О.А. Борисенко

ДИСКРЕТНА МАТЕМАТИКА

Підручник

Затверджено Міністерством освіти і науки України
як підручник для студентів вищих навчальних закладів

Суми
«Університетська книга»
2008

УДК 621.3.037.37(075.8)
ББК 22.174
Б82

Рекомендовано до друку вченою радою Сумського державного університету. Протокол № 9 від 13.04.2006 р.

Рецензенти:
Ф.М. Лиман, д. ф.-м. наук, професор;
Г.С. Воробйов, д. ф.-м. наук, професор;
Г. К. Чепурних, д. ф.-м. наук, професор

Гриф надано Міністерством освіти і науки України.
Лист №14/18.2-104г від 28.04.2006 р.

Борисенко О.А.

Б82 Дискретна математика: Підручник. – Суми: ВТД «Університетська книга», 2007. – 255 с.

ISBN 978-966-680-376-7

У підручнику, який складається з чотирьох частин і додатків, послідовно викладені елементарні питання теорії множин, логіки, комбінаторики та систем числення. Матеріал викладається конспективно за темами лекцій з великою кількістю прикладів.

У першій частині посібника викладені основні поняття теорії множин і операції з ними.

У другій частині розглядаються елементи математичної логіки. Особлива увага звертається на диз'юнктивні і кон'юнктивні нормальні форми логічних функцій та методи їх мінімізації.

Третя частина містить елементарні комбінаторні конфігурації і біном Ньютона.

У четвертій частині подається теорія і практика сучасних позиційних систем числення.

Для студентів за напрямками підготовки «Електроніка», «Інформатика», «Автоматика».

ББК 22.174

ISBN 978-966-680-376-7

ЗМІСТ

ПЕРЕДМОВА

Шановний читачу, вам пропонується підручник з дискретної математики який містить 41 лекцію, контрольні завдання і додатки. Він значною мірою в перших двох частинах „Множини" і „Елементи математичної логіки" є повторенням виданого автором раніше навчального посібника „Лекції з дискретної математики", в якому були виправлені знайдені помилки і внесені редакційні правки. До цих розділів було додано ще два – „Елементи комбінаторики" і „Системи числення".

Вибір цих розділів, як і перших двох, зумовлений тим, що даний підручник розрахований у першу чергу на студентів, для яких важливе значення мають розділи дискретної математики, що використовуються при вивченні спеціальних дисциплін, таких як цифрова схемотехніка, цифрові автомати, електронні системи, тобто студентів, які навчаються в першу чергу за напрямком „Електроніка". Ці розділи також важливі і для майбутніх спеціалістів з інформатики і автоматики.

При виборі і написанні вказаних розділів автор керувався своїм багаторічним практичним досвідом розробника цифрової техніки. Тому матеріал, наданий у підручнику, буде корисний не лише студентам, а й спеціалістам в галузі розроблення й експлуатації цифрової апаратури. Особливо це стосується розділів, що містять оригінальний матеріал, який ще невідомий широкому загалу і який можна використовувати при розробленні сучасної техніки і в науковій роботі.

Хоча навчальний матеріал в підручнику має практичну спрямованість, але в певних формальних межах викладається у вигляді теорем та їх наслідків навіть у тих випадках, коли звичайно він сприймається на інтуїтивному рівні. Така формалізація робить зміст тексту більш строгим і чітким, що дозволяє порівняно легко орієнтуватися в ньому. Цей матеріал вибирався з багатьох джерел, систематизувався, перероблявся, після чого був неодноразово апробований у навчальному процесі.

Одним із завдань цієї книги, було зробити зрозумілим і доступним її зміст для якнайширшого кола читачів, що не мають спеціальної математичної підготовки. Для цього в підручник поряд із викладенням теоретичних питань введено велику кількість різних прикладів з розв'язаннями і вправ, а також контрольні задачі для перевірки й самоперевірки знань студентів. Ці задачі, як і весь навчальний матеріал в цілому, призначені також і для дистанційного навчання, яке набуває сьогодні особливого значення в усьому світі.

Крім того, завданням автора було викладення матеріалу таким чином, щоб студент, особливо заочної форми навчання, був здатен самостійно розібратися й вивчити той чи інший матеріал протягом мінімального часу. Не останню роль при цьому відігравали і міркування щодо доступності книги за вартістю.

У 1998 році цей підручник, що виглядав тоді ще як конспект лекцій, поряд з науковими матеріалами в галузі дискретної математики і теорії інформації виграв конкурс на грант фонду "Відродження" за міжнародною науково-освітньою програмою (МНОП) для наукових співробітників і викладачів серед математиків (№ QSU081019).

Автор висловлює свою подяку фонду "Відродження" і його засновникам – Уряду України та Інституту відкритого суспільства (США) за сприяння закінченню роботи над даним підручником, а також проведенню подальшої наукової роботи в галузі комп'ютерної математики та її практичного застосування.

Автор також щиро дякує керівництву Сумського державного університету за підтримку в написанні даного підручника, його співробітникам та викладачам кафедри електроніки і комп'ютерної техніки, які своїми зауваженнями і побажаннями сприяли покращенню її змісту, а також студентам, що були читачами перших видань цієї книги і без яких вона втратила б свою первісну суть.

Лекція 1
ВСТУПНА

Обмін інформацією в системах комп'ютерної обробки здійснюється за допомогою сигналів, носіями яких можуть бути будь-які фізичні величини, такі як струм, напруга й т. і. Розрізняють такі види сигналів: аналогові (неперервні) сигнали й дискретні сигнали.

1. Аналогові, або неперервні сигнали

Усередині заданого діапазону в довільний момент часу вони набувають будь-яких значень і змінюються згідно з тим чи іншим правилом, тобто є неперервними функціями часу.

2. Дискретні сигнали

Параметри такого сигналу можуть змінюватися стрибкоподібно і набувати лише певних значень у дискретні моменти часу. Це призводить до того, що перешкодам при невеликому їх рівні досить важко спотворити сигнал, оскільки він змінюється лише тоді, коли їх рівень досягне певної межі. Крім того, дискретні сигнали простіше зберігаються й обробляються, і тому сьогодні вони мають більш широке практичне застосування, ніж неперервні.

Існують два види дискретних сигналів:

1. *Дискретні сигнали, отримані способом дискретизації неперервних сигналів.*
2. *Дискретні сигнали, подані у вигляді кодових комбінацій – слів.*

Останні є найбільш універсальними й поширеними. Вони широко застосовуються для обробки інформації в практичній і теоретичній діяльності людини. Це спричинило появу дискретної математики, основною особливістю якої є відсутність у її задачах граничного переходу і неперервності, притаманних класичній математиці.

Дискретна математика – це наука про способи побудови та ефективної обробки слів – послідовностей літер a_1, a_2 ... a_i ... a_n, породжуваних відповідно до заданих правил деяким алфавітом $B = \{b_1, b_2, ..., b_m\}$, $a_i \in B$. Самі слова у свою чергу кодують об'єкти навколишнього світу.

Оскільки більшість дискретних пристроїв здійснює переробку інформації, що подається у вигляді слів, то дискретна математика знаходить широке застосування при їх побудові. Використання слів для кодування інформації не накладає обмежень на сферу застосування дискретних пристроїв, оскільки неперервна інформація може бути з будь-якою точністю апроксимована дискретними сигналами.

Найчастіше використовуються дискретні пристрої, в яких сигнали мають два різних рівня. Це пов`язане з простотою фізичної реалізації таких пристроїв, міркуваннями щодо їх надійності й простоти виконання логічних і арифметичних операцій. Одному з рівнів при цьому присвоюється знак 0, а другому – 1, і, таким чином, пристрій здійснює обробку двійкової інформації – слів у алфавіті $B = \{0,1\}$. Завдяки зазначеним перевагам дискретні пристрої з двійковою інформацією сьогодні займають домінуюче положення в галузі передачі та обробки інформації. Судячи з усього, в перспективі роль цих пристроїв буде тільки зростати.

Електронні обчислювальні машини (ЕОМ) були спочатку аналоговими. Сьогодні цифрові ЕОМ практично витіснили аналогові. Те ж саме відбувається з приймачами, телевізорами та іншими інформаційними пристроями. Протягом найближчих років цифрова електроніка займатиме монопольне положення на ринку електронних систем і пристроїв.

Проте цифрова техніка витіснити аналогову не спроможна в принципі, оскільки фізичні об`єкти, від яких цифрові ЕОМ отримують інформацію, а також об'єкти керування звичайно мають аналогову природу. Тому на вході й виході цифрових ЕОМ потрібні аналогові, цифро-аналогові і аналого-цифрові пристрої. Крім того, часто в найвідповідальніших місцях цифрових схем використовуються аналогові елементи.

Аналогова електроніка має також неперевершену швидкодію, що обмежується тільки швидкістю фізичних процесів, і, як наслідок, її значення зовсім не зменшується, незважаючи на те, що питома вага аналогової електроніки спадає. Тому "неперервна" математика в освіті фахівця з електроніки та автоматики не втрачає свого значення, хоча роль дискретної математики при цьому постійно зростає.

Дискретна математика, крім її застосування для побудови універсальних електронних цифрових пристроїв та систем, також широко використовується для створення та експлуатації комплексних автоматизованих систем обробки інформації, пакетів прикладних програм, банків даних, мікропроцесорних систем спеціального призначення, мереж передачі даних. Тому в наш час методи й засоби дискретної математики розвиваються досить інтенсивно.

Дискретна математика охоплює теорію множин і алгебраїчних систем, математичну логіку, теорію графів, теорію автоматів і формальних граматик, теорію алгоритмів, а також теорію кодування, теорію чисел, комбінаторні обчислення, рішення дискретних екстремальних задач. Серед перелічених розділів дискретної математики особливо вирізняються дискретні задачі на побудову, перебір та нумерацію дискретних об`єктів. Багато інших задач або зводяться до них, або містять їх в явному чи неявному вигляді. До появи обчислювальної техніки ці задачі не мали особливого практичного значення, оскільки для розв'язань задач малої розмірності відшукувались алгоритми, які не потребували великої кількості обчислень, а для розв'язань задач великої розмірності, що потребували значно більшої кількості обчислень, не було технічних засобів їх реалізації.

Особливі труднощі для розв`язування багатьох дискретних задач спричинені відсутністю основоположних теорем, з котрих випливали б методи й алгоритми їх розв`язання. Їх розроблення є однією з центральних проблем теорії дискретної математики.

На сьогодні у сфері розв'язання дискретних задач сформульовані лише загальні принципи, що об`єднують дискретні алгоритми в окрему галузь знань. Це методи решета, універсального розв'язувача задач, пошуків з поверненням. Але безпосереднє застосування цих методів, як правило, веде до алгоритмів, робота з якими потребує багато часу. Тому на практиці широке застосування знаходять евристичні прийоми розв'язання дискретних задач. Їх недолік – вузька сфера застосування, що призводить до необхідності пошуків розв`язків дискретної задачі для кожного конкретного випадку. Однак навіть ці розв'язання мають велику цінність, оскільки дозволяють у разі їх використання підвищити ефективність цифрових систем і пристроїв.

Особливе значення в практичній електроніці мають методи логічного синтезу й комбінаторики, на які звертається особлива увага. Основу цих методів становить теорія множин, елементарні розділи якої є теоретичною базою даного підручника. Тому ці розділи розміщені на початку книги, і лише після них знаходяться деякі найпростіші розділи логіки, а потім комбінаторики, що мають практичне застосування.

У підручнику значну увагу приділено теорії і практиці систем числення, оскільки знання в цій галузі є важливою передумовою підготовки спеціалістів з мікропроцесорної і взагалі обчислювальної техніки. Системи числення потребують свого розвитку, як і інші напрямки дискретної математики, і в перспективі можуть сприяти появі нетрадиційних методів кодування й обробки інформації.

Викладений в підручнику матеріал так чи інакше раніше використовувався як вступні лекції в спеціальних технічних курсах, тому ця книга є не тільки підручником з дискретної математики, а й допоміжним посібником при викладенні спеціальних дисциплін.

Математика – наука молодих.
Н. Вінер

Частина I

МНОЖИНИ

Лекція 2
ОСНОВНІ ОЗНАЧЕННЯ

Під ***множиною*** зазвичай розуміють об`єднання в одне ціле об`єктів, що добре розрізняються нашою думкою або інтуїцією.

Об`єкти, що утворюють множину, будемо називати ***елементами*** множини. Елементи звичайно позначаються малими літерами латинського алфавіту, а множини великими. Якщо m є елемент, який *належить* множині M, то використовується запис $m \in M$, у протилежному випадку – $m \notin M$. Читається так: m належить M.

Множина, яка містить скінченне число елементів, називається *скінченною*, а множина, що містить нескінченне число елементів, – *нескінченною.*

Множина може задаватися різними способами: *перерахуванням* елементів для скінченної множини або зазначенням їх *властивостей.* У разі перерахування використовуються фігурні дужки { }. Наприклад, множину M цифр десяткового алфавіту можна задати у вигляді $M = \{0, 1,..., 9\}$.

Цю ж саму множину можна задати й інакше, як $M = \{i \mid i$ – ціле, $0 \le i \le 9\}$, де справа від вертикальної риски зазначається *властивість* елементів цієї множини.

Якщо ж множина M є множиною, наприклад, парних чисел, то вона записується як $M = \{m \mid m$ – парне число$\}$.

Іноді нескінченні множини задаються простим перерахуванням кількох перших елементів, і тоді характеристична властивість є заданою в неявному вигляді. Наприклад, множину парних чисел можна задати у вигляді $A = \{2, 4, 6, 8, \ldots\}$.

Для того, щоб позначити множину всіх предметів, що є елементами множини A і мають властивість P, замість $\{x \mid x \in A$ і $P(x)\}$ часто пишуть $\{x \in A \mid P(x)\}$.

Наприклад, $\{x \in R \mid 0 \le x \le 1\}$ означає множину всіх дійсних чисел між 0 та 1 включно, а $\left\{x \in Q^{+} \middle| \; x^2 < 2\right\}$ – множину всіх додатних раціональних чисел, квадрати яких менші за число 2.

Множина M' називається ***підмножиною*** множини M тоді й лише тоді, коли будь-який елемент множини M' належить до множини M. Якщо при цьому припускається, що множини M' і M можуть збігатися, тобто M' може дорівнювати M, то тоді пишуть, що

$M' \subseteq M$, де $\subseteq$ – знак *включення* підмножини. Невключення підмножини M' до множини M позначається як $M' \not\subseteq M$.

Множина A, що *строго* включена до B, позначається як $A \subset B$. Це означає, що B містить і інші елементи, крім елементів A, і це особливо підкреслюється.

Дві множини *рівні* в тому і лише в тому разі, коли вони складаються з одних і тих самих елементів. Тому будь яка множина $X = X$.

Рівність двох множин X і Y позначається через $X = Y$, а нерівність множин X і Y – через $X \neq Y$. Для будь-яких рівних множин $X, У$ і X, Y, Z виконуються умови:

1. Якщо $X = Y$, то $Y = X$;
2. Якщо $X = Y$, $Y = Z$, то $X = Z$.

Порядок елементів у множині не є суттєвим. Множини $\{3, 4, 5, 6\}$ і $\{4, 5, 6, 3\}$ являють собою одну й ту саму множину.

Множини не містять однакових елементів. Так, множина простих дільників числа 60 дорівнює $\{2, 3, 5\}$, а не $\{2, 2, 3, 5\}$.

Слід розрізняти об'єкт і множину, єдиним елементом якої є цей об'єкт. Так множина $\{1, 2\}$ становить об'єкт, який є елементом множини $\{\{1, 2\}\}$. Множини $\{\{1, 2\}\}$ і $\{1, 2\}$ не рівні, оскільки перша – одноелементна множина, що має єдиний елемент $\{1, 2\}$, а друга має два елементи 1 і 2.

Якщо множина не містить жодного елемента, то вона називається *порожньою* і позначається $\varnothing$. Інколи її називають ще *пустою* множиною. Наприклад, множина трикутників з двома прямими кутами є порожньою. Також множина простих чисел, які діляться на число чотири, є порожньою. Але множина $\{\varnothing\}$ не є порожньою, тобто $\varnothing \neq \{\varnothing\}$; ця множина є одноелементною. Її єдиним елементом є порожня множина.

Теорема 1. ***Порожня множина є підмножиною будь-якої іншої множини.***

Доведення. Щоб встановити це, потрібно довести, що в тому випадку, коли A є довільною множиною, кожний елемент множини $\varnothing$ є елементом множини A. Оскільки $\varnothing$ не має елементів, то ця умова виконана.

Інше доведення. Припустимо, що $\varnothing \subseteq A$ – хибне твердження. Це може бути лише в тому разі, якщо існує деякий елемент множини $\varnothing$, що не є елементом множини A. Але це неможливо, оскільки $\varnothing$ не має елементів. Отже, умова $\varnothing \subseteq A$ не є хибною, тобто $\varnothing \subseteq A$.

Теорему доведено.

Порожня множина відіграє значну роль в науці. Кількість елементів у порожній множині дорівнює *нулю*. А нуль має особливе значення в математиці і техніці. Без нього, наприклад, не можна побудувати позиційні системи числення, а без них – сучасну обчислювальну техніку. Тому до порожньої множини слід ставитися як до базової множини.

З теореми 1 випливає, що кожна множина $A \neq \varnothing$ має, принаймні, дві підмножини: A і $\varnothing$.

Оскільки будь-яка множина є своєю підмножиною $A \subseteq A$, то і $\varnothing \subseteq \varnothing$.

Кожний елемент a множини A визначає деяку її підмножину. Це означає, що якщо $a \in A$, то і $\{a\} \subseteq A$. Поєднання двох будь-яких випадково взятих елементів множини A також створює її підмножину. Таких підмножин може бути декілька, і їх кількість визначається числом різних поєднань чи комбінацій з двох елементів, узятих із множини A. Тобто, якщо число елементів множини A дорівнює трьом, хай це будуть елементи 1, 2, 3, то можливі такі поєднання двох елементів із трьох: 1, 2; 1, 3; 2, 3.

Також будь-які три елементи множини A створюють її підмножину. Кількість таких трьохелементних підмножин визначається числом різних комбінацій трьох елементів з числа елементів n множини A. Далі, якщо n більше від трьох, можна визначити число комбінацій чотирьох, п'яти і більшого числа елементів множини A з n. У загальному випадку число комбінацій i елементів з n позначається символом C_n^i, $i = 0,1, ..., n$. Ці комбінації називаються ще *сполученнями*. Тобто ***сполучення*** – це будь-яка i-елементна підмножина n-елементної множини.

