 2^A$. Розглянемо множину $B = \{x \in A \mid x \notin f(x)\}$. Оскільки f – сюр'єктивне відображення, то існує $b \in A$, для якого $f(b) = B$. Якщо $b \in B$, то за означенням множини B маємо $b \notin f(b) = B$. Якщо ж $b \notin f(b) = B$, то знову за означенням множини B маємо $b \in f(b) = B$. В обох випадках приходимо до суперечності. Отже, $|A| \neq |2^A|$.

Тепер розглянемо підмножину $B = \{\{x\} \in 2^A \mid x \in A\}$. Відображення $g : A \to B$, $g(x) = \{x\}$ є бієктивним, тому $|A| < |2^A|$ і теорема доведена.

2.3. Елементи комбінаторики

2.3.1. Об'єднання скінченних множин

Для скінченної множини A позначимо через $|A|$ кількість елементів множини A. Зрозуміло, що для двох скінченних множин A і B правильні рівності $|A \times B| = |A| \cdot |B|$ і $|A \cup B| = |A| + |B|$, якщо $A \cap B = \emptyset$.

Для скінченних множин $A_1, \ldots, A_n$ маємо таку формулу.

Теорема 2.4.

$$|A_1 \cup \ldots \cup A_n| = \sum_{i=1}^{n} |A_i| - \sum_{i<j} |A_i \cap A_j| + \ldots + (-1)^{k-1} \sum_{i_1<\ldots<i_k} |A_{i_1} \cap \ldots \cap A_{i_k}| +$$

$$+ \ldots + (-1)^{n-1} |A_1 \cap \ldots \cap A_n|.$$

Доведення. Теорему доводимо методом математичної індукції. Нехай $n = 2$ і $|A_1| = p, |A_2| = q, |A_1 \cap A_2| = r$. Тоді $A_1 \cup A_2 = (A_1 \setminus (A_1 \cap A_2)) \cup (A_2 \setminus (A_1 \cap A_2)) \cup (A_1 \cap A_2)$, причому множини $A_1 \setminus (A_1 \cap A_2))$, $A_2 \setminus (A_1 \cap A_2)$ та $A_1 \cap A_2$ попарно не перетинаються. Тому $|A_1 \cup A_2| = p - r + q - r + r = p + q - r = |A_1| + |A_2| - |A_1 \cap A_2|$.

Припустимо, що $n \geq 2$ і теорема доведена для n множин. Доведемо, що тоді вона правильна і для $n + 1$ множини. Маємо

$$|A_1 \cup \ldots \cup A_n \cup A_{n+1}| = |(A_1 \cup \ldots \cup A_n) \cup A_{n+1}| =$$

$$= |A_1 \cup \ldots \cup A_n| + |A_{n+1}| - |(A_1 \cup \ldots \cup A_n) \cap A_{n+1}| =$$

$$= \sum_{i=1}^{n} |A_i| - \sum_{i<j} |A_i \cap A_j| + \ldots + (-1)^{k-1} \sum_{i_1<\ldots<i_k} |A_{i_1} \cap \ldots \cap A_{i_k}| + \ldots +$$

$$+ (-1)^{n-1} |A_1 \cap \ldots \cap A_n| - \sum_{i=1}^{n} |A_i \cap A_{n+1}| +$$

$$+ (-1)^{k-1} \sum_{i_1<\ldots<i_{k-1}} |A_{i_1} \cap \ldots \cap A_{i_{k-1}} \cap A_{n+1}| +$$

$$+ \ldots + (-1)^{n} |A_1 \cap \ldots \cap A_{n+1}| = \sum_{i=1}^{n+1} |A_i| - \sum_{1 \leq i < j \leq n+1} |A_i \cap A_j| + \ldots +$$

$$(-1)^{k-1} \sum_{1 \le i_1 < \ldots < i_k \le n+1} |A_{i_1} \cap \ldots \cap A_{i_k}| + \ldots + (-1)^n |A_1 \cap \ldots \cap A_{n+1}|.$$

Теорема доведена.

Приклад

У науково-дослідному інституті працюють 67 співробітників. З них французькою мовою володіють 20 співробітників, англійською – 47, німецькою – 35, англійською та французькою – 12, англійською та німецькою – 23, німецькою та французькою – 11, а трьома мовами володіють 5 співробітників. Скільки співробітників не володіє жодною з цих трьох мов?

Нехай x – кількість співробітників, які не знають жодної з трьох мов, A_1, A_2, A_3 – множини співробітників, які знають відповідно англійську, французьку або німецьку мову. Тоді, застосовуючи доведену формулу при $n = 3$, одержимо $67 - x = 47 + 35 + 20 - 23 - 12 - 11 + 5 = 61,\ x = 6$.

Зауваження 2.2. *Доведену формулу часто називають формулою включень і виключень.*

2.3.2. Перестановки

Означення 2.11. *Перестановкою скінченної множини A називають бієктивне відображення множини A в себе.*

Відображення f множини $A = \{a_1, \ldots, a_n\}$ в себе можна записати у вигляді таблиці з двох рядків

$$f = \begin{pmatrix} a_1 & a_2 & \ldots & a_n \\ a_{i_1} & a_{i_2} & \ldots & a_{i_n} \end{pmatrix},$$

де a_{i_k} — образ елемента a_k при відображенні f. Якщо f — перестановка, то $a_{i_1}, a_{i_2}, \ldots, a_{i_n}$ — всі різні елементи множини A, тобто нижній рядок нашої таблиці є «перестановкою» елементів множини A.

Нехай P_n — кількість всіх перестановок множини з n елементів.

Твердження 2.12. $P_n = n!$.

Доведення. Достатньо підрахувати кількість всіх можливих таблиць

$$\begin{pmatrix} a_1 & a_2 & \ldots & a_n \\ a_{i_1} & a_{i_2} & \ldots & a_{i_n} \end{pmatrix}.$$

Якщо ми записуємо таку таблицю, то для запису елемента a_{i_1} маємо n можливостей, для запису елемента a_{i_2} — $n-1$ можливість і т.д., для

запису елемента a_{i_n} існує лише одна можливість. Разом одержуємо $n \cdot (n-1) \cdot \dots \cdot 2 \cdot 1 = n!$ можливостей.

Приклад

Скількома способами семеро дітей можна розсадити на каруселі?

Стати в шеренгу можна $7! = 5040$ способами. Розсаджуючи дітей на каруселі, важливе лише іхнє взаємне розміщення. Тому перестановки, які переходять одна в іншу, при рухомій каруселі треба вважати однаковими. З кожної перестановки можна одержати ще 6 за допомогою обертання. Тому існує 5040:7=720 способів.

2.3.3. Розміщення

Означення 2.12. *Ін'єктивне відображення k-елементної множини A в n-елементну множину B називають розміщенням з n по k.*

Якщо $A = \{a_1, \dots, a_k\}, B = \{b_1, \dots, b_n\}$, то ін'єктивне відображення $f: A \to B$ можна задати за допомогою таблиці

$$f = \begin{pmatrix} a_1 & a_2 & \dots & a_k \\ b_{i_1} & b_{i_2} & \dots & b_{i_k} \end{pmatrix}, \quad \text{де } b_{i_s} = f(a_s). \tag{2.5}$$

Розміщення цілком визначається заданням нижнього рядка цієї таблиці при зафіксованому верхньому рядку.

Позначимо кількість всіх розміщень з n по k через A_n^k.

Твердження 2.13. $A_n^k = \frac{n!}{(n-k)!}$.

Доведення. У впорядкованій послідовності $b_{i_1}, b_{i_2}, \dots, b_{i_k}$ з k різних елементів множини B для елемента b_{i_1} маємо n можливостей, тоді для елемента b_{i_2} маємо $n-1$ можливість і т.д., для елемента b_{i_k} маємо $n-k+1$ можливість. Всього $n \cdot (n-1) \cdot \dots \cdot (n-k+1) = \frac{n!}{(n-k)!}$ можливостей.

Приклад

У чемпіонаті України з футболу беруть участь 16 команд. Перші три призові місця можуть бути розподілені $14 \cdot 15 \cdot 16 = 3360$ способами.

2.3.4. Сполуки

Означення 2.13. *Сполукою з n по k називають k-елементну підмножину n-елементної множини.*

Позначимо кількість всіх сполук з n по k через C_n^k.

Твердження 2.14.

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

Доведення. Розглянемо будь-яке розміщення з n по k, тобто ін'єктивне відображення $f : A_1 \to A$, де $|A_1| = k, |A| = n$. Образ $f(A_1)$ є сполукою. Два різних розміщення f і g визначають ту саму сполуку, якщо $f(A_1) = g(A_1)$. Відображення $\tilde{g} : A_1 \to f(A_1)$, для якого $\tilde{g}(x) = g(x)$, та добуток відображень $\tilde{g}^{-1} \circ f : A_1 \to A_1$ є бієктивними відображеннями. Отже, $\tilde{g}^{-1} \circ f$ — перестановка множини A_1. Звідси одержуємо, що кількість різних сполук з n по k дорівнює кількості розміщень з n до k, розділеній на кількість усіх перестановок з k елементів, тобто дорівнює $\frac{A_n^k}{P_k} = \frac{n!}{k!(n-k)!}$.

Приклад

Скількома способами можна поставити на шаховій дошці три тури? Очевидно, способів такої розстановки існує стільки, скільки є способів вибору трьох полів серед 64 полів шахової дошки, тобто $C_{64}^3 = \frac{64!}{3!61!} = \frac{62 \cdot 63 \cdot 64}{6} = 41664$ різних способів.

2.3.5. Розміщення з повтореннями

Нехай $B = \{b_1, \ldots, b_k\}$ і $A = \{a_1, \ldots, a_n\}$ — дві множини. Розглянемо відображення $f : A \to B$. Позначимо $n_i = |f^{-1}(b_i)|$ — кількість елементів множини A, які відображаються в елемент b_i. З означення відображення випливає, що прообрази $f^{-1}(b_i)$ різних елементів b_i попарно не перетинаються. Звідси випливає, що $n_1 + \cdots + n_k = n$. Зауважимо, що серед множин $f^{-1}(b_i)$, $1 \le i \le k$ можуть бути і порожні множини, тому деякі серед чисел $n_1, \ldots, n_k$ можуть дорівнювати нулю.

Означення 2.14. *Розміщенням з повтореннями назвемо відображення $f : A \to B$.*

Приклад

Нехай $A = \{1, 2, 3, 4\}, B = \{а, л\}$. Існує 16 різних відображень з $f : A \to B$. Виберемо серед них ті, для яких $|f^{-1}(а)| = |f^{-1}(л)| = 2$. Маємо 6 таких відображень:

$$f_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ а & а & л & л \end{pmatrix}, f_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ а & л & а & л \end{pmatrix}, f_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ а & л & л & а \end{pmatrix},$$

$$f_4 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ л & а & а & л \end{pmatrix}, f_5 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ л & а & л & а \end{pmatrix}, f_6 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ л & л & а & а \end{pmatrix}.$$

Позначимо через $C(n; n_1, \dots, n_k)$ кількість розміщень

$$f = \begin{pmatrix} 1 & 2 & \cdots & n \\ b_{i_1} & b_{i_2} & \cdots & b_{i_n} \end{pmatrix}$$

з повтореннями, для яких $|f^{-1}(b_i)| = n_i$.

Твердження 2.15. $C(n; n_1, \dots, n_k) = \frac{n!}{n_1! \cdot \dots \cdot n_k!}$.

Доведення. Число $C(n; n_1, \dots, n_k)$ дорівнює кількості способів вибору в множині $\{1, 2, \dots, n\}$ підмножин $f^{-1}(b_i)$, $1 \le i \le k$, з $|f^{-1}(b_i)| = n_i$. Множину $f^{-1}(b_1)$ можна вибрати $C_n^{n_1}$ способами, тоді множину $f^{-1}(b_2)$ можна вибрати $C_{n-n_1}^{n_2}$ способами і т.д. множину $f^{-1}(b_k)$ можна вибрати $C_{n-n_1-\dots-n_{k-1}}^{n_k}$ способами. Всього існує

$$C_n^{n_1} \cdot C_{n-n_1}^{n_2} \cdot \dots \cdot C_{n-n_1-\dots-n_{k-1}}^{n_k} =$$

$$\frac{n!}{n_1!(n-n_1)!} \cdot \frac{n!}{n_1!(n-n_1)!} \cdot \dots \cdot \frac{(n-n_1-\dots-n_{k-1})!}{n_k!} = \frac{n!}{n_1! \dots n_k!}.$$

способів вибору підмножин $f^{-1}(b_i)$.

Приклад

Знайдемо кількість різних перестановок букв у слові «алла». За доведеною формулою існує $C(4; 2, 2) = \frac{4!}{2!2!} = 6$ таких перестановок.

2.3.6. Формула для $(x_1 + \dots + x_k)^n$

Твердження 2.16. *Нехай $x_1, \dots, x_k$ — елементи довільного комутативного кільця і $n \in \mathbb{N}$. Тоді*

$$(x_1 + \dots + x_k)^n = \sum_{n_1 + \dots + n_k = n} \frac{n!}{n_1! \cdots n_k!} x_1^{n_1} \cdot \dots \cdot x^{n_k}.$$

Доведення.

$$(x_1 + \dots + x_k)^n = (x_1 + \dots + x_k) \cdot \dots \cdot (x_1 + \dots + x_k) = \sum_{1 \le i_1, \dots, i_n \le k} x_{i_1} \cdots x_{i_n}.$$

В останній сумі на доданки $x_{i_1} \cdots x_{i_n}$ можна дивитися як на розміщення з повтореннями елементів $x_1, \dots, x_k$. Після групування та зведення подібних членів одержимо

$$\sum_{1 \le i_1, \dots, i_n \le k} x_{i_1} \cdots x_{i_n} = \sum_{n_1 + \dots + n_k = n} C(n, n_1, \dots, n_k) x_1^{n_1} \cdot \dots \cdot x^{n_k} =$$

$$= \sum_{n_1+\cdots+n_k=n} \frac{n!}{n_1!\cdots n_k!} x_1^{n_1} \cdot \ldots \cdot x^{n_k},$$

що й потрібно було довести.

Наслідок 2.1 (біном Ньютона). *Нехай a, b — елементи довільного комутативного кільця і $n \in \mathbb{N}$. Тоді*

$$(a+b)^n = \sum_{k=0}^{n} \frac{n!}{k!(n-k)!} a^{n-k} b^k.$$

Доведення. Запишемо доведену в попередньому твердженні формулу для випадку $k = 2,\ x_1 = a,\ x_2 = b$. Одержимо

$$(a+b)^n = \sum_{n_1+n_2=n} \frac{n!}{n_1!n_2!} a^{n_1} b^{n_2}.$$

Якщо $n_2 = k$, то $n_1 = n - k$, тому

$$(a+b)^n = \sum_{k=0}^{n} \frac{n!}{k!(n-k)!} a^{n-k} b^k.$$

2.3.7. Сполуки з повтореннями

Нехай A — будь-яка множина (не обов'язково скінченна), нехай задане розбиття цієї множини на n підмножин, тобто $A = A_1 \cup \ldots \cup A_n$, де підмножини $A_i,\ 1 \le i \le n$ попарно не перетинаються. Наочним прикладом такої ситуації може бути ящик з кулями, кожна з яких пофарбована в один з n кольорів.

Означення 2.15. *k-елементною сполукою з повтореннями називають k-елементну підмножину множини A. Елемент сполуки, який міститься у підмножині $A_i,\ 1 \le i \le n$ називатимемо елементом i-го сорту.*

Кількість різних k-елементних сполук з повтореннями позначають через $\overline{C}_n^k$.

Твердження 2.17.

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

Доведення. Закодуємо кожну k-елементну сполуку послідовністю з нулів та одиниць. Нехай наша сполука містить s_1 елементів 1 сорту, s_2 елементів 2 сорту і т.д. s_n елементів n-го сорту. Поставимо у відповідність такій сполуці послідовність з нулів та одиниць: пишемо s_1 одиниць або 0, якщо $s_1 = 0$, потім пишемо s_2 одиниць або 0, якщо $s_2 = 0$, і т.д.

$$\underbrace{11\ldots1}_{s_1}0\underbrace{11\ldots1}_{s_2}0\ldots0\underbrace{11\ldots1}_{s_n}. \tag{2.6}$$

Ця послідовність складається з $k+n-1$ нулів та одиниць. Її можна ототожнити з підмножиною $A' = f^{-1}(1)$ $(k+n-1)$-елементної множини $\{1,2,\ldots,n+k-1\}$. Це робиться за допомогою відображення $f:\ \{1,2,\ldots,n+k-1\} \to \{0,1\}$, табличний запис якого має вигляд

$$f = \begin{pmatrix} 1 & 2 & \ldots & s_1 & s_1+1 & \ldots & n+k-1 \\ c_1 & c_2 & \ldots & c_{s_1} & \ldots & \ldots & c_{s_{n+k-1}} \end{pmatrix},$$

де рядок $c_1c_2\ldots c_{s_1}\ldots c_{s_{n+k-1}}$ — це рядок (2.6). Зрозуміло, що різні k-елементні сполуки з повтореннями відповідають різним послідовностям вигляду (2.6), тобто різним k-елементним підмножинам множини $\{1,2,..,n+k-1\}$. Тому за твердженням 2.14 одержуємо $\overline{C}_n^k = C_{n+k-1}^k$, що й потрібно було довести.

Приклад

У ящику лежать сині, жовті та білі кулі. Набір з п'яти куль можна вибрати $\overline{C}_3^5 = C_7^5 = \frac{7!}{2!\,5!} = \frac{6\cdot7}{2} = 21$ способом.

Математика — знаряддя особливо зручне для роботи з абстрактними поняттями будь-якого типу і немає меж її могутності в цій галузі.

П. Дірак

Розділ 3

Булькові алгебри, висловлення й автомати

3.1. Бульові алгебри

3.1.1. Означення та приклади бульових алгебр

Означення 3.1. *Бульовою алгеброю називається множина B з двома бінарними алгебричними операціями «$\cup$», «$\cdot$» (об'єднання і перетин) та однією унарною алгебричною операцією «$^-$» (доповнення), якщо ці операції мають такі властивості:*

1) $a \cup b = b \cup a$;
2) $a \cup (b \cup c) = (a \cup b) \cup c$;
3) $(a \cup b) \cdot b = b$;
4) $a \cup (b \cdot c) = (a \cdot b) \cup (a \cdot c)$;
5) $(a \cup \bar{a}) \cdot b = b$;

1′) $a \cdot b = b \cdot a$;
2′) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$;
3′) $(a \cdot b) \cup b = b$;
4′) $a \cdot (b \cup c) = (a \cup b) \cdot (a \cup c)$;
5′) $(a \cdot \bar{a}) \cup b = b$.

Приклади

1. Нехай $B = \{0, 1\}$ — множина з двох елементів з операціями

$$a \cup b = \max\{a, b\},\ a \cdot b = \min\{a, b\},\ \bar{0} = 1,\ \bar{1} = 0.$$

За допомогою безпосередньої перевірки легко переконатися, що множина $\{0, 1\}$ є бульовою алгеброю стосовно цих операцій.

2. Нехай $B = 2^M$ – множина всіх підмножин деякої множини M зі звичайними теоретико-множинними операціями об'єднання, перетину та доповнення

$$A \cup B,\ A \cdot B = A \cap B,\ \bar{A} = M \setminus A,$$

де $A, B \in 2^M$ — підмножини множини M. З властивостей операцій над множинами випливає, що це бульова алгебра.

3. Нехай A — непорожна множина, $M = 2^A$ — множина всіх підмножин множини A. Визначимо на 2^A операції: $X \oplus Y = (X \cup Y) \setminus (X \cap Y)$ і $X \odot Y = X \cap Y$ — перетин X і Y. M є кільцем стосовно цих операцій (у першому розділі показано, що всі аксіоми з означення кільця виконуються у цьому випадку).

4. Нехай B — бульова алгебра , M — непорожна множина, а B^M — множина всіх відображень з M в B. Визначимо об'єднання, перетин та доповнення функцій з B^M так:
   $f \cup g$ — функція, для якої $(f \cup g)(x) = f(x) \cup g(x)$;
   $f \cdot g$ — функція, для якої $(f \cdot g)(x) = f(x) \cdot g(x)$;
   $\bar{f}$ — функція, для якої $\bar{f}(x) = \overline{f(x)}$.
   Тепер легко перевірити, що ці операції задовольняють всі аксіоми з означення бульової алгебри. Виконаємо цю перевірку для аксіоми 4.
   Перевірки всіх інших аксіом виконують аналогічно.
   Нехай x довільний елемент множини M, $f, g, h \in B^M$. Тоді

$$(f \cup (g \cdot h))(x) = f(x) \cup (g \cdot h)(x) = (f(x) \cup (g(x)) \cdot (f(x) \cup h(x)) =$$

$$= (f \cdot g)(x) \cup (f \cdot h)(x) = ((f \cup g) \cdot (f \cup h)(x).$$

   Отже, значення функцій $f \cup (g \cdot h)$ та $(f \cup g) \cdot (f \cup h)$ є однаковими для кожного елемента $x \in M$, тому $f \cup (g \cdot h) = (f \cup g) \cdot (f \cup h)$.

5. Розглянемо наступний важливий частковий клас бульових алгебр з попереднього прикладу. Нехай $B = \{0, 1\}$ – бульова алгебра з двох елементів (як у прикладі 1), $M = \underbrace{\{0,1\} \otimes \ldots \otimes \{0,1\}}_{n}$. У цьому випадку бульову алгебру B^M позначатимемо F_n.

   Бульова алгебра F_n складається з усіх функцій $f(x_1, \ldots x_n)$ від n змінних $x_1, \ldots x_n$, кожна з яких набуває значення в множині $\{0, 1\}$, причому значення всіх функцій теж належать до множини $\{0, 1\}$. Можна також розглянути об'єднання $F = \cup F_n$. Множина F теж є бульовою алгеброю.

6. Нехай $B_1, \ldots, B_n$ – бульові алгебри. Розглянемо декартовий добуток $B = B_1 \times \ldots \times B_n$ з операціями об'єднання, перетину та доповнення, визначеними покомпонентно (декартовий добуток бульових алгебр)

$$(x_1, \ldots, x_n) \cup (y_1, \ldots, y_n) = (x_1 \cup y_1, \ldots, x_n \cup y_n);$$

$$(x_1, \ldots, x_n) \cdot (y_1, \ldots, y_n) = (x_1 \cdot y_1, \ldots, x_n \cdot y_n);$$

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

   Безпосередньо перевіряємо, що декартовий добуток бульових алгебр є бульовою алгеброю.

Бульові алгебри тісно пов'язані з елементарною логікою висловлень.

3.1.2. Висловлення

Під висловленням розуміють речення, записане з дотриманням граматики однієї з природних мов (українська, англійська тощо) або штучних (наприклад, одна з мов програмування), про яке можна без двозначності сказати — істинне це речення чи хибне.

Приклади

1. Речення «7 — просте число», «$2 \times 2 = 5$», «після зими починається весна», «існує натуральне число n, що є коренем рівняння $X^2 - 3X + 2 = 0$» є висловленнями.
2. Такі речення як «$X^2 - 3X + 2 = 0$», «змінимо життя на краще», «навіщо Єва зірвала яблуко?», «to be or not to be» не є висловленнями.

Позначимо множину всіх висловлень через S. Поставимо у відповідність кожному істинному висловленню A з множини S число $t(A) = 1$, а кожному хибному висловленню A з множини S число $t(A) = 0$. Одержимо відображення $t : S \to \{0, 1\}$, яке назвемо *функцією істини*.

Висловлення можна розділити на прості та складні. Висловлення називають простим, якщо жодна його частина не є висловленням. Наприклад, «7 — просте число» — просте висловлення, а речення «якщо ціле число a є дільником числа 7, то $a = 1$ або $a = -1$ або $a = 7$ або $a = -7$» — складне висловлення. З'ясуємо, як можна будувати складні висловлення з даних висловлень (зокрема з простих). Позначатимемо висловлення великими буквами латинського алфавіту $P_1, P_2, \ldots, P_n, \ldots, P, Q, R, \ldots A, B, C$. З даних висловлень можна будувати інші (складні) висловлення за допомогою логічних операцій: $\neg$ — заперечення, $\vee$ — диз'юнкція, $\wedge$ — кон'юнкція, $\to$ — імплікація.

Приступаючи до означення логічних операцій над висловленнями, введемо спочатку функцію істини t, визначену на множині всіх висловлень так:

$$t(A) = \begin{cases} 1, & \text{якщо } A \text{ істинне,} \\ 0, & \text{якщо } A \text{ хибне.} \end{cases}$$

Означення 3.2. *1.* **Заперечення***: якщо A – висловлення, то $\neg A$ (читається «не A») — висловлення, для якого $t(\neg A) = 1 - t(A)$.*

Інакше кажучи, висловлення $\neg A$ істинне тоді й лише тоді, коли A хибне.

2. **Диз'юнкція***: якщо A і B — висловлення, то $A \vee B$ (читається «A або B») — висловлення, для якого $t(A \vee B) = \max\{t(A), t(B)\}$.*

Інакше кажучи, висловлення $A \vee B$ істинне тоді й лише тоді, коли хоч одне з висловлень A або B істинне.

3. **Кон'юнкція**: *якщо A і B — висловлення, то $A \wedge B$ (читається «A і B») — висловлення, для якого $t(A \wedge B) = \min\{t(A), t(B)\}$.*
   Інакше кажучи, висловлення $A \wedge B$ істинне тоді й лише тоді, коли висловлення A і B істинні одночасно.
4. **Імплікація**: *якщо A і B — висловлення, то $A \longrightarrow B$ (читається «якщо A то B») — висловлення, для якого $t(A \longrightarrow B) = \max\{1 - t(A),\ t(B)\}$.*
   Інакше кажучи, висловлення $A \to B$ хибне тоді й лише тоді, коли A істинне і B хибне.

Об'єднаємо щойно наведені означення логічних операцій у підсумкову таблицю.

$t(A)$	$t(B)$	$t(\neg A)$	$t(A \vee B)$	$t(A \wedge B)$	$t(A \to B)$
0	0	1	0	0	1
0	1	1	1	0	1
1	0	0	1	0	0
1	1	0	1	1	1

Формули числення висловлень будуються з пропозиційних змінних $P_1, \ldots, P_n, \ldots, P, Q, R, \ldots$ з використанням логічних зв'язок та дужок за допомогою таких правил.

Означення 3.3. *1. Кожна пропозиційна змінна є формулою.*
2. Якщо A і B — формули, то $\neg A$, $(A \vee B)$, $(A \wedge B)$, $(A \to B)$ — формули.

Ось деякі приклади формул:
1) $(((P \to Q) \to R) \to ((P \to Q) \to (P \to R)))$;
2) $((\neg A \vee B) \to (A \to B))$;
3) $((P \vee Q) \to (P \wedge (Q \to R)))$.

Зауважимо, що часто домовляються опускати пару зовнішніх дужок у записі формули. Наприклад, формулу $((A \wedge B) \to A)$ можна записати $(A \wedge B) \to A$.

Істинність або хибність формули A цілком визначається істинністю або хибністю формул, з яких A побудована. Значення функції істини для складних формул зручно обчислювати за допомогою таблиць істинності. Для прикладу побудуємо таку таблицю для формули $A = (P \vee Q) \to (P \wedge (Q \to R))$.

$t(P)$	$t(Q)$	$t(R)$	$t(P \vee Q)$	$t(Q \to R)$	$t(P \wedge (Q \to R)$	t(A)
0	0	0	0	1	0	1
0	0	1	0	1	0	1
0	1	0	1	0	0	0
0	1	1	1	1	0	0
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	0	0	0
1	1	1	1	1	1	1

Означення 3.4. *Формула A називається тавтологією якщо її функція істини $t(A)$ приймає лише значення 1.*

Якщо функція істини $t(A)$ формули A приймає лише значення 0, то формулу A називають суперечністю.

Означення 3.5. *Дві формули A і B називаються еквівалентними, якщо $t(A) = t(B)$.*

Введемо позначення $A \equiv B$ для еквівалентних формул A і B. Легко переконатися в тому, що відношення $A \equiv B$ є відношенням еквівалентності на множині всіх формул числення висловлювань.

Приклад

Перевіримо еквівалентність $(P \to Q) \equiv (\neg P \vee Q)$. Це можна зробити, побудувавши таблиці істини для формул $P \to Q$ і $\neg P \vee Q$

$t(P)$	$t(Q)$	$t(\neg P)$	$t(P \to Q)$	$t(\neg P \vee Q)$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	0
1	1	0	1	1

Порівнюючи два останніх стовпчики цієї таблиці, бачимо, що $t(P \to Q) = (\neg P \vee Q)$, отже, $(P \to Q) \equiv (\neg P \vee Q)$.

Твердження 3.1. *Правильні такі еквівалентності формул числення висловлювань, де A, B і C — довільні формули.*

1) $A \vee B \equiv B \vee A$;

1′) $A \wedge B \equiv B \wedge A$;

2) $A \vee (B \vee C) \equiv (A \vee B) \vee C$;

2′) $A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$;

3) $(A \vee B) \wedge B \equiv B$;
3′) $(A \wedge B) \vee B \equiv B$;
4) $A \vee (B \wedge C) \equiv (A \wedge B) \vee (A \wedge C)$;
4′) $A \wedge (B \vee C) \equiv (A \vee B) \wedge (A \vee C)$;
5) $(A \vee \neg A) \wedge B \equiv B$;
5′) $(A \wedge \neg A) \vee B \equiv B$.

Доведення твердження зводиться до простої перевірки виписаних еквівалентностей за допомогою, наприклад, таблиць істинності.

3.1.3. Алгебра Лінденбаума-Тарського

Легко перевірити, що еквівалентність формул числення висловлювань узгоджена з логічними операціями. Правильне таке твердження.

Твердження 3.2. *Якщо* A_1, A_2, B_1, B_2 *— формули числення висловлювань і* $A_1 \equiv A_2$, $B_1 \equiv B_2$, *то* $\neg A_1 \equiv \neg A_2$, $A_1 \vee B_1 \equiv A_2 \vee B_2$, $A_1 \wedge B_1 \equiv A_2 \wedge B_2$ *і* $A_1 \to B_1 \equiv A_2 \to B_2$.

Доведення. Перевіримо, наприклад, що $A_1 \to B_1 \equiv A_2 \to B_2$. Маємо $t(A_1) = t(A_2)$, $t(B_1) = t(B_2)$, $t(A_1 \to B_1) = \max\{1 - t(A_1), t(B_1)\} = \max\{1 - t(A_2), t(B_2)\} = t(A_2 \to B_2)$. Аналогічно перевіряють інші еквівалентності.

Нехай F — множина всіх формул числення висловлювань, наділена логічними операціями над висловленнями. Позначимо через L фактор-множину $F/_{\equiv}$ множини F за відношенням еквівалентності $\equiv$. Клас всіх формул, які еквівалентні формулі A, позначимо через $[A]$. Множина $[A]$ є елементом множини L. Якщо $A_1 \equiv A_2$, то $[A_1] = [A_2]$. Означимо операції над елементами множини L так:

$$\neg[A] = [\neg A];\ [A] \vee [B] = [A \vee B];\ [A] \wedge [B] = [A \wedge B];\ [A] \to [B] = [A \to B].$$

Попереднє твердження засвідчує таке: коли $[A_1] = [A_2]$ і $[B_1] = [B_2]$, то $\neg[A_1] = \neg[A_2]$, $[A_1] \vee [B_1] = [A_2] \vee [B_2]$, $[A_1] \wedge [B_1] = [A_2] \wedge [B_2]$ і $[A_1] \to [B_1] = [A_2] \to [B_2]$. Це означає, що на множині L коректно визначені операції $\vee$, $\wedge$ та $\neg$. З твердження 3.1 випливає, що ці операції задовольняють такі властивості $1 - 5'$.

Твердження 3.3. *Операції* $\vee$, $\wedge$ *та* $\neg$ *мають такі властивості:*
1) $[A] \vee [B] = [B] \vee [A]$;

1′) $[A] \wedge [B] = [B] \wedge [A]$;
2) $[A] \vee ([B] \vee [C]) = ([A] \vee [B]) \vee [C]$;
2′) $[A] \wedge ([B] \wedge [C]) = ([A] \wedge [B]) \wedge [C]$;
3) $([A] \vee [B]) \wedge [B] = [B]$;
3′) $([A] \wedge [B]) \vee [B] = [B]$;
4) $[A] \vee ([B] \wedge [C]) = ([A] \wedge [B]) \vee ([A] \wedge [C])$;
4′) $[A] \wedge ([B] \vee [C]) = ([A] \vee [B]) \wedge ([A] \vee [C])$;
5) $([A] \vee \neg[A]) \wedge [B] = [B]$;
5′) $([A] \wedge \neg[A]) \vee [B] = [B]$.

Означення 3.6. *Множина A разом з визначеними на ній операціями $\vee$, $\wedge$ та $\neg$ називається алгеброю Лінденбаума-Тарського.*

Попереднє твердження означає, що алгебра Лінденбаума-Тарського є бульовою алгеброю.

3.1.4. Основні властивості бульових алгебр

Доведемо деякі найпростіші наслідки з аксіом бульової алгебри. Зауважимо, що аксіоми $1 - 5'$ симетричні щодо операцій $a \cup b$ і $a \cdot b$; разом з кожною аксіомою ця система аксіом містить ще одну, яку одержують з заданої заміною операцій «$\cup$» на «$\cdot$» і «$\cdot$» на «$\cup$». Тому разом з кожною властивістю операцій бульової алгебри вона має двоїсту властивість, яку одержують з заданої заміною «$\cup$» на «$\cdot$» і «$\cdot$» на «$\cup$». Для спрощення викладок домовимося використовувати знак рівності з індексом, який зазначає властивість, з якої отримуємо цю рівність.

Твердження 3.4. *Операції «$\cup$» і «$\cdot$» ідемпотентні, тобто для довільних елементів $a, b \in B$ виконуються властивості*

6) $a \cup a = a$, 6′) $a \cdot a = a$, 7) $a \cdot b = a \iff a \cup b = b$.

Доведення. Доведемо властивість 6. Маємо

$$a =_{3)} a \cup (a \cdot b) =_{4)} (a \cup a) \cdot (a \cup b) =_{4)} (a \cdot (a \cup b)) \cup (a \cdot (a \cup b)) =_{3)} a \cup a.$$

Аналогічно доводиться рівність 6′. Переходимо до доведення властивості 7. Нехай $a \cdot b = a$. Підставимо це в 3′, одержимо $a \cup b = b$. Навпаки, якщо $a \cup b = a$, то $a \cdot b = a \cdot (a \cup b) =_{1'} (a \cup b) \cdot a =_3 a$.

Означення 3.7. *За означенням для двох елементів a і b бульової алгебри B $a \leq b$, якщо виконується одна з двох рівносильних (за властивістю 7) умов $a \cdot b = a$ або (і) $a \cup b = b$.*

Твердження 3.5 свідчить про те, що відношення $\leq$ є відношенням порядку на бульовій алгебрі B.

Твердження 3.5. *Нехай a, b, c — довільні елементи бульової алгебри B. Тоді*

8) $a \leq a$;

9) $a \leq b \wedge b \leq a \Longrightarrow a = b$;

10) $a \leq b \wedge b \leq c \Longrightarrow a \leq c$.

Доведення. Властивість 8 випливає з 6.

Оскільки $a \leq b$ і $b \leq a$, то $a = a \cdot b$ і $a = a \cup b$. Тому, підставляючи $a = a \cdot b$ в $a = a \cup b$, одержимо властивість 9: $a = a \cup b = (a \cdot b) \cup b =_{3'} b$. Доведемо 10: $a = a \cdot b = a \cdot (b \cdot c) = a \cdot c$, тому $a \leq c$.

Елементи $a \cup b$ та $a \cdot b$ іноді називають відповідно точною верхньою та точною нижньою гранями елементів a, b. Твердження 3.6 обгрунтовує цю термінологію.

Твердження 3.6. *Нехай a і b — довільні елементи бульової алгебри B. Тоді*

11) $a \leq a \cup b,\ b \leq a \cup b$;

12) $a \leq c \wedge b \leq c \Longrightarrow a \cup b \leq c$;

13) $a \cdot b \leq a,\ a \cdot b \leq b$;

14) $c \leq a \wedge c \leq b \Longrightarrow c \leq a \cdot b$.

Доведення. Оскільки згідно з аксіомою 3 $(a \cup b) \cdot b = b$, то $b \leq a \cup b$ і аналогічно $a \leq a \cup b$. Це доводить 11.

За умовою $a \cup c = c$ і $b \cup c = c$. Тому

$$c =_6 c \cup c = (a \cup c) \cup (b \cup c) =_{1,2} (a \cup b) \cup (c \cup c) = (a \cup b) \cup c.$$

За означенням це означає, що $a \cup b \leq c$. Відповідні двоїсті твердження 13, 14 пропонуємо довести самостійно.

Твердження 3.7. *Нехай a і b — довільні елементи бульової алгебри B. Тоді*

15) $a \cup \bar{a} = b \cup \bar{b}$, $\quad 15')\ a \cdot \bar{a} = b \cdot \bar{b}$.

Доведення. З аксіоми 5 $(a \cup \bar{a}) \cdot b = b$ за означенням $\leq$ одержуємо $b \leq (a \cup \bar{a})$. Підставивши сюди $b \cup \bar{b}$ замість b, маємо $(b \cup \bar{b}) \leq (a \cup \bar{a})$.

Оскільки a і b — довільні елементи, то й $(a \cup \bar{a}) \leq (b \cup \bar{b})$. Отже, властивість 15 доведена, двоїсте твердження 15' залишається для самостійного доведення.

Щойно доведене твердження означає, що елементи $a \cup \bar{a}$ і $a \cdot \bar{a}$ не залежать від a. Цей факт дає змогу ввести такі позначення:

$$16)\ \ 1 = a \cup \bar{a}, \quad 16')\ 0 = a \cdot \bar{a},$$

де a — довільний елемент бульової алгебри B. В наступному твердженні підсумуємо основні властивості цих елементів.

Твердження 3.8. *Нехай a — довільний елемент бульової алгебри B. Тоді*

$$17)\ \ 0 \leq a, \ \ a \leq 1, \ \ a \cup 0 = a, \ \ a \cdot 0 = 0, \ \ a \cup 1 = 1, \ \ a \cdot 1 = a.$$

Доведення. Всі ці властивості є безпосередніми наслідками означень елементів 0 та 1 та означення відношення $\leq$.

Зауважимо, що властивість 17 означає, зокрема, що 1 та 0 є відповідно найбільший та найменший елемент бульової алгебри B.

Наступна властивість характеризує операцію доповнення в термінах об'єднання та перетину.

Твердження 3.9. *Нехай a — довільний елемент бульової алгебри B. Тоді*

$$18)\ \ a \cup c = 1 \wedge a \cdot c = 0 \Longleftrightarrow c = \bar{a}.$$

Доведення. Імплікація $\Longleftarrow$ очевидна. Для доведення оберненої імплікації $\Longrightarrow$ треба зробити деякі обчислення.

$$c =_{17} 0 \cup c =_{16} (a \cdot \bar{a}) \cup c =_{1,4'} (a \cup c) \cdot (\bar{a} \cup c) = 1 \cdot (\bar{a} \cup c) =_{17} \bar{a} \cup c.$$

Це означає, що $\bar{a} \leq c$.

$$c =_{17} 1 \cdot c =_{16} (a \cup \bar{a}) \cdot c =_{4} (a \cdot c) \cup (\bar{a} \cdot c) = 0 \cup (\bar{a} \cdot c) =_{17} \bar{a} \cdot c.$$

Це означає, що й $c \leq \bar{a}$, отже, $c = \bar{a}$.

Наслідок 3.1. *Нехай a — довільний елемент бульової алгебри B. Тоді*

$$19)\ \ a = \bar{\bar{a}}.$$

Доведення. Підставимо у 18 $\bar{a}$ замість a і a замість c і використаємо 16, 16'.

Твердження 3.10 (закони де Моргана). *Нехай a і b – довільні елементи бульової алгебри B. Тоді*

20) $\overline{a \cup b} = \bar{a} \cdot \bar{b}$, 20') $\overline{a \cdot b} = \bar{a} \cup \bar{b}$.

Доведення. Нехай $c = \bar{a} \cdot \bar{b}$. Тоді

$$(a \cup b) \cdot c = (a \cup b) \cdot (\bar{a} \cdot \bar{b}) = (a \cdot \bar{a} \cdot \bar{b}) \cup (b \cdot \bar{a} \cdot \bar{b}) = 0 \cup 0 = 0.$$

$$(a \cup b) \cup c = (a \cup b) \cup (\bar{a} \cdot \bar{b}) = (a \cup b \cup \bar{a}) \cdot (a \cup b \cup \bar{b}) = 1 \cdot 1 = 1.$$

Звідси та з 18 одержуємо $\overline{a \cup b} = \bar{a} \cdot \bar{b}$. Інший закон де Моргана $\overline{a \cdot b} = \bar{a} \cup \bar{b}$ пропонуємо довести самостійно.

Вправи. Довести властивості

21) $a \leq b \Longleftrightarrow \bar{b} \leq a$;

22) $\bar{0} = 1$, $\bar{1} = 0$;

23) $a \leq b \Longleftrightarrow a \setminus b = 0$, де $a \setminus b = a \cdot \bar{b}$.

3.1.5. Диз'юнктивна та кон'юнктивна нормальні форми

Повернемося до розгляду скінченних бульових алгебр F_n. Нагадаємо, що елементами алгебри F_n є відображення $f : \{0,1\}^n \to \{0,1\}$, які записуємо у вигляді $f(x_1, \ldots, x_n)$, де $f(x_1, \ldots, x_n)$ можна трактувати як всюди визначену функцію зі значеннями в множині $\{0,1\}$ від n змінних $x_1, \ldots, x_n$, кожна з яких набуває значення з множини $\{0,1\}$.

Означення 3.8. *Нехай $a, b \in \{0,1\}$. Позначимо*

$$b^a = \begin{cases} b, & \text{якщо } a = 1, \\ 1 - b, & \text{якщо } a = 0, \end{cases}$$

тобто $1^1 = 1$, $0^1 = 0$, $1^0 = 0$, $0^0 = 1$. *Інакше кажучи,*

$$b^a = \begin{cases} 1, & \text{якщо } a = b, \\ 0, & \text{якщо } a \neq b. \end{cases}$$

Нагадаємо, що для $x, y \in \{0,1\}$ $x \cap y = \min\{x, y\} = xy$, $x \cup y = \max\{x, y\}$.

Теорема 3.1. *У вищенаведених позначеннях довільну функцію $f \in F_n$ можна записати у диз'юнктивній нормальній формі*

$$f(x_1,\dots,x_n) = \bigcup_{a_1,\dots,a_n\in\{0,1\}^n} f(a_1,\dots,a_n)x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}. \qquad (*)$$

Доведення. Ми перевіряємо, що в результаті підстановки $x_1 = b_1, \dots, x_n = b_n$ у праву частину (*) одержимо $f(b_1,\dots,b_n)$. Справді,

$$\bigcup_{a_1,\dots,a_n\in\{0,1\}^n} f(a_1,\dots,a_n)b_1^{a_1}b_2^{a_2}\cdots b_n^{a_n} =$$

$$= \bigcup_{a_1,\dots,a_n\in\{0,1\}^n} f(a_1,\dots,a_n)\left\{\begin{array}{ll} 1, & \text{якщо } a_1 = b_1 \dots a_n = b_n, \\ 0, & \text{якщо } (a_1,\dots,a_n) \neq (b_1,\dots,b_n) \end{array}\right\}\cdot$$

$$\cdot x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n} = f(b_1,\dots,b_n).$$

Якщо f функція, яка не дорівнює тотожно нулю, то рівність (*) можна записати у вигляді

$$f(x_1,\dots,x_n) = \bigcup_{f(a_1,\dots,a_n)=1} x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}. \qquad (ДНФ)$$

Вправа. Довести, якщо функція $f \in F_n$ не дорівнює тотожно 1, то її можна записати у кон'юнктивній нормальній формі

$$f(x_1,\dots,x_n) = \bigcap_{f(a_1,\dots,a_n)=0} (x_1^{\bar{a}_1} \cup x_2^{\bar{a}_2} \cup \dots \cup x_n^{\bar{a}_n}). \qquad (КНФ)$$

Одержані результати можна застосувати до формул числення висловлень. Нехай $A(P_1,\dots,P_n)$ – формула числення висловлень, побудована з пропозиційних змінних $P_1,\dots,P_n$. Розглянемо відповідну до формули A функцію істинності f_A. Функція f_A є однією з функцій з бульової алгебри F_n, тому її можна записати в диз'юнктивній нормальній формі

$$f_A(x_1,\dots,x_n) = \bigcup_{f_A(a_1,\dots,a_n)=1} x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}.$$

Розглянемо формулу числення висловлень

$$A_1 = \bigvee_{f_A(a_1,\dots,a_n)=1} (P_1^{a_1} \wedge P_2^{a_2} \wedge \dots \wedge P_n^{a_n}),$$

де

$$P^a = \begin{cases} P, & \text{якщо } a = 1, \\ \neg P, & \text{якщо } a = 1. \end{cases}$$

Для функції істинності f_{A_1} формули A_1 маємо при $(b_1, \ldots, b_n) \in \{0,1\}^n$

$$f_{A_1} = \max_{f_A(a_1,\ldots,a_n)=1} \{b_1^{a_1} \cdot \ldots \cdot b_n^{a_n}\} =$$

$$= \bigcup_{f_A(a_1,\ldots,a_n)=1} b_1^{a_1} \cdot \ldots \cdot b_n^{a_n} = f_A(b_1, \ldots, b_n).$$

Це й означає, що $A \equiv A_1$.

Означення 3.9. *Кажуть, що формула $A(P_1, \ldots P_n)$ має досконалу диз'юнктивну нормальну форму, якщо $A = \vee_i X_i$, де $X_i = P_1^{a_{1i}} \wedge P_2^{a_{2i}} \wedge \ldots \wedge P_n^{a_{ni}}$, $a_{ji} \in \{0,1\}$, $P_i^1 = P_i$, $P_i^0 = \neg P_i$.*

Оскільки за попередньою теоремою кожна (не тотожно нульова) бульова функція, зокрема функції істинності формул числення висловлень, записується в диз'юнктивній нормальній формі, то ми маємо таку теорему.

Теорема 3.2. *Кожна формула числення висловлень, яка не є суперечністю, еквівалентна до формули, що має досконалу диз'юнктивну нормальну форму.*

Вправа. Довести двоїсте твердження: кожна формула числення висловлень, яка не є тавтологією, еквівалентна формулі, що має *досконалу кон'юнктивну нормальну форму*

$$A_1 = \bigwedge_{f_A(a_1,\ldots,a_n)=0} (P_1^{\bar{a}_1} \vee P_2^{\bar{a}_2} \vee \cdots \vee P_n^{\bar{a}_n}).$$

Зауваження 3.1. *Якщо формула $A(P_1, \ldots, P_n)$ є тавтологією, то вона еквівалентна формулі $P_1 \vee \neg P_1$, якщо $A(P_1, \ldots, P_n)$ є суперечністю, то вона еквівалентна формулі $P_1 \wedge \neg P_1$,*

3.1.6. Повнота і замкненість систем бульових функцій

Нехай $\mathcal{F}$ – об'єднання всіх множин F_n, $n = 1, 2, \ldots$.

Означення 3.10. *Система функцій $S \subset \mathcal{F}$ називається повною, якщо будь-яку функцію з $\mathcal{F}$ можна записати у вигляді суперпозиції функцій системи S.*

Приклади

1. Система $\mathcal{F}$ повна.
2. Система $S = \{\bar{x}, x_1 \vee x_2, x_1 \wedge x_2\}$ повна. Це випливає з теореми про диз'юнктивну нормальну форму.
3. Якщо до повної системи функцій долучити ще декілька функцій, то одержимо повну систему функцій.
4. Система $\{0, 1\}$ не є повною. Тут 0 і 1 тотожні функції.

Твердження 3.11. *Нехай S_1 та S_2 — дві системи функцій, причому S_2 — повна. Якщо будь-яку функцію системи S_2 можна записати у вигляді суперпозиції функцій системи S_1, то система S_1 повна.*

Доведення цього твердження очевидне. Наведемо декілька прикладів його застосування.

Приклади

1. $S_1 = \{\bar{x}, x_1 \wedge x_2\}$, $S_2 = \{\bar{x}, x_1 \vee x_2, x_1 \wedge x_2\}$. Маємо $x_1 \vee x_2 = \overline{\bar{x}_1 \wedge \bar{x}_2}$. Раніше ми вже бачили, що система S_2 повна, тому й система S_1 повна.
2. Розглянемо систему функцій $S_1 = \{x_1|x_2\}$, яка складається з єдиної функції $x_1|x_2 =_{df} \overline{x_1 \wedge x_2} = \bar{x}_1 \vee \bar{x}_2$. Маємо

$$x_1|x_1 = \bar{x}_1 \vee \bar{x}_1 = \bar{x}_1,$$

$$(x_1|x_2)\big|(x_1|x_2) = \overline{(x_1 \wedge x_2)}\big|\overline{(x_1 \wedge x_2)} = \overline{\overline{(x_1 \wedge x_2)} \wedge \overline{(x_1 \wedge x_2)}} =$$

$$= \overline{\overline{(x_1 \wedge x_2)}} \wedge \overline{\overline{(x_1 \wedge x_2)}} = x_1 \wedge x_2.$$

   Оскільки система S_2 повна за попереднім прикладом, то й система S_1 повна.
3. Нехай $S_1 = \{0, 1, x_1 \wedge x_2, x_1 + x_2$, де

$$x_1 + x_2 = \begin{cases} 1, & x_1 \neq x_2 \\ 0, & x_1 = x_2, \end{cases}$$

   $S_2 = \{\bar{x}, x_1 \wedge x_2\}$. Маємо $0 = x \wedge \bar{x}$, $1 = \bar{0} = \overline{x \wedge \bar{x}}$, $x_1 + x_2 = \overline{(x_1 \wedge \bar{x}_2) \wedge (\bar{x}_1 \wedge x_2)}$, $\bar{x} = 1 + x$. Отже, система S_1 повна тоді й лише тоді, коли система S_2 повна.

Нехай деяка підмножина множини функцій $\mathcal{F}$. *Замиканням* множини S називають множину всіх бульових функцій, які є суперпозиціями

функцій з множини S. Позначимо замикання множини S через $\tilde{S}$. Множина S називається замкненою, якщо вона збігається зі своїм замиканням $\tilde{S}$. Зазначимо, що множина S є повною, якщо $\tilde{S} = \mathcal{F}$.

Приклади

1. Якщо $S = \mathcal{F}$, то й $\tilde{S} = \mathcal{F}$.

2. Нехай $S = \{1, x_1 + x_2\}$. Замиканням цієї множини S є клас всіх лінійних функцій, тобто функцій, які мають вигляд

$$f(x_1, \ldots, x_n) = c_0 + c_1 x_1 + \ldots + c_n x_n \pmod 2,$$

де $c_i \in \{0, 1\}$.

Вправа. Показати, що замикання множини функцій S має такі властивості:

1) $\tilde{S} \supset S$;

2) $\tilde{\tilde{S}} = \tilde{S}$;

3) $S_1 \subset S_2 \Longrightarrow \tilde{S}_1 \subset \tilde{S}_2$;

4) $\widetilde{S_1 \cup S_2} \supset \tilde{S}_1 \cup \tilde{S}_2$.

3.1.7. Спрощення бульових функцій

Бульові функції використовують для спрощення так званих релейно-контактних і вентильних схем. У випадку релейно-контактних схем кожній змінній x (відповідно $\bar{x}$) бульової функції $f(x_1, \ldots, x_n)$ ставлять у відповідність перемикач, який пропускає струм тоді й лише тоді, коли x набуває значення 1 (відповідно 0). Операція $\vee$ інтерпретується як паралельне з'єднання перемикачів, а операція $\wedge$ — як послідовне з'єднання. $f(a_1, \ldots, a_n) = 1$ інтерпретується як «струм проходить через схему», коли стани перемикачів відповідають набору значень $(a_1, \ldots, a_n) \in \{0, 1\}^n$, а $f(a_1, \ldots, a_n) = 0$ інтерпретується як «струм не проходить через схему» при цьому наборі значень станів перемикачів. З теорем про диз'юнктивну та кон'юнктивну нормальні форми випливає, що для кожної бульової функції f можна побудувати електричний ланцюг — паралельно-послідовне з'єднання перемикачів, який пропускає струм тоді й лише тоді, коли f набуває значення 1.

Два електричні ланцюги вважають еквівалентними, якщо вони одночасно пропускають або не пропускають струм при однакових станах відповідних перемикачів. З двох еквівалентних ланцюгів простішим

вважається той, який складається з меншої кількості перемикачів. Наприклад, функції $f(x,y,z) = ((x \vee y)z \vee (\bar{x} \cdot z)$ відповідає ланцюг з п'яти перемикачів. Але

$$f(x,y,z) = ((x \vee y)z \vee (\bar{x} \cdot z) = xz \vee yz \vee \bar{x}z = (x \vee y \vee \bar{x})z = 1 \cdot z = z,$$

тому один з еквівалентних ланцюгів складається з єдиного перемикача z.

Існує ще один важливий тип схем, які трапляються при конструюванні обчислювальних пристроїв — так звані вентильні схеми. Вентильні схеми складаються з елементарних об'єктів — вентилів трьох типів:

1) вентиль $\rightrightarrows\boxed{\vee}\!\!\rightarrow$ з двома входами і одним виходом, на виході одержуємо 1 (наявність напруги), якщо хоч на один з двох входів подається 1;
2) вентиль $\rightrightarrows\boxed{\wedge}\!\!\rightarrow$ з двома входами і одним виходом, на виході одержуємо 1, якщо на обидва входи подається 1;
3) вентиль $\rightarrow\boxed{\neg}\!\!\rightarrow$ з одним входом і одним виходом, на виході одержуємо 1, якщо на вхід подається 0 і на виході одержуємо 0, якщо на вхід подається 1.

При конструюванні вентильних схем виникає проблема їх мінімізації, яка полягає в знаходженні для заданої бульової функції такого виразу, який мав би «простий вигляд». *Вважатимемо, що всі бульові функції, з якими ми працюємо, мають диз'юнктивну нормальну форму*, тобто мають вигляд $f = \cup_i f_i$, де $f_i = x_{i_1}^{\alpha_{i_1}} \cdot \dots \cdot x_{i_s}^{\alpha_{i_s}}$, $\alpha_{i_k} \in \{0,1\}$, $x^1 = x$, $x^0 = \bar{x}$. Функції f_i назвемо *мономами*. Кожне входження змінної у вираз для функції f назвемо *літералом*.

Означення 3.11. *Диз'юнктивна нормальна форма f називається мінімальною за літералами, якщо кожна еквівалентна їй диз'юнктивна нормальна форма g не містить меншої кількості літералів.*

Диз'юнктивна нормальна форма f називається мінімальною за мономами, якщо кожна еквівалентна їй диз'юнктивна нормальна форма g не містить меншої кількості мономів або такої ж кількості мономів, але з меншою кількістю літералів.

Означення 3.12. *Скажемо, що вираз g має наслідком вираз f, якщо не існує такого набору значень змінних, для якого g набуває значення 1,*

а f набуває значення 0. Кожний моном, який має наслідком f, назвемо імплікантом для f. Простим імплікантом назвемо імплікант, який перестає ним бути після вилучення будь-якого літерала.

Отже, прості імпліканти є найкоротшими серед імплікантів.

Твердження 3.12. *Нехай $f = \cup_i f_i$ – диз'юнктивна нормальна форма. Тоді кожний моном f_i є імплікантом для f.*

Доведення. Якщо при заданому наборі значень змінних f_i набуває значення 1, то й f набуває значення 1.

Твердження 3.13. *Кожна бульова функція f дорівнює сумі своїх простих імплікантів.*

Доведення. З теореми про диз'юнктивну нормальну форму випливає, що f дорівнює сумі імплікантів, і для кожного набору змінних, для якого f набуває значення 1 знайдеться деякий імплікант, який теж набуває значення 1.

Приклади

1. Розглянемо функцію $f(x,y) = xy \cup x\bar{y} \cup \bar{x}y$. Зрозуміло, що

$$f(x,y) = \begin{cases} 0 & \text{якщо } x = 0 \text{ і } y = 0; \\ 1 & \text{якщо } x = 1 \text{ або } y = 1. \end{cases}$$

Мономи xy, $x\bar{y}$ та $\bar{x}y$ не є простими імплікантами. Справді, xy та $x\bar{y}$ можна замінити на x, тоді $\bar{x}y$ на y.

2. Моном xy має наслідком функцію $f(x,y,z) = xyz \cup xy\bar{z}$. Він є простим імплікантом, бо жоден з мономів x, y не є імплікантом. Справді, $x = 1$ при $x = 1, y = 0, z = 0$, а $f(1,0,0) = 0$.

Означення 3.13. *Диз'юнктивна нормальна форма g функції f називається ненадлишковою, якщо:*

1) кожний моном, який входить в g, є простим імплікантом функції f;

2) після вилучення будь-якого монома з g одержують $f \neq g$.

Твердження 3.14. *Кожна диз'юнктивна нормальна форма g, мінімальна за літералами й мономами, є ненадлишковою.*

Доведення. Якщо функція g не є ненадлишковою, то або деякий моном не є простим імплікантом, тому його можна замінити коротшим мономом, або один моном можна викреслити, не змінивши g. В обох випадках функція g не є мінімальною за літералами й мономами.

Щоб знайти ненадлишкове мінімальне зображення бульової функції f, потрібно:

1) знайти множину всіх простих імплікантів функції f;

2) знайти ненадлишкові об'єднання простих імплікантів;

3) вибрати серед цих об'єднань мінімальні.

3.1.8. Обчислення простих імплікантів

Почнемо з введення необхідної термінології. Скажемо, що мультиплікативний *моном β накриває моном α* (еквівалентно *моном α накривається мономом β*), якщо кожен літерал, який входить в α входить і в β. Наприклад, моном xy накривається мономами $xyzt$, $xyz\bar{t}$, $xy\bar{z}t$, $xy\bar{z}\bar{t}$ і кожен з цих мономів від чотирьох змінних накриває моном xy. Мультиплікативний *моном β є доповненням монома α стосовно функції f*, якщо β накриває α і кожна змінна, яка входить в f, входить і в β. Наприклад, мономи $xy\bar{z}$ та xyz є доповненнями монома xy стосовно функції $f(x,y,z) = x\bar{y} \cup x\bar{z} \cup \bar{x}z$.

Алгоритм знаходження для заданої функції f її запису у вигляді ненадлишкового об'єднання простих імплікантів ґрунтується на простому твердженні 3.15.

Твердження 3.15. *Нехай g — досконала диз'юнктивна нормальна форма функції f.*

1. *Якщо моном α має наслідком f, то всі його доповнення входять у g.*
2. *Функція f дорівнює сумі h своїх (простих) імплікантів тоді й лише тоді, коли кожний моном з g накривається деяким поліномом з h.*

Доведення. 1. Нехай деяке доповнення β монома α не входить в g. Розглянемо такий набір значень для змінних функції g, для якого α набуває значення 1, β набуває значення 0. Тоді для цього набору α f набуває значення 1, $f = g$ набуває значення 0. Тому α не може бути імплікантом функції f.

2. Нехай f дорівнює сумі h своїх імплікантів. Тоді кожний моном α функції h, очевидно, має наслідком $f = g$. Якби якийсь моном β,

що входить в g не накривався мономом α, то існував би набір значень змінних, для якого $\beta = g = 0$ і $\alpha = f = 1$. Але $f = g$, тому отримуємо суперечність.

Навпаки, нехай при деякому наборі значень змінних функція g набуває значення 1. Тоді один з мономів β, що входять в g, набуває значення 1 для цього набору значень змінних. За припущенням β накривається деяким поліномом α з h. Позаяк для заданого набору значень змінних $\beta = 1 \Longrightarrow \alpha = 1 \Longrightarrow h = 1$, то f (тобто g) має наслідком h. З означення імпліканта випливає, що й h має наслідком f. Тому $f = h$.

Як наслідок з доведеного твердження випливає такий алгоритм переліку всіх простих імплікантів заданої ненульової бульової функції f.

1. *Знаходимо досконалу диз'юнктивну форму g функції f.*
2. *Об'єднуємо в g всілякі пари мономів вигляду αx та $\alpha\bar{x}$; записуємо $\alpha x \cup \alpha\bar{x} = \alpha$.*
3. *До одержаного списку коротших мономів знову застосовуємо крок 2. і т.д.*
4. *Завершивши список, викреслюємо з нього всі мономи, які накривають який-небудь моном зі списку — такі мономи напевно не можуть бути простими імплікантами.*

Приклади

1. Нехай $f(x,y,z) = xy\bar{z} \cup x\bar{y}x \cup x\bar{y}\bar{z} \cup \bar{x}\bar{y}\bar{z}$. Оформимо список, про який йдеться у щойно сформульованих правилах у вигляді такої таблиці.

$xy\bar{z}$	$x\bar{z}$
$x\bar{y}z$	$x\bar{y}$
$x\bar{y}\bar{z}$	$\bar{y}z$
$\bar{x}\bar{y}z$	

У першому стовпчику цієї таблиці ми записали всі мультиплікативні мономи, які входять в f. Далі, об'єднуючи перший і третій мономи, записуємо в другому стовпчику $x\bar{z}$, об'єднуючи другий і третій мономи, записуємо $x\bar{y}$, і, об'єднуючи другий і четвертий — записуємо $\bar{y}z$. Кожний з мономів першого стовпчика накриває якийсь моном з другого стовпчика, тому викреслюємо всі мономи першого стовпчика. Отже, функція f є сумою простих імплікантів $x\bar{z} \cup x\bar{y} \cup \bar{y}z$. Ця сума не мінімальна. Справді, $x\bar{z}$ накриває перший і третій мономи з першого стовпчика, а $\bar{y}z$ накриває другий і четвертий мономи. Отож, $f(x,y,z) = x\bar{z} \cup \bar{y}z$.

2. Розглянемо функцію

$$f(x,y,z,t) = xyz\bar{t} \cup xy\bar{z}t \cup xy\bar{z}\bar{t} \cup x\bar{y}\bar{z}t \cup \bar{x}\bar{y}\bar{z}t \cup \bar{x}yzt \cup \bar{x}\bar{y}zt \cup \bar{x}\bar{y}\bar{z}\bar{t}.$$

Для полегшення обчислень введемо такі позначення. Впорядкуємо змінні, після чого поставимо у відповідність кожному моному послідовність з нулів, одиниць

та символу _ — «пропуск», записуючи 0, якщо змінна входить у моном з рискою, одиницю, якщо вона входить без риски, та _ , якщо ця змінна в моном не входить. У таких позначеннях функцію f запишемо так:

$$f(x,y,z,t) = 1110 \cup 1101 \cup 1100 \cup 1001 \cup 0001 \cup 0111 \cup 0011 \cup 0000,$$

а функцію $g(x,y,z,t) = x\bar{t} \cup \bar{x}z \cup xyzt$ так: $g(x,y,z,t) = 1\_\_0 \cup 0\_1\_ \cup 1111$. Розіб'ємо двійкові вирази, що відповідають мономам функції f, на класи за кількістю одиниць і розмістимо їх у таблицю в порядку зростання кількості одиниць. Для нашої функції f одержимо

0000
0001
0011
1001
1100
0111
1101
1110 .

Правило $\alpha x \cup \alpha\bar{x} = \alpha$ може бути застосоване лише до пар, які розміщені в сусідніх класах. Помістимо всі мономи, які одержують з цієї пари сусідніх класів, в один клас. Повторюємо такі обчислення доки це можливо, помічаючи знаком $\surd$ використані пари: помічені мономи напевно не ввійдуть у список простих імплікантів. У результаті попередня таблиця виглядатиме так:

$\surd$0000	000_
$\surd$0001	00_1
$\surd$0011	_001
$\surd$1001	0_11
$\surd$1100	1_01
$\surd$0111	11_0 .
$\surd$1101	
$\surd$1110	

Отже, простими імплікантами є $\bar{x}\bar{y}\bar{z}$, $\bar{x}\bar{y}\bar{t}$, $\bar{y}\bar{z}t$, $\bar{x}\bar{z}t$, $x\bar{z}t$, $xy\bar{t}$. Для того щоб вибрати серед них мінімальне об'єднання, розглянемо таку таблицю.

	0000	0001	0011	1001	1100	0111	1101	1110
$\surd$000_	+	+						
00_1		+	+					
_001		+		+				
$\surd$0_11			+			+		
$\surd$1_01				+			+	
$\surd$11_0					+			+

У цій таблиці помічаємо знаком $\surd$ ті елементи першого стовпчика, які лежать в рядках, що визначаються стовпчиками з єдиним знаком +. З таблиці видно, що прості імпліканти, помічені знаком $\surd$ в першому стовпчику таблиці, накривають всі мономи функції f, тому $f(x,y,z,t) = \bar{x}\bar{y}\bar{z} \cup \bar{x}zt \cup x\bar{z}t \cup xy\bar{t}$.

3.2. Скінченні автомати

3.2.1. Означення та приклади

Означення 3.14. *Скінченним автоматом M називають набір $M = \{A, Z, S, \nu, \zeta\}$, де*

$A = \{a_0, a_1, \ldots, a_n\}$ *– скінченна множина вхідних символів (вхідний алфавіт);*

$Z = \{z_0, z_1, \ldots, z_m\}$ *– скінченна множина вихідних символів (вихідний алфавіт);*

$S = \{s_0, s_1, \ldots, s_r\}$ *– множина внутрішніх станів;*

$\nu : S \times A \to S$ *– функція переходу (в наступний стан);*

$\zeta : S \times A \to Z$ *– функція виходу.*

Отже, скінченний автомат складається з трьох множин A, S, Z та двох відображень ν і ζ.

Поняття скінченного автомата можна трактувати як математичну абстракцію, яка описує велику кількість різноманітних технічних пристроїв (серед них комп'ютери, цифрові програвачі та інші прилади).

Якщо розглядати скінченний автомат як математичну модель комп'ютера, то на послідовність символів множини A можна дивитися як на програму для комп'ютера. Ця програма за допомогою функції ν відповідним способом послідовно змінює стани для того, щоб за допомогою функції ζ одержати вихідну послідовність символів, яку можна трактувати як бажаний результат (наприклад, результат певних обчислень).

Приклади

1. Нехай $A = Z = \{0, 1\}$. Побудуємо скінченний автомат, який послідовність (x_0, x_1, x_2, x_3), $x_i \in \{0, 1\}$ перетворює (закодовує) в послідовність $(x_0 + 1 \bmod 2,\ x_1, x_2, x_3 + 1 \bmod 2)$. Зазначимо, що такий автомат можна трактувати як певний шифрувальний пристрій. Крім того, якщо «зашифровану» послідовність з нулів та одиниць подати на вхід цього автомата, то одержимо початкову послідовність, тобто цей автомат одночасно буде й «дешифрувальним пристроєм». Спробуємо пошукати такий автомат серед автоматів, для яких множина внутрішніх станів S складається з трьох елементів $S = \{s_0, s_1, s_2\}$. Задамо функції $\nu : S \times A \to S$ та $\zeta : S \times A \to Z$ так:

$$\begin{array}{ll} \nu(s_0, 0) = s_1 & \zeta(s_0, 0) = 1 \\ \nu(s_0, 1) = s_1 & \zeta(s_0, 1) = 0 \\ \nu(s_1, 0) = s_2 & \zeta(s_1, 0) = 0 \\ \nu(s_1, 1) = s_2 & \zeta(s_1, 1) = 1 \\ \nu(s_2, 0) = s_0 & \zeta(s_2, 0) = 1 \\ \nu(s_2, 1) = s_0 & \zeta(s_2, 1) = 0\ . \end{array}$$

Зауважимо, що дію скінченного автомата (тобто означення відображень ν, ζ) зручно записувати у вигляді таблиці (*таблиці станів*).

Функція	ν		ζ	
Стан	0	1	0	1
s_0	s_1	s_1	1	0
s_1	s_2	s_2	0	1
s_2	s_0	s_0	1	0

2. Нехай $A = \{0, 1, *\}$, $Z = \{0, 1, +, -\}$, $S = \{s_0, s_1\}$. Автомат з такою таблицею станів.

Функція	ν			ζ		
Стан	0	1	*	0	1	*
s_0	s_0	s_1	s_0	0	1	+
s_1	s_1	s_0	s_0	0	1	-

Автомат розпочинає роботу, перебуваючи в початковому стані s_0. Він одержує на вхід послідовність з нулів, одиниць і символів *. Якщо на вхід подається 0, то внутрішній стан автомата не змінюється, а символ 1 на вході змінює його стан на інший (з двох можливих). Автомат видруковує на виході ті самі символи, що й на вході, якщо це 0 або 1, і видруковує +, якщо з початку роботи (або з моменту останнього виникнення символу *) він одержав на вхід парну кількість одиниць (і, отже, парну кількість разів змінив свій стан), і видруковує «−» в протилежному випадку. Зокрема, цей автомат переробить послідовність $01100110 * 111 * 1001*$ на вході у послідовність $01100110 + 111 - 1001+$ на виході.

3.2.2. Покриття та еквівалентність автоматів

Якщо ми маємо скінченний автомат M, то на його вхід подаються скінченні впорядковані послідовності $a_0a_1 \ldots a_{n-1}$ символів вхідного алфавіту. Такі послідовності називатимемо *словами* або *рядками* довжини n.

Нехай на вхід подається рядок $\mathbf{a} = a_0a_1 \ldots a_{n-1}$. Тоді на виході одержуємо рядок $\mathbf{z} = z_0z_1 \ldots z_{n-1}$. Корисно також розглядати і рядок відповідних внутрішніх станів $\mathbf{s} = s_0s_1 \ldots s_{n-1}$. Зрозуміло, що рядки $\mathbf{z}$ і $\mathbf{s}$ однозначно визначаються рядком $\mathbf{a}$ та початковим станом s_0. Отже, ми маємо два відображення.

$$\nu_n : \ S \otimes A^n \to S^n, \ \ \zeta_n : \ S \otimes A^n \to Z^n,$$

для яких

$$\nu_i(s_0, \mathbf{a}) = \nu(s_{i-1}, a_{i-1}),$$
$$\zeta_i(s_0, \mathbf{a}) = \zeta(s_{i-1}, a_{i-1}).$$

Це означає, що відображення ν_n і ζ_n «рекурсивно» одержують з відображень ν і ζ, тобто, знаючи ν_i і ζ_i для $i \leq n$, можемо однозначно визначити ν_n і ζ_n.

Якщо скінченний автомат має багато внутрішніх станів, то відповідний йому реальний пристрій конструюється з багатьох електронних схем, які реалізують ці стани. На практиці це призводить до зростання вартості таких пристроїв, до зменшення їхньої надійності та до ускладнення обслуговування і ремонту. Тому важливим завданням є пошук автоматів з якомога меншою кількістю внутрішніх станів. Точніше, нехай задано скінченний автомат $M = \{A, S, Z, \nu, \zeta\}$. Потрібно замінити автомат M автоматом $M' = \{A, S', Z, \nu', \zeta'\}$ з меншою кількістю внутрішніх станів, але так, щоб M' переробляв вхідні послідовності у вихідні так само, як це робить M. У зв'язку з цим дають таке означення.

Означення 3.15. *Автомат M' покриває автомат M, якщо вони обидва мають однакові вхідні та вихідні алфавіти, й існує таке відображення $\varphi : S \to S'$, що*

$$\zeta_r(s, \mathbf{a}) = \zeta'_r(\varphi(s), \mathbf{a}) \ \forall \mathbf{a} \in A^r.$$

Автомат, який не можна покрити автоматом з меншою кількістю внутрішніх станів, називають мінімальним.

Якщо автомат M' покриває автомат M, то пишемо $M' \geq M$.

Означення 3.16. *Автомати M і M' називають еквівалентними, якщо $M' \geq M$ і $M \geq M'$.*

З означень безпосередньо випливає, що відношення еквівалентності автоматів є рефлексивним, симетричним і транзитивним.

Означення 3.17. *Стани s_i та s_j автомата M називають r-еквівалентними і пишуть $s_i \sim_r s_j$, якщо для кожного вхідного слова $\mathbf{a} \in A^r$*

$$\zeta_r(s_i, \mathbf{a}) = \zeta_r(s_j, \mathbf{a}).$$

Якщо $s_i \sim_r s_j$ для всіх r, то стани s_i та s_j називають еквівалентними.

Задача мінімізації заданого автомата зводиться до відшукання еквівалентних станів, а тоді до їх ототожнення. Для роз'яснення цього введемо ще дві функції $\nu^* : S \times A^r \to S$ і $\zeta^* : S \times A^r \to Z$:

$$\nu^*(s_i, \mathbf{a}) = \nu(\cdots(\nu(\nu(s_i, a_0), a_1)\cdots), a_{r-1}),$$

$$\zeta^*(s_0, \mathbf{a}) = \zeta(\cdots(\nu(\nu(s_i, a_0), a_1)\cdots), a_{r-1}).$$

Цим виражається той факт, що $\nu^*(s_i, \mathbf{a})$ є станом автомата, який, перебуваючи в стані s_i зчитав вхідну послідовність $\mathbf{a} = a_0a_1\cdots a_{r-1}$, а $\zeta^*(s_i, \mathbf{a})$ — останній символ слова одержаного на виході автомата, який, перебуваючи в стані s_i, зчитав вхідну послідовність $\mathbf{a} = a_0a_1\cdots a_{r-1}$.

Твердження 3.16. *Якщо $s_i \not\sim s_j$, то або $s_i \not\sim_1 s_j$, або для деякого слова $\mathbf{a} = a_0a_1\cdots a_{r-1}$ маємо $\nu^*(s_i, \mathbf{a}) \not\sim_1 \nu^*(s_j, \mathbf{a})$.*

Доведення. Якщо $s_i \not\sim s_j$, то знайдеться слово $\mathbf{a} = a_0a_1\cdots a_{r-1}$, для якого $\zeta^*(s_i, \mathbf{a}) \neq \zeta^*(s_j, \mathbf{a})$. Можна вважати, відкинувши в разі потреби декілька останніх символів слова $\mathbf{a}$, що вихідні слова, які відповідають станам s_i та s_j відрізняються лише останніми буквами. Якщо при цьому $r = 1$, то $s_i \not\sim_1 s_j$. Якщо $r > 1$, то $s_i \not\sim_r s_j$, і, так як вихідні слова відрізняються лише останніми буквами, $\zeta^*(s_i, \mathbf{a}) \neq \zeta^*(s_j, \mathbf{a})$, отже $\nu^*(s_i, \mathbf{a}) \not\sim_1 \nu^*(s_j, \mathbf{a})$.

Введемо такі позначення:

$$E_r = \{(s, s') \in S \times S \mid s \sim_r s'\},\ \bar{E}_r = S \times S \setminus E_r.$$

Твердження 3.17. *Якщо $s_i \not\sim_r s_j$, але $s_i \sim_k s_j$ для всіх $k < r$, то $\nu(s_i, a_l) \not\sim_{r-1} \nu(s_j, a_l)$ для деякої букви $a_l \in A$.*

Доведення. Якщо $(s_k, s_l) \in \bar{E}_{r-1}$, то $(s_k, s_l) \notin \bar{E}_r \setminus \bar{E}_{r-1}$. Тому достатньо розглянути лише пари (s_k, s_l), для яких існує слово $\mathbf{a} \in A^r$, таке що $\zeta^*(s_k, \mathbf{a}) \neq \zeta^*(s_l, \mathbf{a})$, але $\zeta^*(s_k, \mathbf{a}) = \zeta^*(s_l, \mathbf{a})$ для всіх слів $\mathbf{a} \in A^{r-1}$. Це якраз ті пари, які переводяться в $\bar{E}_1$ $r-1$ вхідним символом a_{r-2}, а тому переводяться в $\bar{E}_{r-1} \setminus \bar{E}_{r-2}$ першим символом a_0.

Наслідок 3.2.

$$\bar{E}_r \setminus \bar{E}_{r-1} = \Big\{(s_i, s_j) \in E_{r-1} \mid \exists a \in A\ (\nu(s_i, a), \nu(s_j, a)) \notin E_{r-1}\Big\}. \qquad (*)$$

Доведення. Це переформулювання попереднього твердження.

Наслідок 3.3. *Якщо $E_{r-1} = E_r$, то $E_{r-1} = E_{r+k}$ для всіх $k \geq 1$.*

Доведення. Якщо $E_{r-1} = E_r$, то й $\bar{E}_{r-1} = \bar{E}_r$. Тому з рівності (*) одержуємо

$$\bar{E}_{r+1} \setminus \bar{E}_r = \Big\{(s_i, s_j) \in E_r \mid \exists a \in A\ (\nu(s_i, a), \nu(s_j, a)) \notin E_r\Big\} =$$

$$= \left\{ (s_i, s_j) \in E_{r-1} \mid \exists a \in A \ (\nu(s_i, a), \nu(s_j, a)) \notin E_{r-1} \right\} = \bar{E}_r \setminus \bar{E}_{r-1} = \emptyset,$$

тому $\bar{E}_{r+1} = \bar{E}_r$, отже й $E_{r+1} = E_r$. Припустимо, що ми вже довели, що $E_{r-1} = E_{r+k}$. Оскільки за доведеним $E_{r+k} = E_{r+k_1}$, то математична індукція завершує доведення.

Доведені результати означають, що стани s_i, s_j, еквівалентні стосовно всіх вхідних слів довжини $r-1$ стають нееквівалентними стосовно деякого слова довжини r лише в тому випадку, коли знайдеться буква a вхідного алфавіту, яка переводить пару (s_i, s_j) у пару $(s_l, s_m) \notin E_{r-1}$. Враховуючи цей факт, можемо сформулювати такий *алгоритм мінімізації кількості станів скінченного автомата.*

1 крок. Знаходимо множину E_1.
Аналізуючи останній стовпчик таблиці скінченного автомата, виписуємо суміжні класи множини станів S. До одного суміжного класу ввійдуть ті елементи $s \in S$, для яких послідовності виходів однакові, якщо на вхід подається впорядкована послідовність усіх символів алфавіту A.

r крок. Знаходимо множину E_r.

До кожної пари елементів $(s_i, s_j) \in E_{r-1}$ і до кожної букви a вхідного алфавіту A знаходимо пару $(s_l, s_m) = (\nu(s_i, a), (\nu(s_j, a))$ і перевіряємо чи $(s_l, s_m) \in E_{r-1}$. Всі пари $(s_i, s_j) \in E_{r-1}$, для яких $(s_l, s_m) \notin E_{r-1}$, вилучаємо з E_{r-1}. В результаті одержуємо множину E_r.

Завершення роботи. Алгоритм припиняє роботу на r-ому кроці, якщо $E_{r-1} = E_r$.

Алгоритм припинить роботу через скінченну кількість кроків. Справді, якщо на r-ому кроці $E_{r-1} = E_r$, то за наслідком 3.3 кількість різних суміжних класів далі не змінюється. Це означає, що одержане розбиття множини станів S є розбиттям, відповідним відношенню еквівалентності станів, а не лише відношенню їх r-еквівалентності. З іншого боку, на деякому кроці обов'язково одержимо рівність $E_{r-1} = E_r$, бо множина E_1 скінченна і $E_1 \supset E_2 \supset \cdots \supset E_r \supset \cdots$.

Для зручності обчислень у прикладах введемо таке позначення:

$$(i, j) \xrightarrow{a} (l, m) \text{ означає } (s_l, s_m) = (\nu(s_i, a), (\nu(s_j, a)).$$

Приклади

1. Нехай скінченний автомат M заданий такою таблицею станів

Стан	Наступний стан			Вихід		
	a_1	a_2	a_3	a_1	a_2	a_3
s_1	s_2	s_2	s_5	1	0	0
s_2	s_1	s_4	s_4	0	1	1
s_3	s_2	s_2	s_5	1	0	0
s_4	s_3	s_2	s_2	0	1	1
s_5	s_6	s_4	s_3	1	0	0
s_6	s_8	s_9	s_6	0	1	1
s_7	s_6	s_2	s_8	1	0	0
s_8	s_4	s_4	s_7	1	0	0
s_9	s_7	s_9	s_7	0	1	1

1 крок. Переглядаючи стовпчик виходів цієї таблиці, виписуємо розбиття множини станів S на класи еквівалентності щодо відношення еквівалентності $\sim_1$ (тобто знаходимо множину E_1)

$$S = \{s_1, s_3, s_5, s_7, s_8\} \cup \{s_2, s_4, s_6, s_9\}. \tag{3.1}$$

2 крок. Для того щоб знайти множину E_2, нам треба вилучити з множини E_1 ті пари $(s_i, s_j) \in E_1$, для яких існує вхідний символ a, такий що $(i,j) \xrightarrow{a} (l,m)$ і $(s_l, s_m) \notin E_1$. З цією метою обчислюємо $(i,j) \xrightarrow{a} (l,m)$ для $(s_i, s_j) \in E_1$

$$1)\begin{cases} (1,3) \xrightarrow{a_1} (2,2) \\ (1,3) \xrightarrow{a_2} (2,2) \\ (1,3) \xrightarrow{a_3} (5,5), \end{cases} \quad 2)\begin{cases} (1,5) \xrightarrow{a_1} (2,6) \\ (1,5) \xrightarrow{a_2} (2,4) \\ (1,5) \xrightarrow{a_3} (3,5), \end{cases} \quad 3)\begin{cases} (1,7) \xrightarrow{a_1} (2,6) \\ (1,7) \xrightarrow{a_2} (2,2) \\ (1,7) \xrightarrow{a_3} (5,8), \end{cases} \quad 4)\begin{cases} (1,8) \xrightarrow{a_1} (2,4) \\ (1,8) \xrightarrow{a_2} (2,4) \\ (1,8) \xrightarrow{a_3} (5,7), \end{cases}$$

$$5)\begin{cases} (2,4) \xrightarrow{a_1} (1,3) \\ (2,4) \xrightarrow{a_2} (4,2) \\ (2,4) \xrightarrow{a_3} (4,2), \end{cases} \quad 6)\begin{cases} (2,6) \xrightarrow{a_1} (1,8) \\ \boxed{(2,6) \xrightarrow{a_2} (4,9)} \\ (2,6) \xrightarrow{a_3} (4,6), \end{cases} \quad 7)\begin{cases} (2,9) \xrightarrow{a_1} (1,7) \\ (2,9) \xrightarrow{a_2} (4,9) \\ \boxed{(2,9) \xrightarrow{a_3} (4,7)}. \end{cases}$$

З проведених обчислень випливає, що елементи $\{s_1, s_3, s_5, s_7, s_8\}$ залишаються еквівалентними щодо відношення еквівалентності $\sim_2$, а результат

$$\boxed{(2,9) \xrightarrow{a_3} (4,7)}$$

свідчить про те, що стани s_2 і s_9 вже не еквівалентні щодо $\sim_2$. Тому суміжний клас $\{s_2, s_4, s_6, s_9\}$ розщеплюється на декілька суміжних класів. Згідно з 5) і 6) s_2, s_4 і s_9 залишаються еквівалентними стосовно $\sim_2$. Це означає, що відношення еквівалентності $\sim_2$ визначає таке розбиття множини станів S на суміжні класи:

$$S = \{s_1, s_3, s_5, s_7, s_8\} \cup \{s_2, s_4, s_6\} \cup \{s_9\}. \tag{3.2}$$

3 крок. Щоб знайти множину E_3, потрібно вилучити з множини E_2 ті пари $(s_i, s_j) \in E_2$, для яких існує вхідний символ a, такий що $(i,j) \xrightarrow{a} (l,m)$ і $(s_l, s_m) \notin E_2$. Результати 1–5 засвідчують, що суміжний клас $\{s_1, s_3, s_5, s_7, s_8\}$ з (3.2) на цьому кроці не розщеплюється, а результат 6 у рамці свідчить про те, що суміжний клас $\{s_2, s_4, s_6\}$ розщеплюється, причому s_2 та s_4 залишаються еквівалентними щодо $\sim_3$; це випливає з (3.2) та з 5. Отже, відношення еквівалентності $\sim_3$ дає нам таке розбиття множини станів S на суміжні класи

$$S = \{s_1, s_3, s_5, s_7, s_8\} \cup \{s_2, s_4\} \cup \{s_6\} \cup \{s_9\}. \tag{3.3}$$

4 крок. Аналогічно шукаємо множину E_4. Результати 5 засвідчують, що суміжний клас $\{s_2, s_4\}$ не розщеплюється. Для того щоб з'ясувати чи розщеплюється суміжний клас $\{s_1, s_3, s_5, s_7, s_8\}$, обчислимо образи щодо функції ν станів s_3 і s_7

$$8)\begin{cases}\boxed{(3,7) \xrightarrow{a_1} (2,6)}\\ (3,7) \xrightarrow{a_2} (2,2)\\ (3,7) \xrightarrow{a_3} (5,8).\end{cases}$$

Це обчислення показує, що s_3 і s_7 належать різним суміжним класам щодо відношення еквівалентності $\sim_4$, а s_1 і s_3 та s_1 і s_8 залишаються в одному суміжному класі згідно з (3.2) і (3.3). Залишається з'ясувати, чи s_5 і s_7 належать різним суміжним класам. Для цього проведемо ще одне обчислення

$$9)\begin{cases}(5,7) \xrightarrow{a_1} (6,6)\\ (5,7) \xrightarrow{a_2} (4,2)\\ (3,7) \xrightarrow{a_3} (3,8).\end{cases}$$

Звідси бачимо, що стани s_5 і s_7 еквівалентні щодо відношення еквівалентності $\sim_4$, тому це відношення дає нам розбиття

$$S = \{s_1, s_3, s_8\} \cup \{s_5, s_7\} \cup \{s_2, s_4\} \cup \{s_6\} \cup \{s_9\}. \tag{3.4}$$

5 крок. Порівнюючи розбиття (3.4) з результатами 1, 4, 9 та 5, бачимо, що $E_5 = E_4$ і побудова мінімального автомата M', еквівалентного автомату M завершена. Мінімальний автомат M' має п'ять станів $\bar{s}_1, \bar{s}_2, \bar{s}_3, \bar{s}_4, \bar{s}_5$ і таку таблицю станів:

Стан	Наступний стан			Вихід		
	a_1	a_2	a_3	a_1	a_2	a_3
$\bar{s}_1$	$\bar{s}_2$	$\bar{s}_2$	$\bar{s}_3$	1	0	0
$\bar{s}_2$	$\bar{s}_1$	$\bar{s}_2$	$\bar{s}_2$	0	1	1
$\bar{s}_3$	$\bar{s}_4$	$\bar{s}_3$	$\bar{s}_1$	1	0	0
$\bar{s}_4$	$\bar{s}_1$	$\bar{s}_5$	$\bar{s}_4$	0	1	1
$\bar{s}_5$	$\bar{s}_3$	$\bar{s}_5$	$\bar{s}_3$	0	1	1

2. Нехай $A = \{0,1\}$, $Z = \{0,1\}$, $S = \{1,2,3,4,5,6,7,8\}$. Розглянемо автомат з такою таблицею станів

Стан	Наступний стан		Вихід	
	0	1	0	1
1	1	2	0	0
2	2	3	1	0
3	3	4	0	0
4	2	4	1	1
5	2	5	0	0
6	6	7	1	1
7	8	7	1	0
8	1	8	1	0

Останній стовпчик таблиці станів свідчить про те, що відношення еквівалентності E_1 на множині станів S розбиває множину S на суміжні класи так:

$$S = \{1,3,5\} \cup \{2,7,8\} \cup \{4,6\}.$$

У позначеннях попереднього прикладу

$$1)\left\{\begin{array}{c}(1,3)\xrightarrow{0}(1,3)\\ \boxed{(1,3)\xrightarrow{1}(2,4)}\end{array}\right.,\quad 2)\boxed{(1,5)\xrightarrow{0}(1,2)},\ 3)\boxed{(3,5)\xrightarrow{0}(3,2)},\ 4)\left\{\begin{array}{c}(2,7)\xrightarrow{0}(2,8)\\ \boxed{(2,7)\xrightarrow{1}(3,7)}\end{array}\right.,$$

$$5)\left\{\begin{array}{c}(2,8)\xrightarrow{0}(2,8)\\ \boxed{(2,8)\xrightarrow{1}(3,8)}\end{array}\right.,\quad 6)\boxed{(7,8)\xrightarrow{0}(8,1)},\ 7)\boxed{(4,6)\xrightarrow{0}(2,6).}$$

Взяті в рамку результати 1), 2), 3) показують, що стани 1, 3, 5 належать різним суміжним класам для відношення еквівалентності E_2, і так само з 4), 5), 6) робимо висновок: стани 2, 7, 8 теж належать різним суміжним класам. Нарешті, 7) означає, що стани 4 і 6 теж нееквівалентні. Отже, жодні два стани розглянутого автомата нееквівалентні, тому він мінімальний.

3.2.3. Суматор

Наведемо один важливий приклад скінченного автомата, який реалізує додавання $x+y$ натуральних чисел x і y, записаних у двійковій системі числення

$$x = x_n x_{n-1} \ldots x_1 \quad \text{і} \quad y = y_n y_{n-1} \ldots y_1,$$

де $x_i, y_i \in \{0,1\}$.

Розглянемо відомий алгоритм додавання чисел x і y «стовпчиком»

$$\begin{array}{cccc} q_{n+1} & q_n & \ldots & q_1 \\ + & x_n & \ldots & x_1 \\ & y_n & \ldots & y_1 \\ \hline z_{n+1} & z_n & \ldots & z_1 \end{array},$$

де $q_{n+1}, q_n, \ldots, q_1$ — результати перенесень з попередніх розрядів: $q_1 = 0$,

$$q_{i+1} = \left\{\begin{array}{ll} 0 & \text{якщо серед } q_i, x_i, y_i \text{ є не більше, ніж одна одиниця,} \\ 1 & \text{якщо серед } q_i, x_i, y_i \text{ є більше, ніж одна одиниця.} \end{array}\right. \tag{3.5}$$

Маємо

$$z_i = x_i + y_i + q_i \pmod 2. \tag{3.6}$$

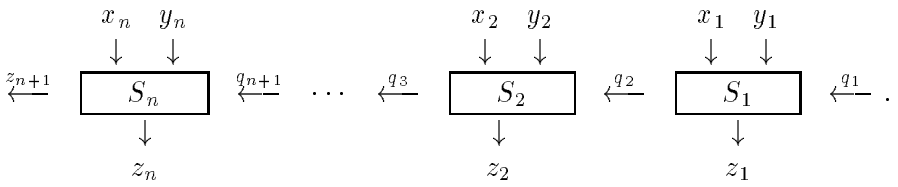
З іншого боку, легко перевірити (наприклад, складаючи таблички значень для бульових функцій, заданих формулами (3.5) і (3.6)), що

$$q_{i+1} = x_i y_i \cup x_i q_i \cup y_i q_i.$$

Так само, складаючи таблички значень для бульових функцій, перевіряємо тотожність

$$x_i + y_i + q_i \pmod 2 = (\overline{(x_i y_i \cup x_i q_i \cup y_i q_i)} \cdot (x_i \cup y_i \cup q_i)) \cup (x_i y_i q_i). \tag{3.7}$$

Використовуючи рівність (3.7), можна легко одержати вентильну схему (як це було пояснено у п. 3.1.7), яка перетворює трійку чисел x_i, y_i, q_i у двійку чисел z_i, q_{i+1}. Позначимо цю вентильну схему через S_i. Тоді суматор — пристрій, що реалізує додавання чисел, можна зобразити у вигляді такої діаграми:



3.2.4. Машини Тьюрінга

Розглянуте нами у п. 3.2.1. поняття скінченного автомата історично розвинулось з близького поняття, яке ввів у 1936 р. логік Тьюрінг. Він розглядав гіпотетичну «машину», яка має скінченну множину внутрішніх станів і одну нескінченно велику стрічку, розділену на комірки, яку машина могла пересувати за такт на одну комірку праворуч чи ліворуч. У кожну комірку машина може записувати символ із скінченного алфавіту A. Початково стрічка має бути порожньою, за винятком скінченної кількості комірок, які заповнюють завчасно. (Ці наперед заповнені комірки можна розуміти як програму запуску машини.)

Основні відмінності між машиною Тьюрінга та скінченним автоматом полягають в тому, що: 1) стрічка машини Тьюрінга нескінченна; 2) машина Тьюрінга може пересуватись по стрічці (чи зміщувати стрічку) в будь-якому напрямі. Це надає машині нескінченну пам'ять, яку можна використовувати під час обчислень. Кожну комірку можна переглядати багатократно. Наведемо формальне означення машини Тьюрінга.

Означення 3.18. *Машиною Тьюрінга назвемо п'ятірку* $[A, S, \nu, \zeta, \delta]$, *де* $A = \{a_0, a_1, \ldots, a_n\}$ — *скінченний алфавіт символів, які можуть бути записані в комірках і одночасно є вхідними та вихідними;* S — *скінченна множина внутрішніх станів,* $S = \{s_0, s_1, \ldots, s_r\}$; ν — *функція з* $S \times A$ *в* S; ζ — *функція з* $S \times A$ *в* A; δ — *функція з* $S \times A$ *в множину* $\{L, R, STOP\}$.

Машина Тьюрінга працює так. Вона починає роботу, перебуваючи в початковому стані s_0. Після зчитування першого символу вона переходить у новий внутрішній стан, який визначається функцією ν. Записує в комірку символ, який є значенням функції ζ. Переміщає стрічку направо (R), наліво (L), чи залишається на місці та закінчує роботу (STOP) залежно від значень функції δ.

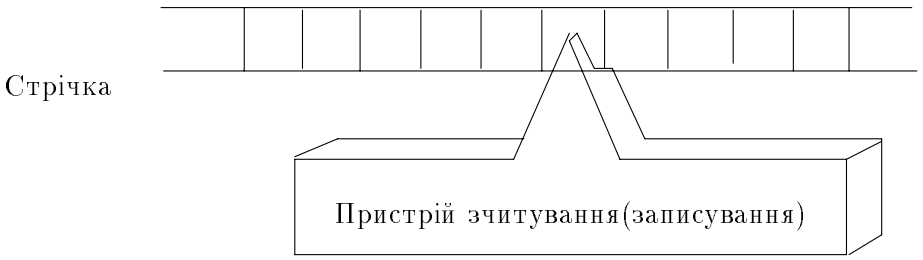


Рис. 3.1. Схематичне зображення стрічки машини Тьюрінга і пристрою записування та зчитування

Ще раз зазначимо, що робота машини полягає у повторенні такого циклу: зчитування символу з комірки, занесення нового символу у цю комірку, який вибирає функція ζ (може трапитись, що це той самий символ), зсув стрічки наліво чи направо або зупинка. Стрічка є нескінченною в обох напрямах, проте напочатку (і, отже, після будь-якого такту) заповнено лише скінченну кількість комірок.

Приклади

Машина Тьюрінга, яку описуємо нижче, зчитує вхідну послідовність нулів і одиниць. Якщо кількість одиниць парна, то в комірку заноситься П, якщо непарна, то Н. По обидві сторони від ряду нулів та одиниць йдуть порожні комірки, які позначатимемо #. Символи Н або П заносяться машиною у першу порожню комірку одразу за вхідним рядом. Алфавіт цієї машини має вигляд

$$A = \{\#, 0, 1, \text{П}, \text{Н}\}.$$

Внутрішні стани: $S = \{s_0, s_1, s_2\}$; s_0 — початковий стан. Машина зупиняється за

сигналом STOP.

$$
\begin{array}{lllllll}
\nu: & (s_0,0)\mapsto s_1 & \zeta: & (s_0,0)\mapsto 0 & \delta: & (s_0,0)\mapsto L\\
& (s_0,1)\mapsto s_2 & & (s_0,1)\mapsto 1 & & (s_0,1)\mapsto L\\
& (s_1,0)\mapsto s_1 & & (s_1,0)\mapsto 0 & & (s_1,0)\mapsto L\\
& (s_1,1)\mapsto s_2 & & (s_1,1)\mapsto 1 & & (s_1,1)\mapsto L\\
& (s_2,0)\mapsto s_2 & & (s_2,0)\mapsto 0 & & (s_2,0)\mapsto L\\
& (s_2,1)\mapsto s_1 & & (s_2,1)\mapsto 1 & & (s_2,1)\mapsto L\\
& (s_0,\#)\mapsto s_0 & & (s_0,\#)\mapsto \# & & (s_0,\#)\mapsto L\\
& (s_1,\#)\mapsto s_1 & & (s_1,\#)\mapsto \Pi & & (s_1,\#)\mapsto STOP\\
& (s_2,\#)\mapsto s_2 & & (s_2,\#)\mapsto \mathrm{H} & & (s_2,\#)\mapsto STOP
\end{array}
$$

Зручно задавати функції ν, ζ, δ, користуючись позначеннями Тьюрінга. У цьому випадку машина Тьюрінга задається скінченною множиною п'ятірок $[s_i,\ a_j,\ s_r,\ z_l,\ t_n]$, де

s_i — стан машини;

a_j — символ, що зчитується з комірки;

s_r — наступний стан машини, $s_r = \nu(s_i, a_j)$;

z_l — символ, що заноситься в комірку, $z_l = \zeta(s_i, a_j)$;

t_n — одна з команд R, L, STOP.

У цих позначеннях описану вище машину задають так:

$$
\begin{array}{lllll}
s_0 & \# & s_0 & \# & \mathrm{L}\\
s_0 & 0 & s_1 & 0 & \mathrm{L}\\
s_0 & 1 & s_2 & 1 & \mathrm{L}\\
s_1 & 0 & s_1 & 0 & \mathrm{L}\\
s_1 & 1 & s_2 & 1 & \mathrm{L}\\
s_2 & 0 & s_2 & 0 & \mathrm{L}\\
s_2 & 1 & s_1 & 1 & \mathrm{L}\\
s_1 & \# & s_1 & \Pi & \mathrm{STOP}\\
s_2 & \# & s_2 & \mathrm{H} & \mathrm{STOP}\ .
\end{array}
$$

Теорема 3.3 засвідчує, що машини Тьюрінга вміють робити все те, що вміють скінченні автомати.

Теорема 3.3. *Нехай* $M = [A,\ S,\ Z,\ \nu,\ \zeta]$ — *деякий скінченний автомат. Приймемо*

$$\overline{A} = A \cup Z \cup \{\Lambda\},$$

де Λ — *символ порожньої комірки, і для всіх* $(s_i, a_k) \in S \times A$

$$
\begin{array}{ll}
\bar{\nu}(s_i,a_k) = \nu(s_i,a_k) & \bar{\nu}(s_i,\Lambda) = s_i\ ;\\
\bar{\zeta}(s_i,a_k) = \zeta(s_i,a_k) & \bar{\zeta}(s_i,\Lambda) = \Lambda\ \ ;\\
\delta(s_i,a_k) = L & \delta(s_i,\Lambda) = STOP\ .
\end{array}
$$

Тоді машина Тьюрінга $T = [\overline{A},\ S,\ \bar{\nu},\ \bar{\zeta},\ \delta]$ *ставить у відповідність вхідній послідовності таку саму послідовність на виході, що і* M.

Доведення. Будь-яку вхідну послідовність $\mathbf{a} = a_0, a_1, \ldots, a_r$ автомата M можна записати на стрічці T так, щоб a_j був записаний у j-у комірку. Описана вище машина T занесе $z_j = \zeta(s_j, a_j)$ в j-у комірку, перейде в стан $s_{j+1} = \nu(s_j, a_j)$ і пересунеться в $(j+1)$ комірку. Дійшовши до $(r+1)$ комірки, вона зупиниться.

3.2.5. Приклади

Приклад 1. Машина Тьюрінга визначає за будь-якою вхідною послідовністю вигляду $\ldots\#\#111\ldots1100\ldots00\#\#\ldots$, (де # — порожні комірки) чи однакова кількість нулів та одиниць у такій послідовності. Її алфавіт складається з символів 0, 1, П, Н, # (зокрема, машина може «записувати» порожні комірки, тобто стирати наявний символ). Внутрішні стани такої машини:

$$S = \{s_0,\ s_1,\ s_2,\ s_3,\ s_4,\ s_5,\ s_6,\ s_7\}.$$

Випишемо тепер п'ятірки, які описують машину за Тьюрінгом

s_0	#	s_0	#	L
s_0	1	s_1	1	L
s_1	1	s_1	1	L
s_1	#	s_7	П	STOP
s_1	0	s_2	1	R
s_2	1	s_2	1	R
s_2	#	s_3	#	L
s_3	1	s_4	#	L
s_4	1	s_5	#	L
s_5	#	s_7	Н	STOP
s_5	0	s_7	П	STOP
s_5	1	s_1	1	L
s_0	0	s_6	0	R
s_6	0	s_6	0	R
s_6	#	s_7	П	STOP
s_6	1	s_1	1	L .

Завершуючи роботу, машина надрукує Н, якщо кількість нулів дорівнює кількості одиниць, П — в іншому випадку, і потім зупиниться.

Нескладно описати машини Тьюрінга, які обчислюють різні функції від чисел, поданих на вхід. Стандартним зображенням невід'ємного числа n в машині Тьюрінга є послідовність $n+1$ одиниць, які стоять підряд. Два таких числа розділені нулем. Отже, послідовність ...##111011##... зображає впорядковану пару (2,1).
Запис ...##111101101011##... зображає послідовність (3, 1, 0, 1).

Приклад 2. Наступна машина Тьюрінга додає два невід'ємних числа, які подаються на вхід

s_0	#	s_0	#	L
s_0	1	s_1	1	L
s_1	1	s_1	1	L
s_1	0	s_2	1	L
s_2	1	s_2	1	L
s_2	#	s_3	#	R
s_3	1	s_4	#	R
s_4	1	s_5	1	STOP .

Вона перетворює дві послідовності одиниць, розділених нулем, на послідовність одиниць, що зображає число, яке дорівнює сумі чисел на вході.

3.3. Деякі класичні алгоритми та складність обчислень

Термін *алгоритм* походить від імені арабського математика Мухаммеда ібн Муси аль-Горезмі, який у IX ст. написав трактат про десяткове позиційне кодування чисел і мистецтво виконання арифметичних операцій над цими кодованими числами.

На початку XII ст. трактат аль-Горезмі переклали латинською мовою, що й стало початком історії розвитку мистецтва «писемних» обчислень у Європі. Довгий час тривали дискусії про переваги методів з цього трактату над іншими, які використовували в обчисленнях рахівниці (abaci) або, наприклад, камінці (pebbles). (Відповідно прихильників методів, викладених у трактаті аль-Горезмі, називали *алгоритмістами*, інших — *абацистами* чи *рахівниками*.)

Вже наприкінці XVI ст. європейські вчителі (університети) рекламували те, що вони можуть навчити виконувати арифметичні операції лише за допомогою ручки та паперу.

Поняття алгоритму — плану обчислень — з часом розширилось, охопило не лише певні обчислювальні процедури, які виникли в античні часи (наприклад, алгоритм Евкліда описаний у п. 3.3.3.), а також аксіоматично означені класи абстрактних математичних об'єктів (наприклад, машина Тьюрінга, алгоритм Маркова).

Формально під алгоритмом розумітимемо описання об'єктів разом з описанням дій, які виконують над цими об'єктами для досягнення визначеної мети. Вважатимемо, що рівень деталізації описання алгоритму відомий апріорі, тобто, строго визначається деяким запасом дій (задається списком їхніх імен) і множиною простих класів об'єктів, яка не потребує подальшого детальнішого пояснення для користувачів алгоритму. Цього здебільшого досягають або аксіоматично (задаючи модель обчислень, наприклад, машину Тьюрінга), або технологічно (задаючи список операцій та операндів вбудованих у комп'ютер його розробниками).

Часто деяку задачу можна розв'язати за допомогою більше ніж одного алгоритму. Тоді виникає запитання, як серед декількох алгоритмів вибрати найкращий. Зрозуміло, що цей вибір має залежати від прийнятих критеріїв, які можуть бути різними для різного типу задач. Найчастіше критерієм є витрати часу, які можна наблизити кількістю елемен-

тарних операцій (таких як множення, додавання, порівняння), потрібних для виконання алгоритму. У багатьох випадках наближений час, потрібний для розв'язання задачі, вдається виразити у вигляді функції $t(n)$ від деякого характеристичного числа n (довжини вхідних даних) задачі. Тоді кажуть, що алгоритм розв'язує задачу за час $t(n)$, якщо на кожному вході довжини n він робить не більше ніж $t(n)$ кроків (елементарних операцій).

Алгоритми називають *поліноміальними*, якщо $t(n) \leq cn^c$, для деякої константи c. Алгоритми, часова складність яких не піддається подібній оцінці, називають *експоненційними*.

Вважається, що поліноміальні алгоритми відповідають швидким, ефективним на практиці алгоритмам, експоненційні — повільним і неефективним алгоритмам. Відповідно задачу, яку можна розв'язати лише експоненційним алгоритмом, чи для якої невідомі поліноміальні алгоритми, називають *важкою*.

3.3.1. Довжина числа та часова оцінка алгоритмів

Часову оцінку складності алгоритмів зручно записувати термінами $O(n)$. Нехай $f(n)$ і $g(n)$ — функції цілого аргумента, які для всіх n набувають додатні значення (не обов'язково цілі). Кажуть, що $f(n) = O\big(g(n)\big)$ (або коротко $f = O(g)$), якщо існує така константа C, що $\forall n \in \mathbb{N}$ число $f(n)$ є менше ніж $C \cdot g(n)$. Наприклад, $2n^2 + 3n - 3 = O(n^2)$. Справді, якщо $C = 3$, то $2n^2 + 3n - 3 < 3n^2$, $\forall n \in \mathbb{N}$. На практиці, вживаючи позначення O-великого, не враховують поведінки функцій f та g для малих значень аргументу. В зв'язку з цим приймають таке означення.

Означення 3.19. *Нехай для всіх аргументів $n \geq n_0$ функції f та g визначені, приймають додатні значення і для деякої константи C виконується нерівність $f(n) \leq C \cdot g(n)$. Тоді говорять, що $f = O(g)$ (f є O-велике від g).*

Зауваження 3.2.

1. *Записуючи $f = O(g)$, ми використовуємо знак рівності, проте маємо його розуміти як нерівність. Для прикладу, запис $n\sqrt{n} = O(n^2)$ є правильним, тоді як запис $n^2 = O(n\sqrt{n})$ є неправильним.*
2. *Очевидно, змінна не завжди має бути позначена літерою n. Позаяк задані вирази можуть містити різні константи, то треба*

чітко розрізнати, яка літера виражає змінну (див. приклад 5).

3. *Функція $g(n)$ повинна добре відображати характер росту функції $f(n)$, тобто бути досить точним обмеженням зверху. Такі твердження формально математично правильні, проте не вживаються на практиці: 1) $n^2 = O(n^3 + n\ln n + 2002)$; 2) $n^2 = O(e^{n^2})$; 3) $e^{-n} = O(n^2)$.*
4. *Припустимо, що $f(n)$ — сума доданків, серед яких один для великих n є значно більшим від інших. Якщо через $g(n)$ позначимо цей «домінуючий доданок», то можемо записати $f(n) = O\big(g(n)\big)$. Наприклад, якщо $f(n)$ — поліном степеня d, то «домінуючим» буде моном $a_d n^d$ і $f(n) = O(n^d)$.*
5. *Якщо $f(n)$ і $g(n)$ — функції, які для $n \geq n_0$ приймають додатні значення крім того, $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ є числом, то нескладно показати, що $f = O(g)$. Якщо ця границя дорівнює нулю, то кажуть, що f є о-мале від g і записують $f = o(g)$.*
6. *Якщо $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$, то записують $f \asymp g$ і кажуть, що f і g є асимптотично рівними.*
7. *Часто використовують ще два символи Ω, Θ, які пов'язані з позначенням О-велике. Запис $f = \Omega(g)$ рівносильний запису $g = O(f)$, вираз $f = \Theta(g)$ означає, що виконуються одночасно $f = O(g)$ і $g = O(f)$, тобто існують такі додатні константи C_1, C_2 і $n_0 \in \mathbb{N}$, що $C_1 g(n) \leq f(n) \leq C_2 g(n)$, для всіх $n \geq n_0$.*

Приклади

1. Якщо $f(n)$ — поліном степеня d зі старшим коефіцієнтом a_d, то $f(n) \asymp a_d n^d$.
2. (Твердження про щільність множини простих чисел.) $\pi(n) \asymp \frac{n}{\ln n}$, де $\pi(n)$ — кількість простих чисел, які не перевищують n.
3. Нехай ε позначає малу додатну константу (наприклад, $\varepsilon = 0,001$). Тоді $\ln n = O(n^\varepsilon)$, більше того $\ln n = o(n^\varepsilon)$. Справді, за допомогою правила Лопіталя можна легко переконатись, що $\lim_{n\to\infty} \frac{\ln n}{n^\varepsilon} = 0$.
4. Нехай $L_b(n)$ — кількість цифр запису числа n у системі числення за основою b. Фіксуємо b і трактуємо n як змінну. Тоді $L_b(n) = 1 + [\log_b n] = 1 + \left[\frac{\ln n}{\ln b}\right]$, де $[a]$ означають цілу частину числа a. Позаяк b — константа, а отже і $\ln b$ — константа, то $L_b(n) = O(\ln n)$.
5. Розглянемо суму $\sum_{i=1}^{n} i^k$. Якщо k зафіксоване і n необмежено зростає, то одержимо
$$f(n) = \sum_{i=1}^{n} i^k \asymp \frac{n^{k+1}}{k+1}.$$

(Щоб це довести, достатньо зауважити, що $\frac{1}{n^{k+1}} f(n)$ є частковою сумою інтеграла Рімана $\int_0^1 x^k dk$.) Якщо ми зафіксуємо n і трактуватимемо k як змінну, то твердження $f(k) = \sum_{i=1}^{n} i^k \asymp \frac{n^{k+1}}{k+1}$ є хибним. Наприклад, для $n = 2$ мало б виконуватись $1 + 2^k \asymp \frac{2}{k+1} 2^k$, що неправильно. Хибним є навіть слабше твердження $1 + 2^k = O\left(\frac{2^{k+1}}{k+1}\right)$.

6. Для функцій f, g від двох змінних m і n позначення $f = O(g)$ розуміємо так: існують константи C, n_0, m_0 такі, що $f(m,n) \le Cg(m,n) \quad \forall m \ge m_0,\ n \ge n_0$. Нехай $f(m,n)$ — кількість точок площини з цілими координатами, які належать області, обмеженій еліпсом з півосями m і n. $f(m,n)$ приблизно дорівнює площі еліпса, тобто πmn. Точне значення $f(m,n)$ залежить від положення еліпса в системі координат. У будь-якому випадку нескладно показати, що $f(m,n) \le 4mn$ для великих m і n, тобто $f(m,n) = O(mn)$.

Далі вважатимемо, що всі цілі числа записані у двійковій системі числення. Розглянемо арифметичні операції над такими числами. Як ми вже зауважили у прикладі 4, довжина $L(n)$ бінарного числа n є такою: $L(n) = L_2(n) = 1 + [\log_2 n] = 1 + \left[\frac{\ln n}{\ln 2}\right]$, тому $L(n) = O(\ln n)$. Останню рівність можна ще інтерпретувати, як оцінку об'єму необхідної комп'ютерної пам'яті для зберігання числа n.

Простежимо якою буде довжина числа, одержаного: а) додаванням; б) множенням n додатних цілих чисел, довжина кожного з яких не перевищує k. Легко зауважити, що довжина суми двох чисел або дорівнює довжині більшого з них, або є більшою від неї на 1. Якщо додаємо n чисел, з яких кожне має довжину щонайбільше k, то сума буде менша за $n2^k$. Тому довжина суми не перевищуватиме $k + L(n)$.

Для того щоб дати відповідь на запитання про довжину добутку, використаємо такий факт: число m довжини k задовольняє нерівність: $2^{k-1} \le m < 2^k$. Якщо $k = L(m_1)$, $l = L(m_2)$, то, перемноживши нерівності $2^{k-1} \le m_1 < 2^k$ і $2^{l-1} \le m_2 < 2^l$, одержимо $2^{k+l-2} \le m_1 m_2 < 2^{k+l}$. Звідси випливає, що довжина добутку $m_1 m_2$ дорівнює сумі довжин m_1 і m_2 або є від неї меншою на 1. Тобто, можна вважати, що при множенні двох чисел їхні довжини додаються. Інакше кажучи, довжина чисел поводиться як логарифм.

Нехай тепер ми хочемо перемножити n чисел $m_1, m_2, \ldots, m_n$ довжини k бітів. Позаяк $2^{k-1} \le m_i < 2^k \quad i = \overline{1,\ldots,n}$, то одержимо $2^{nk-n} \le \prod_{i=1}^{n} m_i < 2^{nk}$. Звідси довжина добутку лежить у межах від $nk - (n-1)$ до nk. Якщо нас цікавить лише точне обмеження зверху, то можемо стверджувати, що довжина добутку n k-бітових множників не перевищує nk.

Приклад

Припустимо, що потрібно оцінити довжину числа $n!$. Зауважимо, що всі множники в $n!$ мають довжину не більшу $L(n) = O(\ln n)$. Тоді за попереднім твердженням $L(n!) \leq n \cdot L(n) = O(n \ln n)$. Позаяк багато множників значно менші від n, то можна засумніватись у тому, що одержана оцінка є найкращою.

Насправді, серед чисел $1, 2, 3, \ldots, n$ є щонайменше $n/2$ чисел довжини $\log_2 n - 1$. Звідси випливає, що довжина $n!$ щонайменше $n/2(\log_2 n - 1)$ і для досить великих n первищує $C_2 n \ln n$, де C_2 — певним чином підібрана константа. Отже, $L(n!) = \Theta(n \ln n)$, тому наведена оцінка є найкращою.

3.3.2. Класичні алгоритми цілочисельної арифметики та їхня складність

Припустимо, що нам треба виконати арифметичні операції з великими цілими числами (наприклад, порядку 2^{1000}). Якщо б ми хотіли передоручити виконання цього завдання комп'ютерові, то довелось би розробляти програмне забезпечення для арифметики цілих чисел. (Це пов'язано з тим, що у комп'ютері апаратно реалізовано оператори $+$, $\cdot$, і т.д. для цілих чисел, які не перевищують 2^{16}, 2^{32} чи 2^{64}, а в деяких мовах програмування для цілих подвійної бінарної довжини, тобто не більше 2^{128}.) З програмістського погляду можна використати такі два підходи.

1. Написати процедуру, назвавши її SumInt, для додавання цілих чисел, — на вхід якої подаються цілі m і n, на виході одержуємо значення їхньої суми.
2. «Переозначити» оператор $+$: тобто, коли потрапляємо на оператор $+$, то перевіряється тип змінних, які є його аргументами і якщо виявлено «довгі» цілі числа, то викликається процедура SumInt. (Такий підхід називають «дружнім до користувача», тому що користувачеві не треба пам'ятати імена всіх процедур, які можуть йому знадобитись.)

Цей підхід можна застосовувати до будь-яких операцій з цілими числами.

Проаналізуємо як виконати додавання двох бінарних чисел, довжина яких не перевищує k. На потрібні міркування ми вже натрапляли у п. 3.2.3., коли описували роботу суматора. Розглянемо такий приклад

додавання «в стовпчик»:

$$\begin{array}{cccccccc}
1 & 1 & 1 & 1 & & & & \\
 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
\hline
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0
\end{array}$$

Якщо одне з чисел коротше від іншого, то дописуємо зліва від нього відповідну кількість нулів і виконуємо k разів такі кроки.

1. Починаючи з менших розрядів, зчитуємо верхній і нижній біт, перевіряємо, чи є над верхнім бітом «перенесення».
2. Якщо обидва біти є нулями і немає перенесення, то записуємо нуль і переходимо до наступної позиції.
3. Якщо (а) обидва біти — нулі і є перенесення, або (б) один з бітів є нулем, а другий дорівнює одиниці і не має перенесення, то записуємо 1 і переходимо до наступної позиції.
4. Якщо (а) один з бітів — нуль, другий дорівнює 1 і є перенесення або (б) обидва біти дорівнюють 1 і немає перенесення, то записуємо 0, запам'ятовуємо перенесення в наступний стовпчик і переходимо до наступної позиції.
5. Якщо обидва біти дорівнюють 1 і є перенесення, то записуємо 1, запам'ятовуємо перенесення в наступний стовпчик і переходимо до наступної позиції.

Однократне виконання описаної процедури називають *бітовою операцією*. Додавання двох k-бітових чисел потребує k бітових операцій. Загалом навіть дуже складні завдання, можна розбити на елементарні (бітові) операції. Час, який потрібний комп'ютерові для виконання завдання, здебільшого пропорційний до кількості елементарних операцій. Очевидно, що коефіцієнт пропорційності — це частка наносекунди, яка потрібна для виконання бітової операції, і залежить від характеристик конкретної комп'ютерної системи. (Зрозуміло, що це дещо спрощений підхід, тому що на час виконання завдання мають вплив «адміністраційні чинники», такі як доступ до пам'яті тощо.) Отже, час (тобто кількість бітових операцій), який затрачається на додавання двох чисел, дорівнює максимуму довжин доданків, що можна записати у вигляді

$$\text{час } \big(\text{(число } k\text{-бітове)} + \text{(число } l\text{-бітове)}\big) = \max(k,l).$$

Якщо використаємо залежність між числами та їхньою довжиною ($L(m) = O(\ln m)$), то одержимо

$$\text{час } \big(m+n\big) = O\big(\max(\ln m, \ln n)\big).$$

Зауважимо, що є відмінність між функціями часу виконання деякої операції записаних у термінах самих чисел (m, n) і термінах їхніх довжин (k, l). Залежно від ситуації обидва підходи можуть використовуватись, тому їх треба чітко розрізняти.

Розглянемо множення k-бітового числа на l-бітове. Використовуючи відомий ще зі школи метод множення «в стовпчик» одержимо, наприклад,

$$\begin{array}{cccccccccc}
 & & & & & 1 & 1 & 1 & 0 & 1 \\
 & & & & & & 1 & 1 & 0 & 1 \\
\hline
 & & & & & 1 & 1 & 1 & 0 & 1 \\
 & & & 1 & 1 & 1 & 0 & 1 & & \\
 & & 1 & 1 & 1 & 0 & 1 & & & \\
\hline
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 &
\end{array}$$

Загалом цей метод дає щонайбільше l рядків (кожний нульовий біт другого числа зменшує кількість рядків на 1). Кожний рядок є копією першого числа зсунотого на певну кількість позицій. Вільні позиції з правого боку можемо заповнити нулями. Далі виконуємо послідовне додавання рядків парами: наприклад, перший до другого, потім суму першого і другого рядків додаємо до третього і т. д. Отже, потрібно буде виконати $l - 1$ додавань. У кожному такому додаванні спочатку списуємо ті біти з верхнього рядка, які лежать над дописаними нулями. Це перенесення бітів не вважаємо бітовою операцією, а відносимо до «адміністраційних чинників», тобто нехтуємо ним оцінюючи складність алгоритму. Отже, кожне додавання потребує лише k бітових операцій. Тому сумарна кількість бітових операцій потрібних для досягнення результату не перевищує $k \cdot l$.

Перш ніж перейти до оцінювання часу, потрібного для виконання інших арифметичних операцій, зробимо декілька зауважень.

1. Якщо ми хочемо одержати простішу та зручнішу оцінку, то повинні розглянути «найгірший» з можливих випадків. Наприклад, у випадку множення можемо мати значно менше ніж $(l - 1)$ додавань, якщо не враховуватимемо нульові рядки. Розглядання таких ча-

сткових випадків не дасть жодної користі, бо наша мета оцінити час у термінах O-великого.

2. Згідно з визначеною домовленістю враховуємо лише кількість бітових операцій, нехтуючи операціями зсуву, доступу до пам'яті тощо.

3. Для оцінювання часу немає єдиної правильної відповіді. Наприклад, якщо оцінюємо час множення k-бітового числа m на l-бітове число n, то кожне з наведених тверджень є правильним: 1) час$=O(kl)$; 2) час$< O(kl)$; 3) час$\leq k(l-1)$; 4) якщо число n має у двійковому записі однакову кількість нулів і одиниць, то час$\leq kl/2$. Надалі використовуватимемо оцінки вигляду 1) та 2).