Усі підмножини множини A, які складаються з n елементів, створюють множину, що називається ***множиною-степенем***, або ***булеаном*** множини A і позначається $P(A)$ або $B(A)$. Якщо, наприклад, $A = \{1, 2, 3\}$, то множина-степінь

$$P(A) = \{\{1, 2, 3\},\{1, 2\},\{1, 3\},\{2, 3\},\{1\},\{2\},\{3\},\{\varnothing\}\}.$$

Теорема 2. ***Множина-степінь P(A) множини $A = \{a_1, a_2, ..., a_n\}$ має 2^n різних підмножин.***

Доведення. Перший елемент, який вибирається, може входити, а може і не входити в підмножину, що будується. Тобто для цього елемента є дві альтернативи – увійти або не ввійти в підмножину. Для другого елемента залишаються ті ж самі дві альтернативи, і це буде властиве всім елементам до останнього із загального їх числа *n*. Якщо перемножити тепер дві перші альтернативи на дві другі, і так *n* раз, то отримаємо потрібний результат – 2^n.

Теорему доведено.

При цьому в одному випадку жоден елемент множини А не зможе ввійти в підмножину, яку будують, а в іншому зможуть увійти всі. У першому з цих випадків маємо справу з порожньою чи пустою множиною, а в другому – з множиною *A*.

Наслідок. Число підмножин множини *A*, що містять $i = 0, 1, ..., n$ елементів, визначається відповідно числом сполучень 0, 1, ..., *n* елементів з *n*. Тому загальне число підмножин дорівнюватиме сумі чисел цих сполучень, тобто 2^n.

Множина-степінь має прикладне значення в інформатиці й цифровій техніці. Наприклад, з її допомогою можна побудувати всі *n*-розрядні двійкові числа або всі рівноважні коди довжини *n*. Цим частково пояснюється та увага, яка приділяється даній множині в підручниках і посібниках з дискретної математики.

Розглянемо відповідно до викладеного матеріалу приклади.

Приклад 1. Довести, що множина *A* всіх додатних парних цілих чисел дорівнює множині *B* усіх додатних цілих чисел, що подаються як сума двох додатних непарних цілих чисел.

Розв`язання. Припустимо, що $x \in A$, і доведемо, що $x \in B$. Якщо $x \in A$, то $x = 2m$, або $x = (2m-1)+1$. Це означає, що $x \in B$.

Припустимо тепер, що $x \in B$, і виведемо звідси, що $x \in A$.

Якщо $x \in B$, то $x = (2p-1)+(2q-1)$, звідки $x = 2(p+q-1)$, з чого випливає, що $x \in A$.

Таким чином, ми довели, що множини *A* і *B* складаються з одних і тих самих елементів, отже доведення отримане.

Приклад 2. Чи є множини $\{2, 4, 6\}$, $\{2, 6, 4\}$ рівними?

Розв`язання. Оскільки множини $\{2, 4, 6\}$ і $\{2, 6, 4\}$ складаються з одних і тих самих елементів, то вони будуть рівними.

Приклад 3. Чи є множини $\{\{1, 2\},\{2, 3\}\},\{1, 2, 3\}$ і $\{1, 2, \{3\}\}$ рівними?

Розв`язання. Ні, не будуть, оскільки елементами першої множини будуть елементи $\{1, 2\}$ і $\{2, 3\}$, другої – 1, 2, 3 і третьої – 1, 2, $\{3\}$.

Приклад 4. Чи є множини $\{\{1, 2\}\}$ і $\{1, 2\}$ рівними?

Розв`язання. Ні, не є рівними, оскільки перша – одноелементна множина, що має своїм єдиним елементом $\{1, 2\}$, а друга має своїми елементами 1 і 2.

Приклад 5. Чи правильно, що $\{1, 2\}\in\{\{1, 2, 3\}, \{1, 3\}, 1, 2\}$?

Розв`язання. Ні, не правильно, оскільки в правій множині відсутній елемент $\{1, 2\}$.

Вправи

1. Які з наведених нижче тверджень правильні, а які ні:

а) якщо $A \neq B$ і $B \neq C$, то $A \neq C$;
б) якщо $A \subset B$ і $B \subseteq C$, то невірно, що $C \subseteq A$;
в) якщо $A \subseteq B$ і $B \subset C$, то $A \not\subset C$?

2. Перелічити всі елементи множини $A = \{\{1,2\},\{3\},1\}$.

3. Довести істинність кожного з тверджень для довільних множин A, B і C:

а) якщо $A \subseteq B$ і $B \subseteq C$, то $A \subseteq C$;
б) якщо $A \subseteq B$ і $B \subset C$, то $A \subset C$;
в) якщо $A \subset B$ і $B \subset C$, то $A \subset C$.

4. Які з наведених нижче співвідношень хибні і чому:

а) $x \in \{2, a, x\}$;
б) $3 \in \{1, \{2, 3\}, 4\}$;
в) $\{x, y\} \in \{a, \{x, y\}, b\}$?

5. Чи рівні між собою множини:

а) $A = \{2, 5, 4\}$ $B = \{5, 4, 2\}$;
б) $A = \{1, 2, 3\}$ $B = \{1, 2, 4\}$;
в) $A = \{2, 4, 5\}$ $B = \{2, 4, 3\}$;
г) $A = \{1, \{2, 5\}, 6\}$ $B = \{1, \{5, 2\}$;
д) $A = \{1, \{2, 5\}, 6\}$ $B = \{1, 2, 5, 6\}$?

6. Як співвідносяться між собою такі множини: $A = \{1, 3\}$; B – множина непарних додатних чисел; C – множина розв`язків рівняння $x^2 - 4x + 3 = 0$?

7. Для множини перших 20 натуральних чисел запишіть такі її підмножини: A – парних чисел; B – непарних чисел; C – квадратів чисел; D – простих чисел. В яких відношеннях перебувають ці підмножини?

Вивчення математики наближає до безсмертних богів.

Платон

Лекція 3
ОПЕРАЦІЇ НАД МНОЖИНАМИ

1. Об'єднання і переріз множин

Розглянуті нижче операції над множинами необхідні для подальшого викладу матеріалу і мають велике значення для розв'язання багатьох задач дискретної математики, особливо тих, що пов'язані із синтезом дискретних автоматів.

***Об`єднанням*** множин $A$ і $B$ називається множина, що складається з усіх тих і лише тих елементів, які належать хоча б одній з множин $A$ або $B$. Позначається $A \cup B = \{x \mid x \in A$ або $x \in B\}$ (називається *сумою* або *об`єднанням*).

Таким чином, за наведеним означенням $x \in A \cup B$ тоді і лише тоді, коли x є елементом хоча б однієї з множин A або B. Наприклад, $\{1, 2, 3\} \cup \{1, 3, 4\} = \{1, 2, 3, 4\}$.

Об'єднання множини A з порожньою множиною буде давати ту ж саму множину A : $A \cup \varnothing = A$.

Аналогічно визначається об`єднання довільної (у тому числі й нескінченної) системи множин. Якщо система містить невелику кількість множин, то їх об`єднання описується явно, наприклад: $A \cup B \cup C \cup D$.

У випадку, якщо всі множини пронумеровані індексами й належать до системи множин $S = \{A_1, A_2, ..., A_к\}$, то їх відображають в вигляді $\bigcup_{i=1}^{k} A_i$, або $\bigcup_{i=1}^{\infty} A_i$, де S – нескінченна система, і її множини пронумеровані підряд натуральними числами.

Для об`єднання множин справедливі *комутативний і асоціативний закони:*

1. Комутативний закон

$$A \cup B = B \cup A.$$

2. Асоціативний закон

$$(A \cup B) \cup C = A \cup (B \cup C).$$

Справедливість цих законів випливає з того, що ліва і права частини наведених рівностей складаються з одних і тих самих елементів, а порядок їх об'єднання для множин не має значення.

Розглянуті вище закони справедливі також і для об'єднання множин з порожньою множиною. Це випливає з того, що порожня множина, хоча вона і не містить у своєму складі елементів, але є взагалі такою ж самою, як і всі інші.

Перетином (*добутком, перерізом*) множин A і B називається множина, що складається з усіх тих і тільки тих елементів, які належать як до множини A, так і до множини B. Позначається $A \cap B$. До цієї множини належать лише спільні елементи множин A і B.

Формальне означення

$$A \cap B = \{ x \mid x \in A \text{ і } x \in B \}.$$

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

Для перетину і об'єднання множин властиві такі *включення*:

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

Вважається, що дві множини A і B не перетинаються, якщо $A \cap B = \varnothing$, і перетинаються, якщо $A \cap B \neq \varnothing$.

Перетин множин має комутативну

$$A \cap B = B \cap A$$

і асоціативну властивість

$$(X \cap Y) \cap Z = X \cap (Y \cap Z) = X \cap Y \cap Z.$$

Для порожньої множини має місце також співвідношення $X \cap \varnothing = \varnothing$, яке твердить, що перетин будь-якої множини з порожньою множиною дає таку ж саму порожню множину.

2. Різниця множини

Різницею множин A і B або відносним доповненням множини B до A називається множина, що складається з усіх тих і лише тих елементів, які належать A і не належать B. Визначається лише для *двох* множин. Наприклад, різниця між натуральними і парними числами являє собою множину всіх непарних натуральних чисел.

Різниця множин A і B позначається як $A \setminus B$, або $A - B$, що відповідає умові $\{a | a \in A$ і $a \notin B\}$, яка визначає ті елементи множини A, що не є елементами множини B. Саму операцію знаходження різниці двох множин називають ***відніманням*** множин.

Нехай $A = \{1, 3, 4, 5\}$, а $B = \{1, 2, 3\}$. Тоді отримана при відніманні множини B від A різниця $A - B = \{4, 5\}$.

Множина $A + B = (A - B) \cup (B - A)$ називається *симетричною* різницею.

Якщо $A = \{1, 3, 4, 5\}$, а $B = (1, 2, 3\}$, то симетрична різниця $A + B = \{2, 4, 5\}$.

Теорема 1. $A \setminus B = A \cap \overline{B}$.

Доведення. Оскільки

$$A \setminus B = \{x \mid x \in A \text{ і } x \notin B\} = \{x \mid x \in A \text{ і } x \in \overline{B}\}, \text{ то } A \setminus B = A \cap \overline{B}.$$

Теорему доведено.

3. Універсальна множина

Множина I (позначається також U), для якої решта всіх інших множин є підмножинами, називається ***універсальною*** множиною, а також ***повною,*** або ***одиничною***. Іноді вона ще називається ***універсамом.***

Універсальна множина є поняттям відносним. Наприклад, в арифметиці універсальною множиною вважається множина раціональних чисел, а її підмножиною буде множина цілих чисел. У той же самий час множина натуральних чисел є підмножиною цілих чисел.

Для універсальної множини виконуються рівності

$$A \cap I = A, \ A \cup I = I.$$

4. Абсолютне доповнення множин

Множина $\overline{A}$, що визначається за співвідношенням $\overline{A} = I \setminus A$, називається ***абсолютним доповненням,*** або просто ***доповненням*** множини A до універсальної множини I. Із приведеної рівності видно, що не тільки $\overline{A}$ є доповненням до I, але й A є доповненням до I, тобто завжди $A \cup \overline{A} = I$. Далі $\overline{\overline{A}} = I \setminus \overline{A} = I - \overline{A} = I - (I - A) = A$. Із цього випливає, що $A = \overline{\overline{A}}$. Також очевидно, що A і $\overline{A}$ не мають спільних елементів. Тому $A \cap \overline{A} = \varnothing$.

5. Розбиття множин

Будь-яка сукупність n множин: A_1, A_2, ..., A_n, що розглядається, називається ***системою*** множин.

Система множин S називається ***розбиттям*** множин M, якщо вона задовольняє таким умовам:

1. Будь-яка множина A системи S є підмножиною множини M: $A \subseteq M$.

2. Будь-які дві множини A і B з S не перетинаються: $A \cap B = \varnothing$.

3. Об`єднання всіх без винятку множин системи S утворює множину M

$$\bigcup_{i=1}^{n} A_i = M \text{ , } A_i \in S.$$

Розбиття множин широко використовується як у математичних теоріях, так і на практиці, особливо в задачах з кодування інформації. Тому спеціаліст у галузі інформатики й цифрової схемотехніки досить часто буде зустрічатися з такими задачами.

Вправи

1. Довести, що для будь-яких множин A і B справедливо

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

2. Показати, що для будь-якої множини A справедливі співвідношення

$$A \cap \overline{A} = \varnothing; \; A \cup \varnothing = A.$$

3. Нехай A – довільна множина. Що являють собою наступні множини

$$A \cap \varnothing, \quad A \cup \varnothing, \; A - \varnothing, \; A - A, \; \varnothing - A?$$

4. Показати, що із співвідношень $A \bigcap B = C$ випливає, що

$$C \subseteq A \text{ і } C \subseteq B.$$

5. Довести, що множина

$$(M - N) \bigcap (N - M) = \varnothing.$$

6. Довести, що

а) $A - (A - B) = B - (B - A)$;
б) $(A - B) - C = (A - C) - (B - C)$.

7. В якому співвідношенні знаходяться множини A і B, якщо

$$A - B = B - A = \varnothing\,?$$

8. Довести, що для довільних множин $A_1, A_2, \ldots, A_n$ $(n \geq 2)$

$$A_1 \bigcup A_2 \bigcup \ldots \bigcup A_n = (A_1 - A_2) \bigcup (A_2 - A_3) \bigcup \ldots$$
$$\bigcup \; (A_{n-1} - A_n) \bigcup (A_n - A_1) \bigcup (A_1 \bigcap A_2 \bigcap \ldots \bigcap A_n).$$

9. Довести, що

а) $A \bigcup (B - A) = A \bigcup B$;
б) $A \bigcap (B - A) = \varnothing$;
в) $A \bigcap (B - C) = (A \bigcap B) - C$;
г) $A - (B \bigcap C) = (A - B) \bigcup (A - C)$;
д) $A - (B \bigcup C) = (A - B) \bigcap (A - C)$;
е) $(A \bigcup B) - C = (A - C) \bigcup (B - C)$.

Знайти доведення математичної теореми важливише, ніж завоювати перське царство.
Демокріт

Лекція 4
ДІАГРАМИ ЕЙЛЕРА

Для наочного зображення операцій над множинами досить часто використовують діаграми (круги) *Ейлера.* Універсальна множина U зображається у вигляді точок деякого прямокутника, а її підмножина – як круг усередині прямокутника (рис.1). Доповнення $\overline{A}$ множини A до U зображається тією частиною прямокутника, яка лежить поза кругом, що зображає A. Якщо на діаграмі Ейлера зобразити кругами множини A і B, що є підмножинами U, то множини $A \cap B$ і $A \cup B$ будуть зображені заштрихованими областями на рис. 2 і 3 відповідно.

Множини для різниць $A - B$ і $B - A$, а також симетричної різниці $A + B$ показані заштрихованими ділянками на рис. 4, 5, 6 відповідно.

Слід зазначити, що, крім кругів чи діаграм Ейлера, на практиці досить часто використовуються і діаграми *Вена.* Вони відрізняються від кругів Ейлера тим, що в них застосовуються довільні криві, тобто діаграми Вена є узагальненням кругів Ейлера. Вони почали застосовуватися на практиці більш як на сто років пізніше, ніж круги Ейлера. Досить часто діаграми Вена називають також діаграмами Ейлера і зворотно – круги Ейлера називають діаграмами Вена, хоча це і не зовсім правильно. Часто круги Ейлера називають також діаграмами Ейлера – Вена. Надалі ми будемо наводити тільки круги (діаграми) Ейлера і так їх будемо називати.

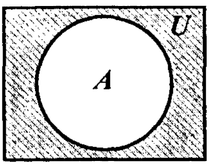


Рис. 1. Діаграма Ейлера для множини A (заштриховано $\overline{A}$)

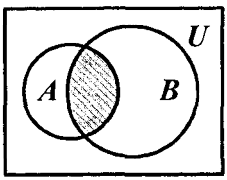


Рис. 2. Діаграма Ейлера для двох множин A і B, які перетинаються (заштриховано $A \cap B$)

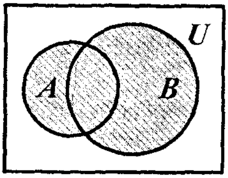


Рис.3. Діаграма Ейлера для двох об'єднаних множин $A \bigcup B$
(заштриховано $A \bigcup B$ *)*

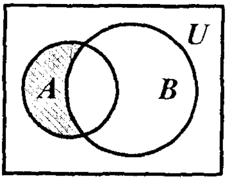


Рис. 4. Діаграма Ейлера для різниці двох множин $A - B$
(заштриховано $A - B$*)*

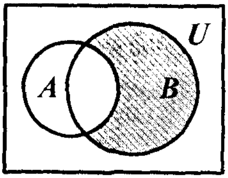


Рис.5. Діаграма Ейлера для різниці двох множин $B - A$
(заштриховано $B - A$*)*

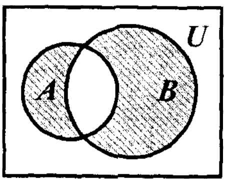


Рис. 6. Симетрична різниця двох множин $A + B$
(заштриховано $A + B$*)*

Приклад 1. Чи існують підмножини A, B і C універсальної множини U, для яких одночасно мали б місце такі співвідношення:

$$C \neq \varnothing, \quad A \bigcap B \neq \varnothing, \quad A \bigcap C = \varnothing, \quad A \bigcap B - C = \varnothing?$$

Розв`язання. З другої умови випливає, що A і B перетинаються, з чого стає зрозумілим, що обидві множини не пусті. Четверта умова стверджує, що $A \cap B \subseteq C$. З цього видно, що перша умова зайва. З одного боку, $A \cap B$ належить C, а з іншого – $A \cap C$ є пустою множиною. Це суперечність. Отже, множин, що задовольняють усім наведеним умовам, не існує. Тому неможливо побудувати відповідну діаграму Ейлера.

Приклад 2. Нехай F, G, L – такі підмножини множини U, що $F \subseteq G$, $G \cap L \subseteq F$, $L \cap F = \varnothing$. Чи існують множини F, G, L, які задовольняли б зазначеній сукупності умов?

Розв`язання. Оскільки $L \cap F = \varnothing$ і $G \cap L \subseteq F$, то $G \cap L = \varnothing$.

З іншого боку, якщо $F \subseteq G$ і $G \cap L = \varnothing$, то виконуються всі умови задачі, і відповідно існують множини F, G, L, які їм задовольняють (див. рис. 7).

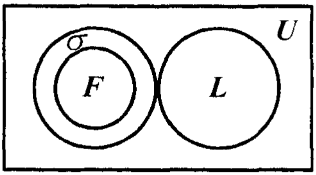


Рис. 7. Діаграми Ейлера для трьох множин F, G, L приклада 2

Розглянуті задачі – це задачі, в яких доводиться існування чи відсутність множин із заданими умовами. Ці задачі можуть бути дуже складними і для свого розв'язання потребувати значних зусиль.

Вправи

1. Побудувати діаграму Ейлера, що відповідає симетричній різниці $A + \overline{B} = (A - \overline{B}) \cup (\overline{B} - A)$ множин A і B.

2. За допомогою діаграми Ейлера довести, що якщо $A \cap B = \overline{C}$ і $A \cup C \subseteq B$, то $A \cap C = \varnothing$.

3. Записати за допомогою операцій над множинами вирази для множин, що відповідають заштрихованим ділянкам діаграм Ейлера 1, 2, 3 на рис. 8, 9, 10.

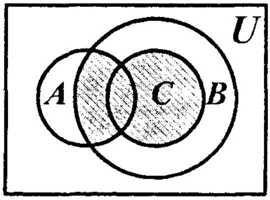


Рис.8. Діаграма Ейлера 1

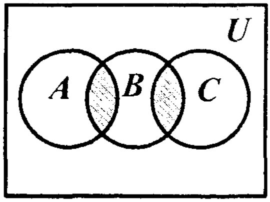


Рис 9. Діаграма Ейлера 2

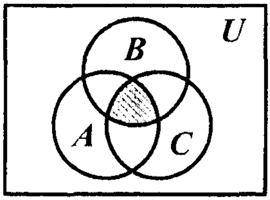


Рис. 10. Діаграма Ейлера 3

4. Довести за допомогою кругів Ейлера тотожність

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

5. Довести за допомогою кругів Ейлера тотожність

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

6. За допомогою кругів Ейлера показати, що

а) $A \cap B \subseteq A \cup B$;
б) $A + \bar{A} = \varnothing$;
в) якщо $A \cap B = C$, то $C \subset A$ і $C \subset B$;
г) $\overline{(M - N)} \cap \overline{(N - M)} = U$.

Лекція 5
АЛГЕБРА МНОЖИН

1. Загальні положення

Алгебра множин створюється з допомогою операцій між підмножинами універсальної множини як сукупність рівностей – тотожностей.

Наприклад, для будь-яких підмножин (множин) A, B і A, B, C універсальної множини U дійсними є рівності:

1. $A \cup (B \cup C) = (A \cup B) \cup C$;
2. $A \cup B = B \cup A$;
3. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$;
4. $A \cup \varnothing = A$;
5. $A \cup \overline{A} = U$; $(\overline{A} = U - A)$.

1'. $A \cap (B \cap C) = (A \cap B) \cap C$;
2'. $A \cap B = B \cap A$;
3'. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$;
4'. $A \cap U = A$;
5'. $A \cap \overline{A} = \varnothing$.

Кожну з наведених рівностей можна довести, показавши, що будь-який елемент множини, що стоїть з одного боку від знака рівності, належить до множини, яка стоїть з іншого боку від цього знака рівності.

Доведемо рівність 3. Доведення складається з двох частин:

1. Нехай $x \in A \cup (B \cap C)$. Тоді $x \in A$ або $x \in B \cap C$. Якщо $x \in A$, то $x \in A \cup B$ і $x \in A \cup C$, і, таким чином, x є елементом перетину цих множин: $(A \cup B) \cap (A \cup C)$.

Якщо $x \in B \cap C$, то $x \in B$ і $x \in C$. Отже, $x \in A \cup B$ і $x \in A \cup C$. У цьому випадку x також є елементом перетину $(A \cup B) \cap (A \cup C)$.

2. Розглянемо вираз

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

Нехай $x \in (A \cup B) \cap (A \cup C)$. Тоді $x \in A \cup B$ і $x \in A \cup C$. Отже, або $x \in A$, або $x \in B$ і $x \in C$. З цього випливає, що $x \in A \cup (B \cap C)$.

Тобто x належить як до першої частини рівності 3, так і до другої, що й доводить її. Доведення решти рівностей провести самостійно за аналогією.

У загальному вигляді рівність 3, а також 3' можна подати в такий спосіб:

$$A \cup (B_1 \cap B_2 \cap \ldots \cap B_n) = (A \cup B_1) \cap (A \cup B_2) \cap \ldots \cap (A \cup B_n);$$
$$A \cap (B_1 \cup B_2 \cup \ldots \cup B_n) = (A \cap B_1) \cup (A \cap B_2) \cup \ldots \cup (A \cap B_n).$$

Рівності 1 и 1' називаються ***асоціативними*** законами для об'єднання і перетину, а рівності 2 і 2' – ***комутативними*** законами для цих операцій. Рівності 3 і 3' – це ***дистрибутивні*** закони для цих операцій.

Для довільних підмножин A і B універсальної множини U, крім вищезазначених рівностей, справедливі також рівності:

1. Якщо $A \cup B = A$, то $B = \varnothing$.
2. $\overline{\varnothing} = U$.
3. $A \cup A = A$.
4. $A \cup U = U$.
5. $A \cup (A \cap B) = A$.
6. $\overline{A \cup B} = \overline{A} \cap \overline{B}$.
7. Якщо $A \cup B = U$ і $A \cap B = \varnothing$, то $B = \overline{A}$.

1'. Якщо $A \cap B = A$, то $B = U$.
2'. $\overline{U} = \varnothing$.
3'. $A \cap A = A$.
4'. $A \cap \varnothing = \varnothing$.
5'. $A \cap (A \cup B) = A$.
6'. $\overline{A \cap B} = \overline{A} \cup \overline{B}$.
7'. Якщо $A \cap B = \varnothing$ і $A \cup B = U$, то $B = \overline{A}$.

Доведення цих рівностей провести самостійно.

Деякі з рівностей відомі під спеціальними назвами. Так 3 і 3' – це закони ***ідемпотентності***; 5 і 5' – закони ***поглинання***; 6 і 6' – закони ***де Моргана***.

Наведені вище в цьому розділі рівності дозволяють спрощувати різні більш складні вирази алгебри множин.

Приклад 1. $\overline{A \cap \overline{B}} \cup B = \overline{A} \cup B \cup B = \overline{A} \cup B.$

Приклад 2.

$$(A\cap B\cap C)\cup(\overline{A}\cap B\cap C)\cup\overline{B}\cup\overline{C} = [(A\cup\overline{A})\cap B\cap C]\cup\overline{B}\cup\overline{C} =$$
$$= (U\cap B\cap C)\cup\overline{B\cap C} = (B\cap C)\cup(\overline{B\cap C}) = U.$$

Приклад 3.

$$(A\cap B\cap C\cap\overline{X})\cup(\overline{A}\cap C)\cup(\overline{B}\cap C)\cup(C\cap X) =$$
$$= (A\cap B\cap C\cap\overline{X})\cup[(\overline{A}\cup\overline{B}\cup X)\cap C] =$$
$$= [A\cap B\cap\overline{X}\cup\overline{A\cap B\cap\overline{X}}]\cap C = U\cap C = C.$$

Приклад 4. Довести, що $A\cup A = A$:

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

2. Принцип двоїстості для алгебри множин

Рівність алгебри множин, отримана з іншої рівності через заміну всіх входжень $\cup$ на $\cap$, $\cap$ на $\cup$, $\varnothing$ на U і U на $\varnothing$, називається ***двоїстою*** (дуальною) відносно вихідної рівності.

Для будь-якого істинного твердження, що формулюється з допомогою знаків операцій $\cup$ та $\cap$, $\varnothing$ та U, двоїсте відносно нього речення також є *істинним*. Це речення виражає принцип *двоїстості* алгебри множин.

З цього принципу випливає, що якщо є деяке твердження з знаками операцій $\cup$ та $\cap$, $\varnothing$ та U, то відповідне йому твердження зі штрихом на підставі двоїстості випливає з цього ж самого твердження. Це означає, що нема потреби доводити, наприклад, рівності $1'-5'$, якщо доведені рівності 1 – 5.

3. Узагальнення операцій над множинами

1. *Об'єднання n множин*

$$A_1 \cup A_2 \cup \ldots \cup A_n = \bigcup_{j=1}^{n} A_j \text{, також } \bigcup_{j=1}^{\infty} A_j \text{ при } n=\infty.$$

2. *Перетин множин*

$$A_1 \cap A_2 \cap \ldots \cap A_n = \bigcap_{j=1}^{n} A_j \text{, також } \bigcap_{j=1}^{\infty} A_j \text{ при } n=\infty.$$

3. *Формула де Моргана*

$$\overline{\bigcup_{j=1}^{n} A_j} = \bigcap_{j=1}^{n} \overline{A_j}\ ;\quad \overline{\bigcap_{j=1}^{n} A_j} = \bigcup_{j=1}^{n} \overline{A_j}.$$

Вправи

1. Довести рівність

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

2. Довести, що для довільних множин A, B, C, D і X

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

б) $(A \cap X) \cup (B \cap \overline{X}) \cup (C \cap X) \cup (D \cap \overline{X}) =$
$= [(A \cup C) \cap X] \cup [(B \cup D) \cap \overline{X}]$;

в) $[(A \cap X) \cup (B \cap \overline{X})] \cap [(C \cap X) \cup (D \cap \overline{X})] =$
$= [(A \cap C) \cap X] \cup [(B \cap D) \cap \overline{X}]$.

3. Показати справедливість рівностей

а) $\overline{\overline{A} \cap \overline{\overline{B}}} \cup B = A \cup B$;

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

КОНТРОЛЬНІ ЗАВДАННЯ І ПИТАННЯ ДО ЧАСТИНИ І

Питання для самоконтролю

1. Що таке множина? З яких елементів і підмножин вона складається?
2. Дайте визначення скінченній і нескінченній множинам.
3. Як задаються множини?
4. Що таке рівність множин?
5. Охарактеризуйте поняття та властивості порожньої множини.
6. Що таке булеан множини і які його властивості?
7. Охарактеризуйте об'єднання множин і його властивості.
8. Що таке перетин множин і які його властивості?
9. Що таке різниця множин і які її властивості?
10. Дайте визначення симетричній різниці.
11. Охарактеризуйте універсальну множину.
12. Дайте визначення абсолютного доповнення множин.
13. Що таке розбиття множин?
14. Дайте визначення діаграми Ейлера.
15. Охарактеризуйте діаграми Ейлера для об'єднання множин.
16. Охарактеризуйте діаграми Ейлера для перетину множин.
17. Охарактеризуйте діаграми Ейлера для симетричної різниці множин.
18. У чому полягають асоціативні закони для операцій об'єднання й перетину?
19. У чому полягають комутативні закони для операцій об'єднання й перетину?
20. У чому полягають дистрибутивні закони для операцій об'єднання й перетину?
21. У чому полягають закони ідемпотентності?
22. У чому полягають закони поглинання?
23. У чому полягають закони де Моргана?
24. Що таке алгебра множин? Які основні її тотожності?
25. У чому полягає принцип двоїстості для алгебри множин?

Контрольні завдання[1]

1. Провести перетворення рівностей для множин відповідно до заданого варіанту в табл. 1.1.

2. Виразити за допомогою символів (використовуючи знаки операцій перетину, об'єднання і доповнення) множини, заштриховані на діаграмах Ейлера, відповідно до табл. 1.2. Номери вашого набору діаграм Ейлера задані в табл. 1.3.

3. Накресліть діаграму Ейлера для чотирьох множин *A*, *B*, *C*, *D*, які утворюють множину *F*, і позначте штриховкою її ділянку (див. табл. 1.4).

Таблиця 1.1. Рівності множин

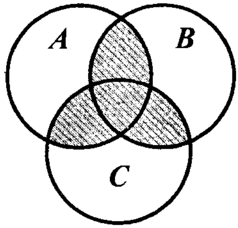
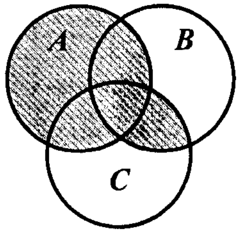
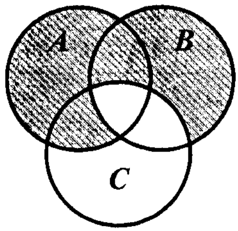
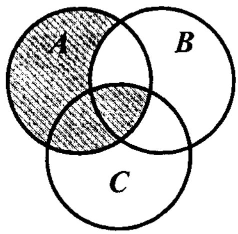
Варіант	Рівності
1	$F=\overline{A\cup\bar{B}}\cup\overline{C\cup\bar{D}}\cup\overline{\bar{A}\cap B\cap\bar{C}\cap D}$
2	$F=(A\cap\bar{B}\cap C)\cup(A\cap\bar{B}\cap C\cap D)\cup(A\cap C\cap\bar{A})$
3	$F=(A\cap B\cap C\cap\bar{D})\cup(\bar{A}\cap C)\cup(\bar{B}\cap C)\cup(C\cap D)$
4	$F=(\overline{A\cup B})\cap(C\cup\bar{D})\cap\overline{A\cup B\cup\bar{C}}$
5	$F=(\overline{A\cup\bar{B}}\cap C)\cup((A\cup\bar{B})\cap C)\cup D$
6	$F=(((B\cup\bar{C})\cap D)\cap((\bar{B}\cap C)\cup\bar{D}))\cup(A\cap C)$
7	$F=(A\cap B\cap C)\cup(\overline{\bar{A}\cup\bar{B}\cup C})\cup(\overline{C\cup\bar{D}})$
8	$F=(\overline{A\cup B})\cap(C\cup\bar{D})\cap\overline{A\cup B\cup\bar{C}}$
9	$F=((A\cup\bar{B}\cup\bar{C})\cap(B\cup C\cup\bar{D}))\cap((\bar{A}\cap B\cap C)\cup$ $\cup(\bar{B}\cap\bar{C}\cap D))$
10	$F=(\bar{A}\cap B\cap C\cap D)\cup(A\cap C)\cup(\bar{B}\cap C)\cup(C\cap\bar{D})$
11	$F=(A\cup B)\cap(A\cup\bar{B}\cup C)\cap(B\cup D\cup\bar{C})\cap(\bar{A}\cap\bar{B})\cap$ $\cap(\bar{B}\cap\bar{D}\cap C)$
12	$F=(\overline{A\cap\bar{B}})\cap(A\cap\bar{C}\cap D)\cap(\bar{A}\cup C\cup\bar{D})$
13	$F=(B\cup C\cup\bar{D})\cap(\bar{B}\cup C\cup\bar{D})\cap(B\cup C\cup D)\cap(A\cup C)$

[1] Контрольні завдання до частини I склала старший викладач кафедри електроніки і комп'ютерної техніки Сумського державного університету Протасова Т.О.

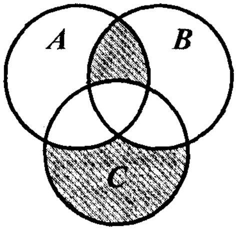
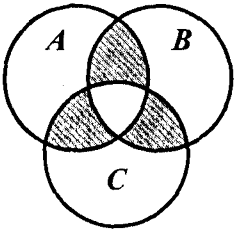
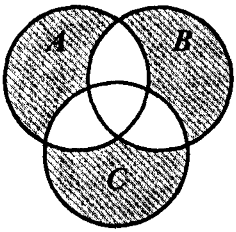
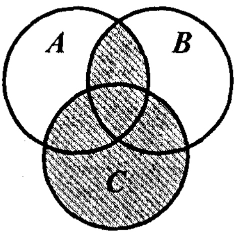
Продовження таблиці 1.1

Варіант	Рівності
14	$F=(D\cap C\cap\bar{A})\cup(\bar{D}\cap C\cap\bar{A})\cup(\bar{D}\cap\bar{C}\cap\bar{A})\cup(\overline{B\cup A})$
15	$F=(\bar{A}\cup C)\cap(\bar{A}\cup C\cup D)\cap(C\cup\bar{D})\cap(\bar{C}\cup\bar{D})\cap(\overline{\bar{B}\cup\bar{D}})$
16	$F=\overline{A\cap C\cap\bar{D}}\cup\overline{A\cap C}\cup D\cup A\cup B$
17	$F=C\cap\bar{D}\cap B\cap(\overline{\bar{A}\cup B\cup\bar{C}})\cap(C\cap D)\cap(\bar{A}\cup B)$
18	$F=\bar{C}\cap[(\overline{A\cup\bar{B}\cap C})\cup((A\cup\bar{B})\cap C)\cup D]$
19	$F=(D\cup C\cup\bar{A})\cap(\bar{D}\cup C\cup\bar{A})\cap(\bar{D}\cup\bar{C}\cup\bar{A})\cap(\overline{B\cap A})$
20	$F=(D\cap C\cap B\cap\bar{A})\cup(\bar{D}\cap B)\cup(\bar{C}\cap B)\cup(B\cap A)$
21	$F=(\overline{\bar{A}\cap\bar{B}})\cup(\bar{C}\cap D)\cup\overline{\bar{A}\cap\bar{B}\cap C}$
22	$F=[(\overline{\bar{A}\cap B\cap\bar{C}})\cup((\bar{A}\cap B)\cup C)]\cap\bar{D}$
23	$F=(\bar{B}\cap C)\cup\bar{D}\cup((B\cup\bar{C})\cap D)\cup(A\cup\bar{C})$
24	$F=(\bar{D}\cap C\cap B\cap A)\cup(D\cap B)\cup(\bar{C}\cap B)\cup(B\cap\bar{A})$
25	$F=(A\cup B\cup C)\cap(\overline{A\cap\bar{B}\cap\bar{C}})\cap\overline{\bar{C}\cap D}$
26	$F=(D\cup\bar{C}\cup A)\cap(D\cup\bar{C}\cup\bar{A})\cap(\bar{D}\cup\bar{C}\cup\bar{A})\cap$ $\cap(\bar{D}\cup\bar{C}\cup A)\cap\bar{B}$
27	$F=(\bar{B}\cup C\cup A)\cap(\bar{B}\cup\bar{C}\cup\bar{A})\cap(\bar{B}\cup\bar{C}\cup A)\cap$ $\cap(\bar{B}\cup C\cup\bar{A})\cap D$
28	$F=(A\cup C\cup\bar{D})\cap(\bar{A}\cup\bar{C}\cup\bar{D})\cap(\bar{A}\cup C\cup\bar{D})\cap$ $\cap(A\cup\bar{C}\cup\bar{D})\cap B$
29	$F=\overline{(A\cup\bar{B})}\cap\overline{C\cup\bar{D}\cup\bar{A}}\cap\overline{\bar{A}\cap B}$
30	$F=(A\cap B\cap C)\cup(A\cap B\cap\bar{C})\cup(A\cap\bar{B}\cap C)\cup$ $\cup(A\cap\bar{B}\cap\bar{C})\cup\bar{D}$

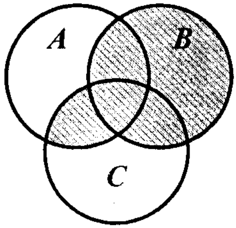
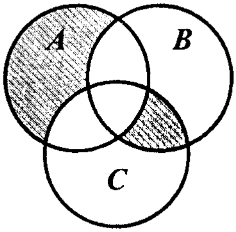
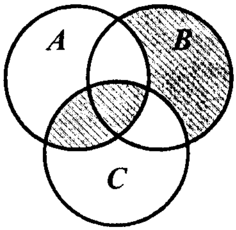
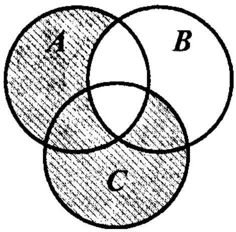
Таблиця 1.2. Діаграми Ейлера