4. Оцінку часу множення можна виразити не лише через довжини множників, а й через них самих: час$(m \times n) = O(\ln m \ln n)$. Якщо ми множимо два числа приблизно однакової довжини, то можемо використовувати оцінку $O(k^2)$ чи $O(\ln^2 m)$.

5. Розглянутий нами метод множення цілих чисел у стовпчик очевидно є набагато швидшим від n-кратного додавання числа m до себе. Розроблено алгоритми множення двох k-бітових чисел, які потребують лише $O(k \ln k \ln\ln k)$ бітових операцій. Зрозуміло, що цей результат кращий, ніж $O(k^2)$ (він кращий ніж $O(k^{1+\varepsilon})$ для довільного $\varepsilon > 0$).

Ми обговорювали додавання та множення цілих чисел. Віднімання дуже схоже на додавання: час віднімання двох k-бітових чисел можна оцінити $O(k)$. Для цього треба розширити поняття бітової операції. Елементарний крок операції віднімання можна визначити аналогічно до додавання, лише операцію «перенесення» біта треба замінити операцією «запозичення» біта зі старшого розряду і утворити новий список всіляких альтернатив.

Розглянувши ділення цілих чисел у стовпчик, нескладно побачити, що кількість бітових операцій потрібних для ділення з остачею k-бітового числа на l-бітове число $(k \geq l)$ має порядок $O\big(l(k-l+1)\big)$. Отже можна вважати, що множення і ділення є арифметичні операції однакової (мультиплікативної) складності $O(k^2)$ і потребують значно більше часу, ніж додавання і віднімання, (адитивна) складність яких є порядку $O(k)$.

3.3.3. Алгоритм Евкліда та теорема Ламе

Одна з основних властивостей цілих чисел — властивість подільності з остачею чи евклідовість.

Твердження 3.18. *Для будь-яких $a, b \in \mathbb{Z}$, $b \neq 0$ існують і єдині (цілі числа) частка q і остача r такі, що $a = bq + r$, $0 \leq r < |b|$.*

Доведення. Розглянемо множину цілих чисел вигляду $a - kb$, де $k \in \mathbb{Z}$, тобто послідовність

$$\ldots,\ a-3b,\ a-2b,\ a-b,\ a,\ a+b,\ a+2b,\ a+3b,\ldots$$

Виберемо найменше невід'ємне число серед чисел цієї послідовності. (Це завжди можна зробити, бо наша множина містить від'ємні числа та (цілком впорядковану) підмножину множини натуральних чисел.) Позначимо вибране число r, а через q позначимо відповідне значення k. Одержимо $r = a - qb \geq 0$. Для доведення єдиності припустимо, що $\exists r' \neq r$ і $a = bq' + r'$, $0 \leq r' < |b|$. Нехай для визначеності $r' < r$, а отже, $0 < r - r' < |b|$. Оскільки $r - r' = (q' - q)b$, то $r - r'$ ділиться на b. Тоді $|b| \leq r - r'$, що суперечить нашому припущенню. Отож, $r = r'$.

Зауваження 3.3. *Для остачі використовуватимемо таке позначення $r = a \mod b$. Число r називатимемо (зведеним) лишком числа a за модулем b. Якщо a ділиться без остачі на b ($r = 0$), то записуватимемо цей факт так $b \mid a$. Кажуть також, що b ділить a, і називають b дільником числа a, число a — кратним числа b.*

Означення 3.20. *Нехай a і b цілі числа, які одночасно не дорівнюють нулю. Ціле число $d > 0$ називається найбільшим спільним дільником числа a і b, якщо*

1) $d \mid a$ *і* $d \mid b$;
2) якщо $c \mid a$ *і* $c \mid b$, *то* $c \mid d$.

Найбільший спільний дільник позначають $d = \text{НСД}(a, b)$ або просто $d = (a, b)$. Умова 2 з означення забезпечує єдиність найбільшого спільного дільника. Його існування випливає з такого твердження.

Твердження 3.19. *Якщо $a, b \in \mathbb{Z}$ і одночасно не дорівнюють нулю, то існують такі цілі числа u і v, що $НСД(a, b) = au + bv$.*

Доведення. Нехай d — найменше додатне ціле число вигляду $au + bv$, наприклад, $d = au_0 + bv_0$. (Тут, як і в твердженні 3.18, існування d випливає з цілком впорядкованості множини натуральних чисел.) Очевидно, що d задовольняє умову 2 означення 3.20. Доведемо від супротивного, що d задовольняє й умову 1. Припустимо, що це не так, і нехай для визначеності, d не ділить b. Тоді $b = dq + r$, $0 < r < d$, а отже, $r = b - dq = b - (au_0 + bv_0)q = a(-qu_0) + b(1 - qv_0)$, що суперечить мінімальності d.

Аналогічно до означення найбільшого спільного дільника можна дати означення найменшого спільного кратного двох цілих чисел.

Означення 3.21. *Нехай $a, b \in \mathbb{Z} \setminus \{0\}$. Ціле число $m > 0$ називається найменшим спільним кратним чисел a і b, якщо*

1) $a \mid m \quad i \quad b \mid m$;
2) *якщо* $a \mid c \quad i \quad b \mid c, \quad$ *то* $\quad m \mid c$.

Найменше спільне кратне чисел a і b позначають НСК(a, b) або $[a, b]$.

Теорема 3.4 (Існування НСК). *Якщо $a, b \in \mathbb{Z} \setminus \{0\}$, то їх найменше спільне кратне існує і правильна рівність НСК$(a, b) = |a \cdot b| /$ НСД(a, b).*

Доведення. Так як $ab \neq 0$ і $b \neq 0$, то $d =$ НСД$(a, b) \neq 0$. З рівностей $\frac{ab}{d} = a(b/d) = b(a/d)$ і того, що числа b/d і a/d цілі, випливає, що $m = \frac{|ab|}{d}$ — додатне спільне кратне чисел a і b. Нехай $c \neq m$ інше спільне кратне чисел a і b. Тоді існують такі цілі числа a' та b', що $c = aa' = bb'$. Оскільки $a \mid bb'$, то $(a/d) \mid b'$ і $b' = s(a/d)$, де $s \in \mathbb{Z}$. Тепер з рівностей $c = bb' = s(ab)/d$ випливає, що $m \mid c$.

Зауважимо, що єдиність НСК випливає з частини 2 означення 3.21 і з того, що m — додатне ціле число.

Тепер можемо викласти класичний *алгоритм Евкліда* обчислення найбільшого спільного дільника двох цілих чисел. Основна ідея, яку тут використовують — це такий факт: якщо $a = bq + r$ і d ділить a і b, то $d \mid r = a - bq$. Це правильно для будь-якого спільного дільника чисел a і b, зокрема для $d =$ НСД(a, b); тому НСД$(a, b) =$ НСД(b, r). Нехай $a_0 = a$ і $a_1 = b$; тоді

$$\begin{aligned} a_0 &= a_1 q_1 + a_2, \qquad 0 < a_2 < |a_1|; \\ a_1 &= a_2 q_2 + a_3, \qquad 0 < a_3 < a_2; \end{aligned}$$

$$\begin{aligned} &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad \text{(AE)}\\ a_{k-2} &= a_{k-1}q_{k-1} + a_k, \qquad 0 < a_k < a_{k-1};\\ a_{k-1} &= a_k q_k + 0. \end{aligned}$$

Позаяк $|a_1| > a_2 > a_3 > \ldots > 0$, то ця процедура на деякому кроці закінчиться й остання ненульова остача a_k дорівнюватиме НСД(a,b). Справді, з міркувань наведених вище, одержуємо

$$\text{НСД}(a_0, a_1) = \text{НСД}(a_1, a_2) = \ldots = \text{НСД}(a_k, 0) = a_k.$$

Приймемо НСД$(0,0) = 0$ і нехай $MOD(a,b)$ і $QUO(a,b)$ — програми, які повертають відповідно r — невід'ємну остачу від ділення a на b і q — частку цього ділення.

Алгоритм Евкліда

Вхід: a і $b \neq 0$;

Вихід: $d = \text{НСД}(a,b)$.

1. (Ініціалізація) $(a_0, a_1) := (a, b)$;
2. (Основний цикл) Доти, доки $a_1 \neq 0$ виконувати $(a_0, a_1) := (a_1, MOD(a_0, a_1))$;
3. (Вихід) Повернути $d := a_0$.

Приклад

Нехай $a = 342$ і $b = -612$. Алгоритм Евкліда обчислює таку послідовність: НСД$(342, -612)$ = НСД$(-612, 342)$ = НСД$(342, 72)$ = НСД$(72, 54)$ = НСД$(54, 18)$ = НСД$(18, 0) = 18$.

Для багатьох прикладних і теоретичних задач важливо вміти виражати найбільший спільний дільник двох цілих чисел a і b у вигляді НСД$(a,b) = au + bv$. Один із способів досягнення цього полягає у застосуванні алгоритму Евкліда і вираженню НСД зворотним використанням рівностей (AE).

Приклад

Для $a = 342,\ b = -612$ одержимо

$$\begin{aligned} 342 &= -612 \cdot 0 + 342;\\ -612 &= 342 \cdot (-2) + 72 \quad \text{або} \quad 72 = -612 - 342 \cdot (-2);\\ 342 &= 72 \cdot 4 + 54 \quad \text{або} \quad 54 = 342 - 72 \cdot 4;\\ 72 &= 54 \cdot 1 + 18 \quad \text{або} \quad 18 = 72 - 54 \cdot 1;\\ 54 &= 18 \cdot 3 + 0. \end{aligned}$$

Проведемо обчислення у зворотному порядку $18 = 72 - 54 \cdot 1 = (-612 - 342 \cdot (-2)) - (342 - 72 \cdot 4) = -612 + 342 + (-612 - 342 \cdot (-2)) \cdot 4 = 9 \cdot 342 + 5 \cdot (-612)$. Тобто, $u = 9$, $v = 5$, і ми розв'язали сформульовану задачу.

Ще один підхід до розв'язання цієї задачі полягає в застосуванні так званого *розширеного алгоритму Евкліда.* Його ідея така: одержимо значення чисел u і v, виконавши кроки алгоритму Евкліда для чисел a і b, якщо на кожному i-му кроці остачу a_i записуватимемо у вигляді $a_i = au_i + bv_i$. Розглянемо таку таблицю, перший стовпчик якої є послідовністю ділень алгоритму Евкліда, розв'язаних щодо остач, а у другому стовпчику остачі подають у вигляді $a_i = au_i + bv_i$.

$$\begin{array}{ll}
a_0 = a, & a_0 = au_0 + bv_0; \\
a_1 = b, & a_1 = au_1 + bv_1; \\
a_2 = a_0 - a_1 q_1, & a_2 = au_2 + bv_2; \\
a_3 = a_1 - a_2 q_2, & a_3 = au_3 + bv_3; \\
\ldots\ldots\ldots & \ldots\ldots\ldots \\
a_i = a_{i-2} - a_{i-1} q_{i-1}, & a_i = au_i + bv_i; \\
\ldots\ldots\ldots & \ldots\ldots\ldots \\
a_k = a_{k-2} - a_{k-1} q_{k-1}, & a_k = au_k + bv_k; \\
0 = a_{k-1} - a_k q_k, & 0 = au_{k+1} + bv_{k+1}.
\end{array}$$

Очевидно, $u_0 = 1$, $v_0 = 0$, $u_1 = 0$, $v_1 = 1$. Порівнюючи значення a_i взяті з різних стовпчиків, одержимо $a_i = au_i + bv_i = a_{i-2} - a_{i-1}q_{i-1} = (au_{i-2} + bv_{i-2}) - (au_{i-1} + bv_{i-1})q_{i-1} = a(u_{i-2} - u_{i-1}q_{i-1}) + b(v_{i-2} - v_{i-1}q_{i-1})$. Звідси випливає рекурентна процедура для обчислення u_i та v_i

$$\left[\begin{array}{l}
q_{i-1} = QUO(a_{i-2}, a_{i-1}); \\
a_i = a_{i-2} - a_{i-1}q_{i-1}; \\
u_i = u_{i-2} - u_{i-1}q_{i-1}; \\
v_i = v_{i-2} - v_{i-1}q_{i-1},
\end{array}\right.$$

яку ми можемо описати такою схемою

Розширений алгоритм Евкліда

Вхід: a і $b \neq 0$;

Вихід: $d, u, v, a', b' \in \mathbb{Z}$ такі, що $d = \text{НСД}(a, b) = au + bv$, $0 = aa' + bb'$.

1. (Ініціалізація). $(a_0, a_1) := (a, b)$; $(u_0, u_1) := (1, 0)$; $(v_0, v_1) := (0, 1)$.
2. (Основний цикл). Доти, доки $a_1 \neq 0$ виконувати
   $q := QUO(a_0, a_1)$; $(a_0, a_1) := (a_1, a_0 - a_1 * q)$;
   $(u_0, u_1) := (u_1, u_0 - u_1 * q)$; $(v_0, v_1) := (v_1, v_0 - v_1 * q)$.

3. (Вихід). Повернути $(d, u, v, a', b') := (a_0, u_0, v_0, u_1, v_1)$.

Зауважимо, що крім сформульованої задачі, розширений алгоритм Евкліда знаходить також такі $a', b' \in \mathbb{Z}$, що $aa' + bb' = 0$. Для прикладу, наведемо проміжні значення розширеного алгоритму Евкліда ($a = 342,\ b = -612$).

Крок	q	a_0	a_1	u_0	u_1	v_0	v_1
0	–	342	−612	1	0	0	1
1	0	−612	342	0	1	1	0
2	−2	342	72	1	2	0	1
3	4	72	54	2	−7	1	−4
4	1	54	18	−7	9	−4	5
5	3	18	0	9	−34	5	−19

На п'ятому кроці розширений алгоритм Евкліда завершується ($a_1 = 0$) і НСД$(a, b) = a_0 = 18$, $u = u_0 = 9$, $v = v_0 = 5$, $a' = u_1 = -34$, $b' = v_1 = -19$.

Проведемо аналіз часу роботи алгоритму Евкліда. Не зменшуючи загальності, можемо вважати, що $0 < b < a$. Оскільки присвоювання у пунктах 1 і 3 алгоритму Евкліда виконуються за час $O(1)$, то достатньо оцінити складність виконання k ділень основного циклу. Нагадаємо, що для виконання ділення $a = b \cdot q_1 + a_2$ потрібно щонайбільше $L(b) \cdot L(q_1)$ бітових операцій, де $L(b)$ — довжина числа b. Подібно, час виконання ділення $a_{i-1} = a_i q_i + a_{i+1}$ можна оцінити $L(a_i)L(q_i) \le L(b)L(q_i)$. Звідси сумарний час всіх ділень є порядку $O\big(\ln b\,(\ln q_1 + \ln q_2 + \ldots + \ln q_k)\big) = O\left(\ln b \cdot \ln \prod_{i=1}^{k} q_i\right)$. Нескладно побачити, що $\prod_{i=1}^{k} q_i \le a$. Отже, оцінка має вигляд $O(\ln a \ln b)$ або $O(\ln^2 a)$.

На завершення розгляду алгоритму Евкліда, доведемо теорему Ламе, яка визначає оцінку найгіршого випадку обчислення НСД(a, b). Така оцінка не найкраща, проте має оригінальне доведення і, напевно, є однією з перших теорем, в яких розглядається складність обчислень.

Теорема 3.5 (Ламе). *Кількість ділень, необхідних для знаходження найбільшого спільного дільника двох цілих чисел, не перевищує кількості (десяткових) цифр меншого з них, домноженої на 5.*

Доведення. Розглянемо послідовність чисел Фібоначчі 1, 1, 2, 3, 5, 8, 13, ..., в якій кожне число дорівнює сумі двох попередніх ($f_0 = 1$, $f_1 = 1$, $f_i = f_{i-1} + f_{i-2}$, $i \ge 2$).

Нескладно показати, що кількість членів послідовності $f_1,\ f_2,\ f_3, \ldots,$ які мають однакову кількість цифр, не менша чотирьох і не більша п'яти. Справді, якщо ми позначимо t_1 перший член з $(k+1)$ цифрою, то $10^k < t_1 < 2 \cdot 10^k$, бо t_1 є сумою двох k-цифрових чисел. Якби $t_0 \le (1/2)10^k$, то, позаяк $t_1 \le 2t_0$, одержали б $t_1 \le 10^k$, а це суперечність, бо t_1 має $k+1$ цифру. Тому $(1/2)10^k < t_0 < 10^k$ і позаяк $t_2 = t_1 + t_0$, то $(3/2)10^k < t_2 < 3 \cdot 10^k$. Продовжуючи далі так діяти, одержимо такі нерівності:

$$(5/2)10^k < t_3 < 5 \cdot 10^k;$$
$$4 \cdot 10^k < t_4 < 8 \cdot 10^k;$$
$$(13/2)10^k < t_5 < 13 \cdot 10^k;$$
$$(21/2)10^k < t_6 < 21 \cdot 10^k.$$

Звідси випливає, що група $(k+1)$-цифрових чисел Фібоначчі має не менше чотирьох і не більше п'яти членів.

Оскільки кількість членів послідовності $f_1,\ f_2,\ f_3, \ldots,\ f_{n-1}$ не перевищує $5L_{10}(f_n) - 1$, то кількість ділень, які треба виконати для знаходження НСД(f_{n+1}, f_n) не перевищує кількості цифр f_n, помноженої на п'ять.

Нехай тепер хочемо знайти найбільший спільний дільник двох цілих чисел a, b $(a > b)$ і нехай $f_{n-1} \le b < f_n$.

Вияснимо, як розподілені остачі відносно інтервалів $[f_{s-1}, f_s)$, утворених послідовністю $f_{n+1},\ f_n, \ldots,\ f_2,\ f_1$. Спочатку розглянемо випадок $q_1 = q_2 = \ldots = q_k = 1$. Якщо дві остачі r_h і r_{h-1} потраплять в один інтервал $[f_{s-1}, f_s)$ так, що $f_{s-1} \le r_h < r_{h-1} < f_s = f_{s-1} + f_{s-2}$, то $r_{h-1} = r_h + r_{h+1}$ і $r_{h+1} = r_{h-1} - r_h < f_s - f_{s-1} = f_{s-2}$. Отже, інтервал $[f_{s-2}, f_{s-1})$ не містить остач.

Отож, якщо всі частки в алгоритмі Евкліда дорівнюють 1, то остачі будуть розподілені так, що в кожному інтервалі, утвореному сусідніми числами Фібоначчі, буде не більше двох остач і кожному інтервалу, який містить дві остачі, передуватиме інтервал без остач.

Розглянемо випадок $q > 1$, тобто на деякому кроці алгоритму Евкліда одержимо $r_i = qr_{i+1} + r_{i+2} \ge 2 \cdot r_{i+1} + r_{i+2}$. Нехай f_j і f_{j+1} — два послідовні числа Фібоначчі, між якими лежить r_i. Тоді $r_i - 2r_{i+1} > 0$, $2f_j - f_{j+1} > 0$ і $2(f_j - r_{i+1}) - (f_{j+1} - r_i) > 0$. Звідси випливає, що $r_{i+1} < f_j$. Якщо r_{i+1} також менше f_{j-1}, то інтервал (f_{j-1}, f_j) не буде містити остач. Якщо ж $r_{i+1} \ge f_{j-1}$ і $r_{i+2} \ge f_{j-2}$, то $r_i \ge 2r_{i+1} + r_{i+2} \ge 2f_{j-1} + f_{j-2} = f_{j-1} + f_j = f_{j+1}$, а це суперечить тому, що $r_i \in [f_j, f_{j+1}]$.

Отже, якщо частка в алгоритмі Евкліда більша 1, то знайдеться хоча б один інтервал у послідовності Фібоначчі, який не містить остач і це не компенсується інтервалом з двома остачами.

Отже, для того щоб послідовність остач $r_1,\ r_2, \ldots,\ r_k$ мала таку саму довжину, що й послідовність $f_n,\ f_{n-1}, \ldots,\ f_2,\ f_1$, частки в усіх операціях ділення повинні дорівнювати 1. Тоді $r_k = 1$, але $f_1 = 1,\ f_2 = 2$ і тому r_{k-1} не може дорівнювати 2, бо інакше ці послідовності однакові і $b = f_{n+1}$, що не так. Звідси випливає, що r_{k-1} дорівнює щонайменше 3 і послідовність остач матиме строго меншу довжину ніж відповідна послідовність чисел Фібоначчі.

3.3.4. Бінарний алгоритм піднесення до степеня

Розглянемо задачу обчислення функції $f(x) = x^N$ в кільці $\mathbb{Z}_m$. Ця задача має прикладне значення: піднесення до степеня за модулем деякого числа m використовують, наприклад, у реалізації криптосистем з відкритим ключем (див. п. 4.3.2.). Користуючись означенням степеневої функції, можна запропонувати прямолінійну програму для її обчислення, яка потребує $(N-1)$ множень.

Вхід: $x,\ N,\ m$;

Вихід: $f := x^N \mod m$.

1. (Ініціалізація). $f := MOD(x, m)$; $i := 1$.
2. (Основний цикл). Доти, доки $i < N$ виконувати $f := MOD(f * x, m)$; $i := i + 1$.
3. (Вихід). Повернути f.

Нехай m — натуральне число довжини k бітів, N — l-бітове натуральне число. Завжди можемо вважати, що $N < m$. Якщо це не так, то використовуючи теорему Ойлера ($x^{\varphi(m)} \equiv 1 \mod m$, де $m > \varphi(m)$ — значення функції Ойлера від m), показник N завжди можна понизити. Якщо прийняти, що N і x є порядку m, то алгоритм обчислення $x^N \mod m$ за наведеною схемою буде експоненційним. Справді, він виконує порядку 2^k множень вхідного числа довжини k, на що потрібно $O(k2^k)$ бітових операцій.

Ще до нашої ери в Індії був відомий ощадливіший алгоритм піднесення до степеня, який називають бінарним чи методом багатократного піднесення до квадрата.

Твердження 3.20. *Існує поліноміальний алгоритм знаходження лишку за модулем m від x^N, складність якого $O(k^2l)$, де k, l — довжини відповідно чисел m і N.*

Доведення. Запишемо показник N у двійковій системі числення $N = b_{l-1} \cdot 2^{l-1} + b_{l-2} \cdot 2^{l-2} + \ldots + b_1 \cdot 2 + b_0$, де $b_i \in \{0, 1\}$ $i = \overline{0, 1, \ldots, l-1}$. Далі послідовно обчислюємо лишки x^{2^i} за модулем m, підносячи до квадрата лишок $x^{2^{i-1}} \mod m$, одержаний на попередньому кроці. Позаяк довжина чисел на кожному кроці є менша k, то час виконання одного кроку є порядку $O(k^2)$.

Нехай тепер $i_1, i_2, \ldots, i_s$ є тими індексами, для яких $b_{i_j} = 1$, $j = \overline{1, \ldots, s}$. Тоді $N = \sum_{j=1}^{s} 2^{i_j}$ і $x^N = \prod_{j=1}^{s} x^{2^{i_j}}$. Тому множимо, наприклад, спочатку $x^{2^{i_1}}$ на $x^{2^{i_2}}$ за модулем m, далі одержаний результат множимо на $x^{2^{i_3}}$ і шукаємо лишок за модулем m і т. д. У результаті одержимо $x^N \mod m$. Очевидно, що таких множень у найгіршому випадку буде $l - 1$. Звідси випливає, що наведений алгоритм бінарного піднесення до степеня потребує часу порядку $O(k^2l)$ чи $O(k^3)$, тобто є поліноміальним.

Наведемо схему бінарного алгоритму піднесення до степеня.

Вхід: $x,\ m,\ N = (b_{l-1}, b_{l-2}, \ldots, b_1, b_0)_2$.
Вихід: $f := x^N \mod m$.

1. (Ініціалізація). $i := 0$; $a := MOD(x, m)$; $f := 1$.
2. (Основний цикл). Доти, доки $i < l$
   виконувати: [якщо $b_i \neq 0$, то $f := MOD(f * a, m)$;
   $a := MOD(a * a, m)$; $i := i + 1$;].
3. (Вихід). Повернути f.

Зауважимо таке: якщо НСД$(x, m) = 1$ і $N = \varphi(m) - 1$, то наведений алгоритм ефективно обчислює $x^{-1} \in \mathbb{Z}_m$.

3.3.5. Типи задач та їхня звідність

Далі для загального опису завдання ми вживатимемо термін *масова задача* чи просто *задача*, тоді як кожний конкретний приклад задачі називатимемо *індивідуальною задачею*.

Приклади

1. *Задача факторизації* цілого числа є завданням знаходження нетривіального дільника числа N або визначення, що такого дільника немає (тобто N — просте число). Коли задане конкретне число N, для якого шукаємо дільник, то маємо справу з індивідуальною задачею.

2. *Задача комівояжера* є завданням знаходження найкоротшого шляху, який виходить з міста A, проходить через всі інші наперед задані міста і повертається до міста A. Індивідуальною задачею комівояжера є конкретний список міст і відстаней між ними. (Залежно від того, що комівояжер хоче мінімізувати, список замість відстаней може містити ціни автобусних білетів між заданими містами, витрати пального тощо.)
3. *Задача трьох фарб* є завданням (якщо це можливо) — розмалювати задану карту за допомогою трьох кольорів так, щоб будь-які дві сусідні області були розмальовані у різні кольори. Насправді, природніше ставити задачу про розмалювання графа, бо ця задача є загальнішою (пояснення цього див. на с. 155). Задача розмалювання графа полягає у зіставленні кожній вершині графа одного з трьох кольорів так, щоб жодне ребро не з'єднувало вершини однакового кольору.

Під терміном *вхідні дані* чи *умова* задачі розумітимемо будь-яку інформацію, яку треба подати для описання індивідуальної задачі. Для задачі розкладу на множники вхідними даними є число N. Нехай *довжина вхідних даних* означатиме кількість символів, потрібних для запису цих даних. Вважатимемо, що зафіксовано деяку систему символів (вхідний алфавіт). Якщо, наприклад, вхідний алфавіт складається з двох символів $\{0, 1\}$, то дані задачі розкладу на множники записують у двійковій системі числення і їхня довжина становить $1 + [\log_2 N]$.

Якщо пронумеруємо в задачі комівояжера міста від 1 до m, то вхідними даними буде конкретне відображення з множини пар (i, j), $1 \le i < j \le m$ у множину натуральних чисел $\mathbb{N}$. (Припускаємо, що всі відстані між містами додатні цілі числа.)

Якщо занумерувати вершини графа в задачі трьох фарб, то вхідні дані можна трактувати як підмножину пар (i, j), $1 \le i < j \le m$. Інакше кажучи, вхідні дані — це граф $G = (V, E)$, де V — множина вершин $\{1, 2, \ldots, m\}$, а $E \subset \{(i, j) | 1 \le i < j \le m\}$ — множина ребер.

Залежно від умови та розв'язку задачі розрізняють декілька типів задач. Покажемо як можна модифікувати наші задачі, щоб отримати *задачі розпізнавання*, тобто такі задачі, результатом розв'язання яких є відповідь 1 («так») або 0 («ні»). Якщо розв'язанням задачі є щось більше ніж відповідь «так» або «ні», то такі задачі називатимемо *задачами пошуку*.

Зауваження. На відміну від задач розпізнавання, задачі пошуку можуть мати декілька правильних розв'язань. Наприклад, у задачі комівояжера шукаємо шлях найменшої довжини, який проходить через всі міста. (Шлях, що проходить через всі міста і повертається до по-

чаткового пункту, називається в теорії графів *гамільтоновим циклом*.) Очевидно, граф може містити декілька різних гамільтонових циклів мінімальної довжини.

Приклади

1. Індивідуальна задача розпізнавання факторизації цілого числа виглядає так.

   ЗАДАНО: додатні цілі числа N і k.

   ЗАПИТАННЯ: чи N має дільник M, для якого $2 \leq M \leq k$?

   Задача знаходження нетривіального дільника M числа N є задачею пошуку факторизації чисел.

2. Індивідуальну задачу комівояжера у формі задачі розпізнавання можна записати так.

   ЗАДАНО: число $m \in \mathbb{N}$, відображення з множини пар (i, j), $1 \leq i < j \leq m$ у множину натуральних чисел і натуральне число k.

   ЗАПИТАННЯ: чи існує гамільтонів цикл довжини $\leq k$?

   Задачею пошуку для комівояжера є завдання знаходження гамільтонового циклу найменшої довжини.

3. Індивідуальна задача розпізнавання для трьох фарб має такий вигляд:

   ЗАДАНО: граф $G = (V, E)$.

   ЗАПИТАННЯ: чи можна розмалювати граф G трьома кольорами, тобто чи існує відображення $c : V \to \{1, 2, 3\}$ таке, що $(i, j) \in E \Longrightarrow c(i) \neq c(j)$?

Для більшості задач, враховуючи наведені нами у прикладах, відповідні задача розпізнавання та задача пошуку є рівносильними. Це означає, що алгоритм, який розв'язує задачу одного типу, можна легко перетворити в алгоритм розв'язання задачі іншого типу. Не наводячи формального доведення, покажемо як це відбувається на прикладі задачі факторизації. Спочатку припустимо, що існує алгоритм, який розв'язує задачу пошуку. Тоді, застосовуючи цей алгоритм, до заданого N можемо знайти його нетривіальний дільник M. За допомогою цього ж самого алгоритму знаходимо нетривіальні дільники чисел M та N/M і т. д., доки не одержимо розклад числа N на прості множники. Тепер можемо дати відповідь на запитання, чи N має дільник у межах від 2 до k. Відповідь буде «так» тоді і тільки тоді, коли найменший простий дільник N лежить в інтервалі $[2, k]$.

Припустимо, що маємо алгоритм розв'язання задачі розпізнавання. Для знаходження значення нетривіального дільника числа N можемо застосувати метод послідовних питань (*бінарного пошуку*). За допомогою цього методу обчислимо нетривіальний дільник числа N

біт за бітом, починаючи з більш значущих. Нехай 2^n — найменший степінь двійки більший, ніж N. Іншими словами, n є довжиною N: $n = 1 + [\log_2 N]$. Спочатку застосуємо алгоритм, який розв'язує задачу розпізнавання для $k = 2^{n-1} - 1$. Якщо відповідь буде «ні», то N є простим числом, бо кожний нетривіальний дільник M числа N повинен задовольняти нерівність $M \leq N/2 < 2^{n-1}$. У цьому випадку ми вже розв'язали задачу пошуку. Припустимо, що відповідь була «так». Тоді повторимо наш алгоритм для процесу розпізнавання з $k = 2^{n-2} - 1$. Якщо одержимо відповідь «ні», то N має нетривіальний дільник $M = 1 \cdot 2^{n-2} + b_{n-3} \cdot 2^{n-3} + \ldots + b_0$, де b_i — біти двійкового запису числа M. Якщо відповідь була б «так», то N має нетривіальний дільник подібного вигляду лише з першим бітом 0, тобто $M = b_{n-3} \cdot 2^{n-3} + \ldots + b_0$. Для того щоб знайти біт b_{n-3}, приймемо

$$k = \begin{cases} 2^{n-2} + 2^{n-3} - 1, & \text{якщо попередня відповідь «так»;} \\ 2^{n-3} - 1\ , & \text{якщо попередня відповідь «ні».} \end{cases}$$

Якщо алгоритм задачі розпізнавання дає відповідь «ні», то $b_{n-3} = 1$. Якщо відповідь «так», то $b_{n-3} = 0$. Продовжуючи виконувати кроки подібно, одержимо всі біти числа M. Після n-кратного застосування алгоритму розв'язання задачі розпізнавання одержимо нетривіальний дільник числа N. Отже, відомий алгоритм задачі розпізнавання було модифіковано в алгоритм, який розв'язує задачу пошуку.

Приклад

Нехай відомо алгоритм задачі розпізнавання факторизації. Знайдемо нетривіальний дільник 119 методом бінарного пошуку.

1. ЗАПИТАННЯ: чи існує дільник 119 в межах від 2 до 63?
   ВІДПОВІДЬ: так.
2. ЗАПИТАННЯ: чи існує дільник 119 в межах від 2 до 31?
   ВІДПОВІДЬ: так.
3. ЗАПИТАННЯ: чи існує дільник 119 в межах від 2 до 15?
   ВІДПОВІДЬ: так.
4. ЗАПИТАННЯ: чи існує дільник 119 в межах від 2 до 7?
   ВІДПОВІДЬ: так.
5. ЗАПИТАННЯ: чи існує дільник 119 в межах від 2 до 3?
   ВІДПОВІДЬ: ні.
6. ЗАПИТАННЯ: чи існує дільник 119 в межах від 2 до 5?
   ВІДПОВІДЬ: ні.
7. ЗАПИТАННЯ: чи існує дільник 119 в межах від 2 до 2?
   ВІДПОВІДЬ: ні.

Звідси випливає, що $(0000111)_2 = 7$ є нетривіальним дільником 119.

Зауважимо, що за допомогою запропонованого методу ми завжди одержимо найменший нетривіальний дільник числа N.

Нехай Π_1 і Π_2 — дві масові задачі розпізнавання.

Означення 3.22. *Говоритимемо, що* Π_1 *поліноміально зводиться до* Π_2 *(або просто* Π_1 *зводиться до* Π_2*), якщо існує алгоритм, який є поліноміальним щодо довжини вхідних даних задачі* Π_1 *і для кожної індивідуальної задачі* P_1 *масової задачі* Π_1 *будує таку індивідуальну задачу* P_2 *масової задачі* Π_2*, що* P_1 *і* P_2 *мають однакову відповідь.*

Припустимо, що ми маємо ефективний алгоритм розв'язання задачі Π_2. Якщо Π_1 зводиться до Π_2, то можемо використати алгоритм Π_2 для розв'язання Π_1. Застосувавши алгоритм з означення 3.22 до деякої індивідуальної задачі з Π_1, можемо знайти за поліноміальний час відповідну індивідуальну задачу з Π_2. Відповідь, яку дає алгоритм розв'язання задачі Π_2, збігається з відповіддю задачі Π_1. Тобто, алгоритм для задачі Π_2 дає нам зразу алгоритм розв'язання задачі Π_1. Якщо алгоритм для задачі Π_2 був поліноміальним, то таким буде й алгоритм, одержаний для задачі Π_1.

Приклад

Нехай Π_1 буде такою задачею.

ЗАДАНО: поліном другого степеня $p(x)$ з цілими коефіцієнтами.

ЗАПИТАННЯ: чи $p(x)$ має два різні дійсні корені?

Нехай Π_2 — це така задача.

ЗАДАНО: ціле число N.

ЗАПИТАННЯ: чи N є додатним числом?

Покажемо, що Π_1 зводиться до Π_2. Нехай $p(x) = ax^2 + bx + c$ буде індивідуальною задачею Π_1. Значення $N = b^2 - 4ac$ можна обчислити за поліноміальний час. Задача Π_2 на вході N має позитивну відповідь тоді і тільки тоді, коли позитивну відповідь на вході $ax^2 + bx + c$ має задача Π_1.

Означення 3.22 можна також використати для доведення того, що задача є важкорозв'язною.

Припустимо, що задача Π_1 важка і ми впевнені, що для неї не існує ефективного алгоритму. Якщо Π_1 зводиться до Π_2, то для Π_2 також не існує ефективного алгоритму.

Наведене означення звідності задач є дещо вузьким. Було б корисно мати таке означення звідності задачі Π_1 до Π_2, яке б давало змогу використовувати для розв'язання індивідуальної задачі з Π_1 кілька різ-

них індивідуальних задач з Π_2. Перед тим як навести нове означення звідності розглянемо поняття *оракула*.

Означення 3.23. *Нехай Π_2 — задача пошуку чи розпізнавання. Під зверненням до Π_2-оракула в описі алгоритму для деякої задачі Π_1 розумітимемо процедуру, за якою цей алгоритм утворює деяку індивідуальну задачу з Π_2, яка розв'язується іншим алгоритмом. Час, який витрачає алгоритм на розв'язання Π_2, не враховується в час роботи алгоритму для Π_1. Іншими словами, приймаємо, що алгоритм для Π_2 є «чорною скринею», яка дає негайну відповідь.*

Користуючись програмістськими термінами, можемо інтерпретувати оракул як програму, яка викликається без врахування часу її роботи.

Означення 3.24. *Нехай Π_1 і Π_2 — дві задачі (пошуку чи розпізнавання). Говоритимемо, що Π_1 поліноміально зводиться до Π_2, якщо існує поліноміальний алгоритм, який звертається до Π_2-оракула щонайбільше поліноміальну кількість разів.*

Приклад

Нехай Π_1 і Π_2 будуть відповідно задачами пошуку та розпізнавання факторизації цілих чисел. Ми показали, що Π_1 можна розв'язати за допомогою n звернень до Π_2-оракула, де n — довжина числа N, яке розкладаємо на множники. (Приклад на с. 116 ілюструє цю процедуру для $N = 119$). Отож, задача Π_1 зводиться до Π_2.

3.3.6. Класи $\mathcal{P}$, $\mathcal{NP}$ і $\mathcal{NP}$-повний

Розглянемо неформальні означення трьох фундаментальних понять теорії складності: класу $\mathcal{P}$ задач розпізнавання, які розв'язують за поліноміальний час; класу $\mathcal{NP}$ задач розпізнавання, які розв'язують недетерміністично за поліноміальний час і класу задач типу $\mathcal{NP}$, які є «повними».

Означення 3.25. *Задача розпізнавання* Π *належить класу $\mathcal{P}$ задач, які розв'язують за поліноміальний час, якщо існує поліном $p(n)$ і алгоритм, що дає правильну відповідь за час $\leq p(n)$ до індивідуальної задачі з* Π*, довжина вхідних даних якої $\leq n$.*

Рівносильне означення: задача розпізнавання Π *належить класу $\mathcal{P}$, якщо існують такі константа c і алгоритм, коли індивідуальна задача з* Π *має вхідні дані довжини $\leq n$, то алгоритм дає відповідь на питання задачі за час $O(n^c)$.*

Зауважимо, що між означенням 3.25 і означенням поліноміального алгоритму (див. с.99) існує тісний зв'язок: задача розпізнавання П належить класу $\mathcal{P}$, якщо існує поліноміальний алгоритм, який розв'язує цю задачу.

Виникає запитання, чи насправді означення 3.25 охоплює клас задач, які на практиці можна швидко й ефективно розв'язати. Наприклад, алгоритм з часовою складністю n^{100}, де n – довжина вхідних даних є повільнішим від алгоритму з часовою складністю $e^{0,001 \cdot n}$ доки n є менше десяти мільйонів. Але перший алгоритм поліноміальний, а другий — експоненційний.

Для багатьох задач класу $\mathcal{P}$, які мають практичне значення, існують алгоритми, час виконання яких можна обмежити низьким степенем довжини вхідних даних. Інколи задачі, які належать класу $\mathcal{P}$ або які на перший погляд є класу $\mathcal{P}$, на практиці мають ефективний експоненційний алгоритм. Оскільки експоненційні алгоритми є повільними лише в найгіршому випадку, то може трапитись, що для більшості індивідуальних задач він потребує значно менше часу. Показовим прикладом є симплекс-метод для задачі лінійного програмування, який має експоненційну часову складність, проте добре працює на практиці.

Отже, для задач класу $\mathcal{P}$, які мають практичне значення, існують ефективні алгоритми, хоча в деяких випадках найкращі алгоритми не є поліноміальними або час їх роботи не піддається аналізу.

Означення 3.26. *Задача розпізнавання* П *належить до класу* $\mathcal{NP}$*, якщо для кожної індивідуальної задачі з* П *особа, яка володіє необмеженими обчислювальними можливостями, може не лише дати відповідь на запитання задачі, а й у випадку відповіді «так», може навести доведення, яке будь-хто інший може використати для перевірки відповіді за поліноміальний час. Таке доведення правильності відповіді «так» називатимемо сертифікатом (або поліноміальним сертифікатом).*

Задача розпізнавання П називається задачею класу со-$\mathcal{NP}$, якщо П задовольняє умови аналогічні до наведених вище, в яких лише відповідь «так» замінено на «ні». Це означає, що для кожної індивідуальної задачі з П, відповідь якої — «ні», існує поліноміальний сертифікат, який доводить правильність відповіді «ні».

Приклади

1. Розглянемо задачу розпізнавання факторизації:

ЗАДАНО: натуральні числа N і k.
ЗАПИТАННЯ: чи N має дільник в межах $[2, k]$?

Ця задача не є з класу $\mathcal{P}$. Вона належить до класу $\mathcal{NP}$. Припустимо, що деяка надпотужна особа (наприклад, яка володіє комп'ютером XXII ст. чи представник позаземної цивілізації) розкладає число N на прості множники і знаходить дільник $M \in [2, k]$. Далі повідомляє нам, що відповідь задачі — «так» і пред'являє число M. Володіючи цією інформацією, можемо перевірити правильність відповіді за поліноміальний час, виконавши ділення N на M.
Задача факторизації належить також до класу со-$\mathcal{NP}$. Якщо відповідь задачі «ні», то надпотужна істота надає нам повний розклад N на прості множники, з якого можемо зразу побачити, що немає простого множника $\leq k$. Разом з розкладом має вона нам також надати сертифікат, який дає змогу за поліноміальний час перевірити, що кожний множник насправді просте число.

2. Задача комівояжера також є важкорозв'язною задачею, тобто не класу $\mathcal{P}$, а належить до класу $\mathcal{NP}$. Припустимо, що деяка позаземна істота знаходить найкоротший гамільтоновий цикл і виявляється, що він коротший ніж k. Тоді ця істота повідомляє комівояжерові відповідь на його задачу «так» і показує йому цей шлях. Після цього можна швидко перевірити правильність відповіді.
3. Аналогічно можна легко переконатися, що задача про розмалювання графа трьома кольорами теж є класу $\mathcal{NP}$.
4. До класу $\mathcal{NP}$ належить також задача дискретного логарифма.
   ЗАДАНО: $x,\ g,\ m,\ k\ \in \mathbb{N}$.
   ЗАПИТАННЯ: чи існує $u \in \mathbb{N}$ таке, що $2 \leq u \leq k, \quad x \equiv g^u \mod m$?

Можна навести формальніше означення класу $\mathcal{NP}$. Задача розпізнавання належить до класу $\mathcal{NP}$, якщо існує поліноміальний недетермінований алгоритм, який розв'язує цю задачу. На відміну від детермінованих алгоритмів, *недетермінований алгоритм* має властивість вгадування. Формальним еквівалентом недетермінованого алгоритму є програма для недетермінованої машини Тьюрінга. Від означеної в п. 3.2.4. машини Тьюрінга, недетермінований її аналог відрізняється тим, що, крім модуля керування з головкою зчитування/запису, володіє модулем вгадування з головкою запису на стрічку.

З наведених означень безпосередньо випливає таке: якщо задача класу $\mathcal{P}$, то вона належить також до класу $\mathcal{NP}$, тобто $\mathcal{P} \subset \mathcal{NP}$. Скоріше за все клас $\mathcal{NP}$ значно більший, ніж $\mathcal{P}$, але цього на сьогодні не доведено. Твердження $\mathcal{P} \neq \mathcal{NP}$ найвідоміша гіпотеза в теоретичних основах інформатики. Це одна з центральних задач не лише теорії складності, а й сучасної математики взагалі.

Означення 3.27. *Задача розпізнавання* Π_1 *класу* $\mathcal{NP}$ *називається* $\mathcal{NP}$*-повною задачею, якщо кожну іншу задачу* Π_2 *класу* $\mathcal{NP}$ *можна звести до* Π_1 *за поліноміальний час.*

Інакше кажучи, якби хтось знайшов поліноміальний алгоритм для деякої $\mathcal{NP}$-повної задачі Π_1, то існував би поліноміальний алгоритм для всіх задач Π_2 класу $\mathcal{NP}$. Це б означало, що клас $\mathcal{NP}$ збігається з класом $\mathcal{P}$, тобто гіпотеза $\mathcal{P} \neq \mathcal{NP}$ — хибна. У зв'язку з цим ніхто не сподівається появи поліноміального алгоритму для будь-якої задачі класу $\mathcal{NP}$-повний. У певному сенсі $\mathcal{NP}$-повні задачі є найважчими задачами в класі $\mathcal{NP}$.

Останнє твердження трактувати треба обережно. Не треба виключати, що можна навести ефективний алгоритм (навіть поліноміальний за часом), який розв'язує більшість індивідуальних задач деякої масової задачі Π класу $\mathcal{NP}$-повний. Проте це не суперечить гіпотезі $\mathcal{P} \neq \mathcal{NP}$.

На практиці часто досить важко розв'язати більшість індивідуальних задач заданої $\mathcal{NP}$-повної задачі розпізнавання, бо з ростом довжини вхідних даних час обчислень всіх відомих алгоритмів росте експоненційно.

Приклади

Наведемо список деяких $\mathcal{NP}$-повних задач.

1. Задача комівояжера, яку розглядали у попередньому пункті, є $\mathcal{NP}$-повною задачею.
2. Задача розмалювання графа трьома фарбами.
3. ЗАДАНО: бульову функцію від n змінних записану у кон'юктивній нормальній формі.
   ЗАПИТАННЯ: чи існують значення змінних, при яких бульова функція набуває значення 1?
4. ЗАДАНО: скінченну підмножину $M \subset \mathbb{N}$ і $k \in \mathbb{N}$.
   ЗАПИТАННЯ: чи існує підмножина $M' \subset M$ така, що сума її елементів дорівнює k?
5. ЗАДАНО: два графи $G_1 = (V_1, E_1)$ і $G_2 = (V_2, E_2)$.
   ЗАПИТАННЯ: чи містить граф G_1 підграф ізоморфний до G_2?

3.3.7. Ймовірнісні алгоритми та класи складності

Алгоритми, які розглядали досі, називають *детермінованими*. Більші обчислювальні можливості мають *ймовірнісні алгоритми*. Крім входу w, ймовірнісний алгоритм отримує випадкову двійкову послідовність $r \in \{0,1\}^l$, далі працює як звичайний детермінований алгоритм і результат роботи u подає на вихід. Довжина випадкової послідовності l залежить від довжини входу $|w|$.

Зауважимо, що вихід $u = u(w, r)$ ймовірнісного алгоритму залежить не лише від входу, а й від випадкової послідовності. Випадкова послідовність вважається рівномірно розподіленою на $\{0,1\}^l$, тобто кожне r вибирається з ймовірністю 2^{-l}. Ймовірнісний алгоритм розв'язує масову задачу Π з ймовірністю помилки ε, якщо, отримавши на вхід індивідуальну задачу P, він подає на вихід її правильний розв'язок з ймовірністю не менше $1 - \varepsilon$. Іншими словами, вихід алгоритму є розв'язком індивідуальної задачі P для всіх, крім щонайбільше $\varepsilon 2^l$ випадкових послідовностей r.

Приклад

Розглянемо ймовірнісний алгоритм, який є найкращим способом тестування простоти заданого непарного числа N. Алгоритм дає змогу стверджувати, що (i) число N ймовірно просте, або (ii) N цілком напевно складене.

Спочатку запишемо $N-1$ у вигляді $N - 1 = 2^s t$, де 2^s найвищий степінь двійки, що ділить $N-1$, t — непарне число. Випадково виберемо число a в межах $1 < a < N-1$ і перевіримо, чи НСД$(a, N) = 1$. Далі обчислюємо $(N-1)$ степінь a за модулем N. Обчислення можемо реалізувати за два кроки: (а) знайдемо лишок a^t за бінарним алгоритмом піднесення до степеня (див. п. 3.3.4.), а потім (б) підносимо a^t до квадрата поки не одержимо $a^{2^s t} = a^{N-1}$:

$$a^t \mod N, \quad a^{2t} \mod N, \quad a^{4t} \mod N, \ldots, a^{2^{s-1}t} \mod N, \quad a^{N-1} \mod N. \quad (3.8)$$

Зауважимо таке: коли N — просте число, то

1) $a^{N-1} \equiv 1 \mod N$ (мала теорема Ферма), тобто останнє число в (3.8) дорівнюватиме 1;

2) якщо не всі числа в (3.8) дорівнюють 1, то першій одиниці в цьому списку буде передувати $N-1$ (бо єдиними квадратними коренями з 1 за простим модулем є ± 1).

Коли одночасно виконуються 1 і 2 то говоримо, що N витримує сильний тест Ферма при основі a. (Цей тест ще називають тестом Міллера або ймовірнісним тестом Рабіна.)

Позаяк цей тест використовує піднесення до степеня за бінарним алгоритмом, то його складність має порядок $O(\ln^3 N)$.

Якщо число N витримує сильний тест Ферма, то з ймовірністю $3/4$ можемо бути певними, що число N — просте. Не будемо доводити, що ймовірність помилки цього тесту дорівнює $1/4$. Якщо повторити k разів сильний тест Ферма для випадково вибраних a і N задовольняє умови 1 і 2 для всіх тих a, то можемо стверджувати, що число N просте з ймовірністю $1 - 4^{-k}$.

Наведемо означення класу складності, яке охоплює випадок попереднього прикладу.

Означення 3.28. *Кажуть, що задача розпізнавання* Π *розв'язується за рандомізаційно поліноміальний час (solvable in randomized polynomial*

time) і записують $\Pi \in \mathcal{RP}$*, якщо існує поліноміальний ймовірнісний алгоритм, який розв'язує задачу* Π*, відповідь «так» на виході алгоритму є завжди правильною, а ймовірність того, що відповідь «ні» є правильною — більша за* $1/2$*.*

Приклади

1. З попереднього прикладу випливає, що наступна задача належить до класу $\mathcal{RP}$:
   ЗАДАНО: непарне натуральне число N.
   ЗАПИТАННЯ: чи число N складене?
2. Іншим прикладом задачі з класу $\mathcal{RP}$ є задача визначення нерівнозначності множників для поліномів.
   ЗАДАНО: дві множини поліномів $\{f_1, f_2, \ldots, f_m\}$ і $\{g_1, g_2, \ldots, g_n\}$, де f_i і g_j — поліноми від однієї чи декількох змінних над деяким полем $\mathbb{F}$.
   ЗАПИТАННЯ: чи $f_1 \cdot f_2 \cdot \ldots \cdot f_m$ і $g_1 \cdot g_2 \cdot \ldots \cdot g_n$ є різними поліномами?

Кожний поліном однозначно задається скінченною сумою мономів з ненульовими коефіцієнтами. Якщо б ми спробували розв'язати задачу за допомогою множення двох множин поліномів і порівняння відповідних коефіцієнтів, то загалом затратили би більше ніж поліноміальний час, бо кількість ненульових доданків у поліномі може зростати експоненційно щодо довжини вхідних даних.

Існує простіший метод тестування, чи $\prod_{i=1}^{m} f_i = \prod_{j=1}^{n} g_j$. Припустимо, що f_i та g_j є поліномами від l змінних $x_1, x_2, \ldots, x_l$. Випадково виберемо $c_1, c_2, \ldots, c_l \in \mathbb{F}$ і обчислимо значення кожного з поліномів для $x_k = c_k$, $k = \overline{1, 2, \ldots, l}$, а потім перевіримо чи

$$\prod_{i=1}^{m} f_i(c_1, c_2, \ldots, c_l) = \prod_{j=1}^{n} g_j(c_1, c_2, \ldots, c_l).$$

Якщо ні, то знатимемо, що дві множини поліномів не є рівнозначними і відповіддю задачі розпізнавання майже напевно буде «так». Якщо значення добутків однакові, то ймовірно правильна відповідь — «ні». Очевидно, якщо однакові значення поліномів в одній точці, то ми не можемо бути певними в тому, що вони рівні. Але якщо їхні значення однакові для досить великої кількості випадково вибраних точок, то цілком імовірно, що такі поліноми рівні. Можемо стверджувати, що ймовірність правильності відповіді «так» дорівнює $1 - \varepsilon$, де ε — додатна константа незалежна від вхідних даних задачі. Отже, задача розпізнавання нерівнозначності множин поліномів належить до класу $\mathcal{RP}$.

Зауваження 3.4. *Якщо* $\Pi \in \mathcal{RP}$*, то для довільної константи* $\varepsilon > 0$ *існує алгоритм, який дає відповідь «ні» з імовірністю більшою, ніж* $1 - \varepsilon$*. Очевидно, достатньо вибрати* k *так, щоб* $2^{-k} < \varepsilon$ *і взяти* k *незалежних ітерацій алгоритму з означення 3.28.*

Означення 3.29. *Кажуть, що задача розпізнавання* Π *розв'язується за рандомізаційно поліноміальний час з ймовірністю помилки обмеженою* $1/2$ *і записують* $\Pi \in \mathcal{BPP}$*, якщо існує константа* $\delta > 0$ *і поліноміальний ймовірнісний алгоритм з ймовірністю помилки меншою* $1/2 - \delta$*.*

Тобто, незважаючи на те, яку відповідь дає на виході алгоритм — «так» або «ні», ймовірність її правильності перевищує $1/2 + \delta$.

Зауваження 3.5. *Як і у випадку класу* $\mathcal{RP}$ *(див. попереднє зауваження), якщо* $\Pi \in \mathcal{BPP}$, *то для будь-якої константи* $\varepsilon > 0$ *існує алгоритм, який дає правильну відповідь з ймовірністю більшою ніж* $1 - \varepsilon$.

Достатньо взяти алгоритм складений з k ітерацій алгоритму (існування якого гарантує означення 3.29) і процедури «голосування більшістю»: новий алгоритм дає на виході «так», якщо щонайменше $[k/2]+1$ кроків ітерації дали відповідь «так». Застосовуючи стандартну техніку числення ймовірності, можна показати, що для будь-якої константи $\delta > 0$ існує таке k, що ймовірність правильної відповіді «так» на виході алгоритму «голосування» перевищує $1 - \varepsilon$. Інтуїтивно така ситуація зрозуміла. Наприклад, якщо ми маємо монету, для якої ймовірність випасти «гербом» дорівнює $1/2 + \delta$ і будемо її підкидувати багато разів, то ймовірність того, що «герб» випаде частіше, ніж «копійка», буде більшою ніж $0,999$.

Легко помітити, що клас $\mathcal{BPP}$ охоплює клас $\mathcal{RP}$. $\mathcal{BPP}$ містить також со-$\mathcal{RP}$, тобто клас задач розпізнавання, які задовольняють означення 3.28, в якому відповіді «так» і «ні» міняються місцями.

Приклад

До класу со-$\mathcal{RP}$ очевидно належить така задача.

ЗАДАНО: непарне натуральне число N.

ЗАПИТАННЯ: чи є N простим числом?

Немає галузі математики, якою б вона не була абстрактною, що не може бути застосована до явищ реального світу.

М. Лобачевський

Розділ 4

Графи, коди та шифри

4.1. Графи

Традиційно вважається, що теорія графів виникла у 1736 р., коли Л. Ойлер розв'язав задачу про кенігсбергські мости. Проте теорія графів як важлива частина математики виникла лише у другій половині XIX ст., коли Г. Кіркгоф застосував її для дослідження електричних схем, а А. Келі — до описання будови молекул вуглеводів.

Граф — це математичний об'єкт, за допомогою якого можна інтерпретувати географічні карти, схеми доріг і молекул хімічних речовин, електричні схеми, відносини між людьми та групами людей і ще багато інших різних конкретних ситуацій.

Теорія графів тепер бурхливо розвивається, її результати застосовують, проектуючи різноманітні електронні пристрої, вивчаючи автомати, у програмуванні, фізиці, хімії, біології, економіці, статистиці, соціології та багатьох інших галузях діяльності людини.

4.1.1. Означення графів і приклади графів

Означення 4.1. *Нехай V – довільна множина, E – яка-небудь підмножина множини $V^{(2)}$, де $V^{(2)}$ – множина всіх двоелементних підмножин множини V. Графом G називають пару $G = \{V, E\}$. Множину V називають множиною вершин графа G, а множину E – множиною його ребер.*

Часто поряд з поняттям графа доводиться розглядати загальніше поняття мультиграфа. *Мультиграф* — це пара $G = \{V, E\}$, де E не підмножина, а *набір* елементів множини $V^{(2)}$. Під набором елементів з деякої множини ми розуміємо таку сукупність, в якій деякі елементи можуть траплятися більше ніж один раз. Граф G називають *скінченним*, якщо множина V скінченна. Ми розглядатимемо лише скінченні графи. Графи можна інтерпретувати геометрично, поставивши у відповідність кожному елементові $v \in V$ точку P_v на площині і провівши дугу (або відрізок прямої) P_vP_w для кожної пари $\{v, w\} \in E$.

Зауважимо, що за нашим означенням множина E ребер графа G не містить елементів вигляду $\{v, v\}$, де $v \in V$. Іноді дають трохи інше означення графа, яке допускає ребра вигляду $\{v, v\}$: їх називають *петлями*, а граф без ребер вигляду $\{v, v\}$ тоді називають *граф без петель*. У такій термінології перші чотири графи *а–г* на рис. 4.3 є графами без петель, а граф *д* має дві петлі (ребра $\{1, 1\}$ та $\{3, 3\}$.

Приклади

1. Нехай $V_1 = \{1, 2, 3, 4, 5\}$, $E_1 = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$. Тоді пара $G_1 = \{V_1, E_1\}$ – граф. Якщо $V_1 = \{1, 2, 3, 4\}$, $E_1 = \{\{1, 2\}, \{1, 2\}, \{2, 3\}, \{2, 3\}, \{1, 4\}, \{2, 4\}, \{3, 4\}\}$, то пара $G_2 = \{V_2, E_2\}$ є мультиграфом.

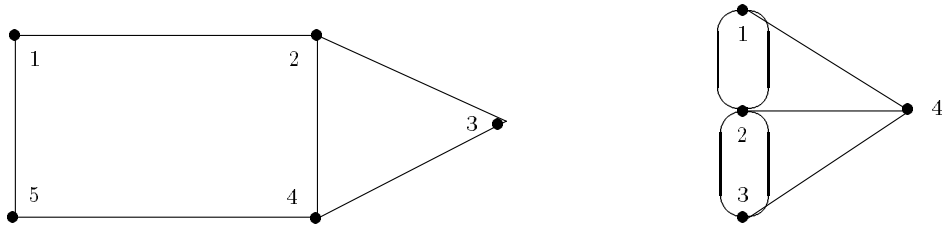


Рис. 4.1.

Рис. 4.1 — це геометрична інтерпретація графів G_1 і G_2. Звернемо увагу на те, що граф G_2 — це граф відповідний до вищезгаданої задачі про кенігсбергські мости. Нагадаємо її суть. Через місто Кенігсберг протікає річка, яка розгалужується на два рукави, крім того, посередині є острів. Цей острів та обидва береги річки сполучені мостами за схемою, яка зображена на рис. 4.2.

Задача про кенігсбергські мости полягає в тому, що треба, вийшовши з деякого місця, пройти по кожному мосту лише один раз і повернутися в початкове місце. Ця задача еквівалентна до наступного питання: чи можна одним розчерком пера намалювати граф G_2 (рис. 4.1), проходячи кожну дугу лише по одному разу. Л.Ойлер показав, що цього зробити не можна і виділив клас графів (які тепер називають *ойлеровими*), геометричну інтерпретацію яких можна намалювати одним розчерком пера, проходячи кожну дугу лише по одному разу. Ойлерові графи ми розглянемо пізніше.

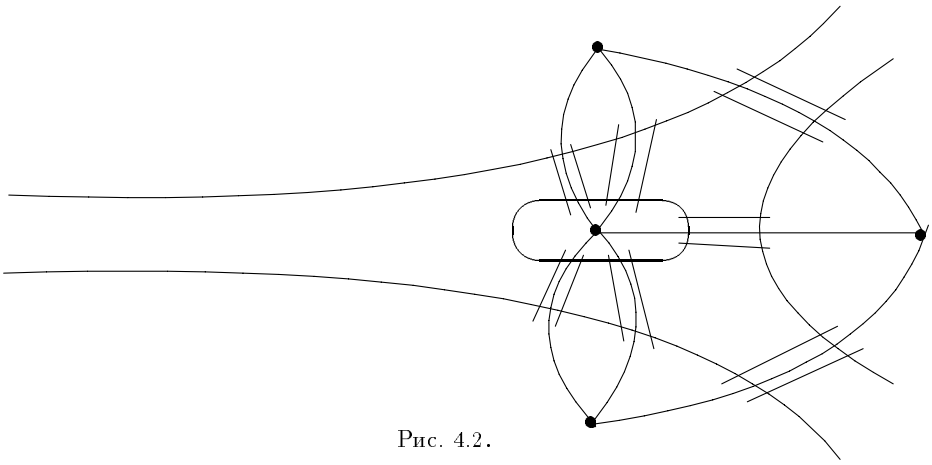


Рис. 4.2.

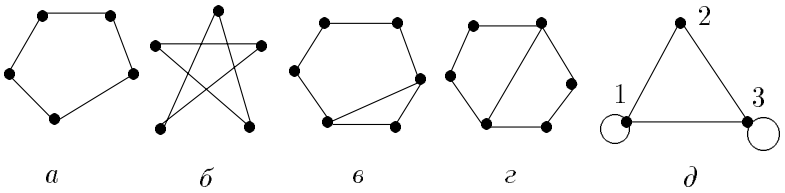
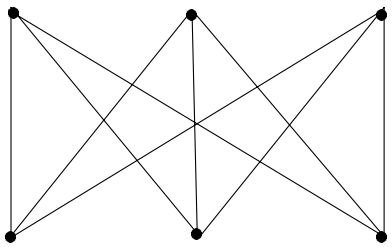


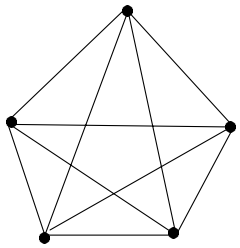
Рис. 4.3.

З ім'ям Ойлера пов'язаний ще один важливий клас графів — *плоскі* або *планарні* графи. Граф G називають плоским, якщо існує таке його зображення на площині, при якому дуги, які відповідають ребрам графа, перетинаються лише у вершинах графа. Зрозуміло, що вивчення плоских графів має важливе практичне значення: досить згадати, що друковані електронні схеми, які широко використовують у різноманітних електронних пристроях, можна інтерпретувати за допомогою плоских графів.

Всі графи, зображені на рис. 4.3, є плоскими, а графи $K_{3,3}$ і K_5 з рис. 4.4 — неплоскі. Ми доведемо це пізніше. З іншого боку, кожен скінченний граф G можна розмістити у тривимірному просторі так, що різні його ребра можуть перетинатися лише у вершинах. Доведення цього факту зовсім просте. Вибираємо яку-небудь пряму l і зображаємо вершини графа G точками цієї прямої. Далі через пряму l проводимо стільки площин, скільки ребер має граф G, кожне ребро розміщуємо у



$K_{3,3}$



K_5

Рис. 4.4.

своїй площині так, як це зображено на рис. 4.5.

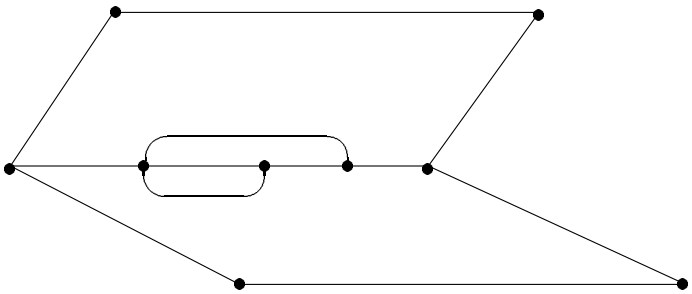


Рис. 4.5.

Приклад

Відома така головоломка пов'язана з графом $K_{3,3}$ на рис. 4.4. Маємо три будинки і три криниці з різною водою. Від кожного будинку до кожної криниці прокладено стежки так, як це зображено на рис. 4.4. Всі власники будинків між собою пересварилися і кожен з них хотів би ходити до криниць довшим шляхом і окружними стежками, аби лише його стежки не перетиналися зі стежками інших. Запитання: чи можна цього домогтися? Пізніше ми переконаємося, що відповідь на це запитання негативна. Ті читачі, які ще не знають розв'язку, можуть спробувати знайти його.

При вивченні графів важливу роль відіграє поняття ізоморфізму графів. У теорії графів ізоморфні графи не розрізняються: кожен з них вважається точною копією іншого.

Означення 4.2. *Нехай $G = \{V, E\}$ і $G' = \{V', E'\}$ — два графи. Графи G і G' називають ізоморфними, якщо існує бієктивне відображення*

$\varphi\colon V \to V'$, яке «зберігає ребра», тобто $\{\varphi(v_i), \varphi(v_j)\} \in E'$ тоді й лише тоді, коли $\{v_i, v_j\} \in E$.

Наприклад, графи *а* і *б*, зображені на рис. 4.3, ізоморфні, а графи *в* і *г* з цього ж рисунка неізоморфні. Сподіваємося, що з рисунка видно чому графи *а* і *б* ізоморфні. Графи *в* і *г* неізоморфні тому, що в першому з них існує замкнений шлях, який складається з трьох ребер, а в другому такого шляху не існує.

Важливу роль у теорії графів відіграє клас орієнтованих графів.

Означення 4.3. *Орієнтованим графом G називають пару $G = \{V, E\}$, де $E \subset V \times V$ — підмножина декартового квадрата множини V. Множину V називають множиною вершин графа G, а множину E — множиною його ребер.*

Звертаємо увагу на те, що орієнтований граф — це, по суті, бінарне відношення на множині його вершин. Отже, теорія орієнтованих графів збігається (з точністю до термінології) з теорією бінарних відношень. Зауважимо про надзвичайно важливу роль бінарних відношень у математиці, адже це і функції та відображення з множини V в себе, відношення еквівалентності та порядку на V та багато інших важливих для математики об'єктів. Це свідчить про особливу роль теорії графів у самій математиці та в її застосуваннях.

Ребрами орієнтованого графа є впорядковані пари $(v_i, v_j) \in V \times V$, тоді як ребрами звичайного графа є пари $\{v_i, v_j\} \in V^{(2)}$: це різні речі. Орієнтований граф називають скінченним, якщо множина його вершин є скінченною. Ми розглядаємо лише скінченні графи та орієнтовані графи, тому далі опускатимемо прикметник «скінченний».

Так само, як і у випадку звичайних графів важливим є поняття ізоморфізму орієнтованих графів.

Означення 4.4. *Нехай $G = \{V, E\}$ і G' — два орієнтованих графи. Графи $G = \{V, E\}$ і $G' = \{V', E'\}$ називають ізоморфними, якщо існує бієктивне відображення $\varphi\colon V \to V'$, яке «зберігає ребра», тобто $(\varphi(v_i), \varphi(v_j)) \in E'$ тоді й лише тоді, коли $(v_i, v_j) \in E$.*

Поряд зі звичайними графами орієнтовані графи теж інтерпретують малюнками на площині, зображаючи вершини у вигляді точок, а ребра у вигляді відрізків прямих або дуг зі стрілками. На рис. 4.6 зображено три орієнтовані графи, кожний з чотирма вершинами. Рис. 4.6,*а* — це

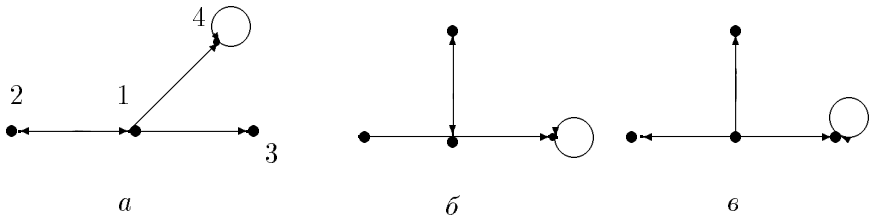


Рис. 4.6.

зображення графа $\{\{1,2,3,4\},\{(1,2),(1,3),(1,4),(2,1),(4,4)\}$. Графи *а* та *б* ізоморфні, а графи *а* і *в* не ізоморфні.

4.1.2. Деякі важливі класи графів

Граф G називають *повним*, якщо кожні дві його вершини з'єднані ребром, тобто множина E ребер графа $G = \{V, E\}$ збігається з множиною всіх двоелементних підмножин множини V. Якщо множина вершин повного графа G складається з n елементів, то його позначають K_n. На рис. 4.7 зображено повні графи з 2,3,4 та 5 вершинами.

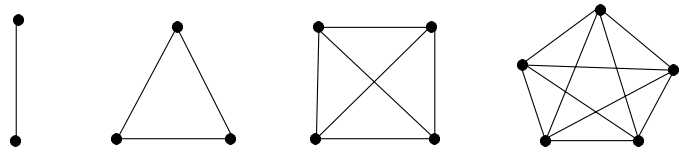


Рис. 4.7.

Кількість 2-елементних підмножин n-елементної множини дорівнює C_n^2, тому повний граф K_n має C_n^2 ребер.

Граф $G = \{V, E\}$ з множиною вершин $\{v_1, \ldots, v_n\}$ називають *циклом* і позначають C_n, якщо множиною його ребер є множина $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$. На рис. 4.8 зображено цикли C_3, C_4 та C_7. Граф $G = \{V, E\}$ з множиною вершин $\{v_1, \ldots, v_n\}$ називають *ланцюгом* і позначають P_n, якщо множиною його ребер є множина $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}\}$. На рис. 4.8 зображено ланцюги P_4 та P_7.

Маршрутом графа називають послідовність ребер $(l_1, \ldots, l_n)$, в якій

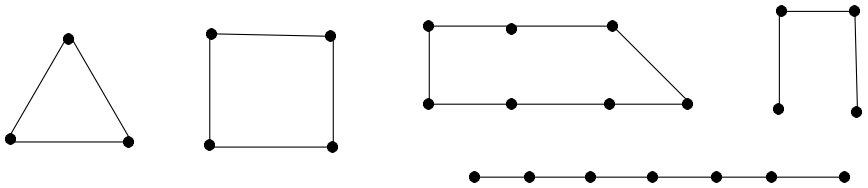
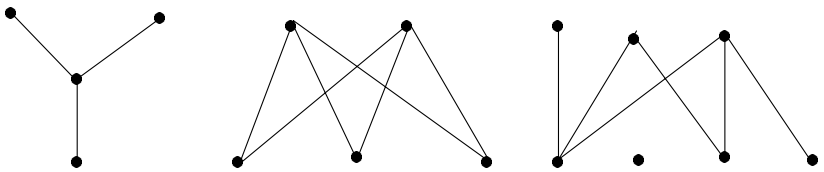


Рис. 4.8.

кожні два послідовних ребра l_i та l_{i+1} є різними і мають (єдину) спільну вершину. Те саме ребро може траплятися в маршруті декілька разів. Вершина ребра l_1, яка не належить до l_2, називається початком маршруту, вершина ребра l_n, яка не належить до l_{n-1}, називається його кінцем. Дві вершини v і w називають *з'єднаними*, якщо існує маршрут з початком у v і з кінцем у w. Якщо вершини v і w з'єднані і $v = w$, то маршрут називають *циклічним*. *Довжина маршруту (циклу, ланцюга)* — це кількість ребер, з яких він складається. *Відстань* між з'єднаними вершинами v і w – це кількість ребер маршруту найменшої довжини, який з'єднує v і w. Очевидно, що відношення «вершини v і w з'єднані» є відношенням еквівалентності на множині вершин графа G. (Ми вважаємо, що за означенням кожна вершина з'єднана сама з собою.) Тому множина вершин графа G розбивається на класи еквівалентності (з'єднаних між собою вершин). Зрозуміло, що вершини різних класів не з'єднані. Тому граф G розбивається на частини $G_1, \ldots, G_k$, які складаються зі з'єднаних вершин і ребер, які їх з'єднують. Ці частини $G_1, \ldots, G_k$ називають *зв'язними компонентами* графа G. Якщо всі вершини графа G з'єднані, то граф називають *зв'язним*. Всі графи, зображені на попередніх рисунках 4.1–4.8, є зв'язними. З іншого боку, рис. 4.8, наприклад, можна розглядати як зображення одного графа, який має 5 зв'язних компонент.

Граф $G = \{V, E\}$ називають *2-графом*, якщо існує таке розбиття $V = V_1 \cap V_2$ на дві підмножини, що кінці кожного ребра належать різним підмножинам. Якщо кожні дві вершини, які належать різним підмножинам, з'єднані ребром, то граф називають *повним 2-графом*. Повний 2-граф, у якому $|V_1| = m, |V_2| = n$ позначають $K_{m,n}$. На рис. 4.9 зображено повні 2-графи $K_{1,3}$ та $K_{2,3}$ (на рис. 4.4 зображено граф $K_{3,3}$) та один неповний 2-граф. Граф $K_{1,n}$ називають *зіркою*.



$K_{1,3}$ $K_{2,3}$

Рис. 4.9.

2-графи трапляються у багатьох розділах математики. Зокрема, кожне відношення між множинами V_1 і V_2 (в тому числі функції та відображення з V_1 у V_2) можна трактувати як 2-граф.

Твердження 4.1. *Граф G є 2-графом тоді й лише тоді, коли всі його цикли мають парну довжину.*

Доведення. Якщо C — цикл 2-графа з підмножинами V_1 і V_2, то C проходить почергово через вершини з V_1 і V_2, тому він мусить мати парну довжину. Навпаки, нехай всі цикли графа G мають парну довжину. Розглянемо будь-яку зв'язну компоненту G_0 графа G і виберемо вершину u графа G_0. Множина вершин V графа G_0 розбивається на дві підмножини: V_1 — множина вершин, відстань від яких до вершини u непарна, V_2 — множина вершин, відстань від яких до вершини u парна. Хочемо довести, що G_0 2-граф з підмножинами вершин V_1 і V_2. Нехай $v, w \in V_2$. Міркуючи від супротивного, припустимо, що існує ребро, яке з'єднує v і w. Нехай $L(u,v)$ та $L(u,w)$ — ланцюги (парної) довжини, що з'єднують, відповідно, u і v та u і w, і нехай u' — перша (рахуючи від u) спільна вершина ланцюгів $L(u,v)$ та $L(u,w)$ (див. рис. 4.10).

Нехай $L(u',v)$ та $L(u',w)$ — підланцюги ланцюгів $L(u,v)$ та $L(u,w)$ з початками у вершині u'. Тоді цикл $L(u',v) \cup \{v,w\} \cup L^{-1}(u',w)$, де $L^{-1}(u',w)$ означає ланцюг $L(u',w)$, пройдений у зворотному порядку, має непарну довжину, що суперечить припущенню. Тому жодні дві вершини множини V_2 не з'єднані ребром. Так само доводиться, що жодні дві вершини множини V_1 не з'єднані ребром. Це означає, що граф G_0 — 2-граф. Оскільки це вірно для кожної зв'язної компоненти графа G, то G — 2-граф.

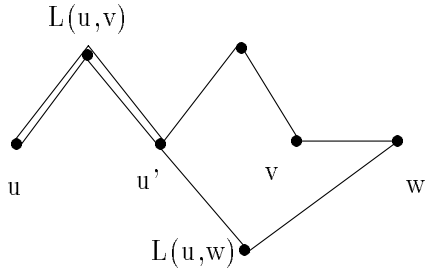


Рис. 4.10.

4.1.3. Лема про рукостискання

Нехай $G = \{V, E\}$ — граф. Якщо $l \in E$ має вигляд $l = \{u, v\}$, то ребро l називають *інцидентним* до вершини u (і v), а кожну з вершин u і v називають *інцидентною* до ребра l. *Степенем вершини* графа називають кількість інцидентних їй ребер. Вершини u і v, які належать деякому ребру, називають *суміжними*.

Лема 4.1. *Сума степенів вершин скінченного графа є парним числом.*

Доведення. Кожне ребро з'єднує дві вершини і тому вносить доданок 2 у суму степенів вершин. Отже, сума степенів вершин дорівнює подвоєній кількості ребер.

Наслідок 4.1. *Кількість вершин непарного степеня довільного скінченного графа є парним числом.*

Доведення. Розіб'ємо суму степенів вершин графа на два доданки $A_1 + A_2$, де A_2 сума степенів вершин парного степеня, а A_1 сума степенів вершин непарного степеня. Число A_2, очевидно, парне. Якби кількість вершин непарного степеня була непарною, то число A_1, а отже, і сума $A_1 + A_2$ була непарною, а це суперечить лемі.

Назву, формулювання та доведення останньої леми можна проінтерпретувати так. Нехай на вечірці зібралася певна кількість людей, серед яких є знайомі та незнайомі між собою люди. Всі знайомі потиснули один одному руки. Тоді кількість потиснутих рук є парним числом, бо при кожному рукостисканні беруть участь дві руки.

4.1.4. Матриці, зв'язані з графами

Нехай $G = \{V, E\}$ — скінченний граф, $V = \{v_1, \ldots, v_n\}$ — множина його вершин. Ми вважаємо, що множина $\{v_1, \ldots, v_n\}$ впорядкована індексами $1, \ldots, n$.

Означення 4.5. *Матрицею суміжності графа G називається квадратна матриця n-го порядку $A = [\alpha_{ij}]$, для якої*

$$\alpha_{ij} = \begin{cases} 1, & \text{якщо ребра } v_i, v_j \text{ суміжні,} \\ 0, & \text{в іншому випадку.} \end{cases}$$

Матрицею суміжності орієнтованого графа G називається квадратна матриця n-го порядку $A = [\alpha_{ij}]$, для якої

$$\alpha_{ij} = \begin{cases} 1, & \text{якщо ребра } v_i, v_j \text{ суміжні,} \\ 0, & \text{в іншому випадку.} \end{cases}$$

Дві вершини u і v орієнтованого графа $G = \{V, E\}$ суміжні, якщо існує ребро $(u, v) \in E \subset V \times V$. Зауважимо, що для звичайних (неорієнтованих) графів відношення суміжності на множині його вершин є симетричним, а для орієнтованих графів це не так. Тому матриця суміжності неорієнтованого графа симетрична, а матриця суміжності орієнтованого графа здебільшого несиметрична.

Тепер припустимо, що задано скінченний граф $G = \{V, E\}$, в якому множина вершин $V = \{v_1, \ldots, v_m\}$ і множина ребер $E = \{l_1, \ldots, l_n\}$ впорядковані індексами $i = 1, \ldots, m$ та $j = 1, \ldots, n$ відповідно.

Означення 4.6. *Матрицею інцидентності орієнтованого графа $G = \{V, E\}$ називається $m \times n$-матриця $A = [\alpha_{ij}]$, для якої*

$$\alpha_{ij} = \begin{cases} 1, & \text{якщо вершина } v_i \text{ є початком ребра } l_j, \\ -1, & \text{якщо вершина } v_i \text{ є кінцем ребра } l_j, \\ 0, & \text{в інших випадках.} \end{cases}$$

Матрицею інцидентності (неорієнтованого) графа $G = \{V, E\}$ називається $m \times n$-матриця $A = [\alpha_{ij}]$, для якої

$$\alpha_{ij} = \begin{cases} 1, & \text{якщо вершина } v_i \text{ належить ребру } l_j, \\ 0, & \text{в іншому випадку.} \end{cases}$$

Крім матриць суміжності та інцидентності, важливе значення у теорії графів мають матриці Кіркгофа.

Означення 4.7. *Матрицею Кіркгофа (неорієнтованого) графа* $G = \{V, E\}$ *називається* $m \times n$*-матриця* $A = [\alpha_{ij}]$, *для якої*

$$\alpha_{ij} = \begin{cases} -1, & \text{якщо вершини } v_i \text{ та } v_j \text{ суміжні,} \\ 0, & \text{якщо вершини } v_i \text{ та } v_j \text{ не суміжні,} \\ \deg v_i, & \text{якщо } i = j, \end{cases}$$

де $\deg v_i$ — степінь вершини v_i.

Зауважимо, що матриця Кіркгофа є симетричною матрицею. Крім того, сума елементів кожного її стовпчика та кожного її рядка дорівнює нулю.

Приклади

1. Нехай $G_1 = \{\{1,2,3,4\}, \{\{1,2\},\{1,3\},\{2,3\},\{2,4\}\}\}$ — (неорієнтований) граф. Матриця
$$A_1 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$
— це матриця суміжності графа G_1.
2. Нехай тепер $G_2 = \{\{1,2,3,4\}, \{(1,3),(3,1),(2,1),(2,3),(4,2),(4,4)\}\}$ — орієнтований граф. Його матрицею суміжності є така матриця:
$$A_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$
3. Тепер наведемо приклад матриці інцидентності орієнтованого графа. Розглянемо граф $G_3 = \{\{1,2,3,4\}, \{(1,3)_1,(2,1)_2,(2,3)_3,(4,2)_4,(4,4)_5\}\}$. Його матрицею інцидентності є така матриця:
$$A_3 = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 \\ -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$
4. Матриця
$$A_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

є матрицею інцидентності (неорієнтованого) графа $G_4 = \{\{1,2,3,4\},\{\{1,2\}_1, \{2,4\}_2, \{2,3\}_3, \{3,4\}_4, \{1,4\}_5\}\}$.

5. Розглянемо граф $G_5 = \{\{1,2,3,4\},\{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\}\}\}$. Матриця Кіркгофа цього графа має такий вигляд:

$$A_5 = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}.$$

Пропонуємо геометрично зобразити графи $G_1 - G_5$.

Твердження 4.2. *Матриці суміжності та Кіркгофа ізоморфних графів подібні.*

Доведення. Нехай A і B — матриці суміжності ізоморфних графів G_1 і G_2. Нам треба довести існування невиродженої матриці C, для якої $B = C^{-1}AC$. Нехай $\{v_1, \ldots, v_n\}$ — множина вершин графа G_1 і $\varphi(v_k) = v'_{i_k}$, де φ — ізоморфізм графів G_1 і G_2. Зрозуміло, що ізоморфізм φ цілком визначається підстановкою

$$\varphi = \begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}$$

множини $\{1, 2, \ldots, n\}$. Розкладемо підстановку φ в добуток транспозицій $\varphi = \pi_k \cdots \pi_1$. Нехай $\pi_s = (i_s, j_s), s = 1, \ldots, k$. Доведемо наше твердження у випадку, коли $\varphi = \pi_1$. У цьому випадку зрозуміло, що матрицю B одержуємо з матриці A за допомогою перестановки місцями i_1-го і j_1-го рядків та i_1-го і j_1-го стовпчиків матриці A. Таку перестановку рядків і стовпчиків матриці A одержуємо домноженням матриці A ліворуч та праворуч на матрицю $\Pi_{i_1 j_1}$, одержану з одиничної матриці перестановкою місцями її i_1-го і j_1-го рядків (відповідні стовпчики при цьому переставляються автоматично).

Отже, маємо $B = \Pi_{i_1 j_1} A \Pi_{i_1 j_1}$. Але $\Pi^2_{i_1 j_1} = E$ — одинична матриця, тобто $\Pi_{i_1 j_1} = \Pi^{-1}_{i_1 j_1}$. Тому в цьому випадку $B = C^{-1}AC$, де $C = \Pi_{i_1 j_1}$.

Загальний випадок довільної підстановки φ зводиться до розглянутого випадку, використовуючи розклад φ в добуток транспозицій. Тут маємо

$$B = (\Pi_{i_k j_k} \cdots \Pi_{i_1 j_1}) A (\Pi_{i_1 j_1} \cdots \Pi_{i_k j_k}) = C^{-1}AC,$$

де $C = \Pi_{i_1 j_1} \cdots \Pi_{i_k j_k}$.

Означення 4.8. *Характеристичним поліномом графа G називається характеристичний поліном його матриці суміжності. Спектром графа G називається множина коренів його характеристичного полінома.*

З попереднього твердження випливає, що характеристичні поліноми та спектри ізоморфних графів збігаються. Зауважимо, що графи можуть мати однакові спектри і не бути ізоморфними. Для того щоб переконатися у цьому, пропонуємо знайти спектри таких (неізоморфних) графів: $G_1 = \{\{1,2,3,4,5\},\{\{2,3\},\{3,4\},\{4,5\},\{2,5\}\}\}$, $G_2 = \{\{1,2,3,4,5\},\{\{1,2\},\{1,3\},\{1,4\},\{1,5\}\}\}$.

Далі нам буде потрібний той факт, що всі алгебричні доповнення матриці Кіркгофа рівні між собою.

Теорема 4.1. *Нехай $A = [a_{ij}]$ – квадратна матриця n-го порядку з елементами з деякого поля P. Припустимо, що*

$$\sum_{k=1}^{n} a_{ik} = \sum_{k=1}^{n} a_{kj} = 0 \quad \forall i,j,\ 1 \le i,j \le n. \tag{4.1}$$

Тоді всі алгебричні доповнення матриці A рівні між собою.

Доведення. Нехай r — ранг матриці A. З умов (4.1) випливає, що рядки (і стовпчики) матриці A лінійно залежні. Тому $r \le n-1$. Якщо $r < n-1$, то всі алгебричні доповнення матриці A дорівнюють нулю, отже рівні між собою.

Тому достатньо розглянути випадок, коли $r = n-1$. Нехай A_{ij} — алгебричні доповнення матриці A і $\tilde{A} = [A_{ij}]^t$ — приєднана матриця. Позаяк рядки матриці A лінійно залежні, то її визначник $\det A$ дорівнює нулю і $A \cdot \tilde{A} = \det A \cdot E = 0$ — нульова матриця. Тому маємо рівності

$$\sum_{k=1}^{n} a_{ik} A_{jk} = 0 \quad 1 \le i,j \le n. \tag{4.2}$$

Зафіксувавши j, можна розглядати (4.2) як систему лінійних однорідних рівнянь з матрицею A щодо невідомих $A_{j1},\dots,A_{jn}$. Оскільки ранг матриці A дорівнює $n-1$, то за теоремою про фундаментальну систему розв'язків ця система має один фундаментальний розв'язок. Умови (4.1) засвідчують, що цим розв'язком є $(1,1,\dots,1)$. Тому $A_{j1} = \dots = A_{jn}$,

тобто всі алгебричні доповнення з довільного j-го рядка матриці $\tilde{A}$ рівні між собою. Розглянувши ще й добуток $\tilde{A} \cdot A = \det A \cdot E = 0$, який теж є нульовою матрицею, аналогічно одержимо систему рівностей

$$\sum_{k=1}^{n} a_{kj} A_{ki} = 0 \quad 1 \leq i, j \leq n. \tag{4.3}$$

Тепер зафіксувавши i, розглянемо (4.3) як систему лінійних однорідних рівнянь з матрицею A^t щодо невідомих $A_{1i}, \ldots, A_{ni}$. Як і раніше, з умов (4.1) випливає, що $A_{1i} = \ldots = A_{ni}$, тобто всі алгебричні доповнення з довільного i-го рядка матриці $\tilde{A}$ рівні між собою.

Остаточно маємо $A_{st} = A_{it} = A_{ij}$ для довільних i, j, s, t, $1 \leq i, j, s, t \leq n$.

Наслідок 4.2. *Всі алгебричні доповнення матриці Кіркгофа скінченного графа рівні між собою.*

4.1.5. Регулярні графи

Означення 4.9. *Скінченний зв'язний граф називають регулярним, якщо степені всіх його вершин однакові. Степенем регулярного графа називають спільний степінь його вершин. Кількість вершин регулярного графа називають його порядком.*

На рис. 4.11 зображено три регулярні графи, відповідні таким правильним многогранникам: тетраедру, кубу та октаедру.

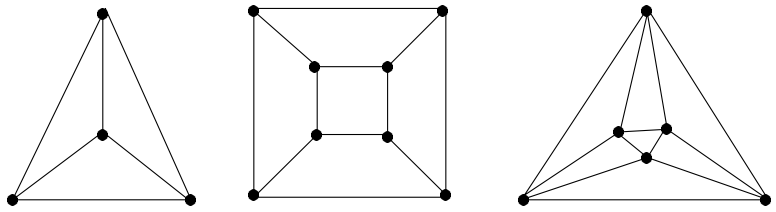


Рис. 4.11.

Теорема 4.2. *Для кожного натурального числа $n \geq 1$ і для кожного натурального числа $d \geq n - 1$ існує регулярний граф степеня d і порядку n.*

Доведення. Достатньо довести існування регулярних графів $G = \{V, E\}$ для яких $d \leq \frac{n-1}{2}$. Звідси випливатиме, що теорема правильна і для $d > \frac{n-1}{2}$. Справді, тоді можна розглянути граф $G' = \{V, E(K_n) \setminus E\}$, який має ті самі вершини, що й граф G, множина ребер якого є доповненням множини ребер графа G до множини ребер повного графа K_n. Граф G' буде регулярним степеня $d' = n - 1 - d > n - 1 - \frac{n-1}{2} > \frac{n-1}{2}$.

Побудуємо регулярний граф G порядку n і степеня $d \leq \frac{n-1}{2}$. Для цього розглянемо адитивну групу класів лишків $\mathbb{Z}_n$, елементи якої вважатимемо множиною вершин графа G.

Зауважимо, що для кожного регулярного графа степеня d і порядку n хоч одне з чисел d або n обов'язково буде парним. Справді, за лемою про рукостискання добуток nd є парним числом, тому числа n і d не можуть бути обидва непарними.

Якщо $d = 2k$ – парне число, то розглянемо підмножину $A \subset \mathbb{Z}_n$, $A = \{\pm\bar{1}, \pm\bar{2}, \ldots \pm\bar{k}\}$ і назвемо дві вершини $x, y \in \mathbb{Z}_n$ суміжними, якщо $x - y \in A$, тобто $x \in y + A$.

Якщо $d = 2k+1$ — непарне число, то число n парне і можна розглянути підмножину $A \subset \mathbb{Z}_n$, $A = \{\pm\bar{1}, \pm\bar{2}, \ldots \pm\bar{k}, \pm\frac{\bar{n}}{2} = \frac{\bar{n}}{2}\}$ і так само назвати дві вершини $x, y \in \mathbb{Z}_n$ суміжними, якщо $x - y \in A$, тобто $x \in y + A$.

В обох випадках одержуємо регулярний граф степеня d і порядку n.

У наступній теоремі відображено деякі цікаві властивості власних значень регулярних графів.

Теорема 4.3. *Нехай G – регулярний граф степеня d. Тоді:*

1) *d — власне значення графа G;*
2) *якщо G — зв'язний граф, то кратність власного значення d дорівнює 1;*
3) *$d \geq |\lambda|$ для кожного власного значення λ графа G.*

Доведення. 1. Нагадаємо, що за означенням для елементів α_{ij} матриці суміжності A графа G маємо: $\alpha_{ij} = 1$, якщо i-а та j-а вершини суміжні, і $\alpha_{ij} = 0$ в іншому випадку. Звідси бачимо, що кожний рядок матриці суміжності A регулярного графа степеня d містить d одиниць, решта його елементів є нулями. Тому для $\overrightarrow{a} = (1, 1, \ldots, 1)$ маємо $\overrightarrow{a}A = d\overrightarrow{a}$, тобто, d — власне значення графа G, відповідне власному вектору $\overrightarrow{a}$.

2. Достатньо довести, що всі координати будь-якого власного вектора $\overrightarrow{b}$, відповідного власному значенню d, рівні між собою. Звідси й

випливатиме твердження 2. Справді, з лінійної алгебри відомо, що для симетричної матриці з елементами з поля $\mathbb{R}$ існує база в $\mathbb{R}^n$, складена з її власних векторів. Тому, якби кратність власного значення d була більшою від 1, то існував би відповідний власний вектор не пропорційний вектору $\overrightarrow{a}$.

Отже, нехай $\overrightarrow{b} = (b_1, \ldots, b_n)$, $\overrightarrow{b}A = d\overrightarrow{b}$ і нехай b_j — координата вектора $\overrightarrow{b}$ з максимальним модулем, тобто $|b_j| \geq |b_i|$ для всіх $i, 1 \leq i \leq n$. Позначаючи через $(\overrightarrow{b}A)_j$ j-у координату вектора $\overrightarrow{b}A$, одержуємо

$$(\overrightarrow{b}A)_j = db_j. \tag{4.4}$$

Нехай N_j означає оточення j-ої вершини графа G, тобто множину всіх вершин, які з'єднані ребрами з j-ою вершиною. Тоді рівність (4.4) можна записати у вигляді

$$\sum_{i \in N_j} b_i = db_j, \tag{4.5}$$

звідки випливає нерівність

$$d|b_j| \leq \sum_{i \in N_j} |b_i|. \tag{4.6}$$

Оскільки права частина нерівності (4.6) має d доданків, то ця нерівність можлива лише тоді, коли $|b_i| = |b_j|$ для всіх $i \in N_j$. Але тоді з (4.5) випливає, що $b_i = b_j$ для всіх $i \in N_j$. Використовуючи зв'язність графа G, бачимо, що $b_i = b_j$ для всіх $i, 1 \leq i \leq n$, і це завершує доведення твердження 2.

3. Нехай λ — власне значення графа G, $\overrightarrow{x} = (x_1, \ldots, x_n)$ — відповідний власний вектор. Нехай x_j — координата вектора $\overrightarrow{x}$ з найбільшим модулем. Позначаючи через $(\overrightarrow{x}A)_j$ j-у координату вектора $\overrightarrow{x}A$, одержуємо $(\overrightarrow{x}A)_j = \lambda x_j$, де N_j — оточення j-ої вершини графа G. Це означає, що $\sum_{i \in N_j} x_i = \lambda x_j$. Звідси за нерівністю трикутника одержуємо $|\lambda||x_j| \leq \sum_{i \in N_j} |x_i| \leq d|x_j|$, що й дає потрібну нерівність $|\lambda| \leq d$.

4.1.6. Дерева

Важливий клас графів становлять дерева.

Означення 4.10. *Скінченний граф, який не містить циклів, називають лісом. Зв'язний ліс називають деревом.*

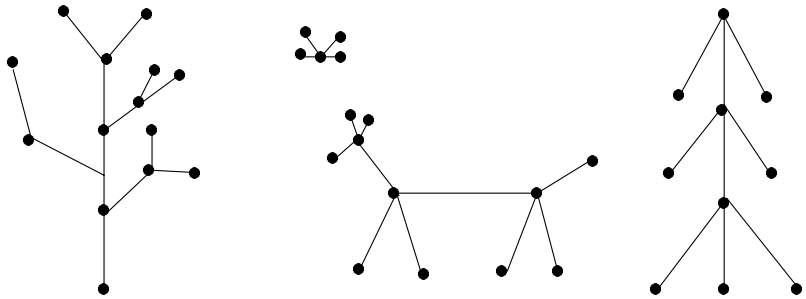


Рис. 4.12.

На рис. 4.12 зображено ліс, який складається з чотирьох дерев.

Лема 4.2. *Нехай $G = \{V, E\}$ — зв'язний граф, $l \in E$. Тоді:*

1) *якщо ребро l належить деякому циклу C графа G, то граф $G - l = \{V, E \setminus \{l\}\}$ зв'язний;*
2) *якщо ребро l не належить жодному циклу графа G, то граф $G - l$ має дві зв'язні компоненти.*

Доведення. 1. Нехай u і v — дві вершини графа $G - l$. Потрібно довести, що існує маршрут, який з'єднує u і v. Такий маршрут L існує у графі G, бо G зв'язний. Якщо ребро l не міститься в L, то воно з'єднує вершини u і v і в графі $G - l$. Якщо ж $L = \{l_1, \ldots, l_k, l, l_{k+2}, \ldots, l_s\}$, то маршрут $L' = \{l_1, \ldots, l_k\} \cup (C - l) \cup \{l_{k+2}, \ldots, l_s\}$ з'єднує u і v у графі $G - l$, де $C - l$ — маршрут, одержаний з циклу C вилученням ребра l.

2. Нехай $l = \{a, b\}$. Передусім граф $G - l$ незв'язний: якби вершини a і b тут були з'єднані за допомогою деякого ланцюга L, то, долучивши до L ребро l, ми одержали б цикл графа G, який містить l. Отож, граф $G - l$ має принаймні дві зв'язні компоненти G_a і G_b, де G_a і G_b — підграфи графа G з множинами вершин, з'єднаних з a та b відповідно. Залишається перевірити, що інших зв'язних компонент немає. Нехай вершина c графа $G - l$ не з'єднана з вершиною a. Доведемо, що c з'єднана з вершиною b. У графі G існує маршрут L з c в a і ребро l повинне входити в цей маршрут: $L = \{l_1, \ldots, l_k, l, \ldots, l_s\}$. За означенням маршруту, ребро l_k має з $l = \{a, b\}$ спільну вершину, якою не може

бути a, тому це b. Отже, кожна вершина графа $G - l$ з'єднана або з a або з b. Лема доведена.

Наступна лема засвідчує, що зв'язний граф не може мати занадто мало ребер.

Лема 4.3. *Нехай $G = \{V, E\}$ — зв'язний граф, $|V| = n > 0, |E| = m$. Тоді $m \geq n - 1$.*

Доведення. Використаємо індукцію за m. Якщо $m = 0$, то зі зв'язності графа G випливає, що $n = 1$ і нерівність $m \geq n - 1$ правильна. Припустимо, що лема доведена для зв'язних графів з меншою, ніж m кількістю ребер. Нехай $l \in E$ — яке-небудь ребро графа G. З попередньої леми випливає, що граф $G - l$ або залишається зв'язним або розбивається на дві зв'язні компоненти G_1, G_2. У першому випадку за припущенням індукції маємо $m - 1 \geq n - 1$, тим більше $m \geq n - 1$. Розглянемо другий випадок. Нехай графи G_1 і G_2 мають відповідно по m_1 і m_2 ребер та n_1 і n_2 вершин. Тоді $m_1 + m_2 = m - 1$, $n_1 + n_2 = n$, і, за припущенням індукції $m_1 \geq n_1 - 1, m_2 \geq n_2 - 1$, тому $m_1 + m_2 \geq n - 2$, тобто $m - 1 \geq n - 2$ і $m \geq n - 1$, що й потрібно довести.

Теорема 4.4. *Нехай $G = \{V, E\}$ — граф, $|V| = n, |E| = m$. Такі умови еквівалентні:*

1) G — дерево;
2) G — зв'язний граф і $m = n - 1$;
3) G — граф без циклів і $m = n - 1$.

Доведення. $1 \Rightarrow 2$. Використаємо індукцію за n. Для $n = 1$ твердження тривіальне. Нехай $n > 1$. Припустимо, що для графів з меншою кількістю вершин наше твердження доведене. Розглянемо граф $G - l$, де $l \in E$ – яке-небудь ребро графа G. За лемою 4.2 граф $G - l$ розбивається на дві зв'язні компоненти G_1 і G_2. Нехай графи G_1 і G_2 мають, відповідно по m_1 і m_2 ребер та n_1 і n_2 вершин. За припущенням індукції $m_1 = n_1 - 1, m_2 = n_2 - 1$, тому $m_1 + m_2 = n - 2$, але $m_1 + m_2 = m - 1$, тому $m - 1 = n - 2$, отже $m = n - 1$.

$2 \Rightarrow 3$. Припустимо (від супротивного), що граф G містить цикл. Нехай l — ребро цього циклу. Тоді за лемою 4.2 граф $G - l$ зв'язний і має $n - 2$ ребра, а це суперечить лемі 4.3.

$3 \Rightarrow 1$. Достатньо довести, що граф G зв'язний. Міркуємо індукцією за n. Для $n = 1$ доводити не треба. Припустимо, що всі графи з меншою

кількістю вершин, які задовольняють умову 3, зв'язні. Якщо граф G незв'язний, то він розбивається на $k > 1$ зв'язних компонент $G_1, \ldots, G_k$. Нехай m_i та n_i — кількість ребер і вершин графа $G_i, 1 \leq i \leq k$. Кожен граф G_i є деревом, тому за доведеним $m_i = n_i - 1, 1 \leq i \leq k$. Додавши почленно ці рівності, одержимо $\sum_{i=1}^{k} m_i = \sum_{i=1}^{k} n_i - k$, тобто $m = n - k$. За припущенням $m = n - 1$, тому $k = 1$ і граф G зв'язний. Теорема доведена.

Вершину v називають *кінцевою*, якщо її степінь дорівнює 1. Виявляється, що кожне дерево завжди має кінцеві вершини.

Наслідок 4.3. *Кожне дерево D порядку $n \geq 2$ має не менш ніж 2 кінцеві вершини.*

Доведення. Нехай $d_1, d_2, \ldots, d_n$ — послідовність степенів вершин дерева D. За лемою про рукостискання маємо $\sum_{i=1}^{n} d_i = 2n - 2$, тому принаймні два числа серед d_i повинні дорівнювати 1.

Важливу роль у вивченні довільних графів відіграють каркасні дерева. Якщо $G = \{V, E\}$ — граф, то граф $G' = \{V, E'\}$, де $E' \subset E$ називають *каркасом* графа G. Каркас, який є деревом, називають *каркасним деревом*. Важливою задачею є відшукання каркасних дерев зв'язних графів. Їх кількість можна обчислювати за допомогою такої теореми.

Теорема 4.5. *(Кіркгоф) Кількість каркасних дерев зв'язного графа G порядку $n \geq 2$ дорівнює алгебричному доповненню довільного елемента його матриці Кіркгофа.*

Для доведення теореми Кіркгофа нам буде потрібно декілька лем. У перших двох з них використовують поняття орієнтації (неорієнтованого) графа. *Орієнтацією* графа $G = \{V, E\}$ називають орієнтований граф $G' = \{V, E'\}$, де $E' \subset V \times V$ і кожне ребро $l' \in E'$ має вигляд $l' = (a, b)$, де $\{a, b\}$ — деяке ребро графа G.

Геометрично це означає, що орієнтації одержують заміною дуг у зображенні графа на стрілки.

Лема 4.4. *Нехай зафіксований перелік вершин та ребер графа G і нехай A та I, відповідно, — матриця Кіркгофа графа G та матриця інцидентності якої-небудь його орієнтації. Тоді $A = I \cdot I^t$, де I^t — транспонована з I матриця.*

Доведення. Нехай m — кількість вершин графа G, n — кількість його ребер. Матриця $I = [\beta_{ij}]$ — це $m \times n$-матриця з елементами

$$\beta_{ij} = \begin{cases} 1, & \text{якщо вершина } v_i \text{є початком ребра } l_j; \\ -1, & \text{якщо вершина } v_i \text{є кінцем ребра } l_j; \\ 0, & \text{в інших випадках.} \end{cases}$$

Пара різних вершин v_i та v_j графа G може належати лише одному (наприклад k-ому) ребру, і в цьому випадку $\beta_{ik} = -\beta_{jk}$. Якщо v_i або v_j не належать k-ому ребру, то $\beta_{ik} \cdot \beta_{jk} = 0$. Звідси одержуємо, що при $i \neq j$

$$\sum_{k=1}^{n} \beta_{ik}\beta_{jk} = \begin{cases} -1, & \text{якщо вершини } v_i \text{ та } v_j \text{ належать одному з ребер;} \\ 0, & \text{в іншому випадку.} \end{cases}$$

З іншого боку, $\sum_{k=1}^{n} \beta_{ik}^2$ дорівнює кількості ребер, яким належить i-а вершина, тобто це $\deg v_i$ — степінь i-ої вершини. Отже, для елементів α_{ij} матриці $I \cdot I^t$ одержуємо

$$\alpha_{ij} = \sum_{k=1}^{n} \beta_{ik}\beta_{jk} = \begin{cases} -1, & \text{якщо вершини } v_i \text{ та } v_j \text{ суміжні;} \\ 0, & \text{якщо вершини } v_i \text{ та } v_j \text{ не суміжні;} \\ \deg v_i, & \text{якщо } i = j. \end{cases}$$

Це й означає, що матриця $A = I \cdot I^t$ є матрицею Кіркгофа графа і завершує доведення леми.

Лема 4.5. *Нехай $G = \{V, E\}$ — граф, $|V| = n$, $|E| = m$, $n = m + 1$. Нехай I — матриця інцидентності якої-небудь його орієнтації і M — довільний мінор m-го порядку матриці I. Тоді:*

1) якщо G — дерево, то $M = \pm 1$;

2) якщо G — не дерево, то $M = 0$.

Доведення. Почнемо з зауваження про те, що після перестановки декількох рядків та декількох стовпчиків визначник квадратної матриці не змінюється або міняє знак на протилежний. Це означає, що в доведенні леми ми можемо перенумеровувати рядки і стовпчики мінора M (це якраз *всі* стовпчики матриці I) так, як нам зручно.

1. Нехай граф G є деревом, a — його вершина, відповідна рядку матриці I, що не входить у мінор M. Присвоїмо цій вершині номер $m+1$. За наслідком 4.3 дерево має не менше ніж дві кінцеві вершини. Виберемо

серед них одну, іншу ніж a (якщо a кінцева), і позначимо її v_1, а єдине ребро, інцидентне вершині v_1, позначимо l_1. Тоді розглянемо дерево D_1, яке одержується з D вилученням вершини v_1 та ребра l_1, і так само знайдемо у D_1 кінцеву вершину v_2 та ребро l_2. Повторюватимемо цей процес доки не вичерпаються всі вершини та ребра графа G, одержимо перелік $v_1, v_2, \ldots, v_m, v_{m+1}$, $l_1, l_2, \ldots, l_m$ вершин і ребер графа G. При такому переліку матриця інцидентності графа G має вигляд

$$\begin{pmatrix} \pm 1 & 0 & 0 & \cdots & 0 & 0 \\ * & \pm 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ * & * & * & \cdots & * & \pm 1 \\ * & * & * & \cdots & * & * \end{pmatrix},$$

$*$ позначає елементи, які не впливають на наше міркування. Мінор M, який тепер проходить через перші m рядків цієї $(m+1) \times m$-матриці, дорівнює ± 1.

2. Нехай граф G не є деревом. Тоді за теоремою 4.4 граф G незв'язний. Нехай G_1 — одна з його зв'язних компонент, що не містить вершини a (вершини, для якої відповідний рядок не входить у мінор M). Розглянемо такий перелік вершин і ребер графа G: спочатку нумеруємо довільним способом вершини $v_1, \ldots, v_k$ та ребра $l_1, \ldots, l_s$ графа G_1, а тоді довільним способом продовжуємо цей перелік $v_{k+1}, \ldots, v_{m+1}$, $l_{s+1}, \ldots, l_m$ доки не вичерпаємо всі вершини та ребра графа G. При такому впорядкуванні вершин і ребер матриця інцидентності I графа G набуває блочно-діагонального вигляду

$$I = \begin{pmatrix} I_1 & 0 \\ 0 & I_2 \end{pmatrix},$$

де I_1 — матриця інцидентності графа G_1.

Сума стовпчиків матриці I_1 дорівнює нулю, бо за її означенням у кожному рядку один раз трапляються $+1$ і -1, а всі інші елементи дорівнюють нулю. Тому й сума перших s стовпчиків матриці I (і мінора M) дорівнює нулю. Звідси випливає, що $M = 0$, бо його стовпчики лінійно залежні. Лема доведена.

Наступна лема — *теорема Біне-Коші* про визначник добутку двох прямокутних матриць. Нехай A — $n \times m$-матриця, B — $m \times n$-матриця. $C = AB$ — квадратна матриця n-го порядку і можна говорити про її

визначник. Нехай $n \le m$, нехай мінор M n-го порядку матриці A розміщений у стовпчиках з номерами $j_1, \dots, j_n$. Тоді мінор M' n-го порядку матриці B, розміщений у рядках з тими самими номерами $j_1, \dots, j_n$, називають *відповідним* мінору M.

Лема 4.6. *У попередніх позначеннях* $\det(AB) = 0$, *якщо* $n > m$, *і*

$$\det(AB) = \sum_{1 \le j_1 \le \dots \le j_n \le m} MM'$$

— сума за всіма наборами номерів стовпчиків, мінорів n-го порядку матриці A на відповідні мінори матриці B.

Доведення. Нехай

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix},$$

Тоді $C = AB =$

$$\begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + \cdots + a_{1m}b_{m1} & \cdots & a_{11}b_{1n} + a_{12}b_{2n} + \cdots + a_{1m}b_{mn} \\ a_{21}b_{11} + a_{22}b_{21} + \cdots + a_{2m}b_{m1} & \cdots & a_{21}b_{1n} + a_{22}b_{2n} + \cdots + a_{2m}b_{mn} \\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots & \cdots & \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots \\ a_{n1}b_{11} + a_{n2}b_{21} + \cdots + a_{nm}b_{m1} & \cdots & a_{n1}b_{1n} + a_{n2}b_{2n} + \cdots + a_{nm}b_{mn} \end{pmatrix}.$$

Стовпчики матриці C є лінійними комбінаціями стовпчиків матриці A з коефіцієнтами $b_{11}, \dots, b_{m1}, \dots, b_{1n}, \dots, b_{mn}$. Тому, використовуючи лінійність визначника за стовпчиками, маємо

$$\det(C) = \sum_{1 \le i_1, \dots, i_n \le m} b_{i_1 1} \cdots b_{i_n n} \begin{vmatrix} a_{1i_1} & \cdots & a_{1i_n} \\ \cdots & \cdots & \cdots \\ a_{ni_1} & \cdots & a_{ni_n} \end{vmatrix}. \tag{4.7}$$

Якщо серед індексів $i_1, \dots, i_n$ однакові, то відповідний доданок у (4.7) дорівнює нулю, бо визначник, що входить в нього має два однакові стовпчики. Викинемо з суми (4.7) такі нульові доданки, решту доданків перегрупуємо: в окрему групу виділимо доданки, які відповідають заданому зафіксованому набору $1 \le j_1 < \dots < j_n \le n$. Одержимо

$$\det(C) = \sum_{1\le j_1<\dots<j_n\le m} \left(\sum_{\left(\begin{smallmatrix} i_1 & \dots & i_n \\ j_1 & \dots & j_n \end{smallmatrix}\right)} b_{i_1 1}\cdots b_{i_n n} \begin{vmatrix} a_{1i_1} & \cdots & a_{1i_n} \\ \cdots & \cdots & \cdots \\ a_{ni_1} & \cdots & a_{ni_n} \end{vmatrix} \right), \quad (4.8)$$

де $\left(\begin{smallmatrix} i_1 & \dots & i_n \\ j_1 & \dots & j_n \end{smallmatrix}\right)$ — підстановка елементів $i_1, \dots, i_n$.

У визначниках, які входять у (4.8), переставимо стовпчики у порядку зростання других індексів $i_1, \dots, i_n$:

$$\det(C) = \sum_{1\le j_1<\dots<j_n\le m} \left(\sum_{\left(\begin{smallmatrix} i_1 & \dots & i_n \\ j_1 & \dots & j_n \end{smallmatrix}\right)} b_{i_1 1}\cdots b_{i_n n}(-1)^{\varepsilon} \begin{vmatrix} a_{1j_1} & \cdots & a_{1j_n} \\ \cdots & \cdots & \cdots \\ a_{nj_1} & \cdots & a_{nj_n} \end{vmatrix} \right), \quad (4.9)$$

де $\varepsilon = 0$, якщо підстановка $\left(\begin{smallmatrix} i_1 & \dots & i_n \\ j_1 & \dots & j_n \end{smallmatrix}\right)$ парна і $\varepsilon = 1$, якщо ця підстановка непарна.

Виносячи в (4.9) за дужки спільний множник, одержуємо

$$\det(C) = \sum_{1\le j_1<\dots<j_n\le m} \begin{vmatrix} a_{1j_1} & \cdots & a_{1j_n} \\ \cdots & \cdots & \cdots \\ a_{nj_1} & \cdots & a_{nj_n} \end{vmatrix} \cdot \begin{vmatrix} b_{j_1 1} & \cdots & b_{j_1 n} \\ \cdots & \cdots & \cdots \\ b_{j_n 1} & \cdots & b_{j_n n} \end{vmatrix},$$

що й треба було довести.

Доведення теореми 4.5. Нехай граф G має n вершин і m ребер, A — матриця Кіркгофа графа G, а I — матриця інцидентності якої-небудь його орієнтації. За лемою 4.4 $A = I \cdot I^t$. Нехай B — матриця, одержана з A, викресленням останнього рядка та останнього стовпчика матриці A, C — матриця, одержана з I, викресленням останнього рядка. Тоді з рівності $A = I \cdot I^t$ одержуємо рівність $B = C \cdot C^t$.

Визначник матриці B дорівнює A_{nn} — алгебричному доповненню елемента α_{nn} матриці A. З іншого боку, за лемою 4.6, $\det(B) = \sum MM'$ — сумі добутків мінорів M $n-1$-го порядку матриці C на відповідні мінори M' матриці C^t. У нашому випадку $M = M'$, і, за лемою 4.5, $M = M' = \pm 1$, якщо підграф графа G, ребра якого відповідають стовпчикам, що входять в M, є каркасним деревом, і $M = M' = 0$ в іншому

випадку. Отже, $A_{nn} = \det(B) = \sum MM' = \sum M^2$ дорівнює кількості каркасних дерев графа G. Для завершення доведення теореми залишається зауважити, що всі алгебричні доповнення матриці Кіркгофа рівні між собою.

Наслідок 4.4. *Нехай $k(G)$ — кількість зв'язних компонент n-вершинного графа G, $n > 1$, $rank A$ — ранг матриці Кіркгофа графа G. Тоді*

$$k(G) = n - rank\, A.$$

Доведення. Нехай спочатку граф G зв'язний, тобто $k(G) = 1$. У цьому випадку G має каркасні дерева, тому $\operatorname{rank} A = n - 1$ і формула правильна.

Якщо $k(G) = k > 1$, то перенумеруємо його вершини так: спочатку нумеруємо вершини $v_1, \ldots, v_{i_1}$ першої компоненти зв'язності, потім другої і т.д., $v_{i_{k-1}}, \ldots, v_n$ — вершини останньої компоненти зв'язності. Тоді матриця Кіркгофа має блочно-діагональний вигляд

$$A = \begin{pmatrix} \boxed{A_1} & & & \\ & \boxed{A_2} & \cdots & 0 \\ & \cdots & & \\ & 0 & \cdots & \boxed{A_k} \end{pmatrix}.$$

Кожна матриця A_i є матрицею Кіркгофа зв'язного графа $G_i, 1 \le i \le k$. Тому, як було зауважено вище, $\operatorname{rank} A_i = n_i - 1$, де n_i – кількість вершин графа G_i. З іншого боку, $\operatorname{rank} A = \sum_{i=1}^{k} \operatorname{rank} A_i = \sum_{i=1}^{k} (n_i - 1) = n - k$, тому $k = n - \operatorname{rank} A$.

Наслідок 4.5. *Кількість каркасних дерев повного графа K_n дорівнює n^{n-2}.*

Доведення. Матриця Кіркгофа повного графа K_n має вигляд

$$A = \begin{pmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & -1 & -1 & \cdots & n-1 \end{pmatrix}.$$

Кількість каркасних дерев повного графа K_n дорівнює алгебричному доповненню будь-якого елемента цієї матриці, зокрема алгебричному доповненню елемента, розміщеного в останньому рядку і останньому стовпчику. Отже, кількість каркасних дерев дорівнює визначнику $n-1$-го порядку

$$\begin{vmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & -1 & -1 & \cdots & n-1 \end{vmatrix}.$$

Для обчислення цього визначника додамо до першого рядка всі інші рядки, а потім до кожного рядка, починаючи з другого, перший рядок. Одержимо

$$\begin{vmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & -1 & -1 & \cdots & n-1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ -1 & n-1 & -1 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & -1 & -1 & \cdots & n-1 \end{vmatrix} =$$

$$= \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & n & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & n \end{vmatrix} = n^{n-2}.$$

4.1.7. Зважені графи. Алгоритми Краскала та Дейкстри

Означення 4.11. *Граф G називають зваженим, якщо кожному його ребру l приписане додатне дійсне число $w(l)$ (інакше кажучи, задане відображення w з множини ребер графа G у множину дійсних чисел). Число $w(l)$ називають вагою ребра l. Суму ваг всіх ребер називають вагою графа і позначають $w(G)$.*

Ось одна з моделей зваженого графа. Нехай $A_1, \ldots, A_n$ — міста. Для кожної пари міст A_i, A_j $w(\{A_i, A_j\})$ означає відстань між ними. Маємо повний зважений граф n-го порядку.

Припустимо, що міста $A_1, \ldots, A_n$ потрібно з'єднати, наприклад, лініями електропередач (або каналами зв'язку, або трубопроводами). Зрозуміло, що найкращими є найдешевші з'єднання, тобто такі, для яких відповідний зважений граф має найменшу вагу. У зв'язку з цим важливою є задача відшукання у повному зваженому графі каркасного дерева

найменшої ваги. Існують алгоритми, які розв'язують цю задачу. Одним з них є алгоритм Краскала, який ми зараз сформулюємо.

Алгоритм Краскала. *Нехай G — зважений зв'язний граф, що має n вершин. Потрібно вибрати каркасне дерево графа G, яке має найменшу вагу.*

Для цього розглянемо граф T_0, що складається лише з вершин графа G і не має ребер.

1. *На першому кроці додамо до графа T_0 будь-яке ребро l_1 з найменшою вагою. Позначимо одержаний граф T_1.*
2. *На другому кроці додамо до графа T_1 будь-яке ребро $l_2 \neq l_1$ з найменшою вагою. Позначимо одержаний граф T_2.*

. . . . . .

i. *На i-му кроці додамо до графа T_{i-1} будь-яке ребро l_i з найменшою вагою, і таке, що:*
   а) l_i має найменшу вагу серед тих ребер графа G, які не входять у T_{i-1};
   б) l_i не утворює циклів з ребрами, які входять у T_{i-1}.
   Позначимо одержаний граф T_{i-1}.

n-1. *T_{n-1} — шуканий граф.*

Теорема 4.6. *Описаний алгоритм є коректним і дає на $n-1$ кроці каркасне дерево найменшої ваги.*

Доведення. Зауважимо, що для кожного i, $1 \leq i < n-1$ знайдеться ребро l_i, яке задовольняє умови *а* і *б*. Справді, граф G зв'язний, тому за лемою 4.3 він має не менше ніж $n-1$ ребро. Далі, якби при долученні кожного ребра $l \neq l_1, \ldots, l_{i-1}$, $i < n-1$ графа G до T_{i-1} виникав цикл, то граф T_{i-1} був би зв'язний, а це суперечить лемі 4.3.

Отже, граф T_{n-1} можна побудувати. З теореми 4.4 випливає, що T_{n-1} дерево. Залишається перевірити, що T_{n-1} — дерево найменшої ваги. Для цього припустимо, що існує каркасне дерево $S \neq T_{n-1}$ і $w(S) < w(T_{n-1})$, де w — вага. Тоді існує хоч одне ребро графа T_{n-1}, яке не є ребром графа S. Серед таких ребер виберемо ребро l_i з найменшим i. Долучимо це ребро l_i до графа S. З теореми 4.4 випливає, що граф S повинен мати цикл. Нехай $l \neq l_i$ – деяке ребро графа S,

яке входить у цей цикл. Розглянемо граф $S' = S + l_i - l$. Це дерево і $w(S') = w(S) + w(l_i) - w(l)$. Якби $w(l) > w(l_i)$, то ми мали б $w(S') < w(S)$. Але S – граф найменшої ваги, тому $w(l) < w(l_i)$, l не утворює циклів з ребрами $l_1, \ldots, l_{i-1}$ (бо $l_1, \ldots, l_{i-1}, l$ – частина дерева S'). Отож, на i-у кроці алгоритму замість ребра l_i повинно було бути вибраним ребро l (або інше ребро з вагою $w(l)$). Одержана суперечність свідчить, що граф T_{n-1} є каркасним деревом найменшої ваги.

Для зважених графів зручно дещо модифікувати означення матриці суміжності.

Означення 4.12. *Нехай G — зважений граф і нехай $v_1, \ldots, v_n$ — деякий перелік вершин графа G. Квадратну матрицю $A = [a_{ij}]$ n-го порядку з елементами*

$$a_{ij} = \begin{cases} w(\{v_i, v_j\}), & \text{якщо вершини } v_i, v_j \text{ суміжні,} \\ 0, & \text{в іншому випадку.} \end{cases}$$

Матрицею суміжності орієнтованого графа G називається квадратна матриця n-го порядку $A = [\alpha_{ij}]$, для якої

$$\alpha_{ij} = \begin{cases} w((v_i, v_j)), & \text{якщо вершини } v_i, v_j \text{ суміжні,} \\ 0, & \text{в іншому випадку.} \end{cases}$$

Зауважимо, що матриця суміжності неорієнтованого зваженого графа симетрична, а матриця суміжності орієнтованого зваженого графа здебільшого несиметрична.