Варіант	Діаграма Ейлера
1	
2	
3	
4	

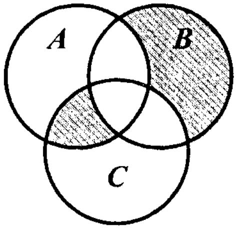
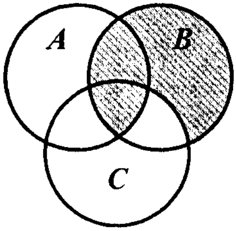
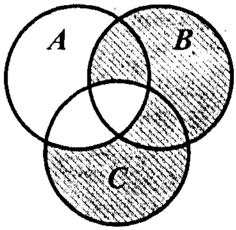
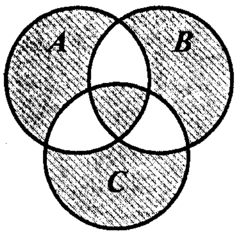
Продовження таблиці 1.2

Варіант	Діаграма Ейлера
5	
6	
7	
8	

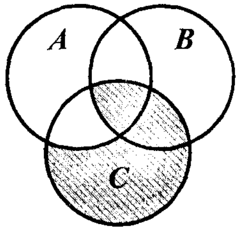
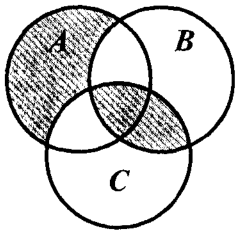
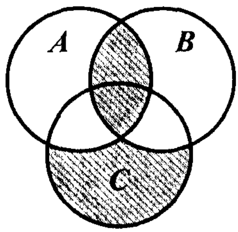
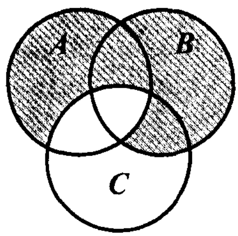
Продовження таблиці 1.2

Варіант	Діаграма Ейлера
9	
10	
11	
12	

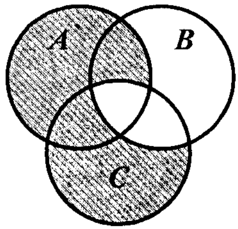
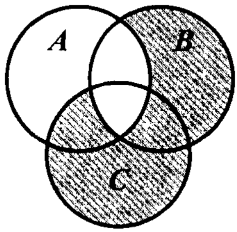
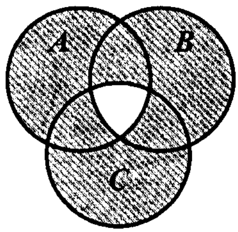
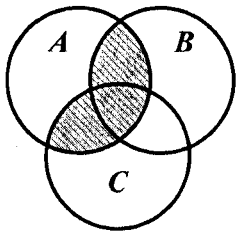
Продовження таблиці 1.2

Варіант	Діаграма Ейлера
13	
14	
15	
16	

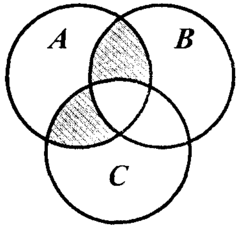
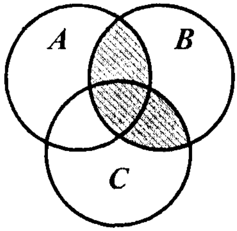
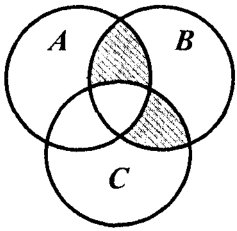
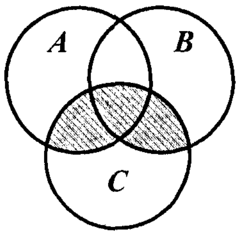
Продовження таблиці 1.2

Варіант	Діаграма Ейлера
17	
18	
19	
20	

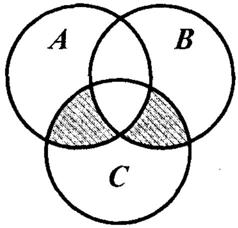
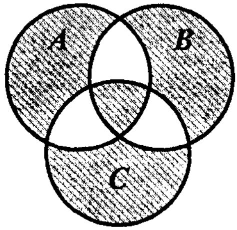
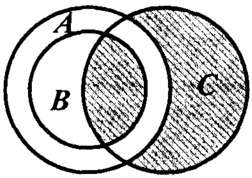
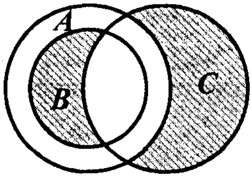
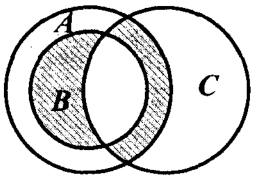
Продовження таблиці 1.2

Варіант	Діаграма Ейлера
21	
22	
23	
24	

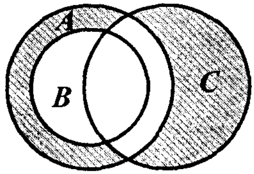
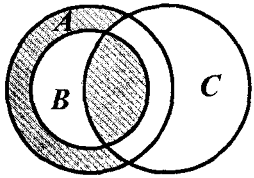
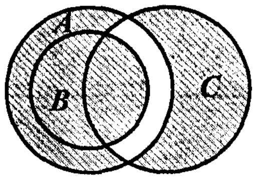
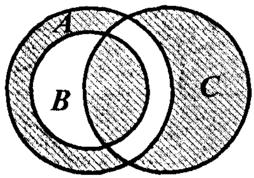
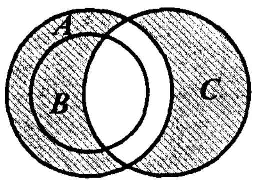
Продовження таблиці 1.2

Варіант	Діаграма Ейлера
25	
26	
27	
28	

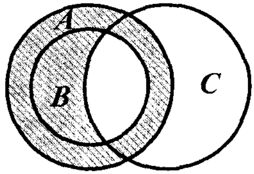
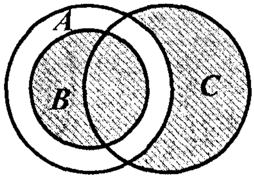
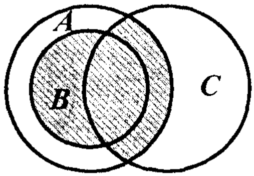
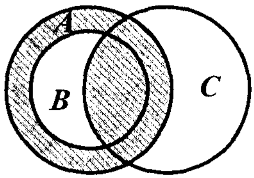
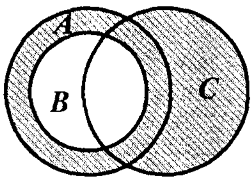
Продовження таблиці 1.2

Варіант	Діаграма Ейлера
29	
30	
31	
32	
33	

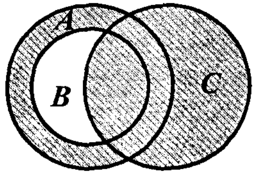
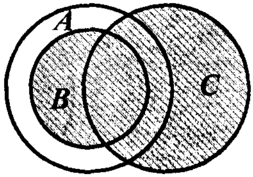
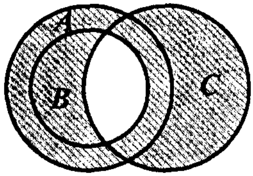
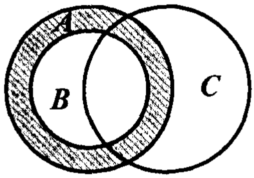
Продовження таблиці 1.2

Варіант	Діаграма Ейлера
34	
35	
36	
37	
38	

Продовження таблиці 1.2

Варіант	Діаграма Ейлера
39	
40	
41	
42	
43	

Продовження таблиці 1.2

Варіант	Діаграма Ейлера
44	
45	
46	
47	

Таблиця 1.3. Набори номерів діаграм Ейлера

Варіант	Номери діаграм Ейлера
1	1, 12, 23, 34
2	2, 13, 24, 35
3	3, 14, 25, 36
4	4, 15, 26, 37
5	5, 16, 27, 38
6	6, 17, 28, 39
7	7, 18, 29, 39
8	8, 19, 30, 41
9	9, 20, 31, 42
10	10, 21, 32, 43
11	11, 22, 33, 44
12	12, 7, 22, 45
13	13, 8, 23, 46
14	14, 9, 24, 47
15	15, 1, 25, 30
16	16, 2, 26, 31
17	3, 17, 27, 32
18	4, 18, 28, 33
19	5, 19, 29, 34
20	6, 20, 30, 35
21	7, 21, 31, 36
22	8, 22, 32, 37
23	9, 12, 33, 38
24	10, 13, 34, 39
25	1, 10, 20, 40
26	2, 11, 21, 41
27	3, 12, 22, 42
28	4, 13, 46, 43
29	5, 14, 47, 44
30	6, 15, 23, 45

Частина II

ЕЛЕМЕНТИ МАТЕМАТИЧНОЇ ЛОГІКИ

РОЗДІЛ 1. ЛОГІЧНІ ОПЕРАЦІЇ І ФУНКЦІЇ

Лекція 6
ЧИСЛЕННЯ ВИСЛОВЛЮВАНЬ

1. Висловлювання

У численні висловлювань об'єктом дослідження є *висловлювання* (висловлення).

Означення 1. Будь-яке твердження, яке може бути *істинним* або *хибним*, називається ***висловлюванням.***

Істинним висловлюванням приписується значення 1, хибним – 0. З одного або кількох висловлювань можна скласти нові висловлювання. Їх ще називають *складеними* висловлюваннями. При цьому окремі висловлювання позначатимемо великими літерами латинського алфавіту A, B, C, ...

Приклад 1. Висловлюваннями будуть: A = "Вісім – парне число"; B = "Вісім ділиться на два"; C = "Вісім ділиться на три"; $Д$ = "Київ – столиця України". З них висловлювання A, B і $Д$ – істинні, а висловлення C – хибне. Це записується як A = 1, B = 1, C = 0, $Д$ = 1.

Приклад 2. Складеним буде висловлення: "Якщо вісім – парне число, то вісім ділиться на два". Це висловлення істинне, тобто дорівнює 1.

Для об`єднання простих висловлювань у складені застосовують логічні *операції*.

2. Операції над висловлюваннями

1. Логічна операція "*Константа нуль*" створює висловлювання, яке завжди є *хибним*. Позначається "$F = 0$".

2. Логічна операція "*Константа одиниця*" створює висловлювання, яке завжди є *істинним*. Позначається "$F = 1$".

3. Логічна операція "*Змінна A*" створює висловлювання $F = A$, яке дорівнює 0 тоді і лише тоді, коли A дорівнює 0, і 1, коли A дорівнює 1. Читається: "Висловлювання залежить лише від A".

4. Логічна операція "*НІ*" створює висловлювання $F = \overline{A}$, яке є *істинним* тоді і лише тоді, коли A хибне, і *хибне*, коли A є істинним. Читається: "Не A" або "Невірно, що A".

5. Логічна операція "*І*" (кон'юнкція, добуток, логічне множення) створює складене висловлювання $F = A \wedge B$ ($A \cdot B$, AB, $A \,\&\, B$), яке є *істинним* тоді і лише тоді, коли обидва висловлювання A і B істинні, і *хибним*, коли хоча б одне з цих висловлювань хибне. Читається: "A і B".

6. Логічна операція "*АБО*" (диз`юнкція, сума, логічне додавання) створює складене висловлювання $F = A \vee B$ ($A + B$, A або B), яке є *хибним* тоді і лише тоді, коли обидва висловлювання A і B хибні, і *істинним*, коли хоча б одне висловлювання A чи B істинне. Читається: "A або B".

7. Логічна операція "*Якщо – то*" (імплікація) створює складене висловлювання $F = A \rightarrow B$, яке є *хибним* тоді і лише тоді, коли A істинне, а B – хибне. Для інших випадків значень, які приймають висловлювання A і B, воно *істинне*. Читається: "Якщо A, то B".

8. Логічна операція "*Заборона з B*" (заперечення імплікації $A \rightarrow B$) створює складене висловлення $F = A \Delta B = \overline{A \rightarrow B}$, яке є *істинним* тоді і лише тоді, коли A істинне, а B хибне. Для інших випадків значень, які приймають висловлювання A і B, воно *хибне*. Читається: "Невірно, якщо A, то B".

9. Логічна операція "*Заборона з A*" (заперечення імплікації $B \rightarrow A$) створює складене висловлювання $F = B \Delta A = \overline{B \rightarrow A}$, яке є *істинним* тоді і лише тоді, коли B істинне, а A хибне. Для інших випадків значень, які приймають висловлювання A і B, воно *хибне*. Читається: "Невірно, якщо B, то A".

10. Логічна операція "*Рівнозначність*" (еквівалентність) створює складене висловлювання $F = A \sim B$ ($A \equiv B$), яке є *істинним* тоді і лише тоді, коли обидва висловлювання A і B істинні або хибні одночасно. Для інших випадків значень, які приймають висловлювання A і B, воно *хибне*. Читається: "A рівнозначне B".

11. Логічна операція "*Нерівнозначність*" (сума за модулем два) створює складене висловлювання $F = A \oplus B$ ($A \neq B$), яке є *істинним* тоді і лише тоді, коли одне висловлювання є істинним, а друге – хибним. Для інших випадків значень, які приймають висловлювання A і B, воно *хибне*. Читається: "A нерівнозначне до B", або "Сума за модулем 2".

12. Логічна операція "*Стрілка Пірса*" (функція Вебба, операція Пірса) створює складене висловлювання $F = A \downarrow B = \overline{A \vee B}$, яке є

істинним тоді і лише тоді, коли обидва висловлювання A і B *хибні* одночасно. Для інших випадків значень, які приймають висловлювання A і B, воно *хибне*. Читається: "Ні A, ні B".

13. Логічна операція "*Операція Шефера*" (Штрих Шефера) створює складене висловлювання $F = A|B = \overline{A \wedge B}$, яке є *хибним* тоді і лише тоді, коли обидва висловлювання A і B є *істинні* одночасно. Для інших випадків значень, які приймають висловлювання A і B, воно *істинне*. Читається: "Невірно, що A і B".

Математика – це жінка, а логіка – її одяг.

М.Клайн

Лекція 7
ЛОГІЧНІ ФУНКЦІЇ

1. Означення логічної функції

Означення 1. Функція F від n аргументів (змінних) $x_1, x_2, ..., x_n$, яка так само, як і її змінні, може приймати лише два значення – 0 і 1, називається ***логічною*** (двійковою, булевою).

Розглянуті вище логічні операції над висловлюваннями можуть бути використані для побудови логічних функцій. Ці операції разом з побудованими з їх допомогою логічними функціями створюють *алгебру логіки.* У ній висловлювання в логічних функціях замінюються логічними змінними, і щодо них виконуються необхідні для розв'язання тієї чи іншої задачі логічні операції. Їх найбільш уживаний склад був розглянутий у попередній лекції.

2. Набори значень змінних логічної функції

Означення 2. Сукупність $a_1, a_2, ..., a_n$ значень n змінних $x_1, x_2, ..., x_n$ називається ***набором*** і позначається $a_1, a_2, ..., a_n$, де $a_i \in \{0,1\}$, $i = 1, 2, ..., n$.

Розмістимо, наприклад, набори для трьох аргументів у вигляді двійкових чисел у спеціальній таблиці – таблиці наборів. Очевидно, що їх число дорівнює 8. Зліва в цій таблиці у вигляді номерів зазначимо десяткові еквіваленти двійкових наборів (див. табл. 2.1). Це дозволяє досить легко перейти від десяткового запису номеру набору до його двійкового вигляду.

Таблиця 2.1. Набори значень двійкових змінних

Номер набору	x_1	x_2	x_3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Через те, що кількість змінних скінченна, кількість можливих наборів обмежена, і, відповідно, ці набори й значення логічної функції можуть бути задані в спеціальній таблиці в порядку зростання від 0 до $2^n - 1$.

Теорема 1. ***Число наборів для аргументів*** $x_1, x_2, ..., x_n$ ***логічної функції*** $N = 2^n$.

Доведення. Оскільки за означенням логічної функції кожна змінна x_i, $i = 1, 2, ..., n$, може приймати значення 0 і 1, кількість наборів, що містять j одиниць і відповідно n-j нулів, дорівнює числу сполучень j з $n - C_n^j$. Оскільки j може приймати значення 0, 1, ..., n, то число всіх можливих наборів дорівнюватиме $N = \sum_{j=0}^{n} C_n^j$, яке, як відомо, становить 2^n. Теорему доведено.

3. Кількість логічних функцій

Кожна логічна функція F приймає у кожному зі своїх наборів значення, яке дорівнюють 0 або 1. Кількість наборів, як було доведено в теоремі 1, відповідає числу $N = 2^n$. Виходячи з цього, має місце така теорема.

Теорема 2. ***Кількість різних логічних функцій від n аргументів***

$$M = 2^N.$$

Доведення. Дійсно, оскільки на кожному наборі $i = 0, 1, ..., N$ логічна функція F_i дорівнює 0 або 1, то число функцій, що приймають 1 на γ наборах і 0 на решті N-γ наборах, дорівнює числу сполучень γ з числа можливих наборів $N - C_N^{\gamma}$.

Оскільки $\gamma = 0, 1, ..., N$, то число всіх можливих функцій

$$M = \sum_{\gamma=0}^{N} C_N^{\gamma} = 2^N.$$

Теорему доведено.

4. Елементарні логічні функції

Розмістимо логічні функції разом з таблицею наборів у порядку зростання їх десяткових індексів, починаючи з нуля, і зазначимо на кожному наборі значень змінних у відповідній клітинці таблиці значення функції, яке відповідає нулю або одиниці (див. табл. 2.2, 2.3). Створені таким чином таблиці називаються таблицями *істинності*. З їх допомогою для однієї і двох змінних побудуємо елементарні логічні функції, на основі яких можна створити інші функції, якими б складними вони не були.

Існує чотири різні логічні функції одного аргументу A (табл. 2.2). Їх назва обирається відповідно до логічної операції, яка їх створює.

Таблиця 2.2. Таблиця істинності логічних функцій однієї змінної

Номер набору	A	F_0	F_1	F_2	F_3
0	0	0	0	1	1
1	1	0	1	0	1

Функція F_0 тотожно дорівнює 0. Її називають "*Константою нуль*".

Функція F_1 повторює значення змінної і тому тотожно дорівнює цій змінній. Її називають "*Змінна A*".

Функція F_2 приймає значення, протилежні до значень аргументу. Тобто це є логічна функція „*НІ*". Її ще називають "*Інверсією A*", або "*Запереченням A*". Позначається $\overline{A}$.

Функція F_3 тотожно дорівнює 1. Її називають "*Константою одиниці*".

Згідно з теоремою 2 існує $M = 2^N = 2^{2^n} = 16$ різних логічних функцій двох аргументів A і B, кожна з яких визначена у $N = 2^2$ наборах змінних (див. табл. 2.3).