Якщо маємо зважений граф G (орієнтований або ні), то виникає задача відшукання в G маршрутів найменшої ваги з початком у заданій вершині u. Вагу маршруту найменшої ваги, який з'єднує вершини u і v, назвемо *відстанню від u до v* і позначимо $d(u, v)$. Існує загальний алгоритм (алгоритм Дейкстри) для знаходження маршрутів найменшої ваги. В алгоритмі Дейкстри вершинам s приписують мітки $\mu(s)$. Мітки — це невід'ємні числа або ∞. Мітки бувають тимчасові та сталі. Якщо мітка $\mu(s)$ стала, то $\mu(s)$ — відстань від u до s. При роботі алгоритму тимчасові мітки зменшуються, доки не стануть сталими. На кожному кроці одна з тимчасових міток стає сталою. Сталі мітки не змінюються під час роботи алгоритму. Крім міток $\mu(s)$, деякі вершини $s \neq u$ при роботі алгоритму наділяються мітками $\theta(s)$: на кожному кроці алгоритму $\theta(s)$ дорівнює номеру вершини, яка передує s у маршруті найменшої

ваги від u до s серед всіх тих маршрутів, які проходять через вершини, що вже мають сталі мітки. Мітки $\theta(s)$ використовують для запису послідовності вершин $u, \ldots, v$, які задають маршрут найменшої ваги від u до v.

Перед початком першого кроку вершина u має сталу мітку $\mu(u) = 0$, а всі інші вершини мають тимчасові мітки ∞. На i-му кроці алгоритм розглядає вершину s, яка одержала сталу мітку на попередньому кроці, і всі вершини t з оточення N_s вершини s з метою зменшення міток $\mu(t)$ цих вершин. Мітка $\mu(t)$ вершини $t \in N_s$ змінюється на $\mu(s) + w(s,t)$, якщо $\mu(s) + w(s,t) < \mu(t)$ і в цьому випадку приймають $\theta(t) = s$. Якщо $\mu(s) + w(s,t) \geq \mu(t)$, то на цьому кроці мітки $\mu(t)$ і $\theta(t)$ не змінюються. Алгоритм завершує роботу, коли мітка вершини v стане сталою.

Наведемо точний опис алгоритму.

Алгоритм Дейкстри

1. Вибрана вершина u одержує сталу мітку $\mu(u) = 0$. Кожна інша вершина s одержує тимчасову мітку $\mu(s) = \infty$. Вершина u одержує назву p: $p := u$.
2. Для всіх вершин $s \in N_p$ з оточення N_p вершини p виконати: якщо $\mu(p) + w(p,s) < \mu(s)$, то $\mu(s)$ замінити на $\mu(p) + w(p,s)$ і прийняти $\theta(s) = p$. В іншому випадку $\mu(s)$ і $\theta(s)$ не змінюються.
3. Нехай V' — множина всіх вершин з тимчасовими мітками μ. Знайти вершину $s^* \in V'$, для якої

$$\mu(s^*) = \min_{s \in V'} \mu(s),$$

   і вважати мітку $\mu(s^*)$ сталою міткою вершини s^*. Вершина s^* одержує назву p: $p := s^*$.
4. Якщо $p = v$, то алгоритм дає маршрут $(u, \ldots, \theta(v), v)$ і зупиняється; якщо $p \neq v$, то він повертається до кроку 2.

Приклад

Проілюструємо роботу алгоритму Дейкстри на прикладі. Розглянемо зважений граф з множиною вершин $\{1,2,3,4,5,6,7\}$ і множиною ребер $\{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{1,6\},\{6,7\},\{7,5\}\}$, $\mu(\{1,2\}) = 1, \mu(\{2,3\}) = 1, \mu(\{3,4\}) = 1, \mu(\{4,5\}) = 100$, $\mu(\{1,6\}) = 5, \mu(\{6,7\}) = 7, \mu(\{7,5\}) = 10$, зображений на рис. 4.13. Матрицею суміж-

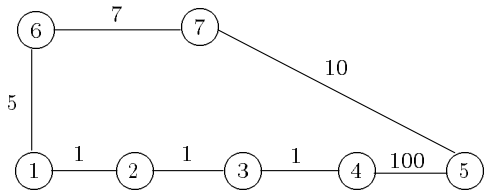


Рис. 4.13.

ності цього графа є матриця

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 5 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 10 \\ 5 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 10 & 7 & 0 \end{pmatrix}.$$

Припустимо, що нам потрібно побудувати маршрут найменшої ваги від вершини 1 до вершини 5. Кроки роботи алгоритму відповідають рядкам таблиці. У i-му рядку таблиці записані мітки μ вершин, що присвоюють вершинам на i-му кроці роботи алгоритму. Постійні мітки підкреслено і надруковано напівжирним шрифтом. В останньому стовпчику таблиці записано побудовані алгоритмом маршрути найменшої ваги.

	1	2	3	4	5	6	7	Маршрут
0	**0**	∞	∞	∞	∞	∞	∞	
1	**0**	**1**	∞	∞	∞	5	∞	(1,2)
2	**0**	**1**	**2**	∞	∞	5	∞	(1,2,3)
3	**0**	**1**	**2**	**3**	∞	5	∞	(1,2,3,4)
4	**0**	**1**	**2**	**3**	103	**5**	∞	(1,6)
5	**0**	**1**	**2**	**3**	103	**5**	**12**	(1,6,7)
6	**0**	**1**	**2**	**3**	**22**	**5**	**12**	(1,6,7,5)

Теорема 4.7. *Описаний алгоритм Дейкстри є коректним і будує маршрут найменшої ваги від u до v.*

Доведення. З означення алгоритму бачимо таке: коли мітка $\mu(s)$ вершини s стала, то вона дорівнює вазі маршруту від u до s. Потрібно довести, що ця мітка дорівнює вазі маршруту найменшої ваги від u до s. Міркуємо за індукцією кількістю k кроків алгоритму. Якщо $k = 1$, то це очевидно. Припустимо, що це правильно для всіх вершин, які одержали

сталі мітки на кроках $1, \ldots, k-1$. Нехай $L(u,s)$ — маршрут від u до s, одержаний на k-му кроці. За побудовою $w(L(u,s)) = \mu(s)$. Нехай $L^*(u,s)$ — маршрут найменшої ваги від u до s. Потрібно довести, що $w(L(u,s)) = w(L^*(u,s))$. Припустимо від супротивного, що $w(L(u,s)) > w(L^*(u,s))$. Якщо маршрут $L^*(u,s)$ проходить не лише через вершини, які одержали сталі мітки на k-му кроці, коли вершина s одержала сталу мітку, то нехай t — перша вершина цього маршруту, яка не одержала сталої мітки, а t' — попередня вершина. Тоді

$$\mu(t') + w(\{t,t'\}) \geq \mu(t) \geq \mu(s).$$

Тим більше, $w(L(u,s)) \leq w(L^*(u,s))$. Отже, маршрут $L^*(u,s)$ проходить лише через вершини, які одержали сталі мітки до k-го кроку.

Розглянемо передостанню вершину s' маршруту $L^*(u,s)$. Позаяк

$$\mu(s) \leq \mu(s') + w((s',s)),$$

то $w(L(u,s)) \leq w(L^*(u,s))$, і теорема доведена.

4.1.8. Ойлерові графи та плоскі графи

Означення 4.13. *Граф G називають ойлеровим, якщо у ньому існує циклічний маршрут, який містить кожне ребро лише один раз.*

З означення, зокрема, випливає, що ойлеровий граф обов'язково зв'язний.

Теорема 4.8. *Зв'язний граф G є ойлеровим тоді й лише тоді, коли всі його вершини мають парний степінь.*

Доведення. Якщо граф G ойлеровий, то кожний циклічний маршрут, який містить кожне ребро лише один раз, входить у кожну вершину і виходить з неї однакову кількість разів. Це й означає, що степені всіх вершин повинні бути парними.

Навпаки, нехай степені всіх вершин парні. Виберемо деяку вершину v, почнемо прокладати з цієї вершини маршрут, який містить кожне ребро не більше ніж один раз, проходячи кожний раз по ребру, яке не було пройденим раніше. Позаяк граф G скінченний, то цей процес повинен закінчитися через скінченну кількість кроків. Оскільки кожна вершина має парний степінь, то при прокладанні такого маршруту, опинившись

у вершині $s \neq v$, завжди є змога вийти з цієї вершини і продовжити маршрут. Отже, процес прокладання маршруту, розпочатий у вершині v повинен закінчитися у цій же вершині v і ми одержуємо деякий цикл C_1.

Якщо C_1 проходить через всі вершини графа G, то доведення закінчене. В іншому випадку знайдеться вершина $w \in C_1$, інцидентна деякому ребру графа G, яке не входить у C_1.

Почнемо новий цикл C_2 з вершини w, використовуючи лише ті ребра, які не входять у C_1. Як і раніше, C_2 повинен закінчитися у цій же вершині w. Тепер, маючи цикли C_1 і C_2, можна побудувати «довший» циклічний маршрут C_3 так: починаємо з вершини w, проходимо C_1, повернувшись у w, а тоді проходимо C_2, повернувшись знову у w.

Якщо C_3 проходить через всі вершини графа G, то доведення закінчене. В іншому випадку знайдеться вершина $w_1 \in C_3$, інцидентна деякому ребру графа G, яке не входить у C_3, і можна, повторюючи попередні міркування, знайти ще довший циклічний маршрут, і так далі, аж поки не отримаємо циклічний маршрут, який містить кожне ребро графа G лише один раз.

Означення 4.14. *Граф G називають плоским, якщо його можна зобразити на площині так, щоб ребра не мали, відмінних від вершин, точок перетину.*

Нехай G — плоский граф, розглянемо деяке його зображення G' на площині, яке задовольняє попередньому означенню. Нехай M — множина всіх точок площини, що не належать G'. Назвемо дві точки A і B множини M еквівалентними, якщо їх можна сполучити неперервною дугою, яка з'єднує ці точки і не має спільних елементів з G'. Одержане відношення на множині M є відношенням еквівалентності. Множини всіх еквівалентних між собою точок назвемо *гранями* графа G. На рис. 4.14 зображений плоский граф, який має грані I, II, III, IV.

Теорема 4.9 (Формула Ойлера). *Нехай G – зв'язний плоский граф, v — кількість вершин, r — кількість ребер, і g — кількість граней графа G. Тоді*

$$v - r + g = 2.$$

Доведення. Розглянемо будь-яке каркасне дерево D графа G. Граф D має одну грань, і, якщо він має v вершин, то за теоремою 4.4 він має

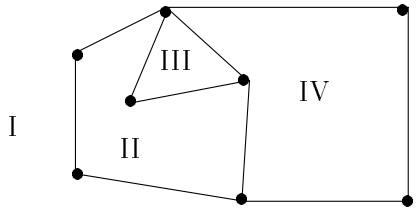


Рис. 4.14.

$v-1$ ребро. Тому $v-r+g=v-(v-1)+1=2$. Тепер, долучаючи до D одне ребро, ми, не збільшуючи кількості вершин, збільшуємо на 1 кількість граней і формула $v-r+g=2$ залишається правильною. Твердження теореми випливає з того, що граф G можна одержати з дерева D, долучаючи до D скінченну кількість ребер.

Твердження 4.3. *Нехай G — зв'язний плоский граф, r — кількість ребер, g — кількість граней графа G. Тоді $3g \leq 2r$.*

Доведення. Нерівність $3g \leq 2r$ випливає з того, що кожна грань обмежена не менше ніж трьома ребрами, а кожне ребро розділяє не більше ніж дві грані.

Наслідок 4.6. *Граф K_5 не є плоским.*

Доведення. Якби цей граф був плоским, то кількість його граней дорівнювала б $g=2-v+r=2-5+10=7$. Підставивши це в формулу з попереднього наслідку, одержуємо суперечність $3\cdot 7 \leq 2\cdot 10$.

Наслідок 4.7. *Граф $K_{3,3}$ не є плоским.*

Доведення. Граф $K_{3,3}$ має 6 вершин і 9 ребер. Якби він був плоским, то для нього $g=2-v+r=2-6+9=5$. З іншого боку, як легко зрозуміти, кількість його граней g повинна задовольняти нерівність $4\cdot g \leq 2r$, тобто $20<18$ і ми одержуємо суперечність.

4.1.9. Правильні многогранники

Нехай G — плоский граф. Побудуємо новий плоский граф G^*, який називають *двоїстим* до G, таким способом. Всередині кожної грані

графа G вибираємо одну точку. Ці точки є вершинами графа G^*. Дві такі точки A і B з'єднуємо ребром тоді й лише тоді, коли вони належать до сусідніх граней, тобто до граней, які мають спільне граничне ребро. На рис. 4.15 суцільними лініями зображений плоский граф G, а пунктирними лініями зображений двоїстий граф G^*.

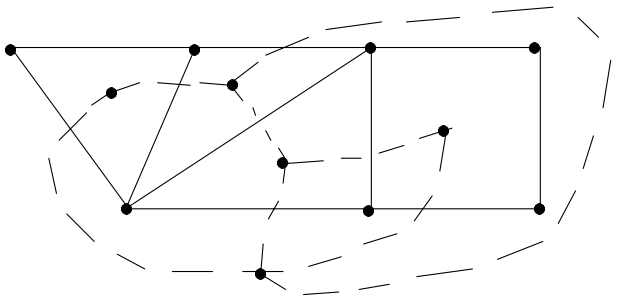


Рис. 4.15.

Нагадаємо, що граф називають регулярним, якщо степені всіх його вершин однакові. Регулярний плоский граф G називають *правильним*, якщо двоїстий граф G^* теж регулярний. Нехай G — правильний граф, ρ — степінь його вершин, а ρ^* — степінь вершин двоїстого графа G^* (ρ^* дорівнює кількості ребер, які обмежують кожну грань графа G). Якщо, як і раніше, v, r і g — кількість вершин, ребер та граней графа G, то $2r = \rho v = \rho^* g$.

Звідси маємо

$$r = \frac{1}{2}\rho v, \quad g = \frac{\rho}{\rho^*} v. \tag{4.10}$$

Підставивши ці значення r і g у формулу Ойлера, одержуємо

$$v(1 + \frac{\rho}{\rho^*} - \frac{1}{2}\rho) = 2,$$

тобто

$$v(2\rho + 2\rho^* - \rho\rho^*) = 4\rho^*. \tag{4.11}$$

Оскільки v і ρ^* — натуральні числа, то

$$2\rho + 2\rho^* - \rho\rho^* > 0.$$

Цю нерівність можна записати у вигляді

$$\rho\rho^* - 2\rho - 2\rho^* < 0, \quad \text{або} \quad (\rho - 2)(\rho^* - 2) < 4. \tag{4.12}$$

Розглянемо випадок, коли $\rho - 2 > 0$ і $\rho^* - 2 > 0$. У цьому випадку $\rho - 2 \leq 3$ і $\rho^* - 2 \leq 3$ і розв'язками нерівності (4.12) можуть бути лише такі 5 пар $(\rho, \rho^*) : (3,3), (3,4), (3,5), (4,3), (5,3)$.

Для кожної такої пари можливих значень (ρ, ρ^*) обчислюємо v, r, g за формулами (4.10) і (4.11) і розміщуємо результати обчислень у таблицю.

Правильні графи

ρ	ρ^*	v	r	g	Тип
3	3	4	6	4	Тетраедр
3	4	8	12	6	Куб
3	5	20	30	12	Додекаедр
4	3	6	12	8	Октаедр
5	3	12	30	20	Ікосаедр

Пропонуємо читачеві самостійно дослідити розв'язки нерівності (4.12) у випадку, коли $\rho - 2 < 0$ і $\rho^* - 2 < 0$.

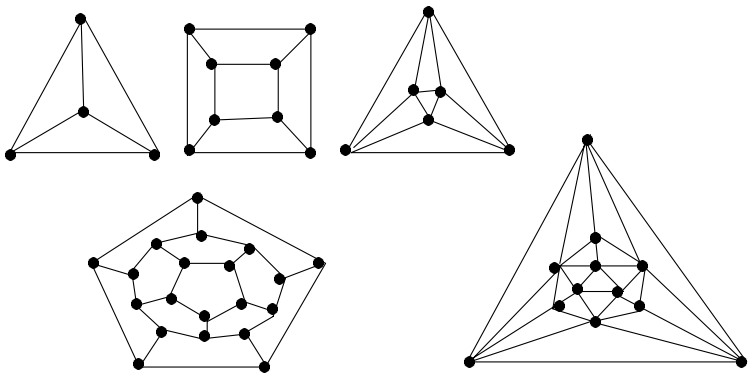


Рис. 4.16.

Отже, існує лише 5 типів правильних графів, для яких $\rho \geq 3$ і $\rho^* \geq 3$. Ці правильні графи зображені на рис. 4.16.

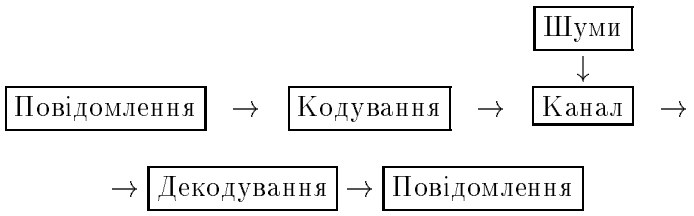
Правильні графи з рис. 4.16 — це графи правильних многогранників. Правильним многогранником називають многогранник, який має рівні ребра, кути між ними, а також однакові грані.

Правильні многогранники вперше систематично вивчали у 13 книзі «Начал» Евкліда, потім їх досліджував Платон (тому їх ще називають платоновими тілами). В давнину та середні віки правильні многогранники вважали символами гармонії всесвіту.

4.2. Коди

Припустимо, що ми передаємо інформацію, використовуючи деякий канал зв'язку (наприклад, телефонну лінію). В процесі передачі інформація може спотворюватися, тому виникає задача розроблення прийомів, які дали б змогу або зробити такі спотворення неможливими або звести можливі спотворення до мінімуму. Суть таких прийомів полягає в тому, що перед відсиланням інформації до каналу зв'язку її закодовують, а після проходженння інформації через канал зв'язку розкодовують.

Загальна схема процесу передачі та прийому інформації має такий вигляд:



Вважатимемо, що інформація, яку ми передаємо, складається зі слів, які є впорядкованими послідовностями з нулів та одиниць (*бітів*). Найпростіший прийом, який дає змогу зменшувати ймовірність одержання неправильної інформації на виході з каналу зв'язку, це повторення декілька разів символів, які передаються.

Наприклад, якщо хочемо передати слово 00101, то повторюючи кожен символ тричі, кодуємо його у слово 000000111000111, яке й відправляємо у канал зв'язку. Припустимо, на виході з каналу зв'язку ми одержали 010000110000111. Розбиваємо одержане слово на групи по три символи: 010 000 110 000 111. Далі кожну групу символів замінюємо тим символом, який найчастіше у цій групі трапляється. Одержуємо початкове слово 00101.

Такі методи кодування називають *кодами з повтореннями*. Зауважимо, що коди з повтореннями дуже неекономні. За можливість зменшувати ймовірність виникнення помилок при передачі доводиться платити збільшенням у декілька разів (у розглянутому випадку втричі) часу, протягом якого передається інформація.

Економніший з цього погляду клас кодів становлять так звані *коди з перевіркою на парність*. Найпростіший код з перевіркою на парність полягає в тому, що ми дописуємо до послідовності з нулів та одиниць, яку збираємося пропустити через канал зв'язку, ще один символ (0 або 1) так, щоб загальна кількість одиниць стала парною. Наприклад, до слова 00101 дописуємо 0, а до слова 10101 дописуємо 1. Одержуємо довші слова 001010 та 101011, які й передаємо. На виході з каналу зв'язку спочатку підраховують парність кількості входжень одиниць в одержаному слові. Якщо ця кількість парна, то роблять висновок, що помилки при передачі не з'явилися, а процес декодування полягає просто у закресленні останнього символу в одержаному слові. Якщо ж кількість одиниць в одержаному слові виявилась непарною, то можна лише зробити висновок, що при передачі з'явилися помилки, але не можна відновити передане слово. Кажуть, що такий код виявляє помилки, тоді як розглянутий вище код з повтореннями не лише їх виявляє, а й виправляє.

Розглянуті два приклади кодів дуже елементарні, тому кожен з них має серйозні недоліки. Предмет теорії кодування полягає у створенні досконаліших кодів, які б поєднували переваги і не мали недоліків щойно розглянутих кодів. Значною мірою побудова таких кодів ґрунтується на поєднанні двох основних прийомів: повторення символів та перевірки на парність.

Ми ознайомимося з важливим класом кодів так званими лінійними кодами. Для лінійних кодів кодовими словами є елементи скінченновимірних лінійних просторів над скінченним полем. Тому ми почнемо з основних фактів про скінченні поля.

4.2.1. Скінченні поля

Нехай p — просте число. Позначимо через $\mathbb{F}_p$ факторкільце $\mathbb{Z}/p\mathbb{Z}$.

Твердження 4.4. *$\mathbb{F}_p$ — поле, яке має p елементів.*

Доведення. Достатньо довести, що кожний ненульовий елемент $\bar{a} \in \mathbb{F}_p$ має обернений щодо множення. Якщо $\bar{0} \neq \bar{a} \in \mathbb{F}_p$, то $p \not| a$, тому p

і a взаємно прості. З твердження 3.19 випливає, що існують $u, v \in \mathbb{Z}$, для яких $ua + pv = 1$. (Знайти такі $u, v \in \mathbb{Z}$ можна використовуючи алгоритм Евкліда або розширений алгоритм Евкліда.) Звідси, переходячи до класів лишків, одержуємо рівність $\overline{ua + pv} = \bar{1}$ у кільці $\mathbb{F}_p$, яку можемо записати у вигляді $\bar{u}\bar{a} + \bar{p}\bar{v} = \bar{1}$, тобто $\bar{u}\bar{a} = \bar{1}$, бо $\bar{p}\bar{v} = \bar{0}$.

Твердження 4.5, у доведенні якого майже дослівно повторюються міркування з доведення попереднього твердження, дає метод побудови полів за допомогою поліномів.

Твердження 4.5. *Нехай K — довільне поле, $K[X]$ — кільце поліномів з коефіцієнтами з поля K. $p(X) \in K[X]$ — незвідний поліном, $(p(X))$ — головний ідеал, породжений поліномом $p(X)$. Тоді факторкільце $K[X]/(p(X))$ є полем.*

Доведення. Достатньо довести, що кожний ненульовий елемент $\overline{a(X)} \in K[X]/(p(X))$ має обернений щодо множення. Якщо $\overline{0} \neq \overline{a(X)}$, то $p(X) \not| \, a(X)$, тому $p(X)$ і $a(X)$ взаємно прості. З наслідку до алгоритму Евкліда знаходження найбільшого спільного дільника випливає, що існують $u(X), v(X) \in K[X]$, для яких $u(X)a(X) + p(X)v(X) = 1$. Звідси, переходячи до суміжних класів за $\bmod p(X)$, одержуємо рівність $\overline{u(X)a(X) + p(X)v(X)} = \overline{1}$ у кільці $K[X]/(p(X))$, яку можемо записати у вигляді $\overline{u(X)a(X)} + \overline{p(X)v(X)} = \overline{1}$, тобто $\overline{u(X)a(X)} = \overline{1}$, бо $\overline{p(X)v(X)} = \overline{0}$.

Приклади

1. Розглянемо поле з двох елементів $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z} = \{0, 1\}$ і поліном $p(X) = X^2 + X + 1 \in \mathbb{F}_2[X]$. Перевіряємо, що 0 і 1 не є коренями полінома $p(X)$, тому цей поліном незвідний. Отже, факторкільце $\mathbb{F}_2[X]/(p(X))$ є полем. Маємо

$$\mathbb{F}_2[X]/(p(X)) = \{\overline{a(X)} \mid a(X) \in \mathbb{F}_2[X]\} =$$

$$\{\overline{d(X)p(X) + a_0 + a_1X} \mid d(X) \in \mathbb{F}_2[X]\} = \{\overline{a_0 + a_1X} \mid a_0, a_1 \in \mathbb{F}_2\} = \{\bar{0}, \bar{1}, \bar{X}, \bar{1} + \bar{X}\}.$$

   Позначимо $\bar{X} = \alpha$. Ми знайшли поле з чотирьох елементів $0, 1, \alpha, \alpha^2$.

2. Знову розглянемо поле $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z} = \{0, 1\}$ і поліном $p(X) = X^3 + X + 1 \in \mathbb{F}_2[X]$. Перевіряємо, що 0 і 1 не є коренями полінома $p(X)$, тому цей поліном незвідний. Отож, факторкільце $\mathbb{F}_2[X]/(p(X))$ є полем.

$$\mathbb{F}_2[X]/(p(X)) = \{\overline{a_0 + a_1X + a_2X^2} \mid a_0, a_1, a_2 \in \mathbb{F}_2\} =$$

$$= \{0, 1, \alpha, 1 + \alpha, \alpha^2, 1 + \alpha^2, \alpha + \alpha^2, 1 + \alpha + \alpha^2\}$$

   — поле з восьми елементів.

Теорема 4.10. *1. Кожне скінченне поле $\mathbb{F}$ містить підполе, ізоморфне полю $\mathbb{Z}/p\mathbb{Z}$ для деякого простого числа p.*

2. Кожне скінченне поле $\mathbb{F}$ містить p^n елементів, де p — деяке просте число, $n \geq 1$.

Доведення. 1. У кожному полі існує нейтральний елемент щодо множення — 1. Розглянемо елементи $1, 1+1 = 2 \cdot 1, \ldots, 1+\cdots+1 = n \cdot 1, \ldots \in \mathbb{F}$. Оскільки $\mathbb{F}$ скінченне поле, то ці елементи не можуть бути всі різними. Тому існують $m, n \in \mathbb{N}$, для яких $m \cdot 1 = n \cdot 1$. Нехай $m > n$, тоді $(m-n) \cdot 1 = 0$, тобто існують додатні натуральні числа k з властивістю $k \cdot 1 = 0$. Нехай p — найменше таке число. Доведемо, що p просте число. Якби $p = p_1 p_2$, де $1 < p_1, p_2 < p$, то ми мали б $0 = (p_1 p_2) \cdot 1 = (p_1 \cdot 1)(p_2 \cdot 1)$. Позаяк у полі немає дільників нуля, то звідси випливає, що $(p_1 \cdot 1) = 0$ або $(p_2 \cdot 1) = 0$, а це суперечить вибору p. Тому p просте число і $\mathbb{F} \cong \mathbb{Z}/p\mathbb{Z}$.

2. За доведеним $\mathbb{F}$ містить поле $\mathbb{F}_p$ з p елементів, для деякого простого числа p. Скінченне поле $\mathbb{F}$ є скінченновимірним лінійним простором над полем $\mathbb{F}_p$. Якщо $\alpha_1, \ldots, \alpha_n$ – база цього простору, то $\mathbb{F} = \{a_1\alpha_1 + \cdots + a_n\alpha_n \mid a_i \in \mathbb{F}_p\}$. Звідси випливає, що $\mathbb{F}$ має p^n елементів.

Далі ми завжди позначатимемо скінченне поле з q елементів через $\mathbb{F}_q$. Зі щойно доведеної теореми випливає, що q є степенем простого числа.

Тепер доведемо такий важливий результат.

Теорема 4.11. *Мультиплікативна група $\mathbb{F}_q^*$ скінченного поля $\mathbb{F}_q$ циклічна.*

Доведення. Група $\mathbb{F}_q^*$ має порядок $q-1$. Розкладемо $q-1$ у добуток простих чисел

$$q - 1 = p_1^{k_1} \cdots p_m^{k_m}.$$

Розглянемо поліном

$$X^{q-1} - 1 \in \mathbb{F}_q[X]. \tag{4.13}$$

Оскільки порядок елемента будь-якої скінченної групи ділить порядок групи (за наслідком з теореми Лагранжа про скінченні групи), то всі елементи групи $\mathbb{F}_q^*$ є коренями полінома (4.13). З іншого боку, позаяк

кількість коренів полінома з коефіцієнтами з поля не може перевищувати степінь полінома, то коренями полінома (4.13) є всі елементи групи $\mathbb{F}_q^*$ і лише вони.

Розглянемо ще поліноми

$$X^{\frac{q-1}{p_i}} - 1 \in \mathbb{F}_q[X],\ 1 \le i \le m. \tag{4.14}$$

Кожен з поліномів (4.14) має не більше ніж $\frac{q-1}{p_i} < q-1$ коренів. Тому для кожного $i, 1 \le i \le m$ існує корінь b_i полінома (4.13), який не є коренем полінома (4.14). Розглянемо елементи

$$a_i = b_i^{p_1^{k_1} \cdots p_{i-1}^{k_{i-1}} p_{i+1}^{k_{i+1}} \cdots p_m^{k_m}}, \quad 1 \le i \le m.$$

Знайдемо порядки $o(a_i)$ елементів a_i. Позаяк

$$a_i^{p_i^{k_i}} = b_i^{p_1^{k_1} \cdots p_m^{k_m}} = b_i^{q-1} = 1,$$

то $o(a_i) \mid p_i^{k_i}$. Якщо ми перевіримо, що $a_i^{p_i^{k_i-1}} \ne 1$, то звідси випливатиме, що $o(a_i) = p_i^{k_i}$. Маємо

$$a_i^{p_i^{k_i-1}} = b_i^{p_1^{k_1} \cdots p_{i-1}^{k_{i-1}} p_i^{k_i-1} p_{i+1}^{k_{i+1}}} = b_i^{\frac{q-1}{p_i}} \ne 1$$

за вибором елементів b_i.

Ми довели, що порядок елемента a_i дорівнює $p_i^{k_i}$. Оскільки числа $p_i^{k_i}$ є попарно взаємно простими, то порядок добутку $a = a_1 \cdots a_m$ дорівнює добутку порядків елементів a_i, тобто дорівнює $p_1^{k_1} \cdots p_m^{k_m} = q-1$. Отже, ми знайшли у групі $\mathbb{F}_q^*$ елемент a, порядок якого дорівнює порядку групи $\mathbb{F}_q^*$. Звідси випливає циклічність групи $\mathbb{F}_q^*$.

Зауваження 4.1. *Міркування, використані у доведенні теореми, дають змогу довести загальніший результат: кожна скінченна підгрупа мультиплікативної групи будь-якого поля є циклічною групою.*

4.2.2. Лінійні коди

Означення 4.15. *Нехай $\mathbb{F}_q$ — скінченне поле. $L = \mathbb{F}_q^n$ — n-вимірний лінійний простір над полем $\mathbb{F}_q$, C — k-вимірний підпростір простору L. Підпростір L називають лінійним (n,k)-кодом. Якщо $q = 2$, то код називають бінарним.*

Приклади

1. Розглянемо ще раз код з повтореннями, введений на початку цього параграфа. Метод кодування тут полягає у тому, що закодовані слова є елементами, які лежать в образі (лінійного) відображення

$$\varphi:\ \mathbb{F}_2^5 \to \mathbb{F}_2^{15},\ a_1a_2a_3a_4a_5 \mapsto a_1a_1a_1a_2a_2a_2a_3a_3a_4a_4a_4a_5a_5a_5.$$

Тут закодовані слова утворюють 5-вимірний підпростір простору $\mathbb{F}_2^{15}$, тому це лінійний $(15,5)$-код.

Цей код можна описати в інший спосіб: як множину розв'язків системи лінійних рівнянь

$$x_1 = x_2 = x_3,\ x_4 = x_5 = x_6,\ x_7 = x_8 = x_9,\ x_{10} = x_{11} = x_{12},\ x_{13} = x_{14} = x_{15}.$$

2. Тепер розглянемо бінарний $(n, n-1)$-код з перевіркою на парність. Це $n-1$-вимірний підпростір n-вимірного простору $\mathbb{F}_2^n$, елементи якого є розв'язками рівняння

$$x_1 + x_2 + \cdots + x_n = 0$$

з коефіцієнтами з поля $\mathbb{F}_2$. Частковий випадок такого коду (для $n = 6$) розглядали на початку цього параграфа.

3. Загальніше, якщо ми маємо однорідну систему лінійних рівнянь від n невідомих з коефіцієнтами зі скінченного поля $\mathbb{F}_q$, і ранг матриці цієї системи дорівнює r, то розв'язки системи утворюють $n-r$-вимірний підпростір простору $\mathbb{F}_q^n$, отже, лінійний $(n, n-r)$-код, який теж називають *кодом з перевіркою на парність*.

4. Розглянемо один частковий випадок коду з перевіркою на парність. Нехай маємо лінійне відображення $\varphi:\ \mathbb{F}_2^3 \to \mathbb{F}_2^7$,

$$\varphi(a_1, a_2, a_3) = (a_1, a_2, a_3, a_1 + a_3, a_2 + a_3, a_1 + a_2, a_1 + a_2 + a_3).$$

Образ лінійного відображення φ є бінарним лінійним $(7,3)$-кодом. Цей код можна розглядати як загальний розв'язок однорідної системи лінійних рівнянь

$$\begin{cases} x_1 + x_3 + x_4 &= 0, \\ x_2 + x_3 + x_5 &= 0, \\ x_2 + x_3 + x_6 &= 0, \\ x_1 + x_2 + x_3 + x_7 &= 0 \end{cases}$$

з матрицею

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

З іншого боку, відображенню φ можна поставити у відповідність 3×7-матрицю G, рядками якої є образи одиничних векторів простору $\mathbb{F}_2^3$

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Кожному лінійному (n,k)-коду ставлять у відповідність дві матриці G і H так як це було зроблено у попередньому прикладі.

Означення 4.16. *Нехай C — лінійний (n,k)-код (тобто k-вимірний підпростір n-вимірного лінійного простору $\mathbb{F}_q^n$ над скінченним полем $\mathbb{F}_q$.) Нехай $\bar{e}_1,\dots,\bar{e}_k$ — база підпростору C. Матриця $G\in M_{k,n}(\mathbb{F}_q)$, рядками якої є вектори $\bar{e}_1,\dots,\bar{e}_k$, називається породжуючою матрицею коду C.*

Нехай G — породжуюча матриця коду C, нехай $\bar{x}=(x_1,\dots,x_n)^t$ (верхній індекс t тут і далі означає транспонування) — стовпчик невідомих. Розглянемо систему лінійних рівнянь

$$G\bar{x}^t=\bar{0}^t. \tag{4.15}$$

Система (4.15) має ранг k, тому її фундаментальна система розв'язків складається з $n-k$ розв'язків. Матриця $H\in M_{n-k,n}(\mathbb{F}_q)$, рядками якої є фундаментальні розв'язки системи (4.15), називається контрольною матрицею коду C.

Зауважимо, що матриці G і H визначаються кодом C неоднозначно.

Приклади

1. Породжуючою матрицею коду з повтореннями

$$\varphi:\ \mathbb{F}_2^3\to\mathbb{F}_2^9,\ a_1a_2a_3\mapsto a_1a_1a_1a_2a_2a_2a_3a_3a_3$$

є 3×9-матриця

$$G=\begin{pmatrix}1&1&1&0&0&0&0&0&0\\0&0&0&1&1&1&0&0&0\\0&0&0&0&0&0&1&1&1\end{pmatrix}.$$

Контрольною матрицею H цього коду є матриця системи лінійних рівнянь

$$x_1=x_2=x_3,\ x_4=x_5=x_6,\ x_7=x_8=x_9,,$$

$$G=\begin{pmatrix}1&1&0&0&0&0&0&0&0\\1&0&1&0&0&0&0&0&0\\0&0&0&1&1&0&0&0&0\\0&0&0&1&0&1&0&0&0\\0&0&0&0&0&0&1&1&0\\0&0&0&0&0&0&1&0&1\end{pmatrix}.$$

2. Нехай C — лінійний $(n,n-1)$-код з перевіркою на парність. Його елементи є розв'язками рівняння $x_1+x_2+\dots+x_n=0$. Контрольна матриця цього коду — матриця $H=(1\ 1\ \dots\ 1)$, а $(n-1)\times n$-матриця

$$G=\begin{pmatrix}1&0&0&\cdots&0&1\\0&1&0&\cdots&0&1\\\cdots&\cdots&\cdots&\cdots&\cdots&1\\0&0&0&\cdots&1&1\end{pmatrix}$$

— породжуюча матриця цього коду.

3. Матриці

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ та } G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

є, відповідно, контрольною та породжуючою матрицею лінійного $(7, 3)$-коду з прикладу 4 у попередньому параграфі.

З означення контрольної та породжуючої матриць випливають такі властивості цих матриць.

Твердження 4.6. *Нехай G і H породжуюча та контрольна матриці (n, k)-коду C.*

1. GH^t і HG^t — нульові матриці відповідно k-го та $(n-k)$-го порядку.

2. $\bar{c} \in C \iff H\bar{c}^t = \bar{0}^t$.

3. $\bar{c} \in C \iff \exists \bar{a} \in \mathbb{F}_q^k \quad \bar{c} = \bar{a}G$.

Доведення. 1. безпосередньо випливає з означень.

2. Зрозуміло, що $\bar{c} \in C$, тоді й лише тоді, коли $\bar{c} = \sum_{i=1}^{k} \alpha_i \bar{g}_i$ є лінійною комбінацією рядків $\bar{g}_1, \ldots, \bar{g}_k$ матриці G. Позаяк HG^t — нульова матриця, то, зокрема, Hg_i^t — нульовий вектор-стовпчик для кожного $i, 1 \le i \le k$. Якщо $\bar{c} \in C$, то $H\bar{c}^t = H(\sum_{i=1}^{k} \alpha_i \bar{g}_i^t) = \sum_{i=1}^{k} \alpha_i H\bar{g}_i^t = \bar{0}$. Навпаки, якщо $H\bar{c}^t = \bar{0}^t$, то вектор $\bar{c}^t$ є розв'язком системи лінійних рівнянь $H\bar{x}^t = \bar{0}$. Позаяк HG^t — нульова матриця, ранг матриці H дорівнює $n-k$ і стовпчики матриці G^t лінійно незалежні, то ці стовпчики $\bar{g}_1, \ldots, \bar{g}_k$ утворюють фундаментальну систему розв'язків системи рівнянь $H\bar{x}^t = \bar{0}$. Тому $\bar{c}$ є лінійною комбінацією векторів $\bar{g}_1, \ldots, \bar{g}_k$, отже, $\bar{c} \in C$.

3. $\Longrightarrow$. Нехай $\bar{c} \in C$. Тоді $\bar{c} = \sum_{i=1}^{k} \alpha_i \bar{g}_i = (\alpha_1, \ldots, \alpha_k)(\bar{g}_1, \ldots, \bar{g}_k)^t = \bar{a}G$, де $\bar{a} = (\alpha_1, \ldots, \alpha_k)$.

$\Longleftarrow$. Якщо $\bar{c} = \bar{a}G$, де $\bar{a} \in \mathbb{F}_q^k$, то $H\bar{c}^t = H(\bar{a}G)^t = HG^t\bar{a}^t = \bar{0}^t$, бо за 1 HG^t є нульовою матрицею. Тому $\bar{c} \in C$ згідно з 2.

З доведеного твердження, зокрема, випливає, що для кодування слова $\bar{a} \in \mathbb{F}_q^k$ його потрібно домножити на породжуючу матрицю G і одержати кодове слово $\bar{a}G$; для перевірки чи належить слово $\bar{c} \in \mathbb{F}_q^n$ коду C потрібно перевірити чи $H\bar{c}^t = 0$.

Вибравши матрицю G так, щоб її перші k стовпчиків становили одиничну матрицю k-го порядку, для декодування потрібно лише відкинути останні $n-k$ координат кодового вектора $\bar{c}$.

Означення 4.17. *Нехай C — лінійний код, $\bar{c} \in C$. Вагою Гемінга $w(\bar{c})$ слова $\bar{c}$ називають кількість ненульових координат вектора $\bar{c}$. Відстанню Гемінга $d(\bar{c}_1, \bar{c}_2)$ між словами $c_1, c_2 \in C$ називають вагу Гемінга $w(\bar{c}_1 - \bar{c}_2)$ слова $\bar{c}_1 - \bar{c}_2$.*

Наприклад, якщо $\bar{c}_1 = 101001, \bar{c}_2 = 100100$, то $w(\bar{c}_1) = 3$, $w(\bar{c}_2) = 2$, $d(\bar{c}_1, \bar{c}_2) = w(001101) = 3$.

Виявляється, що відстань Гемінга має такі самі властивості, як і звичайна відстань.

Твердження 4.7. *Нехай C — лінійний код, d — відстань Гемінга на C, $\bar{x}, \bar{y}, \bar{z} \in C$. Тоді:*

1) $d(\bar{x}, \bar{y}) \geq 0$ і $d(\bar{x}, \bar{y}) = 0 \iff \bar{x} = \bar{y}$;

2) $d(\bar{x}, \bar{y}) = d(\bar{y}, \bar{x})$;

3) $d(\bar{x}, \bar{z}) \leq d(\bar{x}, \bar{y}) + d(\bar{y}, \bar{z})$.

Доведення. Властивості 1 і 2 очевидні. Для доведення властивості 3 нам доведеться використати таку очевидну нерівність для ваги Гемінга: якщо $\bar{a}, \bar{b} \in \mathbb{F}_q^n$, то

$$w(\bar{a} + \bar{b}) \leq w(\bar{a}) + w(\bar{b}).$$

Використовуючи цю нерівність, одержуємо $d(\bar{x}, \bar{z}) = w(\bar{x} - \bar{z}) = w(\bar{x} - \bar{y} + \bar{y} - \bar{z}) \leq w(\bar{x} - \bar{y}) + w(\bar{y} - \bar{z}) = d(\bar{x}, \bar{y}) + d(\bar{y}, \bar{z})$.

Означення 4.18. *Нехай $C \subset \mathbb{F}_q^n$ — лінійний код, r – натуральне число. Код C, називають кодом, який виправляє k помилок, якщо для кожного $\bar{b} \in \mathbb{F}_q^n$ знайдеться не більше, ніж одне слово $\bar{c} \in C$, для якого $d(\bar{c}, \bar{b}) \leq r$.*

Означення 4.19. *Мінімальну відстань між різними словами лінійного коду C називають кодовою відстанню і позначають d_C.*

Теорема 4.12. *Якщо $d_C \geq 2r + 1$, то код C виправляє r помилок.*

Доведення. Скористаємось міркуваннями від супротивного. Якби відстані від деякого вектора $\bar{x} \in \mathbb{F}_q^n$ до двох різних кодових слів були менші від r, то ми одержали б

$$2r + 1 \leq d_C \leq d(\bar{c}_1, \bar{c}_2) \leq d(\bar{c}_1, \bar{x}) + d(\bar{x}, \bar{c}_2) \leq 2r.$$

Одержано суперечність. Тому код C виправляє r помилок.

Отже, побудова кодів, які виправляють багато помилок, зводиться до побудови кодів з великою кодовою відстанню. Наступна теорема дає простий метод оцінки знизу для кодової відстані d_C.

Теорема 4.13. *Нехай d_C — кодова відстань лінійного коду C, H — його контрольна матриця. $d_C \geq s+1$ тоді й лише тоді, коли кожні s стовпчиків матриці H лінійно незалежні.*

Доведення. $\Longrightarrow$. Припустимо, що $d_C \geq s+1$. Нехай $\bar{h}_1, \ldots, \bar{h}_n$ — стовпчики матриці H. Нехай, від супротивного, існують s лінійно залежних стовпчиків цієї матриці. Не зменшуючи загальності, можна вважати, що це перші s стовпчиків $\bar{h}_1, \ldots, \bar{h}_s$ матриці H. Існують елементи $\gamma_1, \ldots, \gamma_s \in \mathbb{F}_q$, для яких

$$\gamma_1 \bar{h}_1 + \ldots + \gamma_s \bar{h}_s = \bar{0}. \tag{4.16}$$

Розглянемо вектор $\bar{c} = (\gamma_1, \ldots, \gamma_s, 0, \ldots, 0)$. Враховуючи (4.16), одержуємо

$$H\bar{c}^t = (\gamma_1 \bar{h}_1 + \ldots + \gamma_s \bar{h}_s + 0 \cdot \bar{h}_{s+1} + \cdots + 0 \cdot \bar{h}_n) = \bar{0}^t.$$

Твердження 4.6 тепер показує, що $\bar{c}^t \in C$. З іншого боку, $d(\bar{c}, \bar{0}) \leq s$, тому $d_C \leq s$.

$\Longleftarrow$. Припустимо, що кожні s стовпчиків матриці H лінійно незалежні. Знову міркуємо від супротивного. Якщо $d_C \leq s$, то знайдеться ненульовий вектор $\bar{c} \in C$, для якого $d(\bar{c}, \bar{0}) \leq s$, тобто вектор $\bar{c} \in C$ має не більше ніж s ненульових координат. Тоді рівність $H\bar{c}^t = \bar{0}$ означає, що деякі $\leq s$ стовпчиків матриці H є лінійно залежні. Одержана суперечність завершує доведення теореми.

4.2.3. Декодування лінійних кодів

Нехай C — лінійний (n,k)-код над полем $\mathbb{F}_q$. Розглянемо факторпростір $\mathbb{F}_q^n / C$. Його елементами є суміжні класи $\bar{a} + C$, де $\bar{a} \in \mathbb{F}_q^n$. Кожний суміжний клас $\bar{a} + C$ складається з q^k елементів. Лінійний простір $\mathbb{F}_q^n$ є об'єднанням $s = q^{n-k}$ різних суміжних класів

$$\mathbb{F}_q^n = C \cup (a_1 + C) \cup \cdots \cup (a_{s-1} + C).$$

Якщо на виході з каналу зв'язку прийнято повідомлення $\bar{x}$, то $\bar{x}$ лежить в одному з суміжних класів, нехай в $a_i + C$. Якщо передали слово $\bar{c}$, то

вектор $\bar{e} = \bar{x} - \bar{c}$ називають *вектором помилок*. Вектор помилок лежить у тому ж суміжному класі, що й $\bar{x}$. Найімовірнішим значенням вектора помилок є вектор з найменшою вагою Гемінга. Вектор з суміжного класу $a_i + C$ з найменшою вагою Гемінга називають *лідером суміжного класу*. Вектор $S(\bar{x}) = (H\bar{x}^t)^t$ називають *синдромом* вектора $\bar{x}$. Очевидно, що всі вектори з одного суміжного класу мають однаковий синдром.

Тепер можемо сформулювати алгоритм декодування.

Алгоритм декодування за лідером суміжного класу. Нехай C — лінійний (n,k)-код, $\bar{x}$ — прийнятий на виході з каналу зв'язку вектор.

1. Обчислюємо синдром $S(\bar{x})$.
2. Знаходимо такий лідер $\bar{e}$ суміжного класу, для якого $S(\bar{e}) = S(\bar{x})$.
3. Декодуємо $\bar{x}$ як $\bar{c} = \bar{x} - \bar{e}$.

Приклади

Розглянемо код C з контрольною матрицею

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Матриця H має ранг 3, тому C є $(5,2)$-кодом. Цей код складається з чотирьох кодових слів: 00000, 11100, 01111, 10011.

Лінійний простір $\mathbb{F}_2^5$ має 32 елементи, тому тут маємо 8 суміжних класів. Випишемо ці суміжні класи, їхні синдроми та лідери у таблиці.

Суміжні класи				Синдром	Лідер
00000	11100	01111	10011	000	00000
10000	01100	11111	00011	100	10000
01000	10100	10111	11011	110	01000
00100	11000	01011	10111	010	00100
00010	11110	01101	10001	101	00010
00001	11101	01110	10010	001	00001
00110	11010	01001	10101	111	00110 або 01001
01010	10110	00101	11001	011	01010 або 00101

Розглянутий код C має кодову відстань $d_c = 3$. Він виправляє одну помилку і виявляє дві помилки.

Проілюструємо на цьому прикладі алгоритм декодування за лідером суміжного класу. Нехай на виході з каналу зв'язку ми одержали слово $\bar{x} = 01001$. Спочатку обчислюємо синдром

$$S(\bar{x}) = (H\bar{x}^t)^t = \bar{x}H^t = (01001)\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = (111).$$

Бачимо, що слово 01001 перебуває у суміжному класі з лідерами 00110 або 01001. Тому робимо висновок, що передавалося одне з двох слів: $01001 + 00110 = 01111$ або $01001 + 01001 = 00000$. У кожному з цих випадків при передачі виникло дві помилки. Отже, ми виявили наявність в одержаному слові двох помилок.

Припустимо, що одержане слово 11000. Його синдром дорівнює

$$S(\bar{x}) = (11000)\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = (010),$$

тому воно знаходиться у суміжному класі з лідером 00100. Робимо висновок, що передавалося слово $11000 + 00100 = 11100$. У цьому випадку виникла одна помилка і код виправив її.

4.2.4. Коди Гемінга

Означення 4.20. *Бінарний код C_k довжини $n = 2^k - 1$, $k \geq 2$ з контрольною $k \times n$-матрицею H, називають бінарним кодом Гемінга, якщо стовпчики матриці H зображають двійкові записи чисел $1, 2, \ldots, 2^k - 1$.*

Приклад

Бінарний код Гемінга C_2 має контрольну матрицю

$$H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix},$$

а контрольною матрицею бінарного коду Гемінга C_3 є матриця

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Твердження 4.8. *Бінарний код Гемінга C_k має розмірність $2^k - k - 1$ і виправляє одну помилку.*

Доведення. Для бінарного коду Гемінга C_k ранг матриці H дорівнює k, бо H містить k одиничних стовпчиків. Тому розмірність коду C_k дорівнює $n - k = 2^k - k - 1$. Стовпчики матриці H всі різні, тому кожні два серед них лінійно незалежні. За теоремою 4.13 кодова відстань коду C_k не менша ніж 3 (насправді, кодова відстань дорівнює 3, бо кожний стовпчик матриці H дорівнює сумі двох інших), а тому за теоремою 4.12 цей код виправляє одну помилку.

Код Гемінга дуже просто виправляє одну помилку. Якщо при передачі виникла одна помилка (наприклад на i-ому місці), то одержане слово $\bar{x}$ відрізняється від переданого слова $\bar{c}$ доданком: $\bar{x} = \bar{c} + \bar{e}_i$, де $\bar{e}_i$ — слово з 1 на i-му місці і з нулями на всіх інших місцях. Обчислюємо синдром $S(\bar{x})$: $S(\bar{x}) = S(\bar{e}_i) = \bar{h}_i^t$, де $\bar{h}_i$ — i-й стовпчик матриці H. Згадавши, що i-й стовпчик матриці H є двійковим записом натурального числа i, бачимо, що синдром $S(\bar{x})$ є номером позиції слова $\bar{x}$, де потрібно виправити помилку (замінити 0 на 1 або 1 на 0).

Приклад

Розглянемо бінарний код Гемінга C_3 з матрицею

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Припустимо, що одержане слово $\bar{x} = 1001111$. Його синдром дорівнює

$$S(\bar{x}) = \bar{x}H^t = (1001111) \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} = (001).$$

Робимо висновок, що помилка трапилася на першому місці, а переданим було слово 0001111.

4.2.5. Циклічні коди

Нехай $\mathbb{F}_q$ — скінченне поле.

Означення 4.21. *Лінійний код $C \subset \mathbb{F}_q^n$ називають циклічним, якщо*

$$(a_1, \ldots, a_{n-1}, a_n) \in C \Longrightarrow (a_n, a_1, \ldots, a_{n-1}) \in C.$$

Вивчення циклічних кодів тісно пов'язане з вивченням факторкілець кільця поліномів над скінченним полем та їх ідеалів. Нехай $(X^n - 1) = (X^n - 1)\mathbb{F}_q[X]$ — головний ідеал у кільці поліномів $\mathbb{F}_q[X]$, породжений поліномом $X^n - 1$. Позначимо факторкільце $\mathbb{F}_q[X]/(X^n - 1)$ через $\mathbb{F}_q[\overline{X}] = \mathbb{F}_q[x]$ (тут і далі $x = \overline{X}$). Це n-вимірний лінійний простір над полем $\mathbb{F}_q$ з базою $\bar{1} = 1 + (X^n - 1)$, $x = X + (X^n - 1), \ldots, x^{n-1} = X^{n-1} +$

(X^n-1), елементами якого є суміжні класи $a_0+a_1x+\cdots+a_{n-1}x^{n-1}=\overline{a_0+a_1X+\cdots+a_{n-1}X^{n-1}}$. Відображення

$$\varphi:\mathbb{F}_q[x]\to\mathbb{F}_q^n,\quad \varphi(a_0+a_1x+\cdots+a_{n-1}x^{n-1}=(a_0,a_1,\ldots,a_{n-1})$$

є ізоморфізмом лінійних просторів. Цей ізоморфізм дає змогу ототожнювати вектори $(a_0,\ldots,a_n)\in\mathbb{F}_q^n$ та елементи $a_0+a_1x+\cdots+a_{n-1}x^{n-1}$ факторкільця $\mathbb{F}_q[x]$. Отже, можемо вважати, що лінійний код C є підпростором лінійного простору $\mathbb{F}_q[x]$, тоді елементи коду C є поліномами; називатимемо їх *кодовими поліномами*.

Враховуючи зроблені домовленості, маємо таку теорему.

Теорема 4.14. *Код C — циклічний тоді й лише тоді, коли C є ідеалом факторкільця $\mathbb{F}_q[x]$.*

Доведення. Нехай $a_0a_1\ldots a_{n-1}$ кодове слово циклічного коду. Ми домовилися ототожнювати його з поліномом $f(x)=a_0+a_1x+\cdots+a_{n-1}x^{n-1}$. Домножимо поліном $f(x)$ на x і поділимо одержаний добуток на x^n-1

$$\begin{aligned}&x(a_0+a_1x+\cdots+a_{n-1}x^{n-1})=a_0x+\ldots+a_{n-2}x^{n-1}+a_{n-1}x^n=\\&=a_{n-1}(x^n-1)+a_{n-1}+a_0x+\ldots+a_{n-2}x^{n-1}=\\&=a_{n-1}+a_0x+\ldots+a_{n-2}x^{n-1}.\end{aligned}\tag{4.17}$$

Позаяк код C циклічний, то поліном (4.17) належить до C. Ми бачимо, що добуток $xf(x)$ кодового полінома $f(x)$ на x є знову кодовим поліномом. Повторюючи щойно зроблене обчислення k разів, одержуємо $x^kf(x)\in C$. Код C є лінійним підпростором, тому для кожного $\alpha\in\mathbb{F}_q$ і кожного $f(x)\in C$ добуток $\alpha f(x)$ лежить у C. Сума двох елементів з C теж належить до C. Звідси випливає, що добуток довільного полінома з $\mathbb{F}_q[x]$ на довільний поліном з C належить до C, тобто C є ідеалом.

Припустимо, що код C є ідеалом. Нехай $f(x)=a_0+a_1x+\cdots+a_{n-1}x^{n-1}\in C$. Тоді, зокрема, $xf(x)=a_{n-1}+a_0x+\cdots+a_{n-2}x^{n-1}\in C$. Записуючи елементи коду C у вигляді векторів, це означає, що

$$(a_0,a_1,\ldots,a_{n-1})\in C\Longrightarrow(a_{n-1},a_0,\ldots,a_{n-2})\in C,$$

що й треба було довести.

Отже, вивчення циклічних кодів зводиться до вивчення ідеалів факторкільця $\mathbb{F}_q[x]$.

Твердження 4.9. *Факторкільце $\mathbb{F}_q[x]$ є кільцем головних ідеалів. Якщо $I = (g(x))$, ідеал в $\mathbb{F}_q[x]$, то $g(X)|(X^n - 1)$.*

Доведення. Відомо, що в кільці поліномів від однієї змінної над довільним полем кожен ідеал є головним. Зокрема, кільце $\mathbb{F}_q[X]$ є кільцем головних ідеалів. Розглянемо канонічний гомоморфізм

$$\varphi : \mathbb{F}_q[X] \to \mathbb{F}_q[x], \quad \varphi(f(X)) = \overline{f(X)} = f(x).$$

Нехай I — ідеал в $\mathbb{F}_q[x]$. Легко перевірити, що його прообраз $I' = \varphi^{-1}(I)$ є ідеалом кільця $\mathbb{F}_q[X]$. Оскільки всі ідеали в $\mathbb{F}_q[X]$ головні, то існує поліном $g(X) \in \mathbb{F}_q[X]$ такий, що $I' = (g(X))$. Тоді $I = (g(x))$. Далі, $\varphi(X^n - 1) = \bar{0} \in I$, тому $g(X)|(X^n - 1)$.

Означення 4.22. *Нехай $C = (g(x))$ — циклічний код, породжений поліномом $g(x)$. Поліном $g(X)$ називають породжуючим поліномом коду C, а поліном $h(X) = \frac{X^n-1}{g(X)}$ називають контрольним поліномом цього коду.*

Твердження 4.10. *Якщо поліном $g(X)$ має степінь k, то код $(g(x))$ має вимірність $n - k$.*

Доведення. Елементи $g(x), xg(x), \ldots, x^{n-k-1}g(x) \in C$ лінійно незалежні. Справді, якби існували $\alpha_0, \ldots, \alpha_{n-k-1}$, для яких $\alpha_0 g(x) + \ldots + \alpha_{n-k-1}x^{n-k-1}g(x) = \bar{0}$, тобто $(\alpha_0 + \ldots + \alpha_{n-k-1}x^{n-k-1})g(x) = \bar{0}$, то це означало б, що поліном $(\alpha_0 + \ldots + \alpha_{n-k-1}X^{n-k-1})g(X)$ степеня $< n$ ділиться на $X^n - 1$. З іншого боку, нехай $f(x) \in C$. Тоді $f(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$ і $f(x) = g(x)d(x)$. Звідси випливає, що елементи $g(x), xg(x), \ldots, x^{n-k-1}g(x) \in C$ утворюють систему твірних підпростору C.

Кодування та декодування для циклічних кодів дуже прості. Для того щоб закодувати поліном $a(x) = a_0 + \ldots + a_{n-k-1}x^{n-k-1}$, потрібно його домножити на породжуючий поліном $g(x)$, а для декодування треба ділити на $g(x)$.

Іноді буває зручніше зображати елементи циклічного коду векторами, а не поліномами. Маючи породжуючий та контрольний поліноми

$$g(x) = g_0 + g_1x + \cdots + g_kx^k$$

та

$$h(x) = h_0 + h_1 x + \cdots + h_{n-k} x^{n-k},$$

можемо записати породжуючу та контрольну матриці G та H:

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_k & 0 & \cdots & 0 & 0 \\ 0 & g_0 & g_1 & \cdots & g_{k-1} & g_k & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & g_0 & g_1 & \cdots & g_{k-1} & g_k \end{pmatrix} \tag{4.18}$$

та

$$H = \begin{pmatrix} 0 & \cdots & 0 & h_{n-k} & h_{n-k-1} & \cdots & h_1 & h_0 \\ 0 & \cdots & h_{n-k} & h_{n-k-1} & \cdots & \cdots & h_0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ h_{n-k} & \cdots & h_0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{4.19}$$

Той факт, що матриця G є породжуючою матрицею циклічного коду з породжуючим поліномом $g(x)$ випливає з того факту, що поліноми $g(x), xg(x), \ldots, x^{k-1}g(x)$ лінійно незалежні. H є контрольною матрицею цього коду. Справді, з одного боку, добуток будь-якого рядка матриці G на будь-який рядок матриці H є сумою добутків вигляду $\sum_{r+s=k} g_r h_s$, тобто це коефіцієнт добутку поліномів $g(x)h(x)$. Позаяк $g(x)h(x) = x^n - 1 = 0$ в $\mathbb{F}_q[x] = \mathbb{F}_q[X]/(X^n - 1)$, то добуток GH^t є нульовою матрицею. З іншого боку, очевидно, рядки матриці H лінійно незалежні і вона має потрібну кількість (а саме, $n - k$) рядків.

Породжуючий і контрольний поліноми циклічного коду є дільниками полінома $X^n - 1 \in \mathbb{F}_q$. Тому при побудові циклічних кодів важливого значення набуває задача розкладу поліномів $X^n - 1 \in \mathbb{F}_q[X]$ (а також довільних поліномів з коефіцієнтами зі скінченного поля) на множники. Якщо ми маємо поліном $f(X)$ степеня n з коефіцієнтами з довільного скінченного поля $\mathbb{F}_q$, то існує лише скінченна кількість поліномів $d(X) \in \mathbb{F}_q$ степеня не більшого, ніж n. Тому задачу розкладу полінома $f(X)$ на множники можна розв'язати методом перебору можливих випадків. Варто зауважити, що існує q^{n+1} поліномів степеня не більшого, ніж n над скінченним полем з q елементів. Тому при великих q або великих n метод перебору стає важкою задачею навіть для сучасних комп'ютерів. У зв'язку з цим розроблено різноманітні методи розкладу поліномів над скінченними полями на множники, які використовують (іноді досить глибокі) властивості скінченних полів і дають

змогу звести до мінімуму безпосередній перебір. На основі цих методів створюють пакети прикладних програм, за допомогою яких такі задачі легко розв'язати, використовуючи навіть не дуже потужні комп'ютери. Одним з таких пакетів є система Maple, яка поєднує символьні та числові методи розв'язування задач практично з усіх розділів математики. Ця система містить, зокрема, пакет GF (поля Галуа), який допомагає розв'язувати різноманітні задачі, пов'язані зі скінченними полями, в тому числі і задачу розкладу поліномів на множники. Наприклад, нехай нам потрібно розкласти на незвідні множники поліном $X^{31}-1 \in \mathbb{F}_2[X]$. Для цього достатньо набрати (у системі Maple V) команду

[> `Factor (x^31-1) mod 2;`

натиснути клавішу «Enter» і миттєво одержати розклад полінома $x^{31}-1 \in \mathbb{F}_2[x]$: на незвідні множники:

$(x^5+x^4+x^3+x^2+1)(x^5+x^3+1)(x^5+x^4+x^2+x+1)(x+1)(x^5+x^3+x^2+x+1)(x^5+x^2+1)(x^5+x^4+x^3+x+1)$.

Аналогічно, для полінома $X^{127}-1 \in \mathbb{F}_2[X]$ так само миттєво одержуємо

[> `Factor (x^127-1) mod 2;`

«Enter»

$(x^7+x^6+x^4+x^2+1)\,(x^7+x^6+x^5+x^4+x^3+x^2+1)\,(x^7+x^6+x^5+x^4+1)$
$(x^7+x^6+x^5+x^4+x^2+x+1)\,(x^7+x^6+x^4+x+1)\,(x+1)\,(x^7+x^3+1)\,(x^7+x+1)$
$(x^7+x^6+x^5+x^2+1)\,(x^7+x^4+x^3+x^2+1)\,(x^7+x^5+x^2+x+1)$
$(x^7+x^6+x^5+x^3+x^2+x+1)\,(x^7+x^5+x^4+x^3+x^2+x+1)\,(x^7+x^6+x^3+x+1)$
$(x^7+x^5+x^4+x^3+1)\,(x^7+x^3+x^2+x+1)\,(x^7+x^6+1)\,(x^7+x^5+x^3+x+1)$
(x^7+x^4+1).

Незважаючи на існування пакета GF системи Maple, студенти повинні вміти вручну розкладати на незвідні множники поліноми невеликих степенів над скінченними полями, з невеликою кількістю елементів.

Приклади

Розкладемо на множники поліном $X^7-1 \in \mathbb{F}_2[X]$. Зрозуміло, що він ділиться на $X+1$ ($X+1 = X-1$ в $\mathbb{F}_2[X]$) і

$$X^7-1 = (X+1)(X^6+X^5+X^4+X^3+X^2+X+1).$$

Інших незвідних поліномів першого степеня в кільці $\mathbb{F}_2[X]$ немає. У цьому кільці існує єдиний незвідний поліном X^2+X+1 другого степеня, але спробувавши поділити X^7-1 на X^2+X+1, доходимо висновку, що X^2+X+1 не є дільником. Тому треба розглянути незвідні поліноми третього степеня. Поліном третього степеня є незвідним тоді й лише тоді, коли 0 і 1 не є його коренями. Зокрема, поліном X^3+X+1

незвідний. Знову спробувавши поділити, одержуємо

$$(X^6+X^5+X^4+X^3+X^2+X+1)=(X^3+X+1)(X^3+X^2+1),$$

тому остаточно

$$X^7-1=(X+1)(X^3+X+1)(X^3+X^2+1).$$

Поліном $g(x)=x^3+x^2+1\in\mathbb{F}_2[X]/(X^7-1)$ (нагадаємо, що $x=X+(X^7-1)\mathbb{F}_2[X]$ — суміжний клас з представником X) є породжуючим поліномом циклічного $(7,4)$-коду з контрольним поліномом $h(x)=(x+1)(x^3+x+1)=x^4+x^3+x^2+1$. Запишемо породжуючу на контрольну матриці цього коду.

$$G=\begin{pmatrix}1&0&1&1&0&0&0\\0&1&0&1&1&0&0\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{pmatrix},\quad H=\begin{pmatrix}0&0&1&1&1&0&1\\0&1&1&1&0&1&0\\1&1&1&0&1&0&0\end{pmatrix}.$$

З означень зрозуміло, що породжуючу на контрольну матриці G та H лінійного (n,k)-коду можна вибрати багатьма різними способами. Зокрема, їх можна вибрати так, щоб перші k стовпчиків матриці G (останні $n-k$ стовпчиків матриці H) були одиничними векторами. Це робиться за допомогою відповідних виборів фундаментальних систем розв'язків систем лінійних однорідних рівнянь з матрицями H та G відповідно. А саме, розглянемо систему рівнянь

$$\begin{pmatrix}0&0&1&1&1&0&1\\0&1&1&1&0&1&0\\1&1&1&0&1&0&0\end{pmatrix}(x_1,x_2,x_3,x_4,x_5,x_6.x_7)^t=(0,0,0,0,0,0,0)^t.$$

Вона має такі чотири фундаментальні розв'язки:

$$\bar g_1=(1,0,0,0,1,0,1),\qquad \bar g_2=(0,1,0,0,1,1,1),$$
$$\bar g_3=(0,0,1,0,1,1,0),\qquad \bar g_4=(0,0,0,1,0,1,1).$$

Аналогічно, розглянувши систему рівнянь

$$\begin{pmatrix}1&0&1&1&0&0&0\\0&1&0&1&1&0&0\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{pmatrix}(x_1,x_2,x_3,x_4,x_5,x_6.x_7)^t=(0,0,0,0,0,0,0)^t,$$

знайдемо три фундаментальні розв'язки

$$\bar h_1=(1,1,1,0,1,0,0),\quad \bar h_2=(0,1,1,1,0,1,0),\quad \bar h_3=(1,1,0,1,0,0,1).$$

Складаємо матриці

$$G'=\begin{pmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{pmatrix},\quad H'=\begin{pmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\1&1&0&1&0&0&1\end{pmatrix},$$

рядками яких є відповідно вектори $\bar g_1,\bar g_2,\bar g_3,\bar g_4$ та $\bar h_1,\bar h_2,\bar h_3$.

Означення 4.23. *Якщо перші k стовпчиків породжуючої матриці G лінійного (n,k)-коду утворюють одиничну матрицю k-го порядку, то кажуть, що матриця G є канонічною породжуючою матрицею. Якщо останні $n-k$ стовпчиків контрольної матриці H лінійного (n,k)-коду утворюють одиничну матрицю $(n-k)$-го порядку, то кажуть, що матриця H є канонічною контрольною матрицею.*

Твердження 4.11. *Канонічна породжуюча та канонічна контрольна матриці лінійного коду C однозначно визначаються кодом C.*

Пропонуємо довести це твердження самостійно.

Циклічні коди можна описувати за допомогою задання коренів породжуючого полінома.

Нехай p — просте число, $q=p^s$, $\mathbb{F}_q$ — скінченне поле з q елементів, і нехай $\mathbb{F}_{q^m}$ — його скінченне розширення степеня m. Якщо $p(X)\in\mathbb{F}_q$ — незвідний поліном степеня r, то ми знаємо (див. твердження 4.5), що факторкільце $\mathbb{F}_q[X]/(p(X)$ є полем з q^r елементів. Поліном $p(X)$ називатимемо *примітивним поліномом* для поля $\mathbb{F}_{q^m}$ над полем $\mathbb{F}_q$, а будь-який корінь α полінома $p(X)$ називатимемо *примітивним елементом* для $\mathbb{F}_{q^m}$ над $\mathbb{F}_q$.

Зрозуміло, що для $n=q^r-1$ і для довільного $\beta\in\mathbb{F}_{q^m}$ з наслідку до теореми Лагранжа (порядок елемента скінченної групи ділить порядок групи) випливає рівність $\beta^n=1$. Нехай $\alpha_1,\ldots,\alpha_s\in\mathbb{F}_{q^m}$, $m_i(X)$ – мінімальні поліноми елементів α_1, $1\le i\le s$, тобто $m_i(X)$ — поліном найменшого степеня серед поліномів з коренем α_i і зі старшим коефіцієнтом 1. Нехай $g(X)$ — найменше спільне кратне поліномів $m_i(X)$. Тоді $g(X)|(X^n-1)$. Якщо $g(x)$ породжує циклічний код C, то легко зрозуміти, що поліном $c(x)\in\mathbb{F}_q[x]$ є елементом коду C тоді й лише тоді, коли $c(\alpha_i)=0$ для всіх i, $1\le i\le s$.

Нагадаємо позначення: $\mathbb{F}_q[x]=\mathbb{F}_q[X]/(X^n-1)$.

Теорема 4.15. *Нехай $C\subset\mathbb{F}_q[x]$ — циклічний (n,k)-код з породжуючим поліномом $g(x)$, $\alpha_1,\ldots,\alpha_{n-k}$ — корені полінома $g(X)$. Поліном $c(x)=c_0+c_1x+\ldots+c_{n-1}x^{n-1}$ є кодовим поліномом тоді й лише тоді, коли для вектора $\bar c=(c_0,\ldots,c_{n-1})$ виконується рівність*

$$\begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-1}\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 1 & \alpha_{n-k} & \alpha_{n-k}^2 & \cdots & \alpha_{n-k}^{n-1}\end{pmatrix}(c_0,\ldots,c_{n-1})^t=(0,\ldots,0)^t. \qquad (4.20)$$

Доведення. Згідно з попередніми зауваженнями $c(x) \in C$ тоді й лише тоді, коли $c(\alpha_i) = 0$ для всіх i, $1 \le i \le n-k$, тобто $c_0+c_1\alpha_i+\ldots+c_{n-1}\alpha_i^{n-1}$ для всіх i, $1 \le i \le n-k$, тобто тоді й лише тоді, коли виконується матрична рівність (4.20).

Твердження 4.12. *Нехай елемент $\alpha \in \mathbb{F}_{q^m}$ породжує мультиплікативну групу $\mathbb{F}^*_{q^m}$. Нехай мінімальний поліном елемента α є породжуючим поліномом циклічного коду C.*

1. Поліном $c(x) \in \mathbb{F}_q[x]$ є елементом коду C тоді й лише тоді, коли $c(\alpha) = 0$.

2. Нехай $q = 2$, $c(x) \in C$, нехай при передачі слова $c(x)$ трапилась одна помилка, тобто прийнятим словом є $c'(x) = c(x) + x^{j-1}$ для деякого j, $1 \le j \le n$. Тоді $c'(\alpha) = \alpha^{j-1}$ і це дає змогу виправити помилку.

Доведення. 1. Зрозуміло, що $c(x) \in C \iff c(X)$ ділиться на $g(X)$. Використовуючи незвідність полінома $g(X)$, одержуємо: $c(X)$ ділиться на $g(X) \iff c(\alpha) = 0$.

2. Якщо прийняте слово має вигляд $c'(x) = c(x) + x^{j-1}$, де $c(x) \in C$, то за доведеним $c(\alpha) = 0$; тому $c'(\alpha) = \alpha^{j-1}$. Тепер залишається зауважити, що елементи $1, \alpha, \alpha^2, \ldots, \alpha^{n-1}$ різні, бо α твірний елемент мультиплікативної групи $\mathbb{F}_{2^m}$. Тому при зафіксованому α, знаючи α^{j-1}, можемо визначити показник $j-1$ і зробити висновок, що коефіцієнт c_{j-1} полінома $c(x)$ неправильно передали. Для виправлення помилки треба замінити коефіцієнт c'_{j-1} полінома $c'(x)$ на 1, якщо він дорівнює 0, і на 0, якщо він дорівнює 1.

4.2.6. БЧХ-коди

Розглянемо мультиплікативну групу $\mathbb{F}^*_{q^m}$. Вона циклічна порядку $n = q^m - 1$. Нехай α її твірна і $m_i(X)$ — мінімальний поліном елемента α^i, тобто поліном найменшого степеня серед поліномів з коренем α^i зі старшим коефіцієнтом 1. Нехай $0 \ne b \in \mathbb{N}$, $2 \le d \le n$, нехай

$$g(X) = \text{НСК}\,(m_b(X), m_{b+1}(X), \ldots, m_{b+d-2}(X)), \tag{4.21}$$

— найменше спільне кратне поліномів $m_b(X), m_{b+1}(X), \ldots, m_{b+d-2}(X)$.

Означення 4.24. *Кодом Боуза-Чоудхурі-Хоквінгема (БЧХ-кодом) з конструктивною відстанню d над полем $\mathbb{F}_q$ називають циклічний код з породжуючим поліномом (4.21).*

Важливість БЧХ-кодів полягає в тому, що для кожного додатного натурального числа d можна побудувати БЧХ-код з мінімальною кодовою відстанню не меншою ніж d. Отже, ми можемо будувати коди, здатні виправляти як завгодно багато помилок. Для цього, правда, доводиться розглядати коди великої довжини, а це відповідно зводиться до розгляду скінченних полів $\mathbb{F}_{q^m}$ з великими m.

Теорема 4.16. *Мінімальна кодова відстань БЧХ-коду з конструктивною відстанню d не менша ніж d.*

Доведення. Достатньо довести, що кожні $d-1$ стовпчиків матриці

$$H = \begin{pmatrix} 1 & \alpha^{b} & \alpha^{2b} & \dots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \dots & \alpha^{(n-1)(b+1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \alpha^{b+d-2} & \alpha^{2(b+d-2)} & \dots & \alpha^{(n-1)(b+d-2)} \end{pmatrix}$$

є лінійно незалежними (ми використовуємо теорему 4.13). Для цього розглянемо визначник матриці, стовпчиками якої є будь-які $d-1$ різних стовпчиків матриці H і покажемо, що він ненульовий.

$$\begin{vmatrix} \alpha^{bi_1} & \alpha^{bi_2} & \dots & \alpha^{bi_{d-1}} \\ \alpha^{(b+1)i_1} & \alpha^{(b+1)i_2} & \dots & \alpha^{(b+1)i_{d-1}} \\ \dots & \dots & \dots & \dots \\ \alpha^{(b+d-2)i_1} & \alpha^{(b+d-2)i_2} & \dots & \alpha^{(b+d-2)i_{d-1}} \end{vmatrix} =$$

$$= \alpha^{b(i_1+i_2+\dots+i_{d-1})} \begin{vmatrix} 1 & 1 & \dots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \dots & \alpha^{i_{d-1}} \\ \dots & \dots & \dots & \dots \\ \alpha^{(d-2)i_1} & \alpha^{(d-2)i_2} & \dots & \alpha^{(d-2)i_{d-1}} \end{vmatrix} =$$

$$= \alpha^{b(i_1+i_2+\dots+i_{d-1})} \prod_{1 \le k < j < d-1} (\alpha^{i_j} - \alpha^{i_k}) \neq 0.$$

Тут ми використали той факт, що одержаний визначник є визначником Вандермонда і тому ненульовий.

Отже, кожні $d-1$ стовпчиків матриці H є лінійно незалежними, тому мінімальна кодова відстань цього коду не менша ніж d.

Приклад

Поліном $X^4 + X + 1$ незвідний над $\mathbb{F}_2$. Справді, ні 0 ні 1 не є його коренями, а тому він не має множників першого степеня. Якби він був звідним, то ділився б на

деякий незвідний поліном другого степеня над полем $\mathbb{F}_2$. Але $X^2+X+1 \in \mathbb{F}_2[X]$ є єдиним незвідним поліномом другого степеня, і $(X^2+X+1) \not| (X^4+X+1)$.

Отож, факторкільце $\mathbb{F}_2[X]/(X^4+X+1)$ є полем $\mathbb{F}_{16}$ з $2^4=16$ елементів. Нехай $\alpha = \bar{X} \in \mathbb{F}_{16}$ — корінь полінома $X^4+X+1 \in \mathbb{F}_2[X]$. Перевіримо, що порядок елемента α дорівнює 15. Маємо $\alpha^3 \neq 1$, бо поліноми X^3-1 та X^4+X+1 взаємно прості. Так само $\alpha^5 = \alpha\alpha^4 = \alpha^2+\alpha \neq 1$, бо поліноми X^2+X+1 та X^4+X+1 взаємно прості. Позаяк порядок елемента α є дільником числа 15 і не дорівнює 1,3,5, то він дорівнює 15, отже, породжує мультиплікативну групу $\mathbb{F}_{16}^*$.

Оскільки $\alpha^4+\alpha+1=0$, то $(\alpha^4+\alpha+1)^2 = \alpha^8+\alpha^2+1 = 0$ і $(\alpha^4+\alpha+1)^4 = \alpha^{16}+\alpha^4+1=0$. Це означає, що α^2 і α^4 — теж корені полінома X^4+X+1. Знайдемо поліном з коренем α^3. Аналогічні міркування засвідчують таке: коли α^3 — корінь деякого полінома над $\mathbb{F}_2$, то $(\alpha^3)^2=\alpha^6$, $(\alpha^6)^2=\alpha^{12}$, $(\alpha^{12})^2=\alpha^{24}=\alpha^9$ — теж корені цього полінома. Тому розглянемо поліном $(X-\alpha^3)(X-\alpha^6)(X-\alpha^9)(X-\alpha^{12})$. Коефіцієнтом при X^3 цього полінома є

$$\alpha^3+\alpha^6+\alpha^9+\alpha^{12} = \alpha^3(1+\alpha^3+\alpha^6+\alpha^9) = \frac{\alpha^3(1-\alpha^{12})}{1-\alpha^3} = \frac{\alpha^3-\alpha^{15}}{1-\alpha^3} = \frac{\alpha^3-1}{1-\alpha^3} = 1.$$

Тому α^3 є коренем полінома $X^4+X^3+X^2+X+1$.

Поліноми $X^4+X^3+X^2+X+1$ та X^4+X+1 взаємно прості, тому найменше спільне кратне цих поліномів збігається з їх добутком. Поліном

$$g(X) = (X^4+X+1)(X^4+X^3+X^2+X+1)$$

має коренями $\alpha, \alpha^2, \alpha^3, \alpha^4$. Тому поліном $g(X)$ породжує код, для конструктивної відстані d якого маємо $b=1$, $b+d-2=4$. Звідси $d=5$, тому розглянутий БЧХ-код виправляє дві помилки.

4.3. Шифри

Припустимо, що двоє збираються обмінятися конфіденційною інформацією, використовуючи деякий ненадійний канал зв'язку. Ненадійність каналу може бути спричинена несанкціонованим користувачем, мета якого полягає в тому (але не завжди тільки в тому), щоб (1) заволодіти цією інформацією; (2) дезінформувати одержувача спотвореним повідомленням; (3) обманути відправника чи одержувача, чи обох стосовно особи протилежної сторони. Традиційно, двох легальних користувачів каналу зв'язку називають Алісою і Бобом, третього недружнього персонажа — суперником (Цезарем). Щоб зберегти таємницю, Боб шифрує своє повідомлення, тобто перетворює його до незрозумілої для суперника форми, застосувавши алгоритм шифрування E. В результаті з повідомлення M, яке ще називають відкритим текстом, виходить криптотекст $C=E(K,M)$, який Боб і посилає Алісі. Отримавши криптотекст

C', Аліса дешифрує його за допомогою алгоритму дешифрування D і отримує повідомлення $M' = D(K', C')$. Тут алгоритми шифрування та дешифрування залежать від ключів K та K' і разом становлять криптосистему або, простіше, шифр. Ключ і криптотекст мають визначати відкритий текст однозначно. Схему наведеної криптосистеми зображено на рис. 4.17.

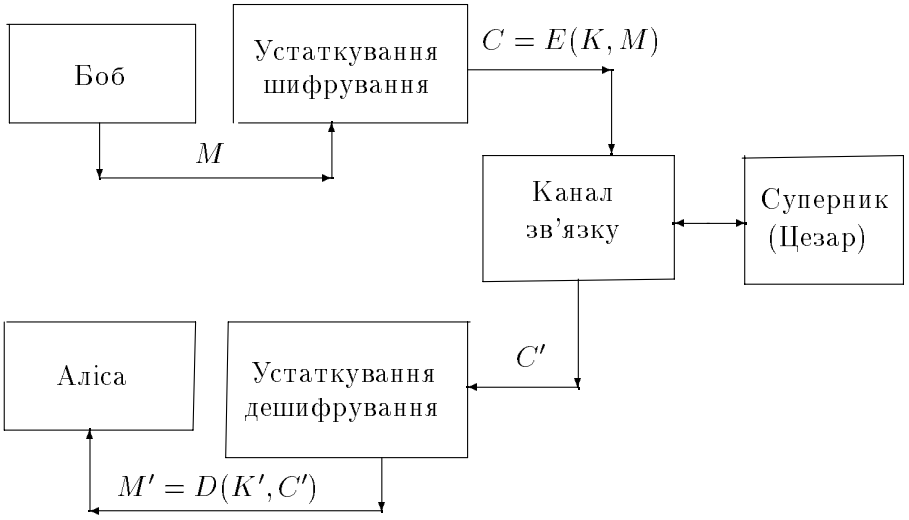


Рис. 4.17. Схема криптосистеми

Якщо суперникові вдалося знайти спосіб знаходження повідомлення за криптотекстом, то кажуть, що він розкрив шифр. Криптографія є мистецтвом створення шифрів, а криптоаналіз — їх розкриття. Такий поділ криптології на її дві складові дещо умовний, адже криптограф не може бути впевнений у надійності шифру без проведення його криптоаналізу. У ширшому трактуванні, завдання криптоаналізу не лише ламати шифри, тобто доводити їх ненадійність, а й навпаки, доводити у вигляді математичної теореми надійність шифру, попередньо означивши, який саме шифр треба вважати надійним.

Отже, перед суперником (криптоаналітиком) стоїть завдання відновити повідомлення M. Згідно з традиційною термінологією він проводить атаку на шифр. Якщо атака певного виду призводить до розкриття шифру, то шифр є вразливим до неї; якщо ж ні, то шифр є стійким до такого виду атаки. Метод повного перебору ключів називатимемо брутальною атакою. Отож, для стійкості криптосистеми до брутальної

атаки кількість її можливих ключів мусить бути досить великою, щоб повний перебір не можна було зробити ні за який розумний час навіть з використанням швидкодіючої обчислювальної техніки.

Зауважимо, що знаходження ключа не є єдино можливим способом досягти успіху суперникові. Прикладом може слугувати частотний метод описаний у пункті 4.3.1..

Для розв'язування свого завдання криптоаналітик може мати різні передумови. Для них прийнята така класифікація.

Атака лише з криптотекстом. Суперник знає лише криптотекст. У гіршому випадку (кращому з погляду суперника), крім $E(K, M)$, відома ще певна кількість криптотекстів $E(K, M_1), \ldots, E(K, M_s)$, зашифрованих з використанням того самого ключа.

Атака з відомим відкритим текстом. Крім $E(K, M)$, суперник знає як додаткові криптотексти $E(K, M_1), \ldots, E(K, M_s)$, так і відповідні їм відкриті тексти $M_1, \ldots, M_s$.

Атака з вибраним відкритим текстом. Суперник має доступ до «шифруючого устаткування» і спроможний отримати криптотексти $E(K, M_1), \ldots, E(K, M_s)$ для вибраних на власний розсуд відкритих текстів $M_1, \ldots, M_s$. Ця атака відповідає мінімальним можливостям суперника у випадку криптосистем з відкритим ключем, яким присвячено пункти 4.3.2.–4.3.6. та розділ V [5].

Атака з вибраним криптотекстом. Суперник має доступ до «дешифруючого устаткування» і спроможний одержати відкриті тексти $D(K', C_1), \ldots, D(K', C_s)$ для вибраних на власний розсуд криптотекстів $C_1, \ldots, C_s$ (однак, як і у випадку попередньої атаки, неспроможний отримати безпосередньо таємний ключ).

У наступних пунктах ми розглянемо концепції побудови криптосистем, наведемо приклади роботи їх алгоритмів шифрування та дешифрування і не будемо торкатись питань криптоаналізу. Зацікавленому криптологією читачеві рекомендуємо книгу [5].

4.3.1. Класичні шифри

У класичній криптографії можна виділити два основних типи перетворення відкритого тексту повідомлення разом з їх комбінаціями.

1. Перетворення *перестановки* змінює порядок групи символів за деяким правилом, не змінюючи їх самих. Тобто, якщо повідомлення

M складається з m блоків, $M = B_1B_2 \dots B_m$, де кожний блок B_i містить n символів, $B_i = b_{i,1}, b_{i,2} \dots b_{i,n}$ $i = 1, 2, \dots, n$, то криптотекст — це $C = C_1C_2 \dots C_m$, де $C_i = b_{i,\pi(1)}b_{i,\pi(2)} \dots b_{i,\pi(n)}$ для кожного $i = 1, 2, \dots, m$, а $\pi \in S_n$ — деяка фіксована перестановка чисел $1, 2, \dots, n$.

2. Перетворення *підстановки* замінює символи відкритого тексту відповідними символами з алфавіту криптотексту. Тобто, якщо $M = a_1a_2 \dots a_n$ — повідомлення, то шифрований текст $C = f_1(a_1)f_2(a_2) \dots f_n(a_n)$ одержуємо за допомогою n відображень f_i, $i = 1, 2, \dots, n$ з алфавіту відкритого тексту в алфавіт криптотексту.

3. Комбінуючи шифри 1 і 2, одержимо шифри перестановки/підстановки (S/P-шифри).

Приклади

1. Нехай ми хочемо зашифрувати повідомлення «дискретна математика». Для цього запишемо цей відкритий текст у зворотному порядку. Одержимо криптотекст

   акитаметам антерксид

   Ще один приклад шифру перестановки — це шифр частоколу. Щоб зашифрувати повідомлення текст записують у вигляді «частоколу» д и с к р е т н а м а т е м а т и к а і, зчитуючи текст рядками, почавши з верхнього, одержимо криптотекст икен аеаиадсртамтмтк. Якщо виберемо «висоту» частоколу, яка дорівнює 3, то криптотекст буде таким сеаамиирнметадкт так.

2. Шифр Цезаря. Давньоримський імператор Юлій Цезар шифрував свої таємні послання, підставляючи замість кожної букви іншу, яку одержували циклічним зсувом алфавіту на три позиції. Якщо цей спосіб адаптувати до текстів записаних українською абеткою, то а міняється на г, б на ґ, в на д,..., я на в. Слову математика в описаній криптосистемі відповідає криптотекст пгхзпгхйнг.

3. Розглянемо шифр, який є комбінацією шифрів перестановки та заміни. Для того щоб зашифрувати повідомлення, яке записане українською абеткою, виконаємо транслітерацію його букв у латинську абетку і застосуємо шифр частоколу висоти 2. Наприклад, слово яблуко перейде у криптотекст alkybuo. Якщо змінити порядок використання цих перетворень, то одержимо інший шифр, в якому слову яблуко відповідатиме криптотекст buoyalk.

Якщо використовується лише один алфавіт для шифрованих повідомлень, то криптосистема називається *одноалфавітною*. Криптосистеми, в яких буква шифрованого тексту може зображати більше однієї букви відкритого тексту, називають *поліалфавітними*.

Одноалфавітні шифри підстановки (заміни) використовують один з $n!$ можливих ключів, де n — кількість символів алфавіту. Тобто, можна стверджувати, що ці шифри є стійкими до брутальної атаки. Проте вони не є надійними, їх можна легко зламати, використовуючи *частотний аналіз*. Цей метод ґрунтується на такому емпіричному факті: у досить довгих текстах кожна буква трапляється з приблизно однаковою частотою, залежною від самої букви і незалежною від конкретного тексту. На підставі цього факту кожному символу можна приписати деяке число, частоту цього символу в мові.

Припустимо, що перехоплено довгий криптотекст (або послідовність багатьох коротких), одержаний за допомогою шифру заміни. Частотним методом можна виконати дешифрування, навіть не знаючи ключа. Для цього обчислюють частоти кожного символу в криптотексті і порівнюють одержані результати з табличкою частот для мови, якою написано повідомлення. Провівши певний аналіз та використовуючи невеликий перебір, можна розпізнати більшість символів повідомлення. Значну роль тут відіграє така властивість мови, як надлишковість, тобто текст можна поновити, коли частина його букв невідома.

Проілюструємо загальну ідею частотного аналізу таким прикладом. Нехай нам відомо, що криптотекст `ьцхфйіотхсцоійухчємхуцхжєчзофхс` одержали шифром зсуву, причому пропуски та розділові знаки ігнорували. Підрахуємо частоти і помічаємо, що найбільша, а саме 6/31, припадає на літеру `х`. Природно припустити, що у відкритому тексті їй відповідає найпоширеніша в українській мові літера `о`. Це означало б, що довжина зсуву дорівнює 7. Виконаємо обернений зсув, тобто на сім позицій вліво, і справді одержимо змістовне повідомлення: `упонеділокпідеморазомпобарвінок`.

Існують досконаліші різновиди частотного аналізу, які, крім частот окремих символів, враховують також частоти пар символів. Ці методи дають змогу ламати певні класи шифрів заміни. Також частотний аналіз допомагає комп'ютерові без участі людини відрізнити осмислений текст від хаотичного набору символів. Завдяки цьому на машину можна перекласти виконання брутальної атаки, тобто повного перебору ключів.

Подання тексту у цифровій формі. Для сучасних засобів передачі, збереження та опрацювання інформації зручнішим є її подання у цифровій формі. Щоб найпростіше досягнути цього, можна замінити симво-

ли тексту їхніми номерами у алфавіті. Наприклад, занумеруємо букви української абетки починаючи з 0.

а	б	в	г	ґ	д	е	є	ж	з	и	···	···	щ	ь	ю	я
0	1	2	3	4	5	6	7	8	9	10	···	···	29	30	31	32

Наприклад, слово `банк` буде подане як 01 00 17 14. Якщо потрібно, то в алфавіт можна ввести, крім букв, також знаки пунктації, пропуск, цифри тощо. Номери букв можемо записувати не в десятковій системі числення, а у двійковій. Для слова `банк` одержимо такий двійковий запис 000001 000000 010001 001110.

Тепер можемо дати означення одного класу шифрів заміни.

Означення 4.25. *Відображення* $n \to an + b \pmod m$ *кільця* $\mathbb{Z}_m$ *в себе при фіксованих* $a, b \in \mathbb{Z}_m$ *і* $НСД(a, m) = 1$ *називається модулярним шифром.*

При $a = 1,\ b = 3,\ m = 33$ одержимо шифр Цезаря для повідомлень записаних мовою, алфавіт якої містить не більше 33 символів. Наприклад, відкритий текст 22 00 17 14 зашифровуємо як 25 03 20 17, або еквівалентно як `хгрн`.

Одноалфавітний шифр може бути підсилений, якщо використовувати поліалфавітний шифр заміни, який приховує частоти букв за рахунок кратних підстановок. При шифруванні повідомлення використовують більше одного алфавіту, і ключ підказує, яку підстановку використовувати для кожного символу. Такі шифри називаються *шифрами Віженера.*

Точніше, поліалфавітний шифр підстановки з періодом s складається з s шифрувальних алфавітів і відображень $f_i : A \to B_i,\ i = 1, 2, \ldots, s$, які визначаються ключем. Переважно як ключ використовують деяке слово $K = k_1 k_2 \ldots k_s$ і $f_i(x) = x + k_i \pmod{33}$. Відкритий текст $M = x_1 x_2 \ldots x_s x_{s+1} \ldots x_{2s} \ldots$ зашифровують як $f_1(x_1) f_2(x_2) \ldots f_s(x_s)\ f_1(x_{s+1})$ $..f_s(x_{2s}) \ldots$, тобто повторенням послідовності відображень $f_1, f_2, \ldots, f_s$ для кожних s символів. Для шифрування можна використовувати квадрат Віженера

а	б	...	ю	я
б	в	...	я	а
.	.	...	.	.
ю	я	...	щ	ь
я	а	...	ь	ю

Процедуру шифрування можна спростити, якщо ми зауважимо, що можна використовувати не всі рядки (алфавіти) квадрата Віженера. Нехай ключ — «УЕФА», тоді $s = 4$ і відображення f_i задають такою таблицею, де ключ стоїть у першому стовпці

а	бвгґдеєжзиіїйклмнопрстуфхцчшщьюя
у	фхцчшщьюяабвгґдеєжзиіїйклмнопрст
е	єжзиіїйклмнопрстуфхцчшщьюяабвгґд
ф	хцчшщьюяабвгґдеєжзиіїйклмнопрсту
а	бвгґдеєжзиіїйклмнопрстуфхцчшщьюя

У цьому випадку повідомлення дискретнаматематика шифрується як шмікиіинутфтщтфтарф, тобто j-та буква відкритого тексту відображається в символ, який є у відповідному стовпці та ($j \mod 4$) рядку (алфавіті). Наприклад, буква с відображається у букву і, що є в стовпці, який починається буквою с і в рядку, який починається буквою ф (3 буквою нашого ключа).

Легко зауважити, що ключ «УЕФА» у запропонованій нами цифровій формі має вигляд 23 06 24 00 і шифрування можна виконати додаванням за модулем 33 відповідних чисел ключа до чисел, що відповідають буквам відкритого тексту.

	5	10	21	14	20	6	22	17	0	16	0	22	6	16	0	22	10	14	0
+	23	6	24	0	23	6	24	0	23	6	24	0	23	6	24	0	23	6	24
	28	16	12	14	10	12	13	17	23	22	24	22	29	22	24	22	0	20	24

Алгоритм дешифрування збігатиметься з алгоритмом шифрування, якщо за дешифруючий ключ взяти числа протилежні за $\mod 33$ до чисел цифрового зображення ключа для шифрування. Для наведеного прикладу, дешифруючий ключ — 10 27 09 00 (ИЧЗА).

Другим варіантом поліалфавітних шифрів заміни є *шифр з автоключем*, який ґрунтується на ідеях Кардано та Віженера. Як і у шифрі Віженера, криптотекст одержують сумуванням відкритого тексту з послідовністю букв такої самої довжини. Цю послідовність формують так: до ключа дописують справа потрібну кількість символів відкритого тексту. Для повідомлення та ключа з попереднього прикладу шифрування відбуватиметься так:

	5	10	21	14	20	6	22	17	0	16	0	22	6	16	0	22	10	14	0
+	23	6	24	0	5	10	21	14	20	6	22	17	0	16	0	22	6	16	0
	28	16	12	14	25	16	10	31	20	22	22	6	6	32	0	11	16	30	0

Афінні шифри. Більшість класичних шифрів заміни можуть бути сформульованими в єдиній моделі лінійних перетворень простору повідомлень. Для цього ототожнимо повідомлення довжини n з n-кою цілих чисел і прирівняємо операції шифрування та дешифрування до пари взаємно обернених лінійних перетворень. Узагальнюючи поняття модулярного шифру, ми одержимо таке означення.

Означення 4.26. *Нехай A — $n \times n$ матриця, елементи якої належать $\mathbb{Z}_m$, X і B — вектори простору $\mathbb{Z}_m^n$. Афінний шифр — це відображення вигляду $X \to AX + B$, яке оборотне тоді і лише тоді, коли НСД$(detA, m)$=1.*

Для шифрування розбиваємо відкритий текст на блоки довжини n і замінюємо кожну букву відповідними елементами з кільця $\mathbb{Z}_m$; утворюємо транспоновані вектор-стовпці X і застосовуємо описане перетворення до кожного такого блоку. Нехай, для спрощення, вектор B в означенні шифру є нульовим. Тоді матриця A — ключ шифруючого перетворення, а обернена матриця A^{-1} буде дешифруючим ключем. (Умова НСД$(\det A, m)$=1 забезпечує оборотність матриці A.)

Приклад

Спочатку, повідомлення `Алісапродавайакції`, яке плануємо шифрувати, запишемо у цифровій формі: 00 15 11 21 00 19 20 18 05 00 02 00 13 00 14 26 11 12.

Нехай афінний шифр задається матрицею $A = \begin{pmatrix} 32 & 21 \\ 3 & 2 \end{pmatrix}$, $(n = 2,\ m = 33)$. Обчислимо криптотекст, тобто помножимо матрицю A на матрицю

$$\begin{pmatrix} 0 & 11 & 0 & 20 & 5 & 2 & 13 & 14 & 11 \\ 15 & 21 & 19 & 18 & 0 & 0 & 0 & 26 & 12 \end{pmatrix},$$

всі операції виконуємо за модулем 33. Одержимо

$$\begin{pmatrix} 18 & 1 & 3 & 28 & 28 & 31 & 20 & 4 & 10 \\ 30 & 9 & 5 & 30 & 15 & 6 & 6 & 28 & 24 \end{pmatrix}$$

або еквівалентно 18 30 01 09 03 05 28 30 28 15 31 06 20 06 04 28 10 24. Нехай тепер маємо криптотекст 30 28 07 01 12 21 28 30 28 15 31 21, який хочемо дешифрувати. Для цього знайдемо дешифруючий ключ — матрицю $A^{-1} = \begin{pmatrix} 2 & 12 \\ 30 & 32 \end{pmatrix}$ і виконаємо потрібне множення. Отримаємо відкритий текст 00 14 26 11 12 09 20 18 05 00 17 18 або `акціїпродано`.

Шифр одноразового блокнота запропонував у 1918 р. американський інженер Гілберт Вернам, який працював у системі телетайпного зв'язку AT&T. Повідомлення передавали тоді за допомогою двійкового коду і

Вернам запропонував додавати до них за модулем 2 деяку випадкову послідовність так, щоб вся частотна інформація, кореляція між символами, періодичність тощо приховувались.

Головний недолік цієї процедури полягає в тому, що вона вимагає завчасного обміну великою кількістю ключів.

Приклад

Нехай ми хочемо зашифрувати слово `банк`. Для цього подамо його у двійковій формі

$$M = 000001\ 000000\ 010001\ 001110.$$

Ключ виберемо так $K = 101100\ 011010\ 001110\ 010100$. Сумуючи побітово за модулем 2 ці послідовності одержимо криптотекст $C = 101101\ 011010\ 011111\ 011010$.

Дешифрування у шифрі одноразового блокнота збігається з шифруванням. Тобто, щоб одержати повідомлення M, треба до криптотексту C побітово додати за модулем 2 той самий ключ K.

Назва шифру походить від того, що процес шифрування/дешифрування використовує списки випадкових чисел («листи блокнота») для отримання ключа, який використовують лише раз. Випадковий вибір ключа забезпечує те, що будь-які два повідомлення однакової довжини з однаковою ймовірністю можуть перейти у цей шифрований текст. Отже, шифр одноразового блокнота абсолютно надійний або, як ще кажуть, надійний у теоретико-інформаційному сенсі.

Шифр вважають надійним в обчислювальному сенсі, якщо для цього шифру невідомо методу його ламання упродовж реалістичного терміну. Донедавна таким вважався стандарт шифрування даних *DES (Date Encryption Standart)*, який у 1976 р. приняли як федеральний стандарт США для захисту комерційної та урядової інформації, не пов'язаної з національною безпекою.

DES-алгоритм — це блочний шифр, розроблений фірмою IBM на основі S/P схеми. DES шифрує 64-бітові блоки відкритого тексту, використовуючи 64-бітові ключі (56 бітів ключа і 8 контрольних бітів). Шифрування відбувається за 16 циклів, на кожному кроці використовують інший 48-бітовий ключ.

З погляду теорії складності DES виглядає надійним, проте найбільше критики зазнавав розмір ключа цього стандарту. Ці недоліки стимулювали подальші дослідження, які призвели до відкриття криптосистем з відкритим ключем.

4.3.2. Концепція шифрів з відкритим ключем

Поняття криптосистеми з відкритим ключем було вперше введено В. Діффі та М. Гелманом у 1976 р. Основна ідея цих криптосистем — застосування до шифрування важкооборотних функцій. Наведемо формальне означення цих функцій.

Означення 4.27. *Бієктивне відображення $f : X \to Y$ називається важкооборотною функцією, якщо для заданого $x \in X$ легко обчислити $f(x)$, а обчислення $f^{-1}(y)$ для випадково вибраного $y \in Y$ є складною задачею.*

Для шифрування використовують спеціальний клас функцій, які залишаються важкооборотними за умови, що певна інформація (дешифруючий ключ) тримають в таємниці. Такі функції називають важкооборотними з секретом. Вони відображають множину символів відкритого тексту в криптотекст, який кожен, хто володіє «відкритим ключем», легко може обчислити; проте оберненої функції (яка дешифрує криптотекст) не можна визначити за реалістичний термін без додаткової інформації («приватного ключа»).

Це означає, що будь-хто може надіслати криптотекст певному абонентові комунікаційної мережі, користуючись його відкритим ключем, який вільно можна знайти у каталозі відкритих ключів користувачів мережі. Тобто, на відміну від криптосистем з таємним ключем, не має потреби попередньо обмінювати ключі за допомогою надійного кур'єра.

Зазначимо, що всі розглянуті нами у попередньому пункті класичні криптосистеми були *симетричними* в тому змісті, що ключі шифрування K і дешифрування K' були однаковими (або легко обчислювались за умови, що відомий один з них), тому мали бути таємними. В цьому контексті криптосистеми з відкритим ключем ще називають *асиметричними*.

Відкриття криптосистем з відкритим ключем значною мірою збільшило роль алгебри та теорії чисел у криптографії. Адже дотепер і напевно надалі, саме ці розділи математики є джерелом важкооборотних функцій із секретом.

Першим на практиці серед шифрів з відкритим ключем почали застосовувати криптосистему RSA (див. п. 4.3.3.), яка використовує апарат теорії чисел, який розробив ще Ойлер. Якщо криптографія з відкритим

ключем не потребує сучасної математики, то чому вона чекала на своє відкриття аж до 1976 р.?

Однією з причин цього було те, що до 70-х років криптографію переважно використовували у військових чи дипломатичних цілях, для яких добре надаються шифри з таємним ключем. Із розвитком техніки і комп'ютеризації виникають нові потреби застосовувати криптографію. Криптосистеми з відкритим ключем успішно починають використовувати у тих галузях, де на відміну від військових чи дипломатів не має такої чіткої службової ієрархії, стабільності кадрів і відлагодженої кур'єрської мережі. Наприклад, новітні криптосистеми забезпечують таємність електронних банківських платежів, використовують для зберігання конфіденційної інформації різного характеру (медичної, фінансово-кредитної тощо).

Інша причина того, чому не було винайдено криптосистем з відкритим ключем за часів Ойлера полягає в тому, що тоді всі обчислення проводили вручну. Для того щоб, наприклад, за допомогою RSA одержати високий рівень надійності, потрібно оперувати великими числами, тобто треба використовувати комп'ютер.

На практиці розвиток криптосистем з відкритим ключем тісно пов'язаний з поширенням потужної комп'ютерної техніки. Наведемо лише список завдань, які стоять перед сучасною криптографією:

1) *таємне пересилання інформації*;
2) *довірливе спілкування* (з підтвердженням того, що повідомлення насправді було відправлене цією особою). Для цього часто використовують вкорочуючу функцію (hash function, див. п. 4.3.3.), цифровий підпис (див. п. 4.3.4.), систему паролів (гасел) та ідентифікації (контроль повноважень при доступі до даних чи пристроїв та засвідчення того, що особа насправді та, за кого себе видає), систему незаперечуваності (запобігання можливості відхилення того, на що хтось дав згоду);
3) *обмін ключем* — коли дві сторони хочуть використати відкритий канал для обміну таємними ключами, для їх використання в симетричних криптосистемах;
4) *підкидання монети по телефону* — коли, наприклад, двоє шахістів, перебуваючи в різних місцях, хочуть визначити по телефону (чи електронною поштою) хто з них гратиме білими;

5) *розподіл сєкрету*, коли деяка таємна інформація (наприклад, код спільного банківського депозиту) розподілена серед групи з k осіб так, що жодні $k-1$ з них не можуть її здобути;
6) *доведення без розголошення* — коли хочуть когось переконати, що розв'язано складну задачу не відкриваючи інформацію про спосіб її розв'язання.

Наведені завдання розв'язують за допомогою різних *протоколів*. Під «протоколом» розуміємо процедуру, яка визначає черговість дій для обміну повідомленнями та досягнення певних цілей.

У наступних пунктах опишемо кілька часто вживаних (базисних) детерміністичних криптосистем, які можуть бути використані в одній або декількох згаданих ситуацій. На практиці, для побудови криптосистем заданого рівня безпеки і надійності потрібно старанно модифікувати і поєднувати певні базисні системи.

4.3.3. Криптосистема RSA

Розглянемо деяку комунікаційну мережу з великою кількістю абонентів, які хочуть налагодити між собою довірливе спілкування. Вважатимемо, що повідомлення, якими вони обмінюються — це цілі числа ω такі, що $0 \le \omega < N$. Наприклад, відкритий текст може складатися з блоків по k букв української абетки, яким відповідають цілі числа в системі числення за основою 33. Тобто, вибираємо $N = 33^k$. На практиці, число N має в криптосистемі RSA від 200 до 600 десяткових знаків.

Кожен користувач мережі A (Аліса) вибирає два досить великі прості числа p і q так, щоб їхній добуток $n = pq$ був більшим від N. Аліса тримає ці два числа в таємниці, а значення n відкрито публікує в каталозі абонентів мережі. Додатково вона анонсує випадково вибране число e, яке взаємно просте як з $p-1$ так і з $q-1$ (того самого порядку, що й число n). Отже, відкритим ключем Аліси є пара чисел (n, e).

Що робить абонент мережі B (Боб), якщо хоче надіслати Алісі повідомлення ω? Він шукає у каталозі її відкритий ключ, обчислює s — лишок від ділення ω^e на n і надсилає його Алісі. Піднесення до степення $s = \omega^e \pmod n$ Боб може виконати досить швидко (див. п. 3.3.4.).

Щоб дешифрувати надіслане їй повідомлення, Аліса використовує свій таємний ключ. Цим дешифруючим ключем є довільне число d, для якого $de \equiv 1 \pmod{p-1}$ і $de \equiv 1 \pmod{q-1}$. Таке число d Алі-

са може легко обчислити, застосовуючи розширений алгоритм Евкліда (див. п. 3.3.3.) до пари чисел e та НСК$(p-1, q-1)$.

Тепер, якщо Аліса обчислить лишок від ділення s^d на n, то одержить повідомлення ω. Справді, $s^d \equiv (\omega^e)^d \equiv \omega^{ed} \pmod{n}$ і, позаяк $de \equiv 1\big(\bmod\ (p-1)(q-1)\big)$, то з теорем Ойлера–Ферма випливає, що $\omega^{ed} \equiv \omega \pmod{n}$.

Приклад

Нехай відкритий ключ Боба в криптосистемі RSA дорівнює

$$(n = 99999999100000001881,\ \ e = 9999999929).$$

Якщо Аліса хоче відправити Бобові повідомлення ЧИЗУСТРІНЕМОСЬМИЗАВТРА, то перетворює відкритий текст у цифрову форму

$$M_{33} = 27\,10\,09\,23\,21\,22\,20\,11\,17\,06\,16\,18\,21\,30\,16\,10\,09\,00\,02\,22\,20\,00.$$

Вона записує це число, наприклад, в десятковій системі числення

$$M_{10} = 21141972072435\ \ 44844996638708578890$$

і розбиває його на блоки $M_{10} = \omega_2\omega_1$ відповідної довжини. Кожен блок ω_i Аліса шифрує, обчислюючи $s_1 \equiv \omega_1^e = 44844996638708578890^e \equiv 70916571639825681883$ і $s_2 \equiv \omega_2^e = 21141972072435^e \equiv 88639038860767980773 \pmod{n}$, і посилає Бобові.

Знаючи розклад числа n на прості множники $p = 9999999967$ і $q = 9999999943$, Боб застосувавши розширений алгоритм Евкліда до пари чисел e і НСК$(p-1, q-1)$ може легко обчислити d за модулем $(p-1)(q-1)$. Оскільки, e взаємно просте з $p-1$ і $q-1$, то їхній найбільший спільний дільник $1 = e \cdot d + \text{НСК}(p-1, q-1) \cdot u$, а отже, $d = 15696465552037422347$. Тепер, обчисливши $\omega_i \equiv s_i^d \pmod{n}$, Боб може прочитати адресоване йому повідомлення.

Зауважимо, що вибрані для цього прикладу параметри криптосистеми (прості числа p і q) володіють майже усіма недоліками, яких треба уникати, будуючи криптосистеми на практиці. По-перше, числа p і q невипадково вибрані (спеціально взяли два найбільші десятизначні прості числа); по-друге, їхня довжина має бути принаймні в 10 разів більшою; по-третє вони не повинні бути «близькими» один до одного. Невиконання цих умов спрощує задачу факторизації n, а отже, і ламання криптосистеми.

Що стримує несанкціонованого користувача мережі C (Цезаря), який володіє відкритим ключем (n, e), від дешифрування повідомлення? Проблема Цезаря полягає в тому, що без знання множників p і q не існує напевно можливого способу знаходження дешифруючого ключа d, який відповідає за ефективну оборотність відображення $\omega \to \omega^e \pmod{n}$. Мабуть, не існує інших методів дешифрування. Останні два твердження залишаються недоведеними. Можна лише сказати, що задача ламання шифру RSA імовірно така складна, як і задача розкладу числа n на множники.

Перше ніж приступити до опису цифрового підпису на основі RSA, хочемо згадати важливий криптографічний інструмент — вкорочуючу функцію (hash function).

Означення 4.28. *Функція $H(x)$ визначена на множині повідомлень довжини l і образ якої — множина слів фіксованої довжини k називається вкорочуючою, якщо для кожного повідомлення M легко обчислити $H(M)$, але*

1) практично неможливо знайти два повідомлення M_1, M_2 таких, що $H(M_1) = H(M_2)$ (тобто $H(x)$ — безколізійна функція);

2) для заданого y_0 з образу функції $H(x)$ майже неможливо знайти такого x_0, що $H(x_0) = y_0$ ($H(x)$ — важкооборотна функція).

Мабуть найважливіше застосування вкорочуючої функції — цифровий підпис. Нехай Аліса і Боб використовують ту ж саму вкорочуючу функцію, яку насправді можуть і не тримати в таємниці від Цезаря. Коли Боб посилає Алісі повідомлення M, то він приєднує до нього вкорочене значення $H(M)$.

Аліса хоче бути певною, що повідомлення прийшло від Боба, що Цезар не підмінив його. Припустимо, що Аліса якимось способом може переконатись, що приєднане значення $H(M)$ походить від Боба. Тоді їй достатньо застосувати вкорочуючу функцію до одержаного повідомлення M' і порівняти результат з $H(M)$. Якщо $H(M') = H(M)$, то Аліса переконується в адресанті повідомлення, бо Цезар практично не може так змінити криптотекст, щоб для зміненого повідомлення M' виконувалась рівність $H(M) = H(M')$.

Залишається нерозв'язаною проблема як Алісі визначити, що $H(M)$ справді надійшло від Боба. Для розв'язання цієї задачі можна використати криптосистему RSA.

Посилаючи Алісі повідомлення M, Боб обчислює його вкорочений варіант $H = H(M)$. Перед тим як вислати його Алісі, Боб підносить H за модулем $n_{\text{Боба}}$ до степеня рівного його дешифруючому ключу $d_{\text{Боба}}$. Боб, крім самого повідомлення M, посилає Алісі блок $H' = H^{d_{\text{Боба}}} \pmod{n_{\text{Боба}}}$, користуючись її відкритим ключем $(e_{\text{Аліси}},\ n_{\text{Аліси}})$, тобто такий фрагмент криптотексту:

$$\left(H^{d_{\text{Боба}}} \pmod{n_{\text{Боба}}}\right)^{e_{\text{Аліси}}} \pmod{n_{\text{Аліси}}}.$$

Аліса дешифрує одержаний криптотекст, потім бере його останній блок і підносить до степеня $e_{\text{Боба}}$ за модулем $n_{\text{Боба}}$. Так вона отримає H. Далі Аліса обчислює вкорочене значення від одержаного відкритого тексту без остатнього блока і порівнює його з H.

Позаяк, лише Боб знає обернене число до $e_{\text{Боба}}$ за модулем $n_{\text{Боба}}$, то Аліса переконується в тому, що H справді надіслав Боб. Вона також стає впевнена в тому, що саме він прислав повідомлення M і в тому, що воно не було сфальшованим.

Приклад

Нехай Боб хоче відповісти Алісі, використовуючи криптосистему RSA і підпис на її основі. Приймемо їхні відкриті ключі, які дорівнюють відповідно ($n_A = 1000000016600\ 00006693$, $e_A = 10^{10} + 61$) і ($n_B = 99999999100000001881$, $e_B = 10^{10} - 71$) і нехай вони для підпису використовують вкорочуючу функцію, яка додає за модулем 10 відповідні цифри кожного блоку повідомлення. Боб перетворює своє повідомлення ТАКОДРУГІЙВБІБЛІОТЕЦІ у цифрову форму

$$M_{33} = 22\ 00\ 14\ 18\ 05\ 20\ 23\ 03\ 11\ 13\ 02\ 01\ 11\ 01\ 15\ 11\ 18\ 22\ 06\ 26\ 11.$$

У десятковій системі числення це число має такий вигляд:

$$M_{10} = 516374945545\ 367852386147823422 37.$$

Розіб'ємо його на блоки w_2, w_1 довжини 20 і просумуємо відповідні цифри кожного блока за модулем 10. Одержимо $H = H(M_{10}) = 367852381200562877 72$. Далі Боб, використовуючи свій таємний ключ d_B, обчислює $H' = H^{d_B} = 8393679242967558940 \pmod{n_B}$. Приєднавши цей блок підпису до блоків повідомлення, Боб шифрує їх підносячи до степеня e_A за модулем n_A. Одержані блоки криптотексту $s_0 = (H')^{e_A} = 51417407710110462013 \pmod{n_A}$, $s_1 = w_1^{e_A} = 853969656288843180 87 \pmod{n_A}$ і $s_2 = w_2^{e_A} = 94375120426889426721 \pmod{n_A}$ Боб посилає Алісі.

Аліса дешифрує отриманий криптотекст, підносячи його блоки до степеня $d_A = 918367361959183733$ за модулем n_A, тобто обчислює $w_i' = s_i^{d_A} \pmod{n_A}$. Далі, відділивши блок w_0', що відповідає за підпис, Аліса обчислює $(w_0')^{e_B} \pmod{n_B}$ і порівнює зі значенням вкорочуючої функції $H(\overline{w_2' w_1'})$ від решти блоків. Якщо остатні два числа рівні, то Аліса переконується в тому, що одержане повідомлення справді від Боба і воно не було сфальшованим.

Варто зауважити, що підпис на основі RSA, крім простої ідентифікації адресанта, розв'язує два інші важливі завдання. По-перше, забезпечується приватність, бо повідомлення шифрується разом з приєднаним блоком H', тому не можна визначити особи відправника. По-друге, підпис забезпечує також і незаперечуваність: Боб не може відмовитись від того, що надіслав підписане повідомлення.

4.3.4. Схема Діффі–Гелмана і DSA

Інший важливий приклад криптосистем з відкритим ключем розроблено на основі дискретного логарифма. Пояснимо в чому полягає задача дискретного логарифмування. Нехай $\mathbb{F}_p{}^* = (\mathbb{Z}_p)^* = \{\bar{1}, \bar{2}, \ldots, \overline{p-1}\}$ — мультиплікативна група скінченного поля з p елементів чи кільця лишків за простим модулем p. Детальніше властивості цієї групи розглядали у п. 4.2.1..

Проблемою дискретного логарифма у $\mathbb{F}_p{}^*$ при основі $g \in \mathbb{F}_p{}^*$ називають задачу знаходження для заданого $y \in \mathbb{F}_p{}^*$ такого натурального числа x, що $y = g^x$ за модулем p (якщо таке x існує; в іншому випадку маємо одержати підтвердження того, що x не належить підгрупі породженій g).

Опишемо систему Діффі–Гелмана (експоненційного) обміну ключем. Припустимо, що Аліса і Боб хочуть домовитись про велике натуральне число, яке вони використовуватимуть як таємний ключ у симетричній криптосистемі. Цей обмін відбувається за допомогою відкритого каналу. Тобто, Цезар, прослуховуючи цей канал, знає зміст всіх повідомлень, якими обмінюються Аліса і Боб.

Спочатку Аліса і Боб відкрито узгоджують досить велике просте число p і первісний корінь g за модулем p. Потім, Аліса випадково вибирає натуральне число k_A в межах від 1 до $p-1$ (але того ж порядку, що і p), обчислює лишок від ділення g^{k_A} на p і посилає це значення Бобові. Подібно чинить Боб: посилає Алісі $g^{k_B} \mod p$ і тримає в таємниці k_B. Узгодженим ключем буде число $g^{k_A k_B} \in \mathbb{F}_p{}^* = \{\bar{1}, \bar{2}, \ldots, \overline{p-1}\}$, яке Боб може обчислити, підносячи за модулем p, число одержане від Аліси до степеня k_B, Аліса — обчислюючи k_A степінь числа, одержаного від Боба. Позаяк $g^{k_A k_B} \equiv \left(g^{k_A}\right)^{k_B} \equiv \left(g^{k_B}\right)^{k_A} \pmod{p}$, то вони володітимуть тим самим числом.

Перед Цезарем, який хоче дізнатись цей таємний ключ, стоїть так звана задача Діффі-Гелмана: за заданими $g, g^{k_A}, g^{k_B} \in \mathbb{F}_p{}^*$ обчислити $g^{k_A k_B}$. Легко зрозуміти, той, хто може розв'язати задачу дискретного логарифма, може також розв'язати задачу Діффі-Гелмана. Нічого невідомо чи правильне зворотне твердження. Тобто, не доведено рівносильність задачі ламання системи обміну ключем Діффі-Гелмана та задачі дискретного логарифма (хоча недавні часткові результати схиля-

ють до думки про справедливість цієї гіпотези). Для практичних цілей приймають, що схема обміну ключем Діффі-Гелмана є надійною доти, доки задача дискретного логарифма вважається складною.

Алгоритм DSA. У 1991 р. американський інститут технічних стандартів (National Institute of Standards and Technology, скорочено NIST) запропонував стандарт цифрового підпису DSS (Digital Signature Standard), який ґрунтується на алгоритмі цифрового підпису DSA (Digital Signature Algorithm). DSS планували використовувати, аналогічно DES (див. п.4.3.1.), як стандартний метод підпису для захисту комерційної та урядової інформації непов'язаної з нацбезпекою. Грунтується цей підпис на задачі дискретного логарифма в полі $\mathbb{F}_p$. Використовує він також стандартну вкорочуючу функцію SHA (Secure Hash Algorithm). Ця функція повідомлення довжини кратної 512 бітів вкорочує до 160 бітів. Вона ґрунтується на ідеях, запропонованих професором Рональдом Райвестом при розробці алгоритму MD4 та значною мірою наслідує його.

Сам алгоритм DSA дуже подібний до системи вперше запропонованої Шнорром, також він подібний до криптосистеми ЕльГамала. Опишемо як діє DSA. Для побудови системи підпису повідомлення кожен користувач спочатку виконує такі кроки:

1) вибирає просте число q довжини порядку 160 бітів (для цього використовують генератор псевдовипадкових бітів і тести простоти);
2) вибирає друге просте число p довжини кратної 64 і таке, що $q \mid (p-1)$ та $2^{512} < p < 2^{1024}$;
3) вибирає довільний твірний елемент єдиної циклічної підгрупи порядку q групи $\mathbb{F}_p{}^*$ (для випадково вибраного числа g_0 обчислює $g_0^{(p-1)/q} \pmod p$. Якщо результат не дорівнює 1, то g_0 — шуканий твірний елемент);
4) випадково вибирає свій таємний ключ — число x у межах $0 < x < q$ та обчислює свій відкритий ключ $y = g^x \pmod p$.