Таблиця 2.3. Таблиця істинності логічних функцій двох змінних

Номер набору	Змінна		Функція F															
	A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
2	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
3	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Ці функції мають однакові назви з логічними операціями, які їх створюють:

$F_0 = 0$ – константа нуль;

$F_1 = A \wedge B$ – логічне множення (кон`юнкція, добуток);

$F_2 = A \Delta B$ – заборона за B;

$F_3 = A$ – змінна A;

$F_4 = B \Delta A$ – заборона за A;

$F_5 = B$ – змінна B;

$F_6 = A \oplus B$ – сума за модулем 2 (логічна нерівнозначність);

$F_7 = A \vee B$ – логічне додавання (диз`юнкція);

$F_8 = A \downarrow B$ – операція Пірса (стрілка Пірса);

$F_9 = A \sim B$ – логічна рівнозначність;

$F_{10} = \overline{B}$ – інверсія B;

$F_{11} = B \to A$ – імплікація від B до A;

$F_{12} = \overline{A}$ – інверсія A;

$F_{13} = A \to B$ – імплікація від A до B;

$F_{14} = A|B$ – операція Шефера (штрих Шефера);

$F_{15} = 1$ – константа одиниця.

5. Закони алгебри логіки

В алгебрі логіки існують логічні *закони*, логічні *суперечності* і *твердження*, що логічно виконуються.

Означення 2. Висловлення, що є *істинним* для всіх можливих комбінацій значень простих висловлювань, з яких воно складається, називається логічним ***законом***.

Означення 3. Висловлення, що є *хибним* для всіх можливих комбінацій значень простих висловлювань, з яких воно складається, називається логічною ***суперечністю***.

Означення 4. Висловлення, що є *істинним* для одних значень простих висловлювань, з яких воно складається, і *хибним* для решти, називається ***твердженням***, що логічно виконується.

Наведемо основні закони алгебри логіки (див. табл. 2.4).

Таблиця 2.4. Основні закони алгебри логіки

Назва закону	Логічний запис
1. Тотожності	$A = A$
2. Суперечності	$\overline{A\overline{A}} = 1$
3. Виключеного третього	$A + \overline{A} = 1$
4. Ідемпотентності	$AA = A; A + A = A$
5. Комутативний	$AB = BA,\ A + B = B + A$
6. Асоціативний	$(AB)C = A(BC)$; $(A + B) + C = A + (B + C)$
7. Дистрибутивний	$A\,(B + C) = AB + AC$; $A + BC = (A+B)(A+C)$
8. Поглинання	$A\,(B + A) = A;\ A + AB = A$
9. Подвійності (теорема де Моргана)	$\overline{AB} = \overline{A} + \overline{B};\ \overline{A + B} = \overline{A}\,\overline{B}$
10. Подвійного заперечення	$A = \overline{\overline{A}}$
11. Властивість одиниці	$A \cdot 1 = A;\ A + 1 = 1$

Шукай простоту і не довіряй їй.

А. И. Уайтхед

Лекція 8
ПЕРЕТВОРЕННЯ ЛОГІЧНИХ ФУНКЦІЙ

1. Базисні функції

Розглянуті вище 16 логічних функцій для двох змінних, як уже зазначалося раніше, мають назву ***елементарних***. Це функції, на основі яких ґрунтується алгебра логіки та її достатньо поширене застосування в науці та техніці. Серед цих функцій відокремимо ***базисні***. За допомогою цих функцій можна одержати будь-які інші логічні функції. У той же самий час базисні функції не можуть бути одержані з більш простих. До них належать:

1. Константа нуль.
2. Константа одиниця.
3. Змінна.
4. Інверсія.
5. Диз`юнкція.
6. Кон`юнкція.

2. Інверсні функції

Означення 1. Логічна функція називається ***інверсною*** відносно іншої, якщо вона може бути отримана з останньої способом інверсії всіх її значень.

Кожна із 16 функцій F_i, $i = 0, 1, ..., 15$ має також інверсну функцію. Так, наприклад, функція F_0 має інверсну функцію F_{15}, а F_3 – інверсну функцію F_{12} (див. табл. 2.3).

3. Перетворення елементарних функцій

Запровадимо ряд важливих формул для елементарних функцій. Доведення правильності формули легко отримати за допомогою безпосередньої перевірки в таблицях істинності за збіжністю значень, що утворюють праву і ліву сторони співвідношень, які доводяться (див. табл. 2.5 – 2.8).

Таблиця 2.5. $A \wedge B = \overline{\overline{A} \vee \overline{B}}$ – правило де Моргана

A	B	$A \wedge B$	$\overline{A}$	$\overline{B}$	$\overline{A} \vee \overline{B}$	$\overline{\overline{A} \vee \overline{B}}$
0	0	0	1	1	1	0
0	1	0	1	0	1	0
1	0	0	0	1	1	0
1	1	1	0	0	0	1

Таблиця 2.6. $A \vee B = \overline{\overline{A} \wedge \overline{B}}$ – правило де Моргана

A	B	$A \vee B$	$\overline{A}$	$\overline{B}$	$\overline{A} \wedge \overline{B}$	$\overline{\overline{A} \wedge \overline{B}}$
0	0	0	1	1	1	0
0	1	1	1	0	0	1
1	0	1	0	1	0	1
1	1	1	0	0	0	1

Таблиця 2.7. $A \to B = \overline{A} \vee B$

A	B	$A \to B$	$\overline{A}$	B	$\overline{A} \vee B$
0	0	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	1	1	0	1	1

Таблиця 2.8. $A \sim B = (\overline{A} \vee B)(A \vee \overline{B})$

A	B	$A \sim B$	$\overline{A}$	B	$\overline{A} \vee B$	A	$\overline{B}$	$A \vee \overline{B}$	$(A \vee \overline{B}) \wedge$ $\wedge (\overline{A} \vee B)$
0	0	1	1	0	1	0	1	1	1
0	1	0	1	1	1	0	0	0	0
1	0	0	0	0	0	1	1	1	0
1	1	1	0	1	1	1	0	1	1

Аналогічно доводяться й співвідношення

1. $A = \overline{\overline{A}}$.
2. $A \sim B = AB \vee \overline{A}\overline{B}$.
3. $A \Delta B = \overline{\overline{A} \vee B} = A \wedge \overline{B}$.
4. $B \Delta A = \overline{\overline{B} \vee A} = B \wedge \overline{A}$.
5. $A \oplus B = \overline{A \sim B} = \overline{(\overline{A} \vee B)(A \vee \overline{B})} = (A \wedge \overline{B}) \vee (\overline{A} \wedge B)$.
6. $A \downarrow B = \overline{A \vee B} = \overline{A} \wedge \overline{B}$.
7. $A | B = \overline{A \wedge B} = \overline{A} \vee \overline{B}$.
8. $B \to A = \overline{B} \vee A$.

У разі, коли однією зі змінних є константа 1 або 0, справедливі співвідношення, що наведені у табл. 2.9.

Таблиця 2.9. Співвідношення для змінних x та 1, x та 0

Співвідношення	Співвідношення
$x \vee 1 = 1$	$x \vee 0 = x$
$x \wedge 1 = x$	$x \wedge 0 = 0$
$x \sim 1 = x$	$x \sim 0 = \overline{x}$
$x \oplus 1 = \overline{x}$	$x \oplus 0 = x$
$x \to 1 = 1$	$x \to 0 = \overline{x}$
$1 \to x = x$	$0 \to x = 1$
$x \mid 1 = \overline{x}$	$x \mid 0 = 1$
$x \downarrow 1 = 0$	$x \downarrow 0 = \overline{x}$

У випадках, коли дві змінні $x_1 = x_2 = x$, а також коли $x_1 = x$ і $x_2 = \overline{x}$, маємо для цих змінних співвідношення, які наведені в табл. 2.10.

Таблиця 2.10. Співвідношення для змінних $x_1 = x_2 = x$ та $x_1 = x$ і $x_2 = \overline{x}$

Співвідношення	Співвідношення
$x \vee x = x$	$x \vee \overline{x} = 1$
$xx = x$	$x\overline{x} = 0$
$x \sim x = 1$	$x \sim \overline{x} = 0$
$x \oplus x = 0$	$x \oplus \overline{x} = 1$
$x \to x = 1$	$x \to \overline{x} = \overline{x}$ $\overline{x} \to x = x$
$x \mid x = \overline{x}$	$x \mid \overline{x} = 1$
$x \downarrow x = \overline{x}$	$x \downarrow \overline{x} = 0$

4. Порядок виконання логічних операцій

Наведені елементарні функції дозволяють будувати нові, у тому числі й більш складні, функції від довільного скінченного числа змінних шляхом підстановки до функцій інших функцій замість змінних. При цьому для запису функцій використовуються дужки.

Передбачається, що спочатку виконуються операції всередині дужок, після цього – операції під знаком заперечення, потім кон`юнкція, а надалі диз`юнкція і всі інші операції в порядку запису зліва направо, наприклад, імплікація.

5. Суперпозиція логічних функцій

Означення 2. Функція $F = F(F_1, F_2, \ldots, F_n)$, яка отримана із функцій $F_1, F_2, \ldots, F_n$, називається ***суперпозицією*** функцій $F_1, F_2, \ldots, F_n$

Приклад 1. Надані функції $A = (\overline{x}_1 \sim x_3)$; $B = x_1 \downarrow x_2$; $C = x_1 \oplus x_2$; $D = x_3$. Потрібно отримати суперпозицію функцій A, B, C, D – $F(A, B, C, D)$, а з неї функцію $F(x_1, x_2, x_3)$.

Розв`язання. Отримаємо суперпозицію $F(A, B, C, D) = ((A \vee B) \wedge C) \to D$. Далі замість функцій A, B, C, D запишемо їх значення:

$$F = ((A \vee B) \wedge C) \to D = F(x_1, x_2, x_3) =$$
$$= \{[(\overline{x}_1 \sim x_3) \vee (x_1 \downarrow x_2)] \wedge (x_1 \oplus x_2)\} \to x_3.$$

Використовуючи таблиці елементарних логічних функцій для $\sim$, $\vee$, $\downarrow$, $\oplus$, $\to$, отримаємо в табл. 2.11, 2.12 всі значення функції $F(x_1, x_2, x_3)$.

Таблиця 2.11. Функція $F(x_1, x_2, x_3)$

Номер набору	x_1	x_2	x_3	$\overline{x}_1$	$\overline{x}_1 \sim x_3$	$x_1 \downarrow x_2$	[ ]
0	0	0	0	1	0	1	1
1	0	0	1	1	1	1	1
2	0	1	0	1	0	0	0
3	0	1	1	1	1	0	1
4	1	0	0	0	1	0	1
5	1	0	1	0	0	0	0
6	1	1	0	0	1	0	1
7	1	1	1	0	0	0	0

Таблиця 2.12. Функція $F(x_1, x_2, x_3)$

Номер набору	x_1	x_2	x_3	$x_1 \oplus x_2$	{ }	x_3	$F(x_1, x_2, x_3)$
0	0	0	0	0	0	0	1
1	0	0	1	0	0	1	1
2	0	1	0	1	0	0	1
3	0	1	1	1	1	1	1
4	1	0	0	1	1	0	0
5	1	0	1	1	0	1	1
6	1	1	0	0	0	0	1
7	1	1	1	0	0	1	1

Таким чином, функція $F(x_1, x_2, x_3) = x_1\bar{x}_2\bar{x}_3$ дорівнює 0 на наборі 4, і 1 на решті наборів.

Можна бути першорядним математиком і не вміти рахувати. Можна рахувати досконало і не мати ані найменшого уявлення про математику.

Новаліс

Лекція 9
БУЛЕВА АЛГЕБРА

1. Основні формули булевої алгебри

Означення 1. Логічні операції над змінними x_1, x_2,..., x_n логічних функцій $F = F(x_1, x_2, ..., x_n)$, що містять у собі операції заперечення, диз`юнкції, кон`юнкції, а також операції константа 0 і константа 1, називаються ***булевими***.

Означення 2. Булеві операції над змінними логічних функцій і співвідношення, які випливають з них, називаються ***булевою алгеброю***.

Важливим є те, що з допомогою цієї алгебри можна реалізувати будь-яку логічну функцію. Ця алгебра отримала свою назву на честь англійського математика 19-го століття Джорджа Буля, що заклав її підвалини.

Змінні x_1, x_2, ..., x_n у булевій алгебрі вважаються довільними логічними функціями, тобто для них справджується принцип *суперпозиції*. Це означає, що будь-який вираз булевої алгебри являє собою логічну функцію і може бути позначений однією літерою, яка є змінною у іншому виразі.

Наведемо в табл. 2.13 основні формули булевої алгебри для диз`юнкції, кон`юнкції та інверсії.

Таблиця 2.13. Основні формули булевої алгебри

Формула для диз`юнкції	Формула для кон`юнкції	Формула для інверсії
1. $0 \vee 0 = 0$	$0 \wedge 0 = 0$	$\overline{0} = 1$
2. $1 \vee 0 = 1$	$1 \wedge 0 = 0$	$\overline{1} = 0$
3. $1 \vee 1 = 1$	$1 \wedge 1 = 1$	$\overline{\overline{x}} = x$
4. $0 \vee x = x$	$0 \wedge x = 0$	
5. $1 \vee x = 1$	$1 \wedge x = x$	
6. $x \vee x = x$	$x \wedge x = x$	
7. $x \vee \overline{x} = 1$	$x \wedge \overline{x} = 0$	
8. $x \vee y = y \vee x$	$x \wedge y = y \wedge x$	
9. $x \vee (y \vee x) = (x \vee y) \vee x$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$	

2. Спеціальні формули булевої алгебри

1. Операція поглинання $x \vee xy = x$ і $x\,(x \vee y) = x$.

2. Операція склеювання $xy \vee x\bar{y} = x$ і $(x \vee y)(x \vee \bar{y}) = x$.

3. Операція з дужками $xy \vee xz = x(y \vee z)$.

4. Формули де Моргана: $\overline{x \vee y} = \bar{x} \wedge \bar{y}$; $\overline{x \wedge y} = \bar{x} \vee \bar{y}$.

5. $x_1 F(x_1, x_2, \dots, x_n) = x_1 F(1, x_2, \dots, x_n)$.

6. $x_1 \vee F(x_1, x_2, \dots, x_n) = x_1 \vee F(0, x_2, \dots, x_n)$.

7. $\bar{x}_1 F(x_1, x_2, \dots, x_n) = \bar{x}_1 F(0, x_2, \dots, x_n)$.

8. $\bar{x}_1 \vee F(x_1, x_2, \dots, x_n) = \bar{x}_1 \vee F(1, x_2, \dots, x_n)$.

9. $\overline{F(x_1, x_2, \dots, x_n, V, \bullet)} = F(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n, \bullet, V)$..

Очевидно, що в формулах 5, 7 функція F представлена в ДНФ, а в формулах 6, 8 – в КНФ.

Доведення формул 1–9 провести самостійно.

Формула 9 означає, що якщо функція F складена таким чином, що змінні пов'язані тільки операціями диз`юнкції і кон`юнкції, то, замінивши всюди у виразі для F знак диз`юнкції на знак кон`юнкції і навпаки, а також узявши заперечення з кожної зі змінних, одержимо заперечення даної функції F.

Формули 8 і 9 з табл. 2.13 являють собою комутативний і асоціативний закони, а формула 3 зі спеціальних формул – розподільний (дистрибутивний) закон. Тому у виразах, які створюють операції диз`юнкції і кон`юнкції, можна розкривати дужки, виносити спільний множник, переставляти місцями члени за правилами звичайної алгебри, вважати формально диз`юнкцію операцією додавання, а кон`юнкцію – операцією множення.

Формули могутні, але сліпі.

Ф.Клейн

Лекція 10
БУЛЕВІ ЛОГІЧНІ ЕЛЕМЕНТИ

1. Інвертор

Елементи цифрової техніки, які застосовують елементарні логічні функції, називаються *логічними* елементами цифрових пристроїв. Серед цих елементів вирізняють універсальні набори, з допомогою яких можна реалізувати логічну функцію будь-якої складності. Такі набори називають ***функціонально-повними*** універсальними логічними базисами. До цих базисів належить булевий набір логічних елементів, який складається з елементів *НІ, І, АБО*, а також констант 0 і 1. Розглянемо цей базис.

У більшості випадків константа 1 реалізується з допомогою деякого значення фізичного параметру, а константа 0 – через відсутність цього значення, хоча можливе й зворотне кодування.

Елемент, що реалізує логічну функцію *НІ* з допомогою одиничних чи нульових значень напруги, струму чи інших фізичних параметрів, називають ***інвертором***. Логіка його роботи зображена в табл. 2.14, а функціональна схема на рис. 2.1.

На функціональних схемах інвертор зображується прямокутником, в якого вхід – зліва, вихід – справа (рис. 2.1 а, б). На *вихідній чи вхідний лінії місце її з'єднання з прямокутником* зображається кружком – символом інверсії. Стрілку на вхідних і *вихідних лініях ставити заборонено.*

Зображення інвертора може бути повернуте на 90° таким чином, що вхід буде зверху, а вихід знизу (рис. 2.1 в, г). Інші повороти заборонені.

Таблиця 2.14. Логіка функціонування інвертора

x	$f=\overline{x}$
0	1
1	0

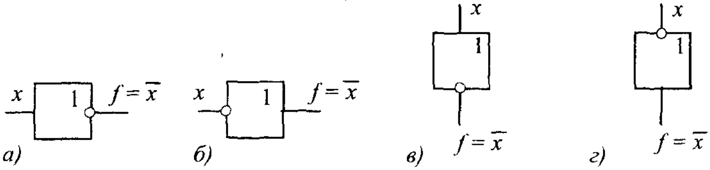


Рис. 2.1 а, б, в, г. Функціональна схема інвертора

2. Спеціальні формули булевої алгебри

1. Операція поглинання $x \vee xy = x$ і $x(x \vee y) = x$.

2. Операція склеювання $xy \vee x\overline{y} = x$ і $(x \vee y)(x \vee \overline{y}) = x$.

3. Операція з дужками $xy \vee xz = x(y \vee z)$.

4. Формули де Моргана: $\overline{x \vee y} = \overline{x} \wedge \overline{y}$; $\overline{x \wedge y} = \overline{x} \vee \overline{y}$.

5. $x_1 F(x_1, x_2, \ldots, x_n) = x_1 F(1, x_2, \ldots, x_n)$.

6. $x_1 \vee F(x_1, x_2, \ldots, x_n) = x_1 \vee F(0, x_2, \ldots, x_n)$.

7. $\overline{x}_1 F(x_1, x_2, \ldots, x_n) = \overline{x}_1 F(0, x_2, \ldots, x_n)$.

8. $\overline{x}_1 \vee F(x_1, x_2, \ldots, x_n) = \overline{x}_1 \vee F(1, x_2, \ldots, x_n)$.

9. $\overline{F(x_1, x_2, \ldots, x_n, V, \bullet)} = F(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n, \bullet, V)$..

Очевидно, що в формулах 5, 7 функція F представлена в ДНФ, а в формулах 6, 8 – в КНФ.

Доведення формул 1–9 провести самостійно.

Формула 9 означає, що якщо функція F складена таким чином, що змінні пов'язані тільки операціями диз`юнкції і кон`юнкції, то, замінивши всюди у виразі для F знак диз`юнкції на знак кон`юнкції і навпаки, а також узявши заперечення з кожної зі змінних, одержимо заперечення даної функції F.