Нехай Аліса виконала наведені кроки і хоче підписати своє повідомлення M. Спочатку вона обчислює вкорочене значення $H = SHA(M)$ — 160 бітове ціле число. Далі випадково вибирає число k (того ж порядку що і q) і обчислює $r = (g^k \mod p) \mod q$. Підпис Аліси буде складатись з пари чисел (r, s), де $s \equiv k^{-1}(H + xr) \mod q$.

Для того щоб підтвердити підпис, адресат (Боб) обчислює $u_1 = s^{-1}H \pmod q$ і $u_2 = s^{-1}r \pmod q$. Якщо $g^{u_1}y^{u_2} \pmod p$ дорівнює r за модулем q, то Боб переконується в тому, що повідомлення надіслала Аліса.

Для того щоб перевірити коректність наведеної схеми, достатньо переконатися, що $g^k \equiv g^{u_1}y^{u_2} \pmod p$. Позаяк $y = g^x \pmod p$, то $g^k \equiv g^{u_1+xu_2} \pmod p$. Оскільки порядок елемента g в $\mathbb{F}_p{}^*$ дорівнює q, то останнє порівняння рівносильне такому $k \equiv u_1 + xu_2 \pmod q$. Використовуючи те, що $u_1 \equiv s^{-1}H \equiv k(H+xr)^{-1}H \pmod q$ і $u_2 \equiv s^{-1}r \equiv k(H+xr)^{-1}r \pmod q$, одержимо $k \equiv u_1 + xu_2 \equiv k(H+xr)^{-1}H + xk(H+xr)^{-1}r = k(H+xr)^{-1}(H+xr) \equiv k \pmod q$, що показує коректність алгоритму підпису DSA.

Очевидно, надійність наведеної криптосистеми залежить від неможливості ефективного розв'язання задачі дискретного логарифма в полі $\mathbb{F}_p$.

4.3.5. Розподіл таємниці, підкидання монети по телефону

Припустимо, що ми хочемо групі осіб довірити секретну інформацію (наприклад, деяке натуральне число N) розподіливши її так, щоб кожна вибрана з них підгрупа k осіб могла це число відновити, але щоб ніяка група з $(k-1)$ учасника не могла цього зробити.

Наведемо три приклади реалізації цього протоколу, які ґрунтуються відповідно на системах лінійних рівнянь, інтерполяційній формулі Лагранжа та китайській теоремі про остачі.

Приклади

1. Виберемо довільну точку $P(x_1, x_2, \ldots, x_k)$ простору $\mathbb{R}^k$ з цілими координатами x_i, та $x_1 = N$. Кожному учасникові протоколу розподілу секрету повідомляємо одне лінійне рівняння k-змінних, яке задовольняють x_i, $i = \overline{1, \ldots, k}$. Так вибираємо k лінійно незалежних рівнянь, які становлять систему Крамера. Очевидно, що така система лінійних рівнянь має єдиний розв'язок, а будь-які $(k-1)$ з них утворюють невизначену систему, тобто з якої не можна одержати розв'язок $x_1 = N$.

2. Інший метод розподілу секрету використовує кільце поліномів $\mathbb{F}_p[x]$ для деякого простого $p > N$. Випадково вибираємо поліном

$$f(x) = \sum_{s=0}^{k-1} a_s x^s \in \mathbb{F}_p[x], \quad \text{де} \quad a_0 = N.$$

Кожному учасникові розподілу таємниці повідомляють пару чисел $(m_i, f(m_i))$, де $m_i \in \mathbb{F}_p{}^*$, і якщо $i \neq j$, то $m_i \neq m_j$.

Тепер, знаючи k значень полінома $f(x)$, його можна відновити за інтерполяційною формулою Лагранжа

$$f(x) = \sum_{i=1}^{k} f(m_i) \prod_{i \neq j} \frac{x - m_j}{m_i - m_j}$$

і знайти секретну інформацію $N = f(0)$. Очевидно, що будь-які $(k-1)$ значень полінома не відновлять секрет, бо для довільного $n \in \mathbb{F}_p{}^*$ за інтерполяційною формулою існує такий поліном, значення якого в нулі дорівнює n.

3. Схематично опишемо протокол розподілу таємниці, який використовує систему порівнянь. Для кожного учасника протоколу вибирають просте число p_i і повідомляють йому лишок від ділення N на p_i. Число N належить деякому проміжку для того, щоб, використовуючи китайську теорему про остачі, його можна було знайти однозначно за k відомими лишками і не можна було відтворити знаючи лише $(k-1)$ лишок.

Розглянемо розв'язання ще одного завдання сучасної криптографії, а саме, реалізацію протоколу підкидання монети по телефону.

Нехай Аліса і Боб хочуть вирішити, хто з них матиме перевагу, наприклад, хто з них гратиме в шахи білими або хто подаватиме першим у тенісному матчі. Вони можуть це зробити підкинувши монету, за умови, що обидвоє в тому самому місці та мають монету, якій вони довіряють. В іншому випадку можна використати криптографічну процедуру *зобов'язання бітів* (bit commitment), яка є складовою протоколу *підкидання монети по телефону.* Ці протоколи можуть бути реалізовані за допомогою різних криптосистем як симетричних, так і асиметричних.

Розглянемо протокол підкидання монети на основі важкооборотної функції. Нехай Аліса і Боб завчасно домовились використовувати деяку важкооборотну функцію f, тоді:

1) Аліса вибирає випадково число x і обчислює $y = f(x)$;
2) Аліса посилає значення y Бобові;
3) Боб відгадує, x парне чи непарне число і посилає результат відгадування Алісі;
4) якщо відгадує Боб правильно, то результатом підкидання монети по телефону є «копійка». Якщо ж Боб не відгадує парність числа x, то результат — «герб». Аліса оголошує результат підкидування монети і посилає значення x Бобові;
5) Боб перевіряє, чи $y = f(x)$.

Надійність цього протоколу залежить від вибору функції f. Якщо Аліса може знайти x і x' такі, що x — парне число, x' — непарне і $y = f(x) = f(x')$, то зможе обманювати Боба в кожному підкидуванні монети. Крім того, Аліса повинна стежити, щоб число x приймало парні і непарні значення з однаковою імовірністю.

У наведеній вище схемі неявно реалізовано процедуру зобов'язання бітів. Аліса і Боб для «підкидання монети по телефону» могли б поступити простіше: Аліса вибирає деяке значення біта, вкладає його у конверт і посилає Бобові. Коли лист вже в дорозі, то Боб по телефону відгадує значення цього біта.

Означення 4.29. *Протоколом зобов'язання бітів називають процедуру, за якою Аліса передає на зберігання в «конверті» таємний біт (тобто, 0 або 1), який Боб намагається відгадати. Боб не може зробити ймовірність вгадування значення біта більшою ніж* $1/2$*, а Аліса не може змінити свого біта після вкладання його у «конверт».*

На основі важкооборотної функції протокол зобов'язання бітів можна реалізувати так.

1. Аліса генерує дві випадкові послідовності бітів R_1 і R_2.
2. Далі вона утворює повідомлення (R_1, R_2, b), яке складається з вибраних чисел R_1, R_2 та біта b, щодо якого хоче виконати зобов'язання бітів.
3. Аліса обчислює вкорочене значення $H(R_1, R_2, b)$ і посилає його Бобові разом з одним із вибраних чисел (R_1).

Ця операція є підтвердженням того, що Аліса виконала «зобов'язання». Позаяк на третьому кроці використовується важкооборотна функція, то Боб не може її обернути і «здобути» біт. Коли приходить час розголошення біта Аліси, протокол продовжується.

4. Аліса посилає Бобові початкове повідомлення (R_1, R_2, b).
5. Боб обчислює його вкорочене значення та порівнює його і R_1 з повідомленням, одержаним на кроці 3. Якщо значення збігаються, то біт правильний.

Зауважимо, що Аліса не може змінити значення біта b на b'. Тобто, вона не може знайти інше повідомлення (R_1, R_2', b') таке, що $H(R_1, R_2', b') =$

$H(R_1, R_2, b)$. Якби Аліса не вислала Бобові R_1, то могла б змінити обидва значення R_1 і R_2 та підмінити значення біта.

Тепер процедуру підкидання монети по телефону можна реалізувати так.

1) Аліса «вкладає у конверт» випадкове значення біта, використовуючи протокол зобов'язання бітів;
2) Боб пробує відгадати значення цього біта;
3) Аліса відкриває біт Бобові. Якщо Боб відгадав значення біта, то він перемагає в цьому раунді підкидування монети.

4.3.6. Доведення без розголошення, ідентифікація

Розглянемо таку ситуацію: припустимо, що Боб зламав систему захисту комп'ютерної мережі деякого банку, знає розв'язання деякої складної задачі, чи володіє іншою секретною інформацією (наприклад, знає зміст четвертого тому комп'ютерної біблії Кнута). Боб хоче похизуватись цим перед Алісою, але зробити це так, щоб вона не отримала жодної секретної інформації. Наміри Боба переконати Алісу в тому, що він володіє секретною інформацією, не розкриваючи при цьому її, здаються суперечливими та нездійсненими. Однак це завдання можна реалізувати, скориставшись протоколом *доведення без розголошення (zero-knowledge proof)*.

«Доведення» ґрунтується на певній процедурі, яка повторюється кілька разів. Припустимо, що інформація, якою володіє Боб — це розв'язання деякої складної задачі. Тоді:

1. Боб використовує свою секретну інформацію і деяке випадкове число для перетворення складної задачі в іншу складну задачу, яка еквівалентна першій. Далі він розв'язує нову задачу за допомогою відомого йому числа і розв'язання першої задачі.
2. Боб виконує зобов'язання щодо розв'язку нової задачі за допомогою деякого протоколу зобов'язання бітів.
3. Боб відкриває Алісі нову складну задачу, причому вона не може на підставі цієї інформації одержати будь-яку інформацію про початкову задачу чи її розв'язання.
4. Аліса просить Боба довести, що:
   а) перша і друга задачі є ізоморфними;

б) відкрити розв'язання, яке міститься в зобов'язанні на кроці 2, і довести, що це розв'язок нової задачі.

5. Боб виконує вимоги Аліси, тобто доводить один з двох пунктів а,б.
6. Аліса і Боб повторюють кроки 1–5 n-кратно.

Математичні доведення такого типу досить складні. Задачу та випадкове число треба вибирати обережно для того, щоб Аліса не могла одержати ніякої інформації про розв'язання початкової задачі, навіть після багатократного повторення протоколу. Зрозуміло, що не кожну «складну задачу» можна використати для реалізації протоколу доведення без розголошення.

Наведемо два приклади доведення без розголошення, які ґрунтуються на задачі відшукання *гамільтонового циклу* в деякому графі та задачі розпізнавання квадратичних лишків.

Приклади

1. Цей приклад вперше запропонував М. Блюм. Припустимо, що Боб знає цикл гамільтона у графі $G = (V, E)$, тобто такий маршрут, який починається і закінчується в деякій вершині та проходить через всі вершини один раз. Для дуже великого графа, знаходження гамільтонового шляху, навіть з використанням комп'ютера, може розтягнутись на роки. Нехай Боб хоче переконати Алісу, що він знає такий маршрут. Тоді:
   1. Боб довільно переставляючи вершини графа G та змінюючи їх назви утворює новий граф G'. Оскільки графи G і G' — ізоморфні, то знаючи гамільтоновий маршрут у графі G, Боб може легко знайти гамільтоновий цикл у графі G'. Оскільки Боб сам утворив граф G', то для нього доведення ізоморфізму графів G і G' є тривіальною задачею. Для кожної іншої особи це складна задача.
   2. Боб посилає Алісі копію графа G'.
   3. Аліса просить виконати Боба одну з двох дій:
      а) довести, що графи G і G' — ізоморфні;
      б) показати гамільтоновий цикл у графі G'.
   4. Боб виконує вимоги Аліси:
      а) доводить, що $G \cong G'$, не зазначаючи гамільтоновий маршрут у жодному графі;
      б) показує гамільтоновий цикл у графі G', не доводячи ізоморфізму G і G'.
   5. Боб і Аліса повторюють кроки 1–4 n разів.

   Якщо Боб не ошукує Алісу, то він її переконує з імовірністю 1, надавши одне з доведень пункту 3. Якщо Боб не знає гамільтонового циклу у графі G, то він не може побудувати граф G', який задовольняє умови а і б одночасно. У кращому випадку Боб може або утворити G' ізоморфний G, або утворити граф G' з

такою ж кількістю вершин, як у G і з відомим гамільтоновим циклом. Тобто, імовірність того, що Аліса викриє Боба в кожному циклі описаної процедури дорівнює $1/2$. Повторивши кроки 1–4 n разів, Аліса може бути впевнена з імовірністю $1-\frac{1}{2^n}$ в тому, що Боб знає розв'язання складної задачі. Очевидно, описаний протокол є доведенням без розголошення, бо Аліса ніколи не отримає інформації, яка б помогла їй знайти гамільтоновий цикл у графі G.

2. Розглянемо реалізацію протоколу доведення без розголошення на основі задачі розпізнавання квадратичних лишків (тобто, визначення чи для пари натуральних чисел (x,n) існує число y таке, що $y^2 \equiv x \pmod{n}$). Якщо невідомо факторизації модуля n, то ця задача є складною.
   Припустимо, що Боб знає y — корінь квадратний з x за модулем n і хоче переконати в цьому Алісу, невідкриваючи їй значення y. Тоді
   1. Боб вибирає довільне взаємно просте з n число v, підносить його до квадрата і результат $u = v^2 \pmod{n}$ посилає Алісі.
   2. Аліса посилає Бобові випадковий біт $b \in \{0,1\}$.
   3. Боб посилає Алісі число $w = \begin{cases} v, & \text{якщо } b=0; \\ vy, & \text{якщо } b=1. \end{cases}$
   4. Якщо $b=0$, то Аліса перевіряє, чи справді $u \equiv w^2 \pmod{n}$, а якщо $b=1$, то перевіряє, чи справді $xu \equiv w^2 \pmod{n}$.
   5. Аліса та Боб повторюють кроки 1–4 m разів.

   Якщо у кожному з m циклів на 4 кроці результат перевірки успішний, то Бобові вдалось переконати Алісу в тому, що він знає розв'язання складної задачі. Ймовірність того, що Боб зможе ошукати Алісу, дорівнює $\frac{1}{2^m}$ і швидко спадає з ростом m.

Зауважимо, що у наведених двох прикладах використовується метод розділяй і вибирай (cut and choose). Вперше використовувати цю процедуру в криптографії запропонував М.Рабін.

Система гасел та ідентифікація. Загальновідомим розв'язком задачі ідентифікації є використання гасла. Недоліком такого найпростішого вирішення проблеми є те, що суперник може підслухати гасло. Як ми вже зазначали (див. п. 4.3.3.), протокол цифрового підпису позбавлений цього недоліку і розв'язує завдання ідентифікації поряд з іншими. Цифровий підпис, як і криптосистеми з відкритим ключем, використовує важкооборотну функцію з секретом. Насправді, систему ідентифікації можна побудувати за допомогою лише важкооборотної функції.

Наприклад, нехай f — деяка важкооборотна функція. Якщо гасла x_i абонентів мережі зберігати в комп'ютері, то їх розкриття стане легкою задачею для будь-якого хакера. Для того щоб список гасел зберігати в таємниці, поряд з іменем i-го абонента зберігають значення $f(x_i)$. Кожного разу, коли користувач хоче отримати доступ до мережі,

вводить свій пароль x_i. Комп'ютер обчислює значення $f(x_i)$, порівнює його з відповідним значенням із списку гасел, надає доступ до відповідних ресурсів чи пристроїв мережі і знищує всі дані про x_i. Очевидно, що задача ламання такої системи гасел зводиться до задачі обчислення значення $x = f^{-1}(y)$, яка є складною задачею в силу важкооборотності функції f.

Процедуру ідентифікації можна також реалізувати на основі протоколу доведення без розголошення. Для цього кожен абонент мережі оприлюднює деяке твердження, доведення якого відоме лише йому.

Якщо деякий абонент (Аліса) хоче пересвідчитися в особі іншого абонента (Боба), то просить співрозмовника довести Бобове твердження за допомогою схеми доведення без розголошення.

Знаючи доведення твердження, Боб переконує Алісу в кожному з n циклів протоколу. Ймовірність того, що це зробить самозванець дорівнює 2^{-n} і є досить малою при великих n. Оскільки доведення було без розголошення, то навіть підслухавши сеанс ідентифікації Боба Алісою, суперник не зможе повторити його замість Боба.

Кожна задача, яку я розв'язував,
стає правилом, яке слугуватиме
до розв'язання інших задач

Р. Декарт

Вправи до Розділу 1

1.1. Множини

1.1.1. Довести тотожності:
1) $\overline{(A \cap B)} = (\overline{A}) \cup (\overline{B})$;
2) $\overline{(A \cup B)} = (\overline{A}) \cap (\overline{B})$;
3) $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$;
4) $A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C)$;
5) $A \setminus (A \setminus B) = A \cap B$;
6) $(A \setminus B) \setminus C = (A \setminus C) \setminus (B \setminus C)$;
7) $A \cap (B \setminus C) = (A \cap B) \setminus (A \cap C) = (A \cap B) \setminus C$;
8) $A \cup (B \setminus C) = (A \setminus C) \cup (B \setminus C)$.

1.1.2. Нехай $A = B \cup C$ і $B \cap C = \emptyset$. Довести, що $B = A \setminus C$ і $C = A \setminus B$.

1.1.3. Чи існують такі множини A, B, C, що $A \cap B \neq \emptyset$, $A \cap C = \emptyset$ і $(A \cap B) \setminus C = \emptyset$?

1.1.4. Нехай $A \subset C$ і $B \subset C$. Довести, що $A \cup B \subset C$ і $A \cap B \subset C$.

1.1.5. За означенням $A \oplus B = (A \cup B) \setminus (A \cap B)$. Знайти необхідні та достатні умови для того, щоб $A \oplus B = A \cup B$.

1.1.6. Довести, що $A = A \cap (A \cup B)$ і $A = A \cup (A \cap B)$.

1.1.7. Довести, що $A \subset B \subset C$ тоді й лише тоді, коли $(C \setminus A) \cup B = C$.

1.1.8. Довести, що з кожного з трьох співвідношень $A \subset B, A \cap B = A$, $A \cup B = B$ випливають два інші.

1.1.9. Довести, що $A \oplus B = C \Leftrightarrow B \oplus C = A \Leftrightarrow C \oplus A = B$.

1.1.10. Довести, що $A \oplus B = B \oplus A, (A \oplus B) \oplus C = A \oplus (B \oplus C), A \oplus A = \emptyset$.

1.1.11. Характеристичною функцією підмножини A множини C називають відображення $f : C \to \{0, 1\}$, де $f(x) = \begin{cases} 1, x \in A, \\ 0, x \notin A. \end{cases}$ Якщо f і g характеристичні функції підмножин A і B множини C, то якими будуть підмножини з характеристичними функціями $1 - f,\ fg,\ f + g - fg$?

1.1.12. Довести, що підмножини будь-якої множини C утворюють кільце щодо операцій додавання $A \oplus B = (A \cup B) \setminus (A \cap B)$ та множення $A \cdot B = A \cap B$. Що є відніманням у цьому кільці?

1.1.13. Для кожного натурального n записати множину A_n з n елементів, якщо $x, y \in A_n$, то $x \in y$ або $y \in x$ або $x = y$.

1.1.14. Розв'язати систему рівнянь $\begin{cases} A \cap X = B \\ A \cup X = C \end{cases}$, де A, B, C – задані множини і $B \subset A \subset C$.

1.1.15. Розв'язати систему рівнянь $\begin{cases} A \setminus X = B \\ X \setminus A = C \end{cases}$, де A, B, C – задані множини і $B \subset A,\ A \cap C = \emptyset$.

1.1.16. Чи правильні для множин A, B, C такі рівності: $A \times B = B \times A$, $A \times (B \times C) = (A \times B) \times C$, $A \times (B \cup C) = (A \times B) \cup (A \times C)$?

1.1.17. Нехай A_i, $(i \in I)$, B підмножини множини X. Довести, що $\left(\bigcup_{i \in I} A_i\right) \cap B = \bigcup_{i \in I} (A_i \cap B)$.

1.1.18. Довести, що $\left(\bigcap_{i \in I} A_i\right) \cup B = \bigcap_{i \in I} (A_i \cup B)$.

1.1.19. Яку максимальну кількість підмножин можна утворити з даних n підмножин за допомогою операцій об'єднання, перетину та доповнення?

1.1.20. Довести, що $(A_1 \cup \ldots \cup A_n) \oplus (B_1 \cup \ldots \cup B_n) \subset (A_1 \oplus B_1) \cup \ldots \cup (A_n \oplus B_n)$.

1.1.21. Довести, що $(A_1 \cap \ldots \cap A_n) \oplus (B_1 \cap \ldots \cap B_n) \subset (A_1 \oplus B_1) \cap \ldots \cap (A_n \oplus B_n)$.

1.1.22. Виразити операції $\cup, \cap$ та $\setminus$ за допомогою $\oplus$ і $\cap$.

1.1.23. Довести, що не можна виразити:
1) операцію $\setminus$ за допомогою операцій $\cup$ і $\cap$;
2) операцію $\cup$ за допомогою операцій $\setminus$ і $\cap$.

1.1.24. Довести, що $A = B \Leftrightarrow A \oplus B = \emptyset$.

1.1.25. Нехай маємо ланцюжок множин $X_0 \supset X_1 \supset \ldots \supset X_n \supset \ldots$. Довести, що перетин всіх множин з кожного нескінченного підланцюжка цього ланцюжка збігається з перетином всіх множин цього ланцюжка.

1.1.26. Для ланцюжка множин з попередньої задачі довести, що будь-який скінченний перетин множин з ланцюжка збігається з однією з цих множин.

1.1.27. Нехай маємо ланцюжок множин $X_0 \subset X_1 \subset \ldots \subset X_n \subset \ldots$. Довести, що об'єднання всіх множин з кожного нескінченного підланцюжка цього ланцюжка збігається з об'єднанням всіх множин заданого ланцюжка.

1.1.28. Довести, що:
1) $2^{A\cap B} = 2^A \cap 2^B$; 2) $2^{A\cup B} = 2^A \cup 2^B$.

1.1.29. Довести, що для довільних множин $A_1, \ldots, A_n$, якщо $A_1 \subset A_2 \subset \ldots \subset A_n \subset A_1$, то $A_1 = A_2 = \ldots = A_n$.

1.1.30. Довести, що існує взаємно однозначна відповідність між такими множинами:
1) $A \times B$ і $B \times A$;
2) $A \times (B \times C)$ і $(A \times B) \times C$;
3) $(A \times B)^C$ і $A^C \times B^C$;
4) $\left(A^B\right)^C$ і $A^{B\times C}$;
5) $A^{B\cup C}$ і $A^B \times A^C$, якщо $B \cap C = \emptyset$.

1.2. Відношення

1.2.1. Скільки бінарних відношень існує на множині з n елементів?

1.2.2. Наведіть приклади відношень, які мають або тільки одну, або тільки дві з трьох властивостей: рефлексивність, симетричність, транзитивність.

1.2.3. Чи є перетин рефлексивних (відповідно симетричних, транзитивних) відношень рефлексивним (відповідно симетричним, транзитивним) відношенням?

1.2.4. Дати відповідь на запитання попередньої вправи для випадку об'єднання відношень.

1.2.5. Нехай R_1, R_2 відношення еквівалентності. Довести, що $R_1 \cap R_2$ теж відношення еквівалентності. Чи є відношенням еквівалентності відношення $R_1 \cup R_2$?

1.2.6. Скільки відношень еквівалентності можна побудувати на множині $\{1,2,3,4\}$?

1.2.7. Чи є перетин та об'єднання двох відношень часткового порядку, визначених на тій самій множині, знову відношенням часткового порядку?

1.2.8. Наведіть приклади відношень часткового порядку, які є одночасно і відношеннями еквівалентності.

1.2.9. Визначити, які з наведених відношень на множині натуральних чисел $\mathbb{N}$ рефлексивні, симетричні, антисиметричні, транзитивні:

1) $m+n$ — парне; 2) $m+n \leq 7$;
3) $m+n$ — непарне; 4) $\frac{m}{n}$ — степінь двійки;
5) $\frac{m}{n}$ — нескоротний дріб; 6) mn — непарне.

1.2.10. Нехай R_1, R_2 рефлексивні та симетричні відношення. Показати, що такі три умови еквівалентні:

1) $R_1 R_2$ — симетричне;
2) $R_1 R_2 = R_2 R_1$;
3) $R_1 R_2 = R_1 \cup R_2$.

1.2.11. Побудувати два симетричних відношення на множині $\{1,2,3\}$, добуток яких несиметричний. Чи утворюють бінарні відношення на заданій множині групу щодо добутку відношень?

1.2.12. Довести таке: коли відношення R симетричне, то і відношення $R \cup R^2 \cup ... \cup R^n$ теж симетричне.

1.2.13. Нехай f і g — функції. За яких умов f^{-1} є функцією? За яких умов gf — взаємно однозначна відповідність?

1.2.14. Показати таке: коли R — відношення еквівалентності, то R^{-1} теж відношення еквівалентності.

1.2.15. Довести таке: коли R_1, R_2 відношення еквівалентності на множині A, то:

1) $R_1R_2 = A \times A \Leftrightarrow R_1 = A \times A$;
2) $R_1R_2 = A \times A \Leftrightarrow R_2R_1 = A \times A$.

1.2.16. Довести, що об'єднання $R_1 \cup R_2$ двох відношень еквівалентності є відношенням еквівалентності тоді і тільки тоді, коли $R_1R_2 = R_2R_1$.

1.2.17. Нехай відношення $\leq, <$ на множині натуральних чисел $\mathbb{N}$ визначені звичайним способом. Довести, що

$$< \circ <\neq<, \quad \leq \circ <=<, \quad \leq \circ \geq = \mathbb{N} \times \mathbb{N}.$$

1.2.18. Нехай $P = (S, \leq)$, $Q = (T, \leq)$ дві частково впорядковані множини. $P \times Q = (S \times T, \leq)$ множина для якої $(s,t) \leq (s',t')$ означає, що $s \leq s'$ в P і $t \leq t'$ в Q. Довести, що $P \times Q$ частково впорядкована множина.

1.2.19. Довести, що $p_{n+1} = \sum\limits_{i=0}^{n} C_n^i p_i$, де p_n — кількість різних відношень еквівалентності на множині з n елементів.

1.2.20. Нехай $P = (S, \leq)$, $Q = (T, \leq)$ дві частково впорядковані множини. $P \times Q = (S \times T, \leq)$ множина для якої $(s,t) \leq (s',t')$ означає, що або $s \leq s'$ в P або $s = s'$ і $t \leq t'$ в Q. Довести, що $P \otimes Q$ частково впорядкована множина.

1.2.21. Дві множини A і B евклідової площини називають ізометричними, якщо існує бієкція $f : A \to B$, що зберігає відстані між точками. Довести, що ізометричність є відношенням еквівалентності.

1.2.22. Довести, що у множини, яка складається з n елементів, існує $2^{n-1} - 1$ розбиттів на два класи еквівалентності.

1.2.23. Нехай $\pi(n,k)$ кількість розбиттів множини, яка складається з n елементів на k непорожніх підмножин. Довести, що $\sum\limits_{k=1}^{l} k!\pi(n,k) = l^n$.

1.2.24. Чи є функціональними відношеннями добуток двох функціональних відношень та відношення, обернене до функціонального?

1.2.25. Довести, що об'єднання (перетин) двох функцій f_1, f_2 з A в B є функцією з A в B тоді і тільки тоді, коли $f_1 = f_2$.

1.2.26. Довести, що для кожної функції f
1) $f(A \cup B) = f(A) \cup f(B)$;
2) $f(\cup_{i\in I} A_i) = \cup_{i\in I} f(A_i)$.

1.2.27. Довести, що для кожної функції f
1) $f(A \cap B) \subset f(A) \cap f(B)$;
2) $f(\cap_{i\in I} A_i) \subset \cap_{i\in I} f(A_i)$.

1.2.28. Нехай f функція. Довести, що $f(A) \setminus f(B) \subset f(A \setminus B)$.

1.2.29. Нехай f функція. Довести рівності
1) $f^{-1}(A \cup B) = f^{-1}(A) \cup f^{-1}(B)$;
2) $f^{-1}(A \cap B) = f^{-1}(A) \cap f^{-1}(B)$.

1.2.30. Довести, що відображення $f : A \to B$ має обернене зліва тоді і тільки тоді, коли воно ін'єктивне.

1.2.31. Довести, що відображення $f : A \to B$ має обернене справа тоді і тільки тоді, коли воно сюр'єктивне.

1.2.32. Довести, що існує бієктивна відповідність між множиною всіх відображень множини A в множину $\{0,1\}$ і множиною 2^A.

1.2.33. Нехай $|A| = m$, $|B| = n$. Визначити кількість ін'єктивних та кількість сюр'єктивних відображень з множини A в множину B.

1.2.34. Задано відображення $f : \mathbb{N} \to \mathbb{N}$, $f(n) = \frac{1}{2}((-1)^n + 1)$ множини натуральних чисел. Знайти $f^{-1}(0)$ та $f^{-1}(1)$.

1.2.35. Нехай $\mathbb{N}$ множина натуральних чисел. $f : \mathbb{N}^2 \to \mathbb{N}$, $f(m,n) = 2^m 3^n$. Перевірити, що відображення f ін'єктивне, але не сюр'єктивне.

1.2.36. Нехай $\mathbb{N}$ множина натуральних чисел. $f : \mathbb{N}^2 \to \mathbb{N}$, $f(m,n) = 2^m 4^n$. Перевірити, що відображення f не ін'єктивне і не сюр'єктивне.

1.2.37. Розглянемо такі відображення множини натуральних чисел в себе: $\mathrm{I}(n) = n$, $f(n) = 2n+1$, $g(n) = n + (-1)^n$, $h(n) = \min\{n, 100\}$, $k(n) = \max\{0, n = 3\}$. Які з цих відображень ін'єктивні, сюр'єктивні, бієктивні?

1.2.38. Нехай f , g , $h : \mathbb{R} \to \mathbb{R}$ відображення множини дійсних чисел в себе: $f(x) = x^3 - 4x$, $g(x) = \left(x^2+1\right)^{-1}$, $h(x) = x^4$. Знайти $f \circ g \circ h$, $f \circ h \circ g$, $h \circ g \circ f$, $f \circ f$, $g \circ g$, $h \circ h$, $g \circ h$, $h \circ g$.

1.2.39. Нехай f , $g : \mathbb{Z} \to \mathbb{Z}$ два відображення з множини цілих чисел в себе. $f(n) = n - 1$, g — характеристична функція підмножини парних

чисел. Знайти $f \circ f$, $f \circ g$, $g \circ g$, $g \circ f$. Чи правильно, що $g \circ f = f \circ f$, $g \circ g = g \circ f$, $f \circ g = g \circ f$?

1.2.40. Нехай A одна з трьох множин: раціональних, дійсних або комплексних чисел. $f : A \to A$ одна з чотирьох функцій: $f(x) = 2x + 3$, $x^3 - 2$, $(x-2)^3$, $x^{\frac{1}{3}} + 7$. Які з цих 12 функцій є відображеннями? Для яких з цих відображень існують обернені?

1.2.41. Нехай $\mathbb{N}$ множина натуральних чисел. $f : 2^{\mathbb{N}} \times 2^{\mathbb{N}} \to 2^{\mathbb{N}}$ одне з чотирьох відображень: $f(A,B) = A \cup B$, $A \cap B$, $A \oplus B$, $A \setminus B$. Які з цих відображень ін'єктивні чи сюр'єктивні? Знайти $f^{-1}(\emptyset)$.

1.2.42. Нехай $\mathbb{N}$ множина натуральних чисел. $f : \mathbb{N}^2 \to \mathbb{N}$ одне з чотирьох відображень: $f(m,n) = m + n$, mn, $\max\{m,n\}$, $\min\{m,n\}$. Які з цих відображень ін'єктивні чи сюр'єктивні? Знайти $f^{-1}(4)$.

1.2.43. Нехай $\mathbb{N}$ множина натуральних чисел. $f, g : \mathbb{N} \to \mathbb{N}$. Показати, що $f(n) = n+1$, $g(n) = \max\{0, n-1\}$ є відображеннями. Які з цих відображень ін'єктивні чи сюр'єктивні? Показати, що $g \circ f = \mathrm{I}_{\mathbb{N}}$, $f \circ g \neq \mathrm{I}_{\mathbb{N}}$.

1.2.44. Нехай $f : S \to T$, $g : T \to S$ відображення для яких $g \circ f = \mathrm{I}_S$. Довести: а) f — ін'єктивне відображення, а g — не обов'язково ін'єктивне; б) f — сюр'єктивне відображення, а g — не обов'язково сюр'єктивне.

1.2.45. Розглянемо відображення $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \times \mathbb{R}$, де $\mathbb{R}$ — множина дійсних чисел. $f((x,y)) = (x+y, x-y)$. Показати, що f бієктивне. Знайти f^{-1}.

1.2.46. Нехай $f : S \to T$ відображення. Довести, що:

1) $f\left(f^{-1}(B)\right) \subset B$;
2) $A \subset f^{-1}(f(A))$;
3) $f^{-1}(B_1 \cap B_2) = f^{-1}(B_1) \cap f^{-1}(B_2)$.

За яких умов у твердженні 1 маємо рівність?

1.2.47. Нехай $f : S \to T$ функція. Довести або спростувати такі твердження:

1) $f(A_1 \cap A_2) = f(A_1) \cap f(A_2)$;
2) $f(A_1 \setminus A_2) = f(A_1) \setminus f(A_2)$;
3) $f(A_1) = f(A_2) \Rightarrow A_1 = A_2$.

1.2.48. Нехай відображення $f : T \to S$ ін'єктивне та відображення $g : S \to T$, $h : S \to T$ такі, що $f \circ g = f \circ h$. Тоді $g = h$.

1.2.49. Навести приклади відображень f, g, h таких, що $f \circ g = f \circ h$. і $g \neq h$.

1.2.50. Навести приклади відображень f, g, h таких, що $g \circ f = h \circ f$, але $g \neq h$. Сформулювати умови, за яких $g \circ f = h \circ f \Rightarrow g = h$.

1.2.51. Розглянемо відображення $f, g, h : \mathbb{Z} \to \mathbb{Z}$ з множини цілих чисел $\mathbb{Z}$ в себе, $f(n) = 3n, g(n) = 3n + 1, h(n) = 3n + 2$. Побудувати відображення, яке було б оберненим зліва до f, g і h одночасно.

1.2.52. Довести таке: коли добуток відображень $g \circ f$ визначений і обидва відображення f і g мають ліві обернені, то і $g \circ f$ має ліве обернене. Підтвердити на прикладі, що обернене твердження загалом неправильне.

1.2.53. Скільки існує сюр'єктивних відображень з трьохелементної множини в двоелементну?

1.2.54. Нехай $|A| = m, |B| = n, m \geq n$. Довести, що кількість сюр'єктивних відображень з множини A в множину B дорівнює $\sum (\text{-}1)^k C_n^k (n\text{–}k)^m$.

1.2.55. Довести, що неперервна дійсна функція $f : [a, b] \to [a, b]$ ін'єктивна тоді і тільки тоді, коли вона строго монотонна.

1.2.56. Довести, що неперервна дійсна функція $f : [a, b] \to [a, b]$ бієктивна тоді і тільки тоді, коли вона строго монотонна, і або $f(a) = c, f(b) = d$ або $f(a) = d, f(b) = c$.

1.2.57. Довести, що ін'єктивне відображення скінченної множини в себе є бієктивним.

1.2.58. Довести, що сюр'єктивне відображення скінченної множини в себе є бієктивним.

1.2.59. Довести таке: якщо добуток відображень $g \circ f$ скінченної множини в себе бієктивний, то f і g бієктивні відображення.

1.2.60. Якщо для цього відображення $f : A \to A$ існує таке натуральне число $n > 0$, що f^n — одиничне відображення, то f бієктивне відображення. У яких випадках правильне обернене твердження?

1.2.61. Нехай $f : M_n(\mathbb{R}) \to M_n(\mathbb{R})$ відображення множини квадратних матриць n-го порядку з дійсними елементами в себе, $f(X) = AX$, де A — фіксована матриця, $A \in M_n(\mathbb{R})$. У яких випадках це відображення

бієктивне? Довести, що для відображення f властивості бути ін'єктивним, сюр'єктивним або бієктивним рівносильні.

1.2.62. Довести, що кількість всіх відображень скінченної множини A з n елементів у множину $\{0,1\}$ збігається з кількістю підмножин множини A.

1.2.63. Нехай K кільце, a — фіксований елемент кільця K. Розглянемо відображення $f : K \to K$, для якого $f(x) = ax$. У яких випадках відображення f ін'єктивне (сюр'єктивне, бієктивне)?

1.2.64. Нехай f — відображення множини A в себе, $a \in A$. Послідовність $a_0 = a, a_1 = f(a_0), \ldots, a_n = f(a_{n-1}), \ldots$ називають орбітою елемента a для відображення f. Довжиною орбіти називають кількість її різних елементів. Знайти найбільше і найменше значення сум довжин різних орбіт для відображень множини з n елементів.

1.2.65. Довести, що відображення f скінченної множини A в себе є бієктивним тоді і тільки тоді, коли сума довжин різних його орбіт (див. попередню задачу) дорівнює $|A|$.

1.2.66. Порядком підстановки $\varphi \in S_n$ називають найменше натуральне число k, таке що φ^k є одиничною підстановкою. Який найбільший порядок можуть мати підстановки на множині з 10 елементів?

1.2.67. Скільки існує підстановок 15 порядку на множині з восьми елементів?

1.2.68. Якщо підстановки φ і ψ комутують при множенні, то довести, що порядок (див. вправу **1.2.66.**) підстановки $\psi \circ \varphi$ є дільником найменшого спільного кратного порядків φ і ψ.

1.2.69. Довести, що для кожної підстановки φ, яка розкладається в добуток l циклів однакової довжини s, знайдеться цикл ψ довжини ls і натуральне число k, для якого $\varphi = \psi^k$. Чи єдиний такий цикл?

1.2.70. Довести, що всі цикли довжини три разом з будь-якою транспозицією утворюють систему твірних симетричної групи S_n.

1.2.71. Чи утворюють систему твірних симетричної групи S_9 підстановки $(1,2,3)$ і $(1,2,3,\ldots,9)$?

1.2.72. Скільки підгруп другого порядку має група S_5?

1.3. Навколо леми Цорна

1.3.1. Довести, що кожна частково впорядкована множина містить не більше ніж один найбільший (найменший) елемент.

1.3.2. Довести, що найбільший (найменший) елемент частково впорядкованої множини є її єдиним максимальним (мінімальним) елементом.

1.3.3. Побудувати приклад частково впорядкованої множини, що не має найменшого елемента, але має єдиний мінімальний елемент.

1.3.4. Довести, що в кожній скінченній частково впорядкованій множині існують максимальний і мінімальний елементи.

1.3.5. Довести, що множину з n елементів можна цілком впорядкувати $n!$ способами.

1.3.6. Дві множини називають подібними, якщо існує бієктивне відображення цих множин, яке зберігає порядок. Довести, що всі скінченні лінійно впорядковані множини подібні.

1.3.7. Довести, що для нескінченних лінійно впорядкованих множин твердження попередньої вправи неправильне.

1.3.8. Довести, що множина всіх відрізків цілком (лінійно) впорядкованої множини A, впорядкована відношенням включення, подібна до множини A.

1.3.9. Довести, що нескінченна лінійно впорядкована множина має порядковий тип ω ($\aleph_0$) (тобто подібна до множини $\mathbb{N}$ натуральних чисел із звичайним впорядкуванням) тоді й лише тоді, коли всі її початкові відрізки скінченні.

1.3.10. Довести, що кожна зліченна лінійно впорядкована множина подібна до деякої підмножини множини $\mathbb{Q}$ раціональних чисел.

1.3.11. Довести, що будь-який інтервал (не сегмент) подібний до множини $\mathbb{R}$ дійсних чисел.

1.3.12. Навести приклади порядкових типів α і β для яких: 1) $\alpha+\beta \neq \beta+\alpha$; 2) $\alpha\beta \neq \beta\alpha$.

1.3.13. Нехай α, β, γ порядкові типи. Довести, що 1) $(\alpha+\beta)+\gamma = \alpha+(\beta+\gamma)$; 2) $(\alpha\beta)\,\gamma = \alpha\,(\beta\gamma)$.

1.3.14. Нехай $\omega, \pi, \eta, \lambda$ порядкові типи, відповідно, множин натуральних чисел, цілих чисел, раціональних чисел і дійсних чисел. Довести, що: а) $1+\omega=\omega$; б) $\omega+1\neq\omega$; в) $\eta+\eta=\eta$; г) $\lambda+1+\lambda=\lambda$; д) $\lambda+\lambda\neq\lambda$; е) $\eta^2=\eta$; є) $1+\lambda+1\neq\lambda$; ж) $\omega\eta\neq\omega\,(\eta+1)$; з) $\eta\cdot 2=\eta$.

1.3.15. Довести, що кожна скінченна лінійно впорядкована множина є цілком впорядкованою.

1.3.16. Довести, що множина натуральних чисел $\mathbb{N}$, де $0<1<2<\ldots$, цілком впорядкована.

1.3.17. Довести, що множина натуральних чисел $\mathbb{N}$, де $\ldots<3<2<1<0$, не цілком впорядкована.

1.3.18. Довести, що множина натуральних чисел $\mathbb{N}$, де $0<2<4<\ldots 1<3<5<\ldots$, не цілком впорядкована.

1.3.19. Чи цілком впорядкована множина $\mathbb{Z}$ цілих чисел щодо звичайного порядку? Означити на множині $\mathbb{Z}$ відношення порядку, щодо якого ця множина є цілком впорядкованою.

1.3.20. Чи є цілком впорядкованою множина $\mathbb{Q}$ раціональних чисел щодо звичайного порядку?

1.3.21. Чи є цілком впорядкованою множина чисел $1-1/n,\ n\geq 1$ щодо звичайного порядку?

1.3.22. Чи можна в цілком впорядкованій множині виділити нескінченний спадний ланцюг елементів $x_1>x_2>x_3>\ldots$?

1.3.23. Довести, що лінійно впорядкована множина є цілком впорядкованою тоді й лише тоді, коли вона не містить підмножини, подібної до підмножини $\ldots<-3<-2<-1<0$.

1.3.24. Довести, що кожне ординальне число дорівнює числу $\alpha+n$, де α граничне ординальне число або нуль, а n натуральне число.

1.3.25. Навести приклади ординальних чисел α, β і γ, для яких $\alpha\neq\beta$ і $\alpha+\gamma=\beta+\gamma$.

1.3.26. Побудувати множину з порядковим типом ω^ω.

Вправи до Розділу 2

2.1. Натуральні числа

2.1.1. Нехай $a, b \in \mathbb{N}$, $a \neq 0$. Довести, що $a + b \neq b$.

2.1.2. Нехай $a, b, c \in \mathbb{N}, c \neq 0$. Довести, що $a \neq b \Rightarrow a + c \neq b + c$.

2.1.3. Означимо a^n для $a, n \in \mathbb{N} : a^0 = 1, a^{n+1} = a^n a$. Довести, що:
1) $1^n = 1$; 2) $a^m a^n = a^{m+n}$;
3) $(ab^n) = a^n b^n$; 4) $(a^m)^n = a^{mn}$.

2.1.4. Довести, що кожна непорожня підмножина множини натуральних чисел має найменший елемент.

2.1.5. Нехай $P(m, n)$ множина висловлень занумерованих парами натуральних чисел $(m, n) \in \mathbb{N}^2$. Припустимо, висловлення $P(0, 0)$ істинне і що з істинності висловлення $P(m, n)$ для всіх $(m, n), m \leq a, n \leq b$, випливає істинність $P(a, b)$. Довести, що тоді висловлення $P(m, n)$ істинне для всіх $(m, n) \in \mathbb{N}^2$.

2.1.6. Довести тотожності:
1) $\sum\limits_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$; 2) $\sum\limits_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2$;
3) $\frac{(2n)!}{n!} = 2^n (2n-1)!!$;
4) $\frac{1}{1\cdot 5} + \frac{1}{5\cdot 9} + \ldots + \frac{1}{(4n-3)(4n+1)} = \frac{n}{4n+1}$;
5) $\frac{1}{n+1} + \frac{1}{n+2} + \ldots + \frac{1}{2n} = 1 - \frac{1}{2} + \frac{1}{3} - \ldots + \frac{1}{2n-1} - \frac{1}{2n}$;
6) $3 + 11 + \ldots + (8n - 5) = 4n^2 - n$;
7) $\sum\limits_{i=1}^{n} i(i+1) = \frac{n(n+1)(n+2)}{3}$;

8) $\left(1-\frac{4}{1}\right)\left(1-\frac{4}{9}\right)\ldots\left(1-\frac{4}{(2n-1)^2}\right)=\frac{1+2n}{1-2n}$;
9) $5+45+\ldots+(4n+1)\,5^{n-1}=n5^n$;

2.1.7. Довести: а) $6 \mid n^3-7n$; б) $7 \mid 6^{2n-1}+1,\ n>0$; в) $19 \mid 7^{n+1}+8^{2n-1},\ n>0$; г) $7 \mid 11^n-4^n$; д) $8 \mid 5^{n+1}+2\cdot 3^n+1$; е) $73 \mid 8^{n+2}+9^{2n+1}$;

2.1.8. Довести нерівності
1) $\sum\limits_{i=1}^{n}\frac{1}{\sqrt{i}}\geq\sqrt{n}$; 2) $\sum\limits_{i=1}^{n}\frac{1}{\sqrt{i}}\leq 2\sqrt{n}-1$;
3) $|\sin nx|\leq n\,|\sin x|$; 4) $\frac{1\cdot 3\cdot 5\cdot\ldots\cdot(2n-1)}{2\cdot 4\cdot 6\cdot\ldots\cdot 2n}\leq\frac{1}{\sqrt{n+1}},\ n>0$;
5) $\frac{(2n)!}{(n!)^2}>\frac{4^n}{n+1},n>1$; 6) $2^n>n^3,\quad n\geq 10$;
7) $2^{n-1}\left(a^n+b^n\right)>(a+b)^n,a,b\in\mathbb{R},n>0$;
8) $2^{n+4}\geq(n+4)^2$; 9) $4^n\geq 3^n+n^2$;
10) $\frac{1}{n+1}+\frac{1}{n+2}+\cdots+\frac{1}{2n}>\frac{13}{24},n>1$;
11) $\sqrt{n}<1+\frac{1}{\sqrt{2}}+\cdots+\frac{1}{\sqrt{n}}<2\sqrt{n}$;
12) $\frac{x_1+\cdots+x_n}{n}\geq\sqrt[n]{x_1\cdot\cdots\cdot x_n},x_i\geq 0$.

2.1.9. Довести, що функція $T_n\left(n\arccos x\right)$ на інтервалі $[-1,1]$ є поліномом степеня n.

2.1.10. Довести, що кожне натуральне число $n>7$ можна записати у вигляді $3k+5l$.

2.1.11. Числами Фібоначчі називають члени такої послідовності a_1, a_2, $\ldots,a_n,\ldots$, $a_1=a_2=1,a_{n+1}=a_n+a_{n-1}$. Довести, що для чисел Фібоначчі виконується рівність $a_{n+1}a_{n+2}-a_na_{n+3}=(-1)^n$.

2.1.12. Довести, що для чисел Фібоначчі виконуються рівності
1) $a_1+\ldots+a_{n-1}+a_n=a_{n+2}-1$;
2) $a_2+a_4+\ldots+a_{2n}=a_{2n+1}-1$;
3) $a_1^2+a_2^2+\ldots+a_n^2=a_na_{n+1}$;
4) $a_{m+n}=a_ma_{n-1}+a_{m+1}a_n$;
5) $a_{2n}=a_{n+1}^2-a_{n-1}^2$;
6) $a_{3n}=a_{n+1}^3+a_n^3-a_{n-1}^3$;
7) $a_{n+1}^2=a_na_{n+2}+(-1)^n$;
8) $a_1a_2+a_2a_3+\ldots+a_{2n-1}a_{2n}=a_{2n}^2$;
9) $a_1a_2+a_2a_3+\ldots+a_{2n}a_{2n+1}=a_{2n+1}^2-1$;
10) $na_1+(n-1)\,a_2+(n-2)\,a_3+\ldots+2a_{n-1}+a_n=a_{n+4}-(n+3)$.

2.1.13. Довести, що для n-го числа Фібоначчі правильна формула

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

2.1.14. Довести, що сторона правильного многокутника, який має 2^n сторін, виражається через радіус R описаного кола формулою

$$a_{2^n} = R\sqrt{2 - \sqrt{2 + \sqrt{2 + \ldots + \sqrt{2}}}},$$

де у правій частині записані $n-1$ знаків кореня.

2.1.15. Довести, що числа $3^{2n+2} - 8n - 9$ та $3^{2n+3} + 40n - 27$ діляться на 64, а число $10^n + 18n - 28$ ділиться на 27 для будь-якого натурального n.

2.1.16. Довести нерівності:

1) $(1+h)^n \geq 1 + nh$ при $h > -1, n \in \mathbb{N}$;
2) $\sum\limits_{k=1}^{n} x_k + \sum x_k^{-1} \geq 2n, \quad x_k \in \mathbb{R}, x_k > 0.$
3) $2! \cdot 4! \cdot \ldots \cdot (2n)! > \left((n+1)!\right)^n$;
4) $\frac{1}{2} \cdot \frac{3}{4} \cdot \ldots \cdot \frac{2n-1}{2n} < \frac{1}{\sqrt{2n+1}}$;
5) $n! < \left(\frac{n+1}{2}\right)^n, \quad n > 1$;
6) $n^{n+1} > (n+1)^n, \quad n \geq 3$;
7) $(2n)! < 2^{2n} \left(n!\right)^2$;
8) $\left(a_1 + \ldots + a_n\right)^2 \leq n\left(a_1^2 + \ldots + a_n^2\right), \quad a_i \in \mathbb{R}$;
9) $\left|\sin\left(\sum\limits_{k=1}^{n} x_k\right)\right| \leq \sum\limits_{k=1}^{n} |\sin x_k|$;
10) $\frac{1}{\sqrt{n}} < \sqrt{n+1} + \sqrt{n-1}, \quad n > 0$;
11) $\frac{1}{2\sqrt{n}} \leq \frac{1}{2} \cdot \frac{3}{4} \cdot \ldots \cdot \frac{2n-1}{2n} < \frac{1}{\sqrt{2n}}$;
12) $\frac{1}{1 \cdot 2 \cdot 3} + \frac{1}{2 \cdot 3 \cdot 4} + \ldots + \frac{1}{n(n+1)(n+2)} < \frac{1}{4}$;
13) $2^{-2} + 3^{-2} + \ldots + n^{-2} < \frac{n-1}{n}$.

2.1.17. Довести, що поліном $(x+y+z)^{2n+1} - x^{2n+1} - y^{2n+1} - z^{2n+1}$ ділиться на поліном $(x+y+z)^3 - x^3 - y^3 - z^3$.

2.1.18. Довести таке: коли $a^{-1} + b^{-1} + c^{-1} = (a+b+c)^{-1}$, то

$$(a^{-1} + b^{-1} + c^{-1})^{2n+1} = \left(a^{2n+1} + b^{2n+1} + c^{2n+1}\right)^{-1}.$$

2.1.19. Нехай $a_0 = 1, a_1 = 2, a_2 = 3, a_n = a_{n-1} + a_{n-2} + a_{n-3}$ для $n \geq 3$. Довести, що $a_n > \left(\frac{3}{2}\right)^n$.

2.1.20. Нехай $a_0 = a_1 = a_2 = 1, a_n = a_{n-1} + a_{n-2} + a_{n-3}$ для $n \geq 3$. Довести, що $a_n \leq 2^{n-1}$.

2.1.21. Нехай $a_0 = 1, a_1 = 3, a_2 = 5, a_n = 3a_{n-2} + 2a_{n-3}$. Довести, що: а) $2^n < a_n < 2^{n+1}$; б) $a_n = 2a_{n-1} + (-1)^{n-1}, \quad n \geq 1$.

2.1.22. Нехай $a_0 = a_1 = a_2 = 1, a_n = a_{n-1} + a_{n-3}, \quad n \geq 3$. Довести, що:

1) $a_n \geq 2a_{n-2}$;
2) $a_n \leq \left(\frac{3}{2}\right)^{n-1}, \quad n \geq 1$;
3) $a_n \geq 2^{\frac{n-2}{2}}, \quad n \geq 2$.

2.1.23. Нехай $\varphi(n)$ — функція Ойлера, тобто кількість натуральних чисел менших від n і взаємно простих з n. Довести, що $\varphi(n) = \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$, де $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$ — канонічний розклад числа n в добуток простих чисел.

2.1.24. Довести, що

1) $\sum\limits_{i=0}^{n} C_n^i = 2^n$;
2) $\sum\limits_{i=0}^{n} (-1)^n C_n^i = 0$;
3) $\sum\limits_{i=0}^{n} i C_n^i = n2^{n-1}$;
4) $\sum\limits_{k=0}^{n} \left(C_n^k\right)^2 = C_{2n}^n$;
5) $\sum\limits_{k=0}^{m} C_{n+k-1}^k = C_{m+n}^m$;
6) $\sum\limits_{k=0}^{m-1} C_{n+k}^k = C_{m+n}^{n+1}$;
7) $C_{2k}^k + C_2^1 C_{2k-2}^{k-1} + C_4^2 C_{2k-4}^{k-2} + \cdots + C_{2k}^k = 2^{2k}$;
8) $\sum\limits_{n_1+\cdots+n_k=n} \frac{n!}{n_1!\cdots n_k!} = k^n$;
9) $\sum\limits_{n_1+\cdots+n_k=n} \frac{n!}{n_1!\cdots n_k!} = \sum\limits_{m=0}^{k} C_k^m (k-m)^n$.

2.2. Потужність множин

2.2.1. Довести, що скінченна множина не може бути рівнопотужна своїй власній підмножині. Показати, що це не так для нескінченних множин.

2.2.2. Довести, що множини точок довільних двох многокутників на площині рівнопотужні.

2.2.3. Довести, що кожна нескінченна підмножина зліченної множини є зліченною.

2.2.4. Довести таке: коли A — зліченна множина, а B — скінченна множина, то $A \setminus B$ також зліченна множина.

2.2.5. Довести, що об'єднання зліченної та скінченної множин є зліченною множиною.

2.2.6. Довести, що об'єднання зліченної родини скінченних множин є зліченною множиною.

2.2.7. Довести, що об'єднання зліченної родини зліченних множин є зліченною множиною.

2.2.8. Нехай область визначення функції є зліченною множиною. Довести, що область значень цієї функції є зліченною або скінченною множиною.

2.2.9. Довести таке: а) якщо A нескінченна множина, а B скінченна або зліченна множина, то $|A \cup B| = |A|$; б) якщо A нескінченна і незліченна множина, а B скінченна або зліченна, то $|A \setminus B| = |A|$.

2.2.10. Довести, що множина раціональних точок довільного відрізка $[a, b]$ є зліченною множиною.

2.2.11. Довести, що множини точок квадрата та відрізка рівнопотужні.

2.2.12. Довести, що множина всіх скінченних підмножин зліченної множини є зліченною множиною.

2.2.13. Довести, що множина всіх підмножин зліченної множини є незліченною множиною.

2.2.14. Довести, що для кожної нескінченної множини A існує власна підмножина B множини A, для якої $|A| = |B|$.

2.2.15. Нехай елементи множини A занумеровані n індексами $i_1, i_2, \ldots, i_n$, кожний з яких незалежно один від одного пробігає зліченну множину значень. $A = \{a_{i_1 \ldots i_n} : i_k \in \mathbb{N},\ 1 \leq k \leq n\}$. Довести, що множина A зліченна.

2.2.16. Нехай елементи множини A занумеровані n індексами $i_1, i_2, \ldots, i_n$, кожний з яких незалежно один від одного пробігає множину I. $A = \{a_{i_1 \ldots i_n} : i_k \in \mathbb{N},\ 1 \leq k \leq n\}$. Довести, що множини A та I рівнопотужні.

2.2.17. Довести, що:
1) $|\mathbb{Q} \times \mathbb{Q}| = |\mathbb{N}|$;
2) $|\mathbb{Q} \times \ldots \times \mathbb{Q}| = |\mathbb{N}|$;
3) $|\mathbb{Q} \setminus \mathbb{N}| = |\mathbb{N}|$.

2.2.18. Довести, що множина $\mathbb{N} \times \ldots \times \mathbb{N} = \mathbb{N}^k$ зліченна.

2.2.19. Довести, що множина всіх поліномів з раціональними коефіцієнтами є зліченною.

2.2.20. Алгебричним числом називають число, яке є коренем полінома з цілими коефіцієнтами. Довести, що множина всіх алгебричних чисел є зліченною.

2.2.21. Довести, що кожний сегмент $[a, b]$, кожний інтервал (a, b) і кожний напівсегмент $(a, b]$ або $[a, b)$ мають потужність континуум.

2.2.22. Довести, що об'єднання скінченної або зліченної родини множин потужності континуум є множиною потужності континуум.

2.2.23. Довести, що об'єднання родини потужності континуум множин потужності континуум є множиною потужності континуум.

2.2.24. Довести, що множина всіх нескінченних послідовностей нулів та одиниць має потужність континуум.

2.2.25. Довести, що множина всіх нескінченних послідовностей натуральних (цілих, раціональних) чисел має потужність континуум.

2.2.26. Довести, що множина всіх ірраціональних чисел має потужність континуум.

2.2.27. Нехай елементи множини A занумеровані n індексами $i_1, i_2, \ldots, i_n$, кожний з яких незалежно один від одного пробігає множину потужності континуум. $A = \{a_{i_1 \ldots i_n} : i_k \in \mathbb{R}, 1 \leq k \leq n\}$. Довести, що множина A має потужність континуум.

2.2.28. Довести, що множина всіх ірраціональних чисел має потужність континуум.

2.2.29. Довести, що множина всіх букв A на площині, які не накладаються, є зліченною.

2.2.30. Довести, що множина всіх точок простору $\mathbb{R}^n$ має потужність континуум.

2.2.31. Довести, що множина всіх дійсних функцій, заданих на відрізку $[0,1]$ має потужність більшу ніж континуум.

2.2.32. Довести, що множина точок розриву монотонної функції, заданої на всій дійсній осі, є скінченною або зліченною.

2.2.33. Довести, що множина всіх функцій, заданих на відрізку $[a,b]$ і розривних хоч в одній точці, має потужність більшу ніж континуум.

2.2.34. Довести, що множина всіх неперервних функцій, заданих на відрізку $[a,b]$ має потужність континуум.

2.3. Елементи комбінаторики

2.3.1. Серед 150 людей 45 плавають, 40 їздять на велосипеді, 50 грають у теніс, 32 грають у теніс і не їздять на велосипеді, 27 грають у теніс і плавають, 10 виконують все це.

1) Скільки людей грають у теніс і не їздять на велосипеді?

2) Якщо відомо, що 21 їздять на велосипеді і плавають, то скільки не вміють нічого?

2.3.2. У класі 35 учнів. З них 20 відвідують математичний гурток, 11 — фізичний, 10 не відвідують жодного гуртка.

1) Скільки учнів відвідують математичний і фізичний гуртки?

2) Скільки учнів відвідують лише математичний гурток?

2.3.3. У класі 45 учнів. З них 25 хлопчиків; 30 учнів навчаються на 4 і 5, з них 16 хлопчиків; 30 займаються спортом, з них 18 хлопчиків та 17 школярів, які навчаються на 4 і 5; 15 хлопчиків навчаються на 4 і 5 і займаються спортом. Показати, що в цій інформації є помилка.