Формули 8 і 9 з табл. 2.13 являють собою комутативний і асоціативний закони, а формула 3 зі спеціальних формул – розподільний (дистрибутивний) закон. Тому у виразах, які створюють операції диз`юнкції і кон`юнкції, можна розкривати дужки, виносити спільний множник, переставляти місцями члени за правилами звичайної алгебри, вважати формально диз`юнкцію операцією додавання, а кон`юнкцію – операцією множення.

Формули могутні, але сліпі.

Ф.Клейн

Лекція 10
БУЛЕВІ ЛОГІЧНІ ЕЛЕМЕНТИ

1. Інвертор

Елементи цифрової техніки, які застосовують елементарні логічні функції, називаються *логічними* елементами цифрових пристроїв. Серед цих елементів вирізняють універсальні набори, з допомогою яких можна реалізувати логічну функцію будь-якої складності. Такі набори називають *функціонально-повними* універсальними логічними базисами. До цих базисів належить булевий набір логічних елементів, який складається з елементів *НІ, І, АБО,* а також констант 0 і 1. Розглянемо цей базис.

У більшості випадків константа 1 реалізується з допомогою деякого значення фізичного параметру, а константа 0 – через відсутність цього значення, хоча можливе й зворотне кодування.

Елемент, що реалізує логічну функцію *НІ* з допомогою одиничних чи нульових значень напруги, струму чи інших фізичних параметрів, називають ***інвертором***. Логіка його роботи зображена в табл. 2.14, а функціональна схема на рис. 2.1.

На функціональних схемах інвертор зображується прямокутником, в якого вхід – зліва, вихід – справа (рис. 2.1 а, б). На вихідній чи вхідний лінії місце її з'єднання з прямокутником зображається кружком – символом інверсії. Стрілку на вхідних і вихідних лініях ставити заборонено.

Зображення інвертора може бути повернуте на 90° таким чином, що вхід буде зверху, а вихід знизу (рис. 2.1 в, г). Інші повороти заборонені.

Таблиця 2.14. Логіка функціонування інвертора

x	$f=\overline{x}$
0	1
1	0

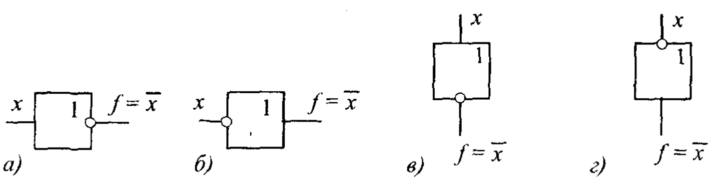


Рис. 2.1 а, б, в, г. Функціональна схема інвертора

У релейно-контактній логіці функцію *НІ* реалізує контакт, який перебуває в замкнутому стані, поки в обмотках реле відсутній струм x, і розімкнутому під час подачі струму x (рис. 2.2). Часова діаграма його роботи зображена на рис. 2.3.

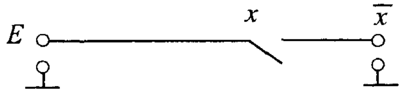


Рис. 2.2. Інвертор в релейному виконанні

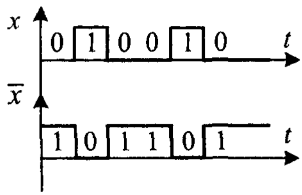


Рис. 2.3. Часова діаграма роботи інвертора

2. Кон'юнктор

Кон'юнктор (схема *І*, схема кон'юнкції, клапан) – двійковий логічний елемент, який реалізує операцію *І* (логічне множення). Зображується, як це показано на рис. 2.4. На його виході з'являється 1 тільки тоді, коли маємо сигнали 1 на всіх його входах (табл. 2.15).

Таблиця 2.15. Логіка функціонування кон'юнктора

x_1 x_2	$f = x_1 \cdot x_2$
0 0	0
0 1	0
1 0	0
1 1	1

Логічна функція, що реалізує кон'юнктор, має вигляд

$$f = x_1 \cdot x_2 = x_1 \wedge x_2 = x_1 \,\&\, x_2 = x_1 x_2 .$$

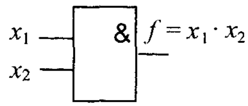


Рис. 2.4. Функціональна схема кон'юнктора

У релейному вигляді кон'юнктор зображений на рис. 2.5, а часова діаграма його роботи наведена на рис. 2.6.

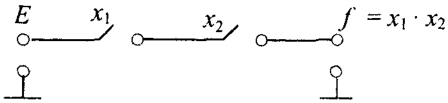


Рис. 2.5. Кон'юнктор у релейному виконанні

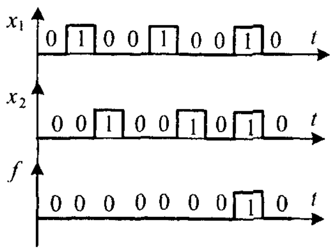


Рис. 2.6. Часова діаграма роботи кон'юнктора

3. Реалізація схеми *І* на основі правила де Моргана

Для реалізації схеми *І* досить часто використовується формула де Моргана

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

Вона дозволяє функціонально замінити операцію *І* операцією *АБО* і інверсією, що можна в технічному плані легше реалізувати на практиці. Функціонування такої схеми наведене в табл. 2.16, а її зображення – на рис. 2.7.

Таблиця 2.16. Логіка роботи схеми І на основі правила де Моргана

x	y	$\overline{x}$	$\overline{y}$	$\overline{x}+\overline{y}$	$\overline{\overline{x}+\overline{y}} = x \cdot y$
0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	0	1

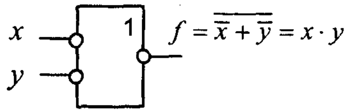


Рис. 2.7.Реалізація елемента І на основі правила де Моргана

4. Диз'юнктор

Диз'юнктор (схема диз'юнкції, схема *АБО*) представляє логічний елемент, який реалізує операцію *АБО* (логічне додавання).

Логічна функція, яку реалізує диз'юнктор

$$f = x_1 + x_2 = x_1 \vee x_2 .$$

Функціонує диз'юнктор відповідно до табл. 2.17, а його функціональна схема наведена на рис. 2.8. Релейний варіант схеми диз'юнкції наданий на рис. 2.9. Часова діаграма роботи подається на рис. 2.10.

Таблиця 2.17. Логіка роботи диз'юнктора

x_1 x_2	$f = x_1 \vee x_2$
0 0	0
0 1	1
1 0	1
1 1	1

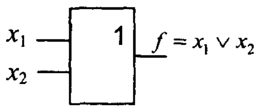


Рис. 2.8. Функціональна схема диз'юнктора

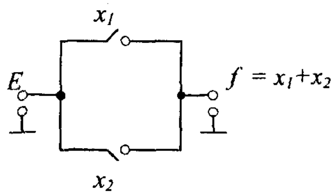


Рис. 2.9. Диз'юнктор у релейному виконанні

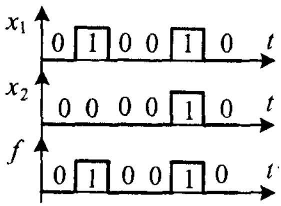


Рис. 2.10. Часова діаграма роботи диз'юнктора

5. Реалізація схеми *АБО* на основі правила де Моргана

Правило де Моргана використовується у вигляді формули:

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

Функціонування схеми відбувається відповідно до табл. 2.18.

Таблиця 2.18. Логіка роботи схеми АБО на основі правила де Моргана

x	y	$\overline{x}$	$\overline{y}$	$\overline{x} \cdot \overline{y}$	$\overline{\overline{x} \cdot \overline{y}} = x + y$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

Реалізується схема *АБО* з допомогою правила де Моргана у вигляді схеми, що подається на рис. 2.11.

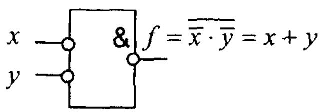
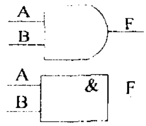
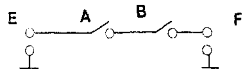
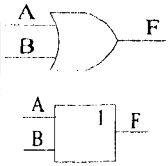
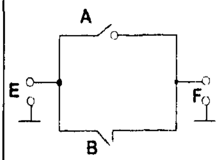
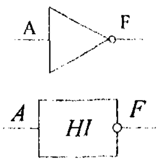
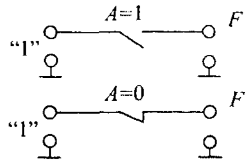
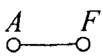
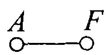
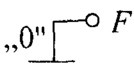
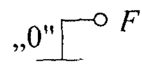
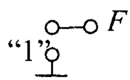
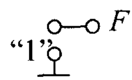


Рис. 2.11. Реалізація елемента АБО на основі правила де Моргана

Таблиця 2.19. Зведена таблиця елементів універсального логічного базису

Найме-нування	Графічне позначення	Реалізація функцій	Таблиця істинності	Релейне виконання
AND I, &		$F = A \cdot B =$ $= AB = A \& B =$ $= A \wedge B$	AB | F 00 | 0 01 | 0 10 | 0 11 | 1	
OR АБО, 1		$F = A + B =$ $= A \vee B$	AB | F 00 | 0 01 | 1 10 | 1 11 | 1	
NOT НІ		$F = \overline{A}$	A | F 0 | 1 1 | 0	
Змінна A		$F = A$	A | F 1 | 1 0 | 0	
Константа 0		$F = 0$	A | F 0 | 0	
Константа 1		$F = 1$	A | F 1 | 1	

Контрольні запитання

1. Розкажіть, як працює інвертор. Наведіть його типове зображення і часову діаграму роботи.
2. Як виглядає зображення диз'юнктора та часова діаграма його роботи?
3. Поясніть, як працює кон'юнктор і наведіть його графічну типову схему.
4. Наведіть релейні схеми інвертора, диз'юнктора та кон'юнктора.

Математика – це лише один з видів прикладної логіки.

М.Г. Чернишевський

Лекція 11
СПЕЦІАЛЬНІ ФУНКЦІОНАЛЬНО ПОВНІ ЛОГІЧНІ ЕЛЕМЕНТИ

1. Реалізація логічних схем на основі елемента Штрих Шефера

Розглянуті вище логічні функції створювали функціонально-повний логічний базис. Такий базис називається ***мінімальним***, якщо видалення з нього хоча б однієї з логічних функцій перетворює його на неповний. До таких базисів належить елемент Штрих Шефера – $A|B = \overline{A \cdot B}$, який реалізує операцію логічного множення з інверсією ($I - HI$). Його функціональна схема наведена на рис. 2.12.

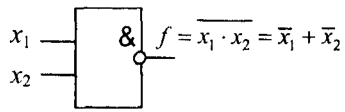


Рис.2.12. Функціональна схема елемента Штрих Шефера

Елемент Штрих Шефера створює *універсальний* функціонально-повний логічний базис, оскільки з його допомогою можна виконати логічні операції інверсії, додавання та множення, а також константи 1 і 0. Виключення з цього елементу можливості виконувати операцію інверсії чи логічного множення робить його *неповним*. Тому він створює мінімальний базис.

З допомогою цього елемента легко реалізуються операції константа 0 і константа 1 (див. рис. 2.13). Для цього лише потрібно постійно подавати на його входи 1 або 0. Тоді на виході відповідно буде постійно знаходитися 0 або 1. Якщо ж 1 і 0 будуть знаходитися на входах як сигнали, тобто деякий час, і потім змінюватися на протилежні, то тоді буде виконуватися операція інверсії.

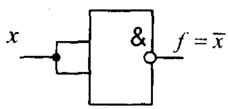


Рис.2.13. Реалізація інвертора на елементах Штрих Шефера

Реалізація схеми I з допомогою елемента Штрих Шефера показана на рис. 2.14, а схеми *АБО* на рис. 2.15.

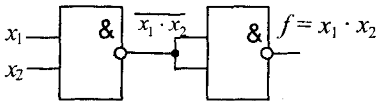


Рис. 2.14. Реалізація схеми І на елементах Штрих Шефера

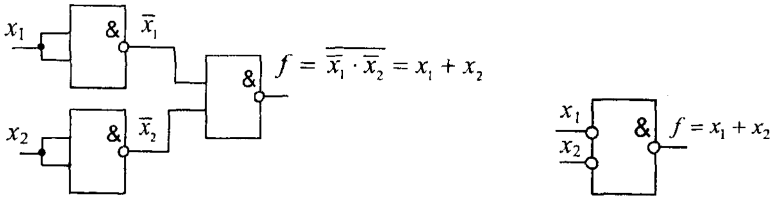


Рис. 2.15. Реалізація схеми АБО на елементах Штрих Шефера

2. Реалізація логічних схем на основі елемента Стрілка Пірса

Стрілка Пірса $A \downarrow B = \overline{A + B}$ реалізує операцію множення с інверсією (АБО – НІ) і за аналогією з елементом Штрих Шефера створює функціонально повний універсальний мінімальний базис. Схема цього елемента показана на рис. 2.16.

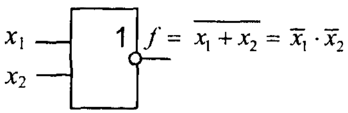


Рис. 2.16. Функціональна схема елемента Стрілка Пірса

Виконання інверсії з допомогою цього елемента наведене на рис.2.17. Як бачимо, для цього потрібно лише перемкнути входи цього елемента між собою.

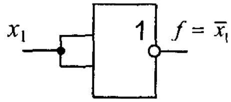


Рис. 2.17. Реалізація інвертора на елементі Стрілка Пірса

Операція логічного множення *I* зображена на рис. 2.18. Для її виконання потрібно три елементи Стрілка Пірса, два з яких виконують функції схеми *НІ*.

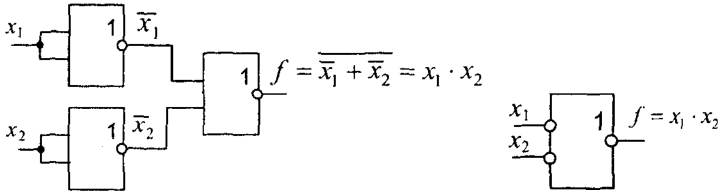


Рис. 2.18. Реалізація схеми I на елементах Стрілка Пірса

Простіше реалізується схема логічного додавання АБО, оскільки для своєї реалізації потребує лише два елемента Стрілка Пірса (див. рис. 2.19).

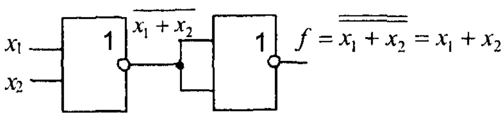


Рис. 2.19. Реалізація схеми АБО на елементах Стрілка Пірса

Контрольні завдання:

1. Побудуйте таблиці функціонування універсальних елементів Штрих Шефера і стрілка Пірса.
2. Реалізуйте з допомогою елемента Штрих Шефера і Стрілка Пірса логічні операції імплікації, рівнозначності і нерівнозначності.

Жодне людське дослідження не може назватись істинною наукою, якщо воно не пройшло через математичне доведення.

Леонардо да Вінчі

РОЗДІЛ 2. НОРМАЛЬНІ ФОРМИ ЛОГІЧНИХ ФУНКЦІЙ

Лекція 12
ДОСКОНАЛІ ДИЗ'ЮНКТИВНІ НОРМАЛЬНІ ФОРМИ

1. Елементарні добутки

Логічні функції можуть бути подані в різних формах. Серед них досить поширеними є *нормальні* форми. Їх вивчення важливе, оскільки з їх допомогою відбувається синтез логічних схем у цифровій техніці. Практично не існує такого цифрового пристрою чи системи (наприклад, комп'ютера чи мобільного телефону), де б не застосовувалася логічна схема, синтез якої не відбувався би на основі нормальних форм, серед яких важливе місце посідають *диз'юнктивні нормальні форми* (ДНФ). Тому вивченню методів синтезу логічних схем на основі диз'юнктивних нормальних форм у посібниках з цифрової техніки приділяється значна увага. У даному підручнику методам синтезу логічних схем приділяється також значна увага як у практичному аспекті, так і теоретичному. Такий підхід дозволить майбутнім спеціалістам в галузі цифрової й комп'ютерної техніки більш професіонально розробляти різні цифрові пристрої, особливо використовуючи їх у структурах програмно-логічних інтегральних схем.

Означення 1. Добуток кількох змінних, узятих із запереченням або без, називається елементарним ***добутком***, або ***кон`юнкцією***.

Приклад 1. Елементарним добутком є вирази $x_1\bar{x}_2$, $x_1\bar{x}_2x_3$, $\bar{x}z$, xyz, а вирази $\overline{xy}$ та $xy \vee x\bar{z}$ не є ним, оскільки в першому виразі член $\overline{xy}$ має спільне заперечення, а в другому між добутками змінних знаходиться знак, який не відповідає операції логічного множення.

Означення 2. Логічна функція, що подається диз'юнкцією елементарних добутків (кон'юнкцій), називається ***диз'юнктивною нормальною формою*** (ДНФ).

Теорема 1. ***Елементарний добуток змінних, чи кон'юнкція, дорівнює одиниці тоді і лише тоді, коли її змінні із запереченням дорівнюють нулю, а змінні без заперечення – одиниці.***

Доведення. Поява хоча б одного нуля в логічному добутку змінних із запереченням і без згідно з властивістю $0 \wedge x = 0$ спричинить нульове значення всього добутку.

Тепер припустимо, що в заданому елементарному добутку хоча б одній змінній із запереченням $\bar{x}$ присвоєна одиниця з будь-якого набору. Тоді ця змінна із запереченням в силу $\bar{1} = 0$ набуде значення, що дорівнює нулю, і, відповідно, увесь елементарний добуток буде дорівнювати 0.

Припустимо далі, що хоча б одна змінна x без заперечення набула нульового значення. Тоді згідно з властивістю добутку $0 \wedge x = 0$ логічний добуток змінної, що набула нульового значення, з рештою змінних незалежно від їх значень також дорівнюватиме нулю.

У випадку, коли в елементарному добутку змінні із запереченням $\bar{x}$ набувають значення нуля, а змінні без заперечення x – одиниці, то, оскільки $\bar{0} = 1$ і $1 \wedge x = x$, добуток усіх змінних дорівнюватиме одиниці.

Таким чином, елементарний добуток дорівнює одиниці лише в тому випадку, коли всім змінним із запереченням присвоєно нуль, а всім змінним без заперечення – одиницю.

Теорему доведено.

Наслідок. Кожному елементарному добутку змінних відповідає один і лише один набір їх значень, на якому цей добуток дорівнює одиниці.

Таким єдиним набором, як це випливає з теореми 1, є набір, в якому змінним із запереченням присвоюється нуль, а змінним без заперечення – одиниця.

Приклад 2. Набір значень змінних x_1, x_2, x_3, на якому елементарний добуток $\varphi = \bar{x}_1 x_2 x_3$ дорівнює одиниці, буде 011, оскільки $\varphi = \bar{0} \wedge 1 \wedge 1 = 1 \wedge 1 \wedge 1 = 1$. Цей набір згідно з наслідком теореми 1 є єдиним. Будь-який інший набір перетворює добуток φ у нуль. Наприклад, якщо змінні x_1, x_2, x_3 набудуть відповідно значення 0, 0, 1, то $\varphi = \bar{0} \wedge 0 \wedge 1 = 1 \wedge 0 \wedge 1 = 0$.

Теорема 2. ***Для кожного набору значень існує один і лише один елементарний добуток змінних, який приймає на ньому значення, що дорівнює одиниці, за умови, що кількість змінних у добутку буде дорівнювати кількості їх значень у наборі.***

Доведення. Зафіксуємо довільний набір значень і побудуємо на ньому елементарний добуток змінних з такою ж самою їх кількістю, як і кількість значень у їх початковому наборі. Розставимо на добутку змінних заперечення таким чином, щоб він дорівнював одиниці. Це означає, що в цьому добутку змінних кожному нулю з набору їх значень відповідатиме одна з них із запереченням, а одиниці – змінна без заперечення. Будь-який інший добуток змінних буде відрізнятися від добутку, який розглядається, наявністю заперечення хоча б в одній з його змінних. У результаті елементарний добуток змінних стане дорівнювати нулю.

Тобто набору значень змінних, який розглядається, відповідає лише один набор змінних, який на цьому наборі буде дорівнювати одиниці.

Теорему доведено.

2. Конституенти одиниці

Означення 3. Логічна функція n змінних, що приймає значення, яке дорівнює одиниці, лише на одному їх наборі, називається ***конституента одиниці***.

Інколи плутають поняття конституента одиниці з поняттям константа одиниці. Це, проте, різні речі. Константа одиниці дорівнює одиниці на *всіх* без винятку наборах значень, а конституента одиниці лише на *одному*. На це слід звернути особливу увагу, оскільки ці поняття відіграють важливу роль в алгебрі логіки.

Теорема 3. ***Елементарний добуток (кон'юнкція) усіх n змінних, що належать функції F, створює конституенту одиниці.***

Доведення. Оскільки відповідно до наслідку теореми 1 будь-який елементарний добуток змінних приймає значення, яке дорівнює 1, лише на одному наборі значень, то й добуток усіх n змінних дорівнює одиниці лише на одному наборі їх значень. Такий добуток саме й створює конституенту одиниці.

Теорему доведено.

Теорема 4. ***Число конституент одиниці n змінних дорівнює 2^n.***

Доведення. Доведення випливає з того, що кількість наборів значень для n змінних дорівнює 2^n, а кожному такому набору значень відповідає лише один добуток змінних, який самий й створює конституенту одиниці. Отже, число конституент одиниці буде дорівнювати 2^n.

Теорему доведено.

Відповідно до змісту теореми 1, для того щоб представити конституенту одиниці з n змінними на m-му наборі, слід подати число m у вигляді n-розрядного двійкового числа і в добутку змінних узяти із запереченням лише ті змінні, яким у двійковому числі відповідають нулі, а інші залишити без змін.

Приклад 3. Записати конституенту одиниці 17-го набору.

***Розв`язання.*** Подамо число 17 у двійковому вигляді: $17_{\langle 10\rangle} = 10001_{\langle 2\rangle}$. Запишемо кон`юнкцію п`яти змінних: $x_1x_2x_3x_4x_5$. Візьмемо із запереченням ті змінні, яким у двійковому числі відповідають нулі: $x_1\overline{x}_2\overline{x}_3\overline{x}_4x_5$. Одержаний результат є конституента одиниці 17-го набору.

3. Подання логічних функцій у ДДНФ

Одна й та ж сама логічна функція може бути подана з допомогою диз'юнкції різних добутків змінних, але найбільш простою й уживаною є досконала диз'юнктивна нормальна форма.

Означення 4. Диз`юнкція конституент одиниці, які дорівнюють одиниці на тих самих наборах, що й задана функція, називається ***досконалою диз`юнктивною нормальною формою (ДДНФ)*** логічної функції.

Теорема 5. ***Будь-яка логічна функція F, крім константи нуля, може бути поданою в ДДНФ, до того ж єдиним способом.***

Доведення. Дійсно, будь-яка логічна функція F, крім константи нуля, характеризується тим, що на $k\,(1 \le k \le 2^n)$ наборах вона дорівнює одиниці, а на решті – нулю. Конституента одиниці дорівнює одиниці лише на одному наборі, а на решті – приймає нульове значення. Тому, якщо всі конституенти одиниці K_i , $i = 1, 2, ..., k$, які дорівнюють одиниці на тих самих наборах, що й логічна функція, об`єднати знаком диз`юнкції, то відповідно до правила $1 \vee x = 1$ функція F буде дорівнювати одиниці на цих же самих наборах і нулю, якщо жодна з конституент одиниці K_i не буде дорівнювати одиниці. Це можливе лише в разі наборів значень змінних, які не належать жодній з цих конституент.

З цього випливає, що будь-яка функція F може бути подана з допомогою диз`юнкції конституент одиниці. Таке подання єдине, оскільки в протилежному випадку дві або більше різних конституенти будуть дорівнювати одиниці на одному й тому ж самому наборі змінних, або два різних набори будуть призводити до того, що конституента дорівнюватиме одиниці. Перший випадок виключається теоремою 1, а другий – теоремою 2. При цьому константа нуля на всіх наборах змінних дорівнює нулю. Тому її не можна подати за допомогою конституент одиниці, тобто у ДДНФ.

Теорему доведено.

Виходячи з доведеної теореми 5, розглянемо задачу подання логічних функцій у ДДНФ.

Для її розв`язання потрібно скласти диз`юнкцію конституент одиниці, які дорівнюють одиниці на тих самих наборах, що й задана функція, за таким алгоритмом:

1. Виписати за числом одиниць у логічній функції добутки всіх аргументів від першого до n-го і поєднати їх знаком диз`юнкції.

2. Записати під кожним аргументом його значення у відповідному до даного добутку наборі, що дорівнює 1 або 0, і над аргументами, що дорівнюють нулю, поставити знаки заперечення.

Даний алгоритм носить назву запису логічної функції в досконалій диз'юнктивній формі або за одиницями.

Приклад 4. Подати у ДДНФ логічну функцію п`яти змінних $F = F\ (x_1, x_2, x_3, x_4, x_5)$, що дорівнює одиниці на наборах з номерами 5, 14, 16, 31 і нулю – на решті наборів.

Розв`яжемо цю задачу за наведеним вище алгоритмом:

1. Випишемо чотири конституенти одиниці й поєднаємо їх знаками диз`юнкції

$$x_1x_2x_3x_4x_5 \vee x_1x_2x_3x_4x_5 \vee x_1x_2x_3x_4x_5 \vee x_1x_2x_3x_4x_5.$$

2. Під кожним добутком поставимо набори значень змінних, поданих у двійковому виді числа 5, 14, 16, 31, і розставимо заперечення над змінними, що дорівнюють нулю.

Тоді отримаємо ДДНФ:

$$F = \overline{x}_1\,\overline{x}_2\,x_3\,\overline{x}_4\,x_5 \vee \overline{x}_1\,x_2\,x_3\,x_4\,\overline{x}_5 \vee x_1\,\overline{x}_2\,\overline{x}_3\,\overline{x}_4\,\overline{x}_5 \vee x_1\,x_2\,x_3\,x_4\,x_5 ..$$
$$0\,0\,1\,0\,1 \quad 0\,1\,1\,1\,0 \quad 1\,0\,0\,0\,0 \quad 1\,1\,1\,1\,1$$

ДДНФ є найбільш загальною формою подання логічних функцій у диз'юнктивній нормальній формі і тому містить найбільш можливу кількість літер. Кількість літер, що належать до ДНФ логічної функції, *дорівнює* або *більше* від кількості змінних, які містяться в цій функції, оскільки одна і та ж сама змінна може входити до функції кілька разів. Наприклад, ДНФ функції $F = xy \vee x\overline{y}z \vee yz$ містить три змінних x, y, z і сім літер унаслідок того, що x входить до функції двічі, y – тричі (один раз із запереченням) і z – двічі.

Знати кількість літер під час синтезу логічних функцій важливо, оскільки кожна з них реалізується відповідною, як правило, електронною схемою, і тому від кількості літер залежить величина апаратурних затрат, а отже, і вартість цифрових пристроїв, але не тільки. Від цієї кількості залежить швидкодія і надійність цифрової апаратури. Тому питання зменшення кількості літер під час синтезу логічних функцій є досить важливим, і на нього завжди слід звертати увагу. На практиці й відповідно в теорії постає проблема скорочення кількості літер у ДДНФ. Розв'язання цієї задачі розглядається в наступних лекціях підручника.

Математика – це мова природи.

Дж. Гіббс

Лекція 13
СКОРОЧЕНІ ДИЗ'ЮНКТИВНІ НОРМАЛЬНІ ФОРМИ

1. Імпліканти логічної функції

Означення 1. Якщо деяка логічна функція φ (в окремому випадку елементарний добуток) дорівнює нулю на тих наборах, на яких дорівнює нулю інша функція *F*, то функція φ називається ***імплікантою*** функції *F*. При цьому функція φ може дорівнювати нулю на тих наборах, на яких $F = 1$, але не навпаки.

У разі, коли функція φ є імплікантою функції *F*, то кажуть, що функція φ *належить функції F*. Умова входження записується як $\varphi \subset F$.

Застосування терміну *імпліканта* пов`язане з логічною функцією $F_{13} = A \to B$. Вираз $\varphi \to F$ дорівнює одиниці лише тоді, коли $\varphi \subset F$, тобто на наборах, де значення для φ і *F* відповідно мають вид: 0 – 0, 0 – 1, 1 – 1. Якщо ж $\varphi \not\subset F$, то на одному з наборів функція φ має прийняти одиничне значення, а функція *F* – нульове. У цьому разі вираз $(\varphi \to F) = 0$.

Приклад 1. У табл. 2.20 дані функції

$$F = F(x_1, x_2, x_3),\ \ \varphi = \varphi(x_1, x_2, x_3), Z = Z(x_1, x_2, x_3).$$

Яка з функцій φ і Z є імплікантою функції *F*?

Таблиця 2.20. Функції трьох змінних для прикладу 1

Номер набору	x_1	x_2	x_3	F	φ	Z
0	0	0	0	1	1	0
1	0	0	1	0	0	0
2	0	1	0	1	0	0
3	0	1	1	1	1	0
4	1	0	0	0	0	1
5	1	0	1	0	0	0
6	1	1	0	1	0	0
7	1	1	1	1	1	1

Відповідь. Імплікантою є функція φ, оскільки вона дорівнює нулю на тих наборах, де функція F також дорівнює нулю. Функція Z не є імплікантою функції F, оскільки на наборі 4 функція $Z = 1$, а функція $F = 0$.

Теорема 1. ***Будь-яка конституента одиниці, що належить до ДДНФ логічної функції F, є її імплікантою.***

Доведення. Дійсно, конституента одиниці дорівнює одиниці лише на одному наборі змінних. А оскільки ДДНФ логічної функції F складається з диз`юнкції конституент одиниці, то одиничне значення будь-якої конституенти одиниці, що належить до ДДНФ, збігається з одним із значень одиниці логічної функції F. Решта значень конституенти одиниці, як відомо, буде дорівнювати нулю. Тобто неможливо, щоб імпліканта набула значення одиниці, а відповідне їй значення функції дорівнювало нулю. Тому конституента одиниці, що належить до ДДНФ, є її імплікантою.

Теорему доведено.

Теорема 2. ***Константа нуля є імплікантою будь-якої логічної функції.***

Доведення. Дійсно, функція константа нуля за означенням дорівнює нулю на всіх її можливих наборах значень, і тому обов`язково дорівнює нулю на всіх тих наборах, на яких функція F дорівнює нулю. Цього досить для того, щоб константа нуля мала всі ознаки імпліканти.

Теорему доведено.

Кожна логічна функція є імплікантою самої себе. Це випливає з того, що ця функція збігається сама з собою й, відповідно, дорівнює нулю на тих наборах значень, де вона вже дорівнювала нулю раніше ще до того, як проводився її аналіз щодо належності до імпліканти, і ніде не дорівнює одиниці, де функція дорівнює нулю. Саме це і є ознакою імпліканти.

Теорема 3. ***Будь-яка логічна функція є імплікантою константи одиниці.***

Доведення. Константа одиниці не дорівнює нулю на жодному наборі. Це означає, що в будь-якої логічної функції, на яких наборах не стояли б у неї одиниці, існуватиме відповідність її одиничних значень до одиничних значень константи одиниці, і ніде не можливий набір, для якого логічна функція дорівнюватиме одиниці, а константа одиниці в цей час дорівнюватиме нулю.

Теорему доведено.

2. Прості імпліканти

Означення 2. Елементарний добуток, одержаний шляхом виключення з початкового добутку однієї або кількох змінних, називається ***власною*** частиною останнього.

Приклад 2. Нехай елементарний добуток – $x\overline{y}z$. Тоді його власними частинами будуть добутки

$$x\overline{y},\ \overline{y}z,\ xz,\ x,\ \overline{y},\ z.$$

Означення 3. Елементарні добутки, які самі належать даній функції в ДДНФ, але ніяка їх власна частина самостійно їй не належить, називаються ***простими*** імплікантами.

Приклад 3. Припустимо, що добутки $\varphi = x_1x_2x_3\overline{x}_4$ і $x_2\overline{x}_4$ належать деякій логічній функції $F = F(x_1, x_2, x_3, x_4)$, а змінні x_2 і $\overline{x}_4$ самостійно до неї не входять. Тоді добуток $x_2\overline{x}_4$ буде простою імплікантою функції F, оскільки $x_2\overline{x}_4 \subset \varphi \subset F$, а $\overline{x}_2 \not\subset F$ і $\overline{x}_4 \not\subset F$. Разом з тим добуток φ не буде простою імплікантою, оскільки його власна частина $x_2\overline{x}_4 \subset F$.

Якщо добуток належить даній функції, то з появою в ньому додаткових змінних новий добуток також буде належати цій функції, оскільки він перетворюється на нуль завжди, коли вихідний добуток дорівнює нулю, незалежно від того, дорівнюють чи не дорівнюють нулю додаткові змінні. Це випливає з того, що добуток 0 з будь-якою змінною дорівнює 0.

3. Подання логічних функцій у скорочених ДНФ

Означення 4. Диз`юнкція простих імплікант називається ***скороченою ДНФ.***

Теорема 4. ***Будь-яку логічну функцію F можна подати у вигляді скороченої ДНФ, тобто диз'юнкції простих імплікант.***

Доведення. Оскільки прості імпліканти належать самостійно функції F, то вони повинні дорівнювати нулю на тих самих наборах, що й функція F. Якщо це не так і проста імпліканта дорівнює одиниці там, де функція F має дорівнювати нулю, то на підставі рівності $1 \vee x = 1$ функція F дорівнюватиме одиниці, що неприпустимо для імплікант.

Для кожного набору, де функція F дорівнює одиниці, має знайтись хоча б одна проста імпліканта, що дорівнює одиниці, інакше функція F у цьому випадку не буде дорівнювати одиниці. В гіршому випадку такою імплікантою є конституента одиниці.

Серед імплікант у функції F можуть знайтись прості імпліканти u, які є власною частиною імплікант y. У цьому випадку $y = uv$. На підставі рівностей $uv + u = u(v + 1) = u$ і відповідно $y + u = u$ відбудеться поглинання імпліканти y її власною частиною u, яка є простою імплікантою.

Це означає, що функція F після всіх можливих поглинань врешті-решт буде складатися з одних простих імплікант.

Теорему доведено.

З метою подання логічної функції F у вигляді скороченої ДНФ розглянемо операції повного і неповного склеювання, а також поглинання і розгортання в ДНФ.

Операція повного склеювання визначається співвідношенням

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

Це випливає з того, що

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

Склеювання добутків xy і $x\overline{y}$ відбувається в цьому разі за змінною y.

Операція неповного склеювання має вигляд:

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

Це можливе тому, що логічне додавання змінної x з виразом $xy \vee x\bar{y}$ ніяк не впливає на останній. Він залишається незмінним, скільки б разів змінна x до нього не додавалась.

Операція поглинання визначається з рівностей

$$x \vee xy = x \text{ і } x \vee x\bar{y} = x.$$

У цьому випадку x поглинає весь вираз. Це випливає з того, що

$$x \vee xy = x\,(1 \vee y) = x.$$

Відповідно

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

Оскільки далеко не завжди вихідна логічна функція подається в ДДНФ, розглянемо операцію її розгортання. Вона перетворює будь-яку просту імпліканту в диз`юнкцію конституент одиниці.

Нехай, наприклад, $x\bar{y}$ – проста імпліканта логічної функції чотирьох аргументів: x, y, z, u. Тоді, застосовуючи двічі операцію розгортання, одержимо

$$x\bar{y} = x\bar{y}\,(z \vee \bar{z}) = x\bar{y}z \vee x\bar{y}\bar{z} = x\bar{y}z\,(u \vee \bar{u}) \vee x\bar{y}\bar{z}\,(u \vee \bar{u}) =$$
$$= x\bar{y}zu \vee x\bar{y}z\bar{u} \vee x\bar{y}\bar{z}u \vee x\bar{y}\bar{z}\bar{u}.$$

При розгортанні різні імпліканти можуть утворювати одну й ту ж саму конституенту одиниці. У цьому разі на основі тотожності $x \vee x = x$ треба залишити одну конституенту одиниці. Унаслідок цього одержимо ДДНФ вихідної логічної функції.

Від часів греків говорити "математика"
означає говорити "доведення".

Н. Бурбакі

Лекція 14
ДОСКОНАЛІ КОН'ЮКТИВНІ НОРМАЛЬНІ ФОРМИ

1. Елементарні суми

Означення 1. Логічна сума кількох різних змінних, узятих із запереченням або без нього, називається ***елементарною сумою,*** або ***диз`юнкцією***.

Означення 2. Логічна функція, що подається кон`юнкцією елементарних сум (диз`юнкцій), називається ***кон`юнктивною нормальною формою (КНФ)***.

Приклад 1. Елементарними сумами будуть вирази $x_1 \vee \overline{x}_2$, $x_1 \vee \overline{x}_2 \vee x_3$, $x \vee y \vee z$, а вирази $\overline{x \vee y}$ і $(x \vee y)(x \vee \overline{z})$ ними не є, оскільки в першому виразі член $\overline{x \vee y}$ має загальне заперечення, а в другому є знак кон`юнкції.

Теорема 1. ***Елементарна сума, або диз'юнкція, дорівнює нулю тоді і лише тоді, коли її змінним із запереченням присвоєна одиниця, а змінним без заперечення – нуль.***

Доведення теорем тут і в лекції 15 не подаються, оскільки вони аналогічні доведенням відповідних теорем у лекціях 13, 14.

Наслідок. Кожній елементарній сумі відповідає один і лише один набір значень змінних, що належать до неї, на якому вона дорівнює нулю.

Приклад 2. Набір значень змінних x_1, x_2, x_3, на якому елементарна сума $\varphi = \overline{x}_1 + x_2 + x_3$ дорівнює нулю, буде дорівнювати 100, оскільки $\varphi = \overline{1} \vee 0 \vee 0 = 0 \vee 0 \vee 0 = 0$.