2.3.4. Нехай $|X| = m, \quad |Y| = n, \quad m \geq n$. Довести, що кількість сюр'єктивних відображень з X в Y дорівнює $\sum\limits_{k=0}^{n} (-1)^k C_n^k (n-k)^m$.

2.3.5. По пустелі іде караван з 9 верблюдів. Скількома способами можна переставити верблюдів так, щоб попереду кожного верблюда ішов інший ніж перед тим?

2.3.6. Серед 100 студентів англійську мову знають 28, німецьку — 30, французьку — 42, англійську та французьку — 10 , англійську та німецьку — 8, німецьку та французьку — 5. Скільки студентів не знають жодної мови?

2.3.7. Скількома способами можна розставити 20 нулів і 15 одиниць так, щоб дві одиниці не стояли поруч? Відповідь обґрунтувати.

2.3.8. У шерензі стоїть 30 людей. Скількома способами з неї можна вибрати 12 людей, які не стоять поруч?

2.3.9. За круглим столом сидить 30 людей. Скількома способами можна вибрати 12 людей так, щоб до вибраних не потрапили два сусіди?

2.3.10. Дано 25 різних предметів і 5 ящиків. Потрібно в кожний ящик покласти по 5 предметів. Скількома способами це можна зробити?

2.3.11. Двоє людей повинні поділити між собою 100 гривень, 100 доларів та 100 євро. Скількома способами вони можуть це зробити?

2.3.12. Скількома способами можна розділити 1000 гривень між 5 людьми?

2.3.13. Скількома способами можна розділити 1000 гривень між 5 людьми за умови, що кожна людина одержує не менше ніж 50 гривень?

2.3.14. Скількома способами можна розділити 50 гривень, 80 доларів та 100 євро між 4 особами?

2.3.15. Є 20 чоловіків і 20 жінок. Скількома способами їх можна розсадити за круглим столом так, щоб дві особи одної статі не сиділи поруч?

2.3.16. Скількома способами можна розсадити на карусель 20 чоловіків і 20 жінок так, щоб дві особи одної статі не сиділи поруч?

2.3.17. Скільки існує мономів степеня n від трьох змінних x, y і z?

2.3.18. Скільки існує мономів степеня n від k змінних?

2.3.19. Скільки різних слів можна одержати, переставляючи букви в словах «парабола», «математика», «моном»?

2.3.20. Скільки різних намист можна скласти з 20 червоних та 10 чорних кульок?

2.3.21. У місті проживає 25000 жителів. Чи правильно, що принаймні двоє з них мають однакові ініціали?

2.3.22. Серед 5 українців 7 росіян та 10 англійців треба вибрати декілька людей так, щоб серед вибраних були особи всіх трьох національностей. Скількома способами можна це зробити?

2.3.23. Скількома способами можна розбити 100 предметів на 5 груп по 20 предметів?

2.3.24. Людина має 7 друзів і протягом 7 днів запрошує їх до себе так, щоб компанія ні разу не повторилася. Скількома способами це можна зробити?

2.3.25. Скількома способами можна переставити букви слова «математика» так, щоб дві голосні букви не стояли поруч?

2.3.26. Скількома способами можна вибрати декілька букв із фрази «Око за око, зуб за зуб»? Скількома способами можна вибрати з цієї фрази 3 букви?

2.3.27. Скільки існує семизначних чисел у яких 4 цифри парні?

2.3.28. Скільки існує семизначних чисел у яких сума цифр парна?

2.3.29. Скільки існує семизначних чисел у яких всі цифри різні?

2.3.30. Скільки семизначних чисел можна скласти з цифр числа: а) 51315113; б) 19717331; в) 18091948?

2.3.31. Скільки є послідовностей $1 \leq x_1 < x_2 < \ldots < x_k \leq n$? $(x_i \in \mathbb{N})$.

2.3.32. Скільки підмножин множини $\{1, 2, \ldots, 100\}$ містять хоч одне непарне число?

2.3.33. Скільки є підстановок $\pi \in S_7$ таких, що $\pi(1) \neq 2$?

2.3.34. Скількома способами сукупність із $2n$ предметів можна розбити навпіл?

2.3.35. Скількома способами можна розбити множину $\{1, 2, \ldots, kn\}$ на k підмножин з n елементів?

2.3.36. Скількома способами можна вибрати на шаховій дошці два квадрати — чорний і білий, що не лежать на і тій самій горизонталі і вертикалі?

2.3.37. Автомобільні номери складаються з двох букв та п'яти цифр.

Знайти кількість таких номерів, якщо використовують 32 букви українського алфавіту.

2.3.38. Скількома способами можна розбити $m+n+p$ предметів на три групи так, щоб в одній було m, у другій n, а в третій p предметів?

2.3.39. Четверо студентів складають іспит. Скількома способами можуть бути поставлені їм оцінки, якщо відомо, що ніхто з них не одержав оцінки «незадовільно»?

2.3.40. З групи, яка складається з 7 чоловіків та 4 жінок, потрібно вибрати 6 людей так, щоб серед них було не менше ніж 2 жінки. Скількома способами можна це зробити?

2.3.41. Скількома способами можна посадити за круглий стіл 8 чоловіків і 8 жінок так, щоб дві жінки не сиділи поруч?

2.3.42. Скількома способами можна переставляти букви слова «молоко», щоб три букви «о» не стояли поряд?

2.3.43. Знайти суму всіх чотиризначних чисел, які одержують при довільних перестановках цифр $1, 2, 3, 4$?

2.3.44. Скільки існує шестизначних чисел, в яких три цифри парні і три непарні?

2.3.45. Довести, що непарну кількість предметів можна вибрати з n предметів 2^{n-1} способами.

2.3.46. У скількох точках перетинаються діагоналі опуклого n-кутника, якщо будь-які три з них не перетинаються в одній точці?

2.3.47. Скількома способами можна скласти букет з 7 однакових або різних квітів, якщо є 11 сортів квітів?

2.3.48. Скількома способами можна утворити групу з 3 солдатів і 1 офіцера, якщо є 20 солдатів і 5 офіцерів?

2.3.49. Скільки кіл можна провести через 12 точок, розміщених так, що будь-які 4 з них не належать одному колу?

2.3.50. Скількома способами можна одночасно з'єднати три пари серед n абонентів телефонної мережі?

2.3.51. Скільки чисел, які менші від 1000000, можна записати за допомогою цифр 5,7,8?

2.3.52. Скільки п'ятизначних чисел можна скласти з цифр числа: а) 1315113; б) 977331; в) 25565889?

2.3.53. Маємо необмежену кількість монет вартістю 10, 25 і 50 коп. Скількома способами можна вибрати 20 монет?

2.3.54. Скількома способами можна вибрати серед натуральних чисел від 1 до 200 два числа так, щоб їхня сума була непарною?

2.3.55. Монету кидають $2n$ разів. Довести, що кількість варіантів, у яких герб за кожним разом не випав частіше ніж решітка, дорівнює $1 + \left(C_n^1\right)^2 + \ldots + \left(C_n^n\right)^2 = C_{2n}^n$.

2.3.56. Скількома способами можна розмістити на шаховій дошці 8 однокольорових тур, щоб будь-які дві з них не били одна одну?

2.3.57. Довести, що при $n \geq 4$ існує розміщення n однокольорових ферзів на шаховій дошці з n^2 клітинами, для якого будь-які два ферзі не б'ють один одного.

2.3.58. У місті є 100 міліціонерів. Кожного вечора на чергування виходять троє. Довести, що не можна так скласти графік чергування, щоб будь-які двоє міліціонерів чергували разом лише один раз.

2.3.59. На шкільному вечорі було 17 дівчат і 15 хлопців. Скількома способами можна вибрати серед них 4 пари?

2.3.60. Скількома способами можна вибрати 20 осіб з 30, якщо задані двоє осіб не можуть бути вибрані разом?

2.3.61. Скількома способами можна розділити колоду з 36 карт пополам так, щоб у кожній пачці було по два тузи?

2.3.62. Скількома способами можна витягнути 4 карти з колоди з 36 карт так, щоб було 3 королі? Один король і 3 дами?

2.3.63. Є 12 чоловіків і 16 жінок. Скількома способами можна скласти компанію: а) з 7 осіб; б) з 3 чоловіків та 4 жінок; в) з 7 чоловіків або 7 жінок?

2.3.64. Є 12 чоловіків і 16 жінок. Скількома способами серед них можна вибрати пару (чоловік, жінка)?

2.3.65. Скільки підмножин множини $\{1, 2, \ldots, 2n\}$ містять принаймні одне парне число?

2.3.66. З міста A в місто B веде n доріг, а з міста B в місто C — k доріг. Скількома способами можна добратися з A в C і повернутися назад? А якщо назад повертатися іншими дорогами?

2.3.67. Учні вивчають 10 предметів. У понеділок 6 різних уроків. Скількома способами можна скласти розклад на понеділок?

2.3.68. Скількома способами можна впорядкувати множину $\{1, 2, ..., 2n\}$ так, щоб кожне парне число мало парний номер?

2.3.69. На площині є n різних точок. Кожні дві точки сполучені відрізком. Скільки відрізків утвориться при цьому?

2.3.70. Скільки можна зробити перестановок з n елементів, у яких задані два елементи не стоять поряд?

2.3.71. Скільки можна зробити перестановок з n елементів, у яких між двома заданими елементами стоять k елементів?

2.3.72. Скількома способами можна за k днів скласти n іспитів (в один день — один іспит)?

2.3.73. У скількох точках перетинаються діагоналі опуклого n-кутника, якщо будь-які 3 з них не перетинаються в одній точці?

2.3.74. Скільки існує шестизначних чисел, у яких цифри розміщені у неспадному порядку?

2.3.75. Скільки існує шестизначних чисел, у яких цифри розміщені у незростаючому порядку?

2.3.76. Скільки існує шестизначних чисел, у яких цифри розміщені у зростаючому порядку?

2.3.77. Скількома способами можна з колоди з 52 карт вибрати 10 карт?

2.3.78. У скількох випадках серед цих карт є хоч один туз? У скількох випадках серед цих карт є лише два тузи? У скількох випадках серед них є не менше двох тузів?

2.3.79. Комісія складається з n осіб. Скільки замків повинен мати сейф, скільки ключів до них треба зробити і як їх розподілити, щоб доступ до сейфа був можливий тоді і тільки тоді, коли збереться не менше ніж k членів комісії?

2.3.80. У кімнаті є n ламп. Скільки є таких способів освітлення кімнати, коли горять лише k ламп? Скільки є різних способів освітлення кімнати?

2.3.81. Скількома способами можна розділити n одинакових подарунків серед m дітей? Скільки існує таких способів, за яких кожна дитина отримає хоч один подарунок?

2.3.82. Скільки існує n-значних чисел, у яких цифри розміщені у неспадному порядку? Скільки серед них є таких, в яких кожна цифра трапляється хоч раз?

2.3.83. У класі вивчають $2n$ предметів. Всі вчаться на 4 і 5. Кожні два учні не вчаться однаково. Про жодного з них не можна сказати, що він навчається краще від інших. Довести, що кількість учнів у класі не перевищує C_{2n}^n .

2.3.84. Скількома способами можна розмістити 10 білих, 20 чорних і 15 синіх куль у 5 різних ящиків? Скільки є таких способів, якщо у другому ящику поміщаємо 3 білих, 5 чорних і 7 синіх куль?

2.3.85. За круглим столом сидять n лицарів. Скількома способами можна вибрати k лицарів так, щоб до їхньої групи не потрапило два сусіди?

2.3.86. Є $2n$ предметів. Скількома способами можна розбити ці предмети на пари, якщо не розрізняти порядок пар і порядок елементів у парах?

2.3.87. Скількома способами можна вибрати 7 однакових або різних тістечок у кав'ярні, де є 15 різних сортів тістечок?

2.3.88. Скільки цілих невід'ємних розв'язків має нерівність $x_1 + x_2 + \ldots + x_m \leq n$?

2.3.89. Скількома способами можна роздати 18 різних предметів 5 особам так, щоб 4 одержали по 4 предмети, а п'ятий — 2 предмети?

2.3.90. Розв'язати попередню задачу за умови, що троє отримують по 4 предмети, а двоє по 2 предмети.

2.3.91. Скількома способами можна роздати 52 карти чотирьом гравцям так, щоб кожний отримав по 3 карти трьох мастей і 4 карти четвертої масті?

2.3.92. Маємо по $2n$ предметів чотирьох сортів. Скількома способами їх можна розділити на дві групи по $4n$ предметів?

2.3.93. Скількома способами можна розділити по n предметів трьох сортів між трьома людьми так, щоб кожний одержав по n предметів?

2.3.94. Є 10 подружніх пар. Вони розбиваються на 5 груп по 4 осіб для прогулянки на човнах. Скількома способами можна розбити їх так, щоб у кожному човні було 2 чоловіків і 2 жінок? У скількох випадках чоловік не буде в одному човні зі своєю дружиною? У скількох випадках задані двоє чоловіків не будуть в одному човні зі своїми дружинами?

2.3.95. Нехай $X = \{x_1, x_2, \ldots, x_m\}$, $Y = \{y_1, y_2, \ldots, y_n\}$, $A_i = \left\{\varphi : X \to Y,\ \varphi^{-1}(y_i) = \emptyset\right\}$, $(i = 1, 2, \ldots, n)$.

1) Довести, що $|A_i| = (n-1)^m$, $|A_i \cap A_j| = (n-2)^m$.

2) Обчислити $\left|\cup_{i=1}^{n} A_i\right|$.

2.3.96. По колу біжать n людей. Скількома способами можна поміняти їх місцями так, щоб попереду кожного була інша людина ніж раніше?

2.3.97. Деяка комісія збиралася 40 разів. Кожний раз у засіданнях брало участь по 10 членів, причому будь-які два члени не були разом на засіданні більше ніж один раз. Доведіть, що кількість членів комісії більша ніж 60.

2.3.98. Довести, що з 25 осіб не можна скласти більше ніж 30 комісій по 5 осіб у кожній так, щоб кожні дві комісії не мали більше одного спільного члена.

2.3.99. На кожній з планет деякої системи є астроном, який спостерігає за найближчою планетою. Відстані між планетами попарно різні. Довести таке: коли кількість планет непарна, то за деякою планетою ніхто не спостерігає.

2.3.100. Довести, що в опуклий многокутник площі S і периметра P можна помістити круг радіуса S/P.

2.3.101. Кожна з 9 прямих розбиває квадрат на два чотирикутники, площі яких відносяться як $2/3$. Довести, що принаймні три з цих 9 прямих проходять через одну точку.

2.3.102. Яку найменшу кількість чисел треба викреслити з сукупності чисел $1, 2, 3, \ldots, 2002$ так, щоб кожне з чисел, які залишаться, не дорівнювало добутку двох інших?

2.3.103. Гра «Морський бій» відбувається в квадраті 7 на 7 клітин. Яку найменшу кількість пострілів потрібно зробити, щоб напевно поранити чотирипалубний корабель, якщо відомо, що він має вигляд: а) □□□□;

б) складається з чотирьох клітинок, які примикають одна до одної сторонами.

2.3.104. Яку найменшу кількість кутиків  треба розмістити у квадраті 8×8 клітин, щоб у нього не можна було помістити без накладання більше такої фігури?

2.3.105. У квадраті з 1987×1987 клітин вирізана одна довільна клітина. Довести, що частину, яка залишилася, завжди можна розрізати на триклітинні «кутики» .

2.3.106. Доведіть, що (2,3,5) та (3,5,7) єдині трійки послідовних простих чисел.

2.3.107. Доведіть, що для кожного простого числа p у послідовності $1, 2, 2^2, 2^3, \ldots$ знайдуться два числа, різниця яких ділиться на p.

2.3.108. Скільки різних пар підмножин, які не перетинаються, має множина, що складається з n елементів?

2.3.109. Для кожного $n \in \mathbb{N}$ знайти k таке, що в множині з n елементів існує k різних підмножин, які попарно не перетинаються.

2.3.110. На площині є нескінченна множина точок. Відстань між будь-якими двома з них є цілим числом. Довести, що всі точки лежать на одній прямій.

2.3.111. Довести, що існує нескінченна кількість простих чисел, у яких три останні цифри однакові.

2.3.112. У квадраті, сторона якого дорівнює 1, взято 51 точку. Довести, що деякі три з цих точок обов'язково містяться всередині круга радіуса $1/7$.

2.3.113. Доведіть, що серед будь яких 39 послідовних натуральних чисел обов'язково знайдеться таке, в якого сума цифр ділиться на 11.

2.3.114. У прямокутник зі сторонами 20 і 25 кидають 120 квадратів зі стороною 1. Доведіть, що в прямокутник можна помістити круг діаметра 1, який не перетинається з жодним квадратом.

2.3.115. Задано правильний сорокап'ятикутник. Чи можна розставити у його вершинах цифри $0, 1, \ldots, 9$ так, щоб для кожної пари різних цифр знайшлася сторона, кінці якої занумеровані цими цифрами?

Вправи до Розділу 3

3.1. Бульові функції

3.1.1. Чи існує бульова алгебра з шести елементів? Відповідь поясніть.

3.1.2. Нехай для заданого натурального числа $n \geq 1$ D_n буде множиною дільників числа n. Визначимо операції $\cup$, $\cdot, -$ на множині D_n так: $a \cup b = \text{НСК}(a,b)$, $a \cdot b = \text{НСД}(a,b)$ і $\bar{a} = n/a$.

а) Множина $D_6 = \{1,2,3,6\}$ з визначеними вище операціями є бульовою алгеброю. Які елементи є нулем і одиницею?

б) Знайдіть множину S таку, щоб алгебри D_6 і 2^S були ізоморфними та запишіть ізоморфізм між ними.

в) Покажіть, що множини D_4 і D_8 з введеними операціями не є бульовими алгебрами.

3.1.3. Побудувати таблиці істинності для таких формул. Звести ці формули до диз'юнктивної нормальної форми та до кон'юнктивної нормальної форми:

а) $(P \to Q) \vee (P \longrightarrow (Q \wedge P))$;

б) $\neg(X \wedge Y) \wedge (Z \longrightarrow X)$;

в) $\neg(X \wedge \neg Z) \longrightarrow Z \wedge X \vee \neg Y \wedge \neg Z$;

г) $\neg(X \longrightarrow Z) \vee \neg Y \wedge (X \vee Y)$;

д) $(((A \longrightarrow B) \longrightarrow (C \longrightarrow \neg A)) \longrightarrow (\neg B \longrightarrow \neg C))$;

е) $(((((A \longrightarrow B) \longrightarrow \neg A) \longrightarrow \neg B) \longrightarrow \neg C) \longrightarrow C)$;

є) $((A \longrightarrow (B \longrightarrow C)) \longrightarrow ((A \longrightarrow \neg C) \longrightarrow (A \longrightarrow \neg B)))$;

ж) $(A \wedge B) \vee (A \wedge \neg B) \vee (C \wedge B)(\neg a \wedge (B \wedge C))$;

з) $\neg((A \longrightarrow B) \wedge (B \longrightarrow \neg A))$.

3.1.4. Знайдіть ДДНФ і ДКНФ для таких бульових функцій:

а) $f(x_1, x_2, x_3) = \begin{cases} 1, & \text{якщо } x_1 + x_2 + x_3 \leq 1 \\ 0, & \text{інакше;} \end{cases}$

б) $f(x_1, x_2, x_3) = \begin{cases} 1, & \text{якщо } x_1 = x_3 \\ 0, & \text{інакше;} \end{cases}$

в) $f(x_1, x_2, x_3) = \begin{cases} 1, & \text{якщо } x_1 + x_2 + x_3 \geq 2 \\ 0, & \text{інакше;} \end{cases}$

г) $f(x_1, x_2, x_3) = \begin{cases} 1, & \text{якщо } x_1 \neq x_3 \\ 0, & \text{інакше;} \end{cases}$

д) $f(x_1, x_2, x_3) = \begin{cases} 1, & \text{якщо } x_1 = x_2 = x_3 \\ 0, & \text{інакше;} \end{cases}$

е) $f(x_1, x_2, x_3) = x_1 + x_2 \cdot x_3 \mod 2$;

є) $f(x_1, x_2, x_3) = x_2 \cdot (x_1 + x_3) \mod 2$;

ж) $f(x_1, x_2, x_3) = x_1 + x_2 + x_3 \mod 2$.

3.1.5. Які з цих формул є тавтологіями?

а) $(B \longleftrightarrow C) \longrightarrow (D \longleftrightarrow A) \longrightarrow (B \vee D \longrightarrow (A \vee C))$;

б) $A \wedge C \vee B \wedge D \longrightarrow (B \vee C) \wedge (A \vee D)$;

в) $(A \longrightarrow C) \longrightarrow (D \longrightarrow A) \longrightarrow (B \vee D \longleftrightarrow A \vee C)$;

г) $((P \longrightarrow R) \longrightarrow ((Q \longrightarrow R) \longrightarrow ((P \vee Q) \longrightarrow R)))$;

д) $((P \longrightarrow Q) \longrightarrow ((P \longrightarrow (Q \longrightarrow R))) \longrightarrow (P \longrightarrow R))$;

е) $((P \longrightarrow Q) \longrightarrow ((Q \longrightarrow R) \longrightarrow (P \longrightarrow R)))$;

є) $((\neg Q \longrightarrow \neg P) \longrightarrow ((\neg Q \longrightarrow P) \longrightarrow Q))$;

ж) $(A \longrightarrow B) \longrightarrow ((A \vee C) \longrightarrow (C \vee B))$.

3.1.6. Спростіть такі вирази:

а) $\neg(B \wedge C) \wedge \neg(A \wedge \neg B \wedge C) \wedge \neg(\neg A \wedge \neg B \wedge C)$;

б) $\neg(A \vee B \vee C \vee D) \vee \neg(A \vee B \vee D) \vee \neg(A \vee C)$;

в) $\neg\big(\neg(A \vee \neg \vee C) \wedge (B \vee \neg D)\big)$;

3.1.7. Довести повноту систем функцій:

а) $\{\wedge, \vee, -\}$; б) $\{-, \wedge\}$; в) $\{-, \vee\}$;

г) $\{-, \to\}$, де $a \to b = \overline{a} \vee b$; д) $\{\ |\ \}$, де $a|b = \overline{x \wedge b}$;

е) $\{\ \downarrow\ \}$, де $a \downarrow b = \overline{a \vee b}$; є) $\{0, \to\}$, де $0(a) = 0\ \forall a$;

ж) $\{0, 1, [\ ,\ ,\ ]\}$, де $0(a) = 0\ \forall a,\ 1(a) = 1\ \forall a,\ [a, b, c] = (b \wedge a) \vee (\bar{b} \wedge c)$;

з) $\{\equiv, \vee, 0\}$, де $a \equiv b = (a \to b) \wedge (b \to a)$.

3.1.8. Довести, що такі системи функцій неповні:

а) $\{-\}$; б) $\{\wedge, \to\}$; в) $\{\to\}$; г) $\{\vee, \to\}$; д) $\{\wedge\}$; е) $\{\vee\}$; є) $\{\wedge, \vee\}$;

ж) $\{\equiv, +\}$, де $a + b = \begin{cases} 1, & \text{якщо } a \neq b, \\ 0, & \text{якщо } a = b. \end{cases}$, $a, b \in \{0, 1\}$; з) $\{\equiv\}$.

3.1.9. Система функцій S називається *незалежною*, якщо жодна функція $f \in S$ не може бути подана як суперпозиція функцій з $S \setminus f$. Довести, що такі системи функцій незалежні:

а) $\{-, \equiv\}$; б) $\{-, +\}$; в) $\{\equiv, +\}$;

г) $\{\vee, \equiv\}$; д) $\{\to, -\}$; е) $\{0, \vee, \equiv\}$.

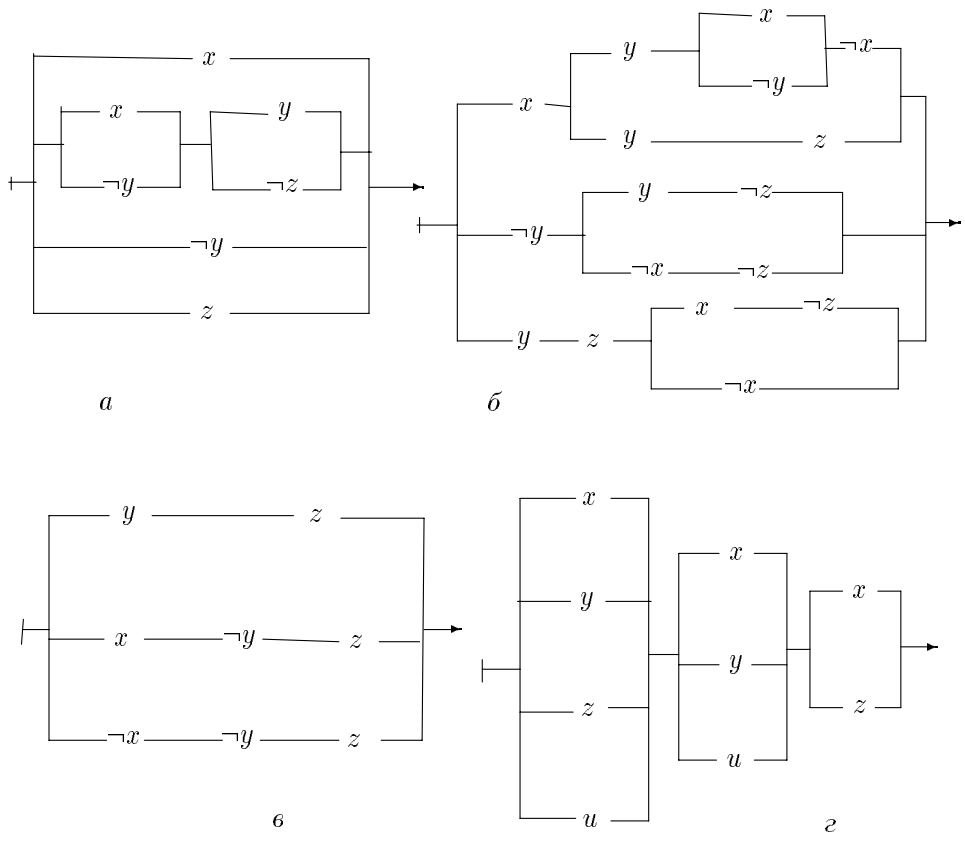
3.1.10. Запишіть логічно еквівалентну формулу з 2 зв'язками:

а) $(\neg(B \vee C) \longrightarrow B \wedge C \wedge D) \vee (\neg B \wedge D)$;
б) $(A \wedge B) \vee (A \wedge \neg C) \vee (\neg A \longrightarrow B) \vee A \vee (B \wedge \neg C)$;
в) $(A \vee (B \longrightarrow C)) \wedge (A \vee B \vee C) \wedge (A \vee C \vee D)$;
г) $(\neg A \wedge B \wedge C \wedge A \wedge \neg B \wedge C) \vee (A \wedge B \wedge B \wedge C)$;

3.1.11. Для таких функцій складіть релейно-контактні схеми та спростіть їх:

а) $(x \longrightarrow y) \wedge (y \longrightarrow z)$;
б) $((x \longrightarrow y) \wedge (y \longrightarrow z)) \longrightarrow (x \longrightarrow z)$;
в) $(x \longrightarrow y) \longrightarrow (\neg x \wedge (y \vee z))$;
г) $(x \longrightarrow (y \longrightarrow z)) \longrightarrow (y \longrightarrow \neg x)$.

3.1.12. Спростіть такі релейно-контактні схеми:



3.2. Скінченні автомати

3.2.1. Розглянемо автомат $M = [A,\ S,\ \nu,\ Z,\ \zeta]$, в якого $A = \{0,1\}$, $Z = \{0,1\}$, $S = \{1,2,3,4,5\}$.

M	ν		ζ	
	0	1	0	1
1	1	2	0	1
2	1	3	0	1
3	5	1	0	1
4	4	2	0	1
5	4	3	1	1

1. Знайдіть автомат $\overline{M}$ з мінімальною кількістю станів, який при всіх

вхідних даних довжини 2 дає ті ж самі виходи, що й автомат M.

2. Чи еквівалентний цей автомат $\overline{M}$ автомату M?

3.2.2. Побудуйте мінімальний автомат $\overline{M}$ для автомата M.

M	ν		ζ	
	a	b	a	b
1	1	6	0	0
2	1	4	0	0
3	2	5	1	0
4	5	8	1	1
5	1	3	1	0
6	8	5	1	1
7	6	3	1	1
8	2	5	1	0

3.2.3. Опишіть машину Тьюрінга, яка додає три натуральних числа.

3.2.4. Побудуйте машину Тьюрінга, яка правильно обчислює такі функції: а) $f(x) = x + 1$; б) $o(x) = 0$; в) $f(x) = \left[\frac{x}{2}\right]$.

3.3. Алгоритми та складність обчислень

3.3.1. Для кожної функції $f(n)$ 1)–10) виберіть найкращу оцінку з такого списку:

а) $f(n) = O(\ln n)$; б) $f(n) = O(\ln^2 n)$; в) $f(n) = O(\ln^3 n)$;
г) $f(n) = O(n)$; д) $f(n) = O(n^2)$; е) $f(n) = O(n^3)$;
є) $f(n) = O(2^n)$; ж) $f(n) = O(n!)$; з) $f(n) = O(n^n)$;

1. C_n^3.
2. $10 \ln^3 n + 15n^2$.
3. Кількість мономів вигляду $x^a y^b z^c$, для яких $a + b + c \leq n$.
4. Кількість поліномів у кільці $\mathbf{Z}_2[x]$, степінь яких не перевищує n.
5. Кількість поліномів у кільці $\mathbf{Z}_n[x]$, степінь яких не перевищує $n-1$.
6. Об'єм пам'яті комп'ютера потрібної для зберігання числа n.

7. Об'єм пам'яті комп'ютера потрібної для зберігання числа n^2.
8. Сума перших n цілих додатних чисел.
9. Сума квадратів перших n цілих додатних чисел.
10. Кількість бітів у сумі квадратів перших n цілих додатних чисел.

3.3.2. Знайти час, потрібний для запису k-бітового числа n у десятковій (шістнадцятковій) системі числення.

3.3.3. Оцінити кількість бітових операцій потрібних для обчислення $n!$.

3.3.4. Знайти функцію $g(n)$, яка асимптотично дорівнює довжині n-го числа Фібоначчі.

3.3.5. Знайти функцію $f(n)$, яка асимптотично дорівнює довжині $n!$.

3.3.6. У термінах O-великого оцінити кількість бітових операцій потрібних для обчислення 3^n.

3.3.7. Оцінити складність задачі обчислення добутку 2×2 цілочисельних матриць. Знайти алгоритм, який використовує менше 8 цілочисельних множень.

3.3.8. Нехай m і n — натуральні числа довжини відповідно k і l бітів. Знайти оцінку вигляду $O(g(k,l))$ кількості бітових операцій потрібних для обчислення m^3n^4. Функція $g(k,l)$ має бути як можна простішою.

3.3.9. Нехай n дорівнює кількості комарів у Карпатах. Розмістити такі числа в зростаючому порядку:

а) час потрібний для знаходження значення полінома п'ятого степеня з 20-бітовими коефіцієнтами в точці n;
б) час потрібний для записування числа n у шістнадцятковій системі числення;
в) час потрібний для знаходження лишку $m!$ за модулем p, де m є число довжини порядку $\ln n$, а p — просте число довжини порядку $2 \ln n$;
г) час потрібний для обчислення лишку b^n за модулем m, де b і m є числами приблизно тієї самої довжини що й n.

3.3.10. Розмістити задачі а–г у порядку зростання довжини їхніх вхідних даних:

а) добуток 20 цілих чисел, порядок яких 10^{100};
б) задача комівояжера з 20 містами, причому відстані між містами цілі числа з проміжку $[1, 100]$;
в) задача знаходження коренів квадратного рівняння, коефіцієнти якого є цілими числами довжини порядку 50;
г) задача знаходження всіх простих дільників цілого числа, довжина якого є порядку 40.

3.3.11. Пояснити, як можна використати задачу розпізнавання комівояжера для розв'язання задачі пошуку комівояжера.

3.3.12. Припустимо, що Π_1 є такою задачею.
ЗАДАНО: два цілих числа.
ЗАПИТАННЯ: чи є вони рівними?
Нехай Π_2 є такою задачею.
ЗАДАНО: два рівняння $ax + by = 0$ і $cx + dy = 0$, де $a, b, c, d \in \mathbb{Z}$;
ЗАПИТАННЯ: чи мають ці рівняння спільний розв'язок, крім $(0, 0)$?
Показати, що Π_2 зводиться до Π_1, побудувавши звідність індивідуальної задачі з Π_2 до індивідуальної задачі з Π_1.

3.3.13. Нехай Π_1 є такою задачею.
ЗАДАНО: два вектори в просторі;
ЗАПИТАННЯ: чи є вони колінеарними?
Нехай задача Π_2 є такою.
ЗАДАНО: дві пари неколінеарних векторів у просторі;
ЗАПИТАННЯ: чи збігаються площини цих векторів?
Показати, що Π_2 зводиться до Π_1, побудувавши звідність індивідуальної задачі з Π_2 до індивідуальної задачі з Π_1.

3.3.14. Нехай $f_1(x, y)$, $f_2(x, y)$, $g_1(x, y)$, $g_2(x_y), g_3(x, y) \in \mathbb{F}_3[x, y]$ — поліноми від двох змінних, степінь яких не перевищує 2. Знайти ймовірність того, що $f_1 \cdot f_2 = g_1 \cdot g_2 \cdot g_3$.

3.3.15. Чи правильне таке твердження: якщо деяка $\mathcal{NP}$-повна задача має експоненційний алгоритм, то будь-яка задача з класу $\mathcal{NP}$ має експоненційний алгоритм. Відповідь пояснити.

Вправи до Розділу 4

4.1. Графи

4.1.1. Довести, що графи на рис.1 неізоморфні, а графи на рис.2 (що мають по 7 вершин по периметру) ізоморфні.

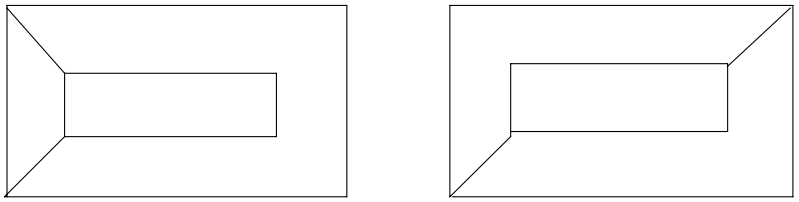


Рис.1

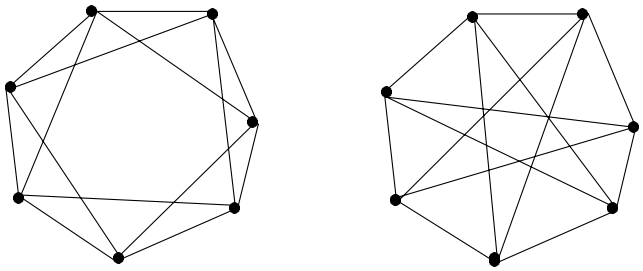


Рис.2

4.1.2. Переконайтеся, що такі об'єкти можна розглядати як графи (орієнтовані графи):

а) вершини та ребра многогранника;
б) план лабіринту;
в) дружні відносини між людьми, які запрошені на вечірку;
г) стадії гри в хрестики-нулики;
д) футбольний турнір;
е) дільники заданого натурального числа;
є) країни на карті.

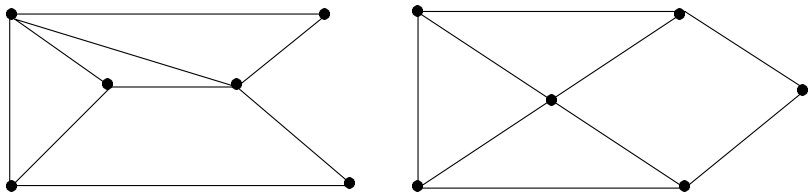
4.1.3. Вивести формулу, яка виражає кількість ребер графа через степені його вершин.

4.1.4. Довести, шо кількість вершин непарного степеня будь-якого графа є парним числом.

4.1.5. Довести, що кількість ребер однорідного графа степеня r, який має n вершин, дорівнює $nr/2$.

4.1.6. Для того щоб у зв'язному графі існував ланцюг ab, який містить всі ребра лише один раз, необхідно і достатньо, щоб a і b були єдиними вершинами непарного степеня цього графа.

4.1.7. Показати, що графи зображені на рисунку не ізоморфні.



4.1.8. Граф називають гамільтоновим, якщо в ньому існує простий цикл, який містить кожну вершину цього графа. Зобразити декілька гамільтонових графів.

4.1.9. Узагальнення теореми Ойлера. У кожному зв'язному графі з $2k$ вершинами непарного степеня існує набір k циклів, які в сукупності містять всі ребра графа лише один раз.

4.1.10. Торговець, який живе у місті A_1, збирається відвідати міста A_2, A_3, A_4. Відстань між містами такі: $A_1A_2 = 120$, $A_1A_3 = 140$, $A_1A_4 = 180$, $A_2A_3 = 70$, $A_2A_4 = 100$, $A_3A_4 = 110$. Знайти найкоротший циклічний шлях з A_1, який проходить через три інші міста.

4.1.11. Скільки ходів конем можна зробити на шаховій дошці?

4.1.12. Скільки ходів королем можна зробити на шаховій дошці?

4.1.13. Довести, що граф, який відповідає ходам коня на шаховій дошці, не є ойлеровим.

4.1.14. Скільки існує простих орієнтованих графів, які мають 5 вершин?

4.1.15. Побудувати граф, який відповідає частково впорядкованій множині підмножин множини з 3 елементів.

4.1.16. Довести, що кількість вершин непарного степеня будь-якого графа є парним числом.

4.1.17. Довести, що не існує графа, який має 24 вершини, з них 11 вершин степеня 7, 10 вершин степеня 5 і 3 вершини степеня 3.

4.1.18. Нехай граф G_n має n вершин, які занумеровані натуральними числами $1, 2, \ldots, n$. У графі G_n i-а та j-а вершини з'єднані ребром, якщо НСД$(i, j) = 1$. Записати матриці Кіркгофа, суміжності та інцидентності графів G_4, G_5, G_6. Чи є граф G_n зв'язним?

4.1.19. Циклічним порядком зв'язного графа називають найменше число його ребер, які потрібно вилучити, щоб одержати дерево. Довести таке: коли граф має l ребер і n вершин, то його циклічний порядок дорівнює $l - n + 1$.

4.1.20. Доведіть, що з точністю до ізоморфізму існує 4 простих графа з 3 вершинами, 11 — з 4 вершинами і 34 з 5 вершинами.

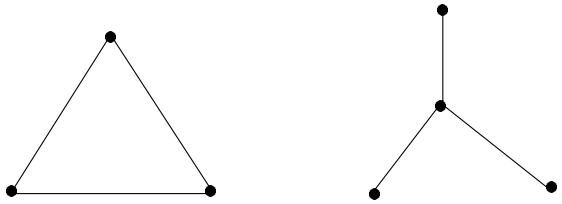
4.1.21. Описати матриці суміжності, Кіркгофа та інцидентності повних графів і дводольних графів. Що можна сказати про ці матриці простого графа та його доповнення?

4.1.22. Довести, що реберний граф повного графа K_n з n вершинами має $n(n-1)/2$ вершин і є регулярним степеня $2n-4$. Одержати аналогічні результати для графів $K_{m,n}$.

4.1.23. Довести, що в простому графі, який має принаймні дві вершини, знайдуться дві вершини з однаковими степенями.

4.1.24. Реберним графом простого графа G називають граф $L(G)$, вершини якого однозначно зіставлені ребрам графа G, а дві вершини в $L(G)$ суміжні тоді й лише тоді, коли відповідні ребра мають спільну вершину

в G. Показати, що реберні графи графів на рисунку ізоморфні. Знайдіть вираз для числа ребер графа $L(G)$ через степені вершин графа G.



а *б*

4.1.25. Зобразити всі кубічні (регулярні степеня 3) графи з не більше ніж 8 вершинами.

4.1.26. Наведіть приклади (якщо це можливо):
а) дводольного графа , що є регулярним;
б) кубічного графа з 9 вершинами;
в) простого графа з n вершинами і $(n-1)(n-2)/2$ ребрами;
г) простого графа, ізоморфного своєму реберному графу;
д) простого графа, доповнення якого ізоморфне реберному графу;
е) чотирьох зв'язних графів, які є регулярними степеня 4.

4.1.27. Довести, що простий граф ізоморфний своєму реберному графу тоді і тільки тоді, коли він є регулярним степеня 2.

4.1.28. Простий граф, ізоморфний своєму доповненню (доповнення $\overline{G}$ графа G має ті самі вершини, і вершини в $\overline{G}$ суміжні тоді і тільки тоді, коли вони не суміжні в G) називаються самодоповнювальним. Довести, що кількість вершин самодоповнювального графа дорівнює $4k$ або $4k+1$. Знайти самодоповнювальні графи з 4 і 5 вершинами.

4.1.29. Довести, що серед 6 осіб завжди знайдуться троє таких, які знають один одного або жоден з них не знає двох інших.

4.1.30. Довести, що реберний граф простого ойлерового графа є ойлеровим. Якщо відомо, що реберний граф простого графа G є ойлеровим, то чи можна звідси вивести, що сам граф G є ойлеровим?

4.1.31. Довести таке: коли граф G зв'язний і має $k > 0$ вершин непарного степеня, то мінімальна кількість ланцюгів, які не мають спільних ребер і об'єднання яких містить кожне ребро графа G, дорівнює $k/2$; як

частковий випадок вивести, що граф є напівойлеровим (не обов'язково замкненим ланцюгом) тоді і тільки тоді, коли в ньому не більше двох вершин мають непарний степінь.

4.1.32. Довести, що в зв'язному графі два простих ланцюги максимальної довжини мають спільну вершину.

4.1.33. Розглянемо скінченний зв'язний граф $G = (V, E)$ на площині (V — множина вершин графа G, E — множина його ребер, $|V| = n$, $|E| = m$). Припустимо, що ребра цього графа не перетинаються поза вершинами. Довести, що обмежена графом G область складається з $m - n + 1$ многокутників.

4.1.34. Узагальнення теореми Ойлера. Нехай G плоский граф з n вершинами, m ребрами, f гранями і k компонентами зв'язності. Довести, що $n + f = m + k + 1$.

4.1.35. Нехай G зв'язний плоский граф з $n \geq 3$ вершинами і m ребрами. Довести, що $m \geq 3n - 6$.

4.1.36. Довести, що у кожному плоскому графі існує вершина, степінь якої не більший ніж 5.

4.1.37. Довести, що графи K_5 і $K_{3,3}$ не плоскі.

4.1.38. Довести, що не існує графа, степені всіх вершин якого різні.

4.1.39. Кожний з чотирьох сусідів з'єднав свій будинок з трьома іншими за допомогою стежок, які не перетинаються. Довести, що п'ятий сусід не зможе так з'єднати свій будинок з чотирма іншими. Довести, що з трьома будинками його будинок з'єднати можна.

4.1.40. Лісом називають граф, який не містить циклів. Зв'язний ліс називають деревом. Довести, що дерево з $n - 1$ ребром має n вершин.

4.1.41. Довести, що зв'язний граф з n вершинами і $n - 1$ ребром є деревом.

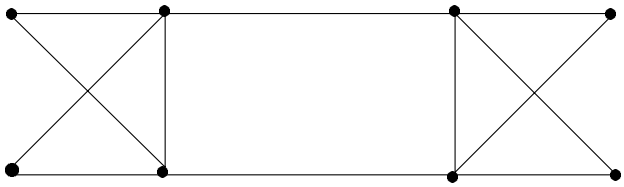
4.1.42. Нехай G ліс з n вершинами і k деревами. Довести, що G має $n - k$ ребер.

4.1.43. Довести, що існує 512 орієнтованих графів з трьома вершинами. Скільки існує орієнтованих графів з n вершинами?

4.1.44. Алгоритм Флері. Нехай G ойлеровий граф. Тоді наступна процедура завжди можлива і призводить до ойлерового ланцюга графа G.

Виходимо з довільної вершини і рухаємось по ребрах графа згідно з такими правилами: а) стираємо ребра по мірі їх проходження, а також стираємо ізольовані вершини, які при цьому утворюються; б) на кожному кроці йдемо мостом лише тоді, коли немає інших можливостей. (Ребро зв'язного графа є мостом, якщо після його вилучення граф стає незв'язним.)

За допомогою алгоритму Флері знайти ойлеровий ланцюг у графі на рисунку



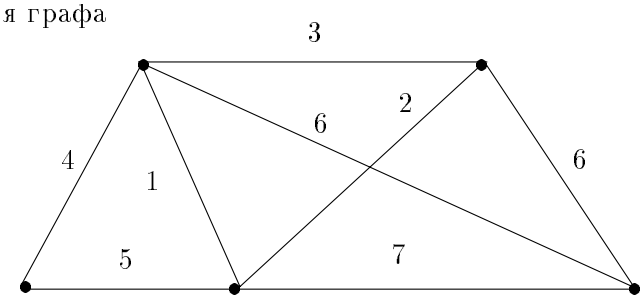
4.1.45. Для яких чисел m і n графи K_n та $K_{m,n}$ є ойлеровими? Чи серед платонових графів (графів, відповідних правильним многогранникам: тетраедр, куб, октаедр, додекаедр, ікосаедр) є ойлерові? Якщо так, то знайдіть у них ойлерові ланцюги.

4.1.46. Довести, що кожне дерево є дводольним графом. Які дерева є повними дводольними графами?

4.1.47. Алгоритм Краскала для знаходження в зв'язному зваженому графі (G, w) каркаса мінімальної ваги 1. Будуємо граф T_1, який складається з ребра найменшої ваги графа (G, w) та відповідних цьому ребру вершин. 2. Якщо граф T_i вже побудований і $i \leq n+1$, то будуємо граф T_{i+1}, долучаючи до графа T_i ребро графа G, яке має найменшу вагу серед ребер, що не входять в T_i, яке не утворює циклів з ребрами з T_i. Довести, що ця побудова завжди можлива і дає каркас найменшої ваги.

4.1.48. Знайдіть каркасні дерева у графах K_5, $K_{3,3}$?

4.1.49. Застосовуючи алгоритм Краскала, знайти каркасне дерево найменшої ваги для графа



4.2. Коди

4.2.1. Довести, що поліноми x^2+x+1 та x_3+x+1 незвідні над полем $\mathbb{F}_2$ і що факторкільця $\mathbb{F}_2[x]/(x^2+x+1)$ та $\mathbb{F}_2[x]/(x^3+x+1)$ є полями, відповідно з 4 та 8 елементів.

4.2.2. Довести, що поліноми x^2+1 та x^2+x+4 незвідні над полем $\mathbb{F}_{11}$ і що факторкільця $\mathbb{F}_{11}[x]/(x^2+1)$ та $\mathbb{F}_{11}[x]/(x^2+x+4)$ є ізоморфними полями з 121 елемента.

4.2.3. Довести, що сума всіх елементів скінченного поля (крім поля $\mathbb{F}_2$) дорівнює нулю.

4.2.4. Нехай $a,b\in\mathbb{F}_{2^n}$, $n=2k+1$. Довести, що з $a^2+ab+b^2=0$ випливає $a=b=0$.

4.2.5. Нехай $\alpha\in\mathbb{F}_q$, $n\in\mathbb{N}$. Довести, що поліном $x^{q^n}-x+n\alpha$ ділиться на $x^q-x-\alpha$ в кільці $\mathbb{F}_q[x]$.

4.2.6. Знайти всі автоморфізми скінченного поля з q елементів.

4.2.7. Знайти примітивні елементи полів $\mathbb{F}_7$, $\mathbb{F}_{17}$, $\mathbb{F}_8$ та $\mathbb{F}_9$.

4.2.8. Знайти примітивний елемент поля $\mathbb{F}_{25}$. Записати всі елементи поля $\mathbb{F}_{25}$ у вигляді лінійних комбінацій базових елементів. Кожний елемент $\alpha\in\mathbb{F}_{25}$, $\alpha\neq 0$ подати у вигляді степеня примітивного елемента.

4.2.9. Для $p=2,3,5,7,11,13,17$ знайдіть найменше натуральне число, що породжує групу $\mathbb{F}_p^*$ і визначте скільки серед чисел $1,2,3,\ldots p-1$ є твірних елементів.

4.2.10. Для кожного з наступних полів $\mathbb{F}_q$, $q=p^f$ знайдіть незвідний поліном над простим підполем $\mathbb{F}_p$, корінь α якого є примітивним і всі степені α запишіть як поліноми від α степеня меншого f:
(а) $\mathbb{F}_4$; (б) $\mathbb{F}_8$; (в) $\mathbb{F}_{27}$.

4.2.11. Доведіть таке: якщо b — твірний елемент $\mathbb{F}_{p^f}^*$ і $d\mid f$, то $b^{(p^f-1)/(p^d-1)}$ є твірним елементом $\mathbb{F}_{p^d}^*$.

4.2.12. Доведіть, що кількість коренів з одиниці степеня k в $\mathbb{F}_{p^f}$ дорівнює НСД(k,p^f-1).

4.2.13. Доведіть таке: якщо $a\in\mathbb{F}_q$ і $a^r=1$, то $a^d=1$, де $d=$ НСД$(r,q$–$1)$.

4.2.14. Нехай $m(x) = x^2 + 1$; знайдіть примітивний елемент β поля $\mathbb{F}_3[x]/(m(x))$. Знайдіть також мінімальний поліном елемента β в $\mathbb{F}_3[x]$.

4.2.15. Скільки коренів 15 степеня з одиниці є в полі з 7^3 елементів.

4.2.16. Доведіть, що у векторному просторі над $\mathbb{F}_2$ відстань Хеммінга — це метрика, а вага Хеммінга — норма.

4.2.17. Покажіть, якщо кодова відстань $\geq r + t + 1$, то цей код може виправляти $\leq r$ помилок і виявляти $r + t$ помилок.

4.2.18. Побудуйте код, який складається з восьми слів довжини 7 такий, що відстань між будь-якими різними кодовими словами не менша 4.

4.2.19. Побудувати коди Хеммінга з $m = 2$ та з $m = 4$.

4.2.20. Нехай H — матриця, стовпчики якої є двійковими записами чисел від 1 до m. Довести, шо лінійний код з контрольною матрицею H виправляє одну помилку. Описати правило виправлення.

4.2.21. Знайти всі кодові слова, визначити кодову відстань і знайти контрольну матрицю бінарного лінійного $(5, 3)$ коду з матрицею

$$G = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

4.2.22. Довести, що лінійний код C може виправляти s або меншу кількість помилок тоді й лише тоді, коли його кодова відстань $d \geq s + 1$.

4.2.23. Нехай H — контрольна матриця деякого лінійного коду. Довести, що кодова відстань цього коду дорівнює d тоді й лише тоді, коли кожні $d - 1$ стовпчики матриці H лінійно незалежні та існує d лінійно залежних стовпчиків.

4.2.24. Довести таке: коли лінійний (n, k)-код має кодову відстань d, то $n - k + 1 \geq d$.

4.2.25. Нехай G_1 породжуюча матриця лінійного (n_1, k)-коду з кодовою відстанню d_1, а G_2 породжуюча матриця лінійного (n_2, k)-коду з кодовою відстанню d_2. Довести, що лінійні коди з породжуючими матрицями $\begin{pmatrix} G_1 & 0 \\ 0 & G_2 \end{pmatrix}$ та $(G_1\, G_2)$ є $(n_1{+}n_2, 2k)$-кодом та $(n_1{+}n_2, k)$-кодом з кодовими відстанями $\min\{d_1, d_2\}$ та $d \geq d_1 + d_2$ відповідно.

4.2.26. Нехай C лінійний (n,k) код над полем F_q. Тоді відповідний двоїстий (або ортогональний) код C^* визначається як $C^* = \{(u_1,\ldots,u_n) \in F_q^n \mid u_1v_1 + \ldots + u_nv_n = 0\ \forall (v_1,\ldots,v_n) \in C\}$. Довести, що вимірність C^* дорівнює $n-k$.

4.2.27. Довести, що для кожного лінійного коду C виконується рівність $C = (C^*)^*$.

4.2.28. Довести, що для довільних лінійних кодів C_1, C_2 виконується рівність $(C_1 + C_2)^* = C_1 \cap C_2$.

4.2.29. Нехай C — бінарний $(n,1)$-код з повторенням. Довести, що код C^* є $(n, n-1)$-кодом з перевіркою на парність.

4.2.30. Знайти породжуючу матрицю і всі кодові слова $(7,3)$-коду, двоїстого до бінарного коду Хеммінга C_3.

4.3. Шифри

4.3.1. Доведіть, що відображення $\varphi : x \to ax + b \pmod n$ із $\mathbb{Z}_n$ в себе ін'єктивне тоді і тільки тоді, коли НСД$(n,a) = 1$.

4.3.2. Скільки модулярних шифрів існує в n-символьному алфавіті.

4.3.3. Використайте слово АЛГЕБРА як ключ у шифрі Віженера для шифрування тексту СЕКРЕТНИЙКОД.

4.3.4. Припустимо, що перехоплено три криптотексти: ЕГАРАЛБ, КНЯІФУЦ і ЬЛАВБУО. Як стало відомо перші два відповідають повідомленням АЛГЕБРА і ФУНКЦІЯ. Розшифруйте третій криптотекст. Для цього ж шифру знайдіть криптотекст повідомлення ЛІНІЙНА.

4.3.5. Побудуйте табличку частот літер української абетки (за деяким досить великим текстом) та з її допомогою дешифруйте таке повідомлення ЧГФЗФІУНЮФЦФБЕХФЗФІЕ, якщо відомо, що використовувався шифр зсуву.

4.3.6. Підкидаючи монету, одержіть випадкову послідовність нулів та одиниць і використайте її в криптосистемі одноразового блокнота для шифрування тексту ТАК.

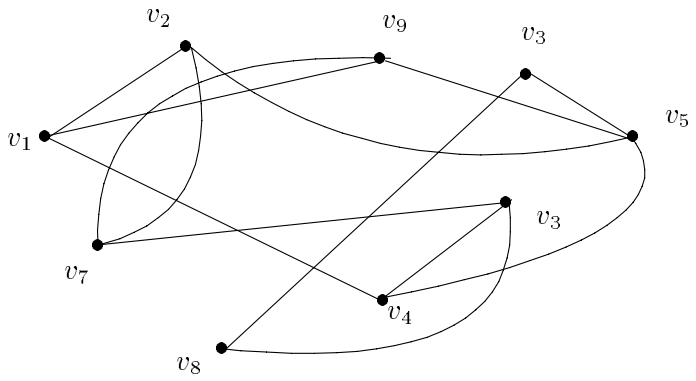
4.3.7. Доведіть коректність шифру RSA: якщо $n = pq$, де p q — різні прості числа, а e і d такі цілі числа, що $ed \equiv 1 \pmod{(p-1)(q-1)}$, то для кожного цілого m виконується $m^{ed} \equiv m \pmod{n}$.

4.3.8. Сформуйте відкритий і таємний ключі для системи RSA на основі $p = 1097$, $q = 97$, $e = 11$. Цифрове повідомлення розбийте на блоки по 5 цифр і
а) зашифруйте повідомлення `МАТЕМАТИКА`;
б) розшифруйте криптотекст 103765 39846.

4.3.9. Припустимо, що Аліса, Боб і Віктор є абонентами комунікаційної мережі, в якій для захисту інформації використовується криптосистема RSA. Знаючи відкриті ключі Боба (e_1, n) і Віктора (e_2, n) такі, що НСД$(e_1, e_2) = 1$, Аліса посилає кожному з них те ж саме повідомлення M. Пояснити, як можна визначити M за криптотекстами $C_1 = M^{e_1} \pmod{n}$ та $C_2 = M^{e_2} \pmod{n}$.

4.3.10. Розділившись на три підгрупи по m чоловік, проробіть m циклів протоколу доведення без розголошення. Нехай перша група знає розв'язання деякої складної задачі і хоче переконати в цьому другу групу. Спілкування відбувається «відкритим каналом зв'язку», тобто відкрито для третьої сторони; завдання якої, незнаючи розв'язку задачі, теж переконати другу групу у тому, що вони його знають.

1. Нехай перша група знає розв'язання задачі відшукання гамільтонового циклу у деякому графі G (напр., див. рис.).



2. Розглянути протокол доведення без розголошення, що ґрунтує-

ться на задачі розпізнавання квадратичних лишків за деяким модулем, наприклад $n = 1000029099973$.

4.3.11. Чи можна певну бульову функцію використовувати як вкорочуючу? Як на основі бульових функцій побудувати хеш-функцію, яка вкорочує 64 бітові блоки до 8 бітів? Скільки є таких функцій?

4.3.12. Розділіть секрет $N = 29$ між трьома особами так, щоб кожні двоє з них не могли його відтворити.

4.3.13. Припустимо, що вчитель хоче дізнатися середній час, який затрачає учень протягом тижня на виконання домашнього завдання з математики. Опишіть протокол, який вирішує це завдання, причому кожен учень зберігає своє число (кількість хвилин) в таємниці.

4.3.14. Як на основі симетричної криптосистеми реалізувати протокол ідентифікації?

Список літератури

[1] *Акимов О.Е.* Дискретная математика, логика, группы, графы. – М., 2001.

[2] *Акритас А.* Основы компьютерной алгебры с приложениями. – М., 1994.

[3] *Бардачов Ю., Соколова Н.А., Ходаков В.Є.* Дискретна математика. – К., 2002.

[4] *Биркгоф Г., Барти М.* Современная прикладная алгебра. – М., 1977.

[5] *Вербіцький О.В.* Вступ до криптології, ВНТЛ. – Львів, 1998.

[6] *Гаврилов Г.П., Сапоженко А.А.* Сборник задач по дискретной математике. – М., 1977.

[7] *Гиндикин С.Г.* Алгебра логики в задачах. – М., 1972.

[8] *Ершов Ю.Л., Палютин Е.А.* Введение в математическую логику. – М., 1979.

[9] *Завало С.Т.,* Алгебра і теорія чисел. Практикум: В 2-х ч. — К., 1983.

[10] *Лавров И.А., Максимова Л.Л.* Задачи по теории множеств, математической логике и теории алгоритмов. – М., 1975.

[11] *Лидл Р., Нидеррайтер Г.* Конечные поля. В 2-х т. – М., 1988.

[12] *Колмогоров А.Н., Драгалин Л.Г.* Введение в математическую логику. – М. 1982.

[13] *Колмогоров А.Н., Драгалин Л.Г.* Математическая логика: дополнительные главы. – М., 1984.

[14] *Кузнєцов О.П., Адельсон-Вольский Г.М.* Дискретная математика для инженеров. – М., 1978.

[15] *Курош А.Г.* Лекции по общей алгебре. – М., 1962.

[16] *Томас Дж.Йех.* Об аксиоме выбора, С.35–63: Справочная книга по математической логике, Ч.II, Теория множеств. – М., 1982.

[17] *Фейс К.* Алгебра: Кольца, модули и категории. Т.1, – М., 1977.

[18] *Яблонский С.В.* Введение в дискретную математику. – М., 1986.

[19] *Ядренко М.Й., Оленко А.Я.* Дискретна математика. Навч.-методичний посібник. – К., 1995.

[20] *Koblitz N.* Algebraic aspects of cryptography, Springer-Verlag. – 1998.

[21] *Ross K., Wright C.* Discrete Mathematics. – 1992.

Предметний покажчик

Навчальне видання

Андрійчук Василь Іванович,
Комарницький Микола Ярославович,
Іщук Юрій Богданович

Вступ до дискретної математики

Редактор Н.Плиса
Технічний редактор С.Сеник
Коректор І.Крук

Підп. до друку . Формат 60 × 84/16. Папір друк. Друк на
Умовн. друк. арк. . Обл.–вид.арк. Тираж 300 прим. Зам.

Львівський національний університет імені Івана Франка
79000 Львів, вул. Університетська, 1

Жовківська книжкова друкарня видавництва Отців
Василіян "Місіонер".
80300, м.Жовква Львівської області, вул. Василіянська, 8