Теорема 2. ***Для кожного набору значень існує одна й лише одна елементарна сума змінних, яка набуває на ньому значення, що дорівнює нулю, за умови, що кількість змінних у сумі буде дорівнювати кількості значень у наборі.***

2. Конституенти нуля

Означення 3. Логічна функція n змінних, що приймає значення, яке дорівнює нулю, лише на одному їх наборі, називається ***конституентою нуля***.

Теорема 3. ***Елементарна сума (диз'юнкція) усіх n змінних, що належать до функції F, є конституентою нуля.***

Теорема 4. ***Число конституент нуля n змінних дорівнює*** 2^n.

Згідно з теоремою 1, щоб подати конституенту нуля від n змінних, що дорівнює 0 на m-му наборі, потрібно подати число m у вигляді n-розрядного двійкового числа і в диз`юнкції всіх змінних узяти із запереченням лише ті змінні, яким у двійковому числі відповідають одиниці.

Приклад 3. Записати конституенту нуля 17-го набору.

***Розв`язання.*** Подамо десяткове число 17 у двійковому вигляді: $17_{(10)} = 10001_{(2)}$. Запишемо диз`юнкцію 5 змінних: $x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$. Візьмемо із запереченням ті змінні, яким у двійковому коді відповідають одиниці: $\overline{x}_1 \vee x_2 \vee x_3 \vee x_4 \vee \overline{x}_5$. Одержаний результат є конституентою нуля 17-го набору.

3. Подання логічних функцій у ДКНФ

Означення 4. Кон`юнкція конституент нуля, що дорівнює нулю на тих самих наборах, що й задана функція, називається ***досконалою кон`юнктивною нормальною формою (ДКНФ)*** логічної функції.

Теорема 5. ***Будь-яка логічна функція, крім конституенти одиниці, може бути подана в ДКНФ, до того ж єдиним способом.***

Виходячи з теореми 5, розглянемо алгоритм подання логічних функцій в ДКНФ.

З цією метою треба скласти добуток тих конституент нуля, що дорівнюють нулю на тих самих наборах, що й задана функція, за таким алгоритмом:

а) виписати за числом нулів у логічній функції диз`юнкції всіх аргументів від першого до $n$-го й по`єднати їх знаками кон`юнкції;

б) записати під кожним аргументом його значення у відповідному до даного добутку наборі, яке дорівнює одиниці або нулю, і над аргументами, що дорівнюють одиниці, поставити знаки заперечення.

Даний алгоритм носить назву запису логічної функції в досконалій кон`юктивній нормальній формі або за нулями.

Приклад 4. Подати у ДКНФ логічну функцію п`яти аргументів $F = F(x_1, x_2, x_3, x_4, x_5)$, що дорівнює нулю на наборах з номерами 5, 14, 16, 31 і одиниці – на решті наборів.

Розв`яжемо цю задачу за наведеним вище алгоритмом:

1. Випишемо чотири суми аргументів і поєднаємо їх знаками кон`юнкції

$$(x_1 + x_2 + x_3 + x_4 + x_5) \wedge (x_1 + x_2 + x_3 + x_4 + x_5) \wedge$$
$$\wedge (x_1 + x_2 + x_3 + x_4 + x_5) \wedge (x_1 + x_2 + x_3 + x_4 + x_5).$$

2. Під кожною сумою поставимо подані у двійковому вигляді числа 5, 14, 16, 31 і розставимо заперечення над змінними, які відповідають одиниці.

Тоді отримаємо ДКНФ

$$F = (\underset{0}{x_1} + \underset{0}{x_2} + \underset{1}{\overline{x}_3} + \underset{0}{x_4} + \underset{1}{\overline{x}_5}) \wedge (\underset{0}{x_1} + \underset{1}{\overline{x}_2} + \underset{1}{\overline{x}_3} + \underset{1}{\overline{x}_4} + \underset{0}{x_5}) \wedge$$
$$\wedge (\underset{1}{\overline{x}_1} + \underset{0}{x_2} + \underset{0}{x_3} + \underset{0}{x_4} + \underset{0}{x_5}) \wedge (\underset{1}{\overline{x}_1} + \underset{1}{\overline{x}_2} + \underset{1}{\overline{x}_3} + \underset{1}{\overline{x}_4} + \underset{1}{\overline{x}_5}).$$

У математиці сила невідомого незмірна.

Наполеон

Лекція 15
СКОРОЧЕНІ КОН'ЮНКТИВНІ НОРМАЛЬНІ ФОРМИ

1. Імпліценти логічної функції

ДКНФ є найбільш загальною формою подання логічної функції у кон`юнктивній формі і тому містить найбільше можливе кількість літер. Розглянемо питання зменшення їх числа.

Означення 1. Якщо деяка логічна функція φ (в окремому випадку – елементарна сума) дорівнює *одиниці* на всіх тих наборах, на яких дорівнює одиниці інша функція F, то функція φ називається ***імпліцентою*** функції F. Кажуть, що функція φ належить функції F. При цьому функція φ може дорівнювати одиниці і на тих наборах, на яких $F = 0$, але не навпаки – не може дорівнювати нулю на наборах, де функція F дорівнює одиниці.

Приклад 1. У в табл. 2.21 подані функції

$$F = F(x_1, x_2, x_3),\ \varphi = \varphi(x_1, x_2, x_3),$$
$$Z = Z(x_1, x_2, x_3).$$

Яка з функцій: φ чи Z – є імпліцентою функції F?

Таблиця 2.21. Функції трьох змінних для прикладу 1

	x_1	x_2	x_3	F	φ	Z
0	0	0	0	1	1	0
1	0	0	1	0	0	0
2	0	1	0	1	0	0
3	0	1	1	1	1	0
4	1	0	0	0	0	1
5	1	0	1	0	0	0
6	1	1	0	1	0	0
7	1	1	1	1	1	1

Відповідь. Ані функція φ, ані функція Z не є імпліцентами функції F, оскільки перша дорівнює нулю на наборах 2 і 6, а друга – на наборах 0, 2, 3 і 6. На цих наборах функція F дорівнює одиниці, що суперечить означенню імпліценти.

Теорема 1. ***Будь-яка конституента нуля, що належить до ДКНФ логічної функції F, є її імпліцентою.***

Теорема 2. ***Константа одиниці є імпліцентою будь-якої логічної функції.***

Теорема 3. ***Будь-яка логічна функція є імпліцентою константи нуля.***

2. Прості імпліценти

Означення 2. Елементарна сума, одержана виключенням з вихідної суми однієї або кількох змінних, називається ***власною*** частиною останньої.

Приклад 2. Нехай є елементарна сума $x+\overline{y}+z$. Тоді її власними частинами будуть суми:

$$x+\overline{y};\ \overline{y}+z;\ x+z;\ x;\ \overline{y};\ z.$$

Означення 3. Елементарні суми, які самі належать даній функції, але жодна їх власна частина не належить, називається ***простими імпліцентами***.

Приклад 3. Припустимо, що сума $\varphi = x_1 + x_2 + x_3 + \overline{x}_4$ і її частина $x_2 + \overline{x}_4$ належать як самостійні члени функції $F = F(x_1, x_2, x_3, x_4)$, а змінні x_2 і $\overline{x}_4$ не належать. Тоді сума $x_2 + \overline{x}_4$ буде простою імпліцентою функції F, оскільки $(x_2 + \overline{x}_4) \subset \varphi \subset F$, а $x_2 \not\subset F$ і $\overline{x}_4 \not\subset F$. У той же самий час сума φ не буде простою імпліцентою, оскільки $(x_2 + \overline{x}_4) \subset F$.

Якщо будь-які елементарні суми належать даній функції, то при додаванні до них будь-яких змінних нова сума також буде належати цій функції, оскільки вона перетворюється в одиницю разом з початковою сумою.

3. Подання логічних функцій у скорочених КНФ

Означення 4. Кон`юнкція простих імпліцент називається ***скороченою КНФ***.

Теорема 4. ***Будь-яку логічну функцію F можна подати у вигляді скороченої КНФ, тобто кон'юнкції простих імпліцент.***

З метою подання логічної функції F у вигляді скороченої КНФ розглянемо операції повного і неповного склеювання, а також поглинання і розгортання в КНФ.

Операція повного склеювання визначається співвідношенням

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

Це випливає з того, що

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

Склеювання сум $x+y$ і $x+\overline{y}$ відбувається в цьому випадку за змінною y.

Операція неповного склеювання має вигляд

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

Операція поглинання визначається з рівностей $x(x+y)=x$ і $x(x+\overline{y})=x$. У цьому випадку x поглинає весь вираз. Це випливає з того, що $x(x+y)=x+xy=x(1+y)=x$.

Відповідно

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

Операція розгортання перетворює будь-яку просту імпліценту в диз'юнкцію конституент нуля.

Нехай, наприклад, $x+\overline{y}$ – проста імпліцента логічної функції чотирьох аргументів: x, y, z, u. Тоді, застосовуючи двічі операцію розгортання, отримуємо:

$$\begin{aligned}
&x+\overline{y}=(x+\overline{y})+z\overline{z}=(x+\overline{y}+z)(x+\overline{y}+\overline{z})=\\
&=((x+\overline{y}+z)+u\overline{u})((x+\overline{y}+\overline{z})+u\overline{u})=\\
&=(x+\overline{y}+z+u)\,(x+\overline{y}+z+\overline{u})\,(x+\overline{y}+\overline{z}+u)(x+\overline{y}+\overline{z}+\overline{u}).
\end{aligned}$$

При розгортанні різні імпліценти можуть утворювати одну й ту саму конституенту нуля. У цьому разі на основі тотожності $xx = x$ треба залишити одну конституенту нуля. У результаті отримаємо ДКНФ початкової логічної функції.

Через логіку доводять, через інтуїцію винаходять.

А.Пуанкаре

РОЗДІЛ 3. МІНІМІЗАЦІЯ ЛОГІЧНИХ ФУНКЦІЙ

Лекція 16
МІНІМІЗАЦІЯ ЛОГІЧНИХ ФУНКЦІЙ У ДНФ

Метод мінімізації логічних функцій, що є предметом розгляду в цій лекції, базується на теоремі Квайна.

Теорема Квайна для ДДНФ. ***Якщо в ДДНФ логічної функції F виконати всі операції неповного склеювання, а потім усі операції поглинання, то одержимо скорочену ДНФ цієї функції, тобто диз`юнкцію всіх її простих імплікант.***

Доведення. Припустимо, що після виконання всіх операцій неповного склеювання, а потім поглинання отримана ДНФ буде містити у вигляді кон'юнкції член q, який не є простою імплікантою. Тоді до цієї функції, крім члена q, самостійно входить також у вигляді кон'юнкції якась його частина p, яка є простою імплікантою. Це означає, що функція F буде містити імпліканту q і просту імпліканту p у вигляді їх диз`юнкції $p \vee q$. Але $q = q_1 p$. Тоді член $q$ згідно з рівністю $p \vee q = p \vee q_1 p = p$ поглинатиметься простою імплікантою $p$, і, відповідно, ДНФ буде містити лише цю імпліканту. Отже, ця ДНФ складатиметься лише з простих імплікант, об`єднаних операціями диз`юнкції, тобто наше вихідне припущення неправильне, і вона буде надана у вигляді скороченої ДНФ.

Теорему доведено.

Особливістю метода мінімізації за Квайном є те, що його робота починається після подання логічної функції, яка мінімізується, в ДДНФ. Тому, якщо логічна функція задана в довільній ДНФ, то перед мінімізацією необхідно перетворити логічну функцію в ДДНФ шляхом її розгортання, як це було показане раніше в лекції 13.

Потім для проведення безпосередньо вже самої мінімізації необхідно виконати такі кроки:

1. У ДДНФ $F = F(x_1, x_2, \ldots, x_n)$ здійснюються всі операції неповного склеювання конституент одиниці.

Неповне склеювання, як уже зазначалося раніше, характеризується тим, що конституенти одиниці після склеювання їх з іншими конституентами, які належать до ДДНФ логічної функції, залишаються в ній для подальшої мінімізації. Це викликане тим, що кожна конституента одиниці може склеюватися з кількома іншими. Тоді її після першого склеювання не поглинають, а використовують для інших операцій склеювання з конституентами, з якими ще не виконувалося склеювання.

У результаті одержують конституенти одиниці й імпліканти, в які входять $(n-1)$ змінна. При цьому ймовірне також отримання й простих імплікант, якими також можуть бути й конституенти одиниці.

2. Відбувається поглинання імплікантами всіх конституент одиниці, які беруть участь у неповному склеюванні.

Конституенти одиниці, що беруть участь в операціях неповного склеювання, обов`язково поглинаються, оскільки кожна з них має у своєму складі імпліканту. Після першого склеювання, з якого починається мінімізація, імпліканти містять в собі $n-1$ літеру.

Конституенти одиниці, які не були задіяні в операціях склеювання, не можуть поглинатися, оскільки вони являють собою прості імпліканти з n змінними. Тому вони залишаються у функції як прості імпліканти.

3. Здійснюються операції неповного склеювання і поглинання імплікант з $n-1$ змінною, одержаних на першому кроці склеювання, за аналогією з пунктами 1 та 2.

Ця процедура повторюється доти, поки існує ймовірність операцій неповного склеювання. Отримана в результаті неї ДНФ відповідно до теореми Квайна буде скороченою, тобто міститиме в собі лише прості імпліканти.

Приклад 1. Знайти скорочену ДНФ логічної функції

$$F = F(x, y, z) = xy \vee \bar{x}z \vee \bar{y}\bar{z}.$$

Розв`язання. 1 Застосовуючи операцію розгортання, одержимо ДДНФ функції $F = F(x, y, z)$:

$$F = F(x, y, z) = x\,y(z \vee \bar{z}) \vee \bar{x}\,z(y \vee \bar{y}) \vee \bar{y}\,\bar{z}(x \vee \bar{x}) =$$
$$= xyz \vee xy\bar{z} \vee \bar{x}yz \vee \bar{x}\,\bar{y}z \vee x\bar{y}\,\bar{z} \vee \bar{x}\,\bar{y}\,\bar{z}.$$

2. Здійснимо в ДДНФ функції $F = F(x, y, z)$ усі можливі операції неповного склеювання конституент одиниці.

Для цього пронумеруємо всі конституенти одиниці функції

$$F = F(x, y, z) = \overset{1}{xyz} \vee \overset{2}{xy\overline{z}} \vee \overset{3}{\overline{x}yz} \vee \overset{4}{\overline{x}\,\overline{y}\,z} \vee \overset{5}{x\,\overline{y}\,\overline{z}} \vee \overset{6}{\overline{x}\,\overline{y}\,\overline{z}}$$

і виконаємо щодо цих конституент усі можливі операції неповного склеювання.

Результати такого склеювання занесемо до табл.. 2.22, в якій перший стовпчик показує номери конституент одиниці, другий – результат склеювання, третій – змінні, за якими відбулося склеювання.

Таблиця 2.22. Склеювання конституент одиниці в прикладі 1

Номера конституент, які склеюються	Імпліканта	Змінна, яка склеюється
1. 1 – 2	xy	z
2. 1 – 3	yz	x
3. 2 – 5	$x\overline{z}$	y
4. 3 – 4	$\overline{x}z$	y
5. 4 – 6	$\overline{x}\,\overline{y}$	z
6. 5 – 6	$\overline{y}\,\overline{z}$	x

3. Проведемо поглинання отриманими в табл. 2.22 імплікантами відповідних їм конституент одиниці. У результаті одержимо функцію

$$F = F(x,y,z) = xy \vee yz \vee x\overline{z} \vee \overline{x}z \vee \overline{xy} \vee \overline{yz}.$$

Очевидно, що подальше склеювання отриманих у табл. 1 імплікант неможливе, і тому вони будуть простими. Тобто одержана ДНФ буде скороченою.

Цей вираз містить шість простих імплікант. Серед них знаходяться й добутки, що належать початковій функції F. Тому інші імпліканти можна вважати зайвими.

Отриманий розв`язок лише підтвердив, що добутки вихідної функції F є простими імплікантами і вона була з самого початку скороченою, але не мінімальною.

Таким чином, ми бачимо, що скорочена ДНФ – це далеко не завжди мінімальна ДНФ, оскільки вона хоч і містить прості імпліканти, проте серед них можуть бути й надлишкові.

Означення. Диз`юнкція простих імплікант, жодна з яких не є зайвою, називається ***тупиковою*** ДНФ логічної функції.

Тупикових функцій у загальному випадку може бути декілька. Кожна з цих функцій може містити кількість літер, відмінну від решти функцій.

Тоді виникає проблема пошуку такої тупикової логічної функції, яка б мала мінімальну кількість літер. Така функція називається ***мінімальною ДНФ.***

Деякі логічні функції можуть мати кілька мінімальних ДНФ, що містять однакову кількість літер. У цьому випадку вибирається мінімальна ДНФ, яка більш придатна порівняно з рештою для технічної реалізації в цифровому пристрої або керуючій програмі.

З метою визначення мінімальної ДНФ використовуються імплікантні матриці.

Нижче в табл. 2.23 наведена імплікантна матриця для розглянутої вище в прикладі логічної функції. У ній за вертикальними входами записуються конституенти одиниці, які належать заданій функції, а за горизонтальними – прості імпліканти цієї функції, одержані зі скороченої ДНФ.

Якщо імпліканта є власною частиною деякої конституенти одиниці, то клітинка імплікантної матриці, що відповідає цій імпліканті та конституенті одиниці, помічається позначкою. У даному випадку цією позначкою буде хрестик. Щоб одержати мінімальну ДНФ заданої функції, достатньо знайти мінімальне число імплікант, які разом накриють хрестиками всі стовпці імплікантної матриці.

У табл. 2.23 кожний стовпець помічений двома хрестиками. Тому зі скороченої ДНФ функції, що розглядається, можна виключити будь-яку імпліканту з двох, що накривають одну й ту ж саму конституенту одиниці. Мінімальна кількість імплікант, що накривають хрестиками всі стовпці, дорівнює 3: xy накриває перший і другий, $\bar{x}z$ – третій і четвертий, $\overline{y}\overline{z}$ – п`ятий і шостий стовпці таблиці. Можна накрити й інакше: $yz$ накриває перший і третій, $x\bar{z}$ – другий і п`ятий, $\bar{x}\bar{y}$ – четвертий і шостий стовпці.

Отже, дана логічна функція має дві мінімальні ДНФ з однаковою кількістю (6) літер

$$1.\ F = xy \vee \bar{x}z \vee \overline{y}\overline{z}.$$

2. $F = yz \vee x\bar{z} \vee \overline{xy}$.

Таблиця 2.23. Імплікантна матриця для прикладу 1

Проста імпліканта	Конституента					
	1	2	3	4	5	6
	xyz	$xy\bar{z}$	$\bar{x}yz$	$\bar{x}yz$	$\overline{xy}z$	$\overline{xyz}$
1. xy	×	×				
2. yz	×		×			
3. $x\bar{z}$		×			×	
4. $\bar{x}z$			×	×		
5. $\overline{xy}$				×		×
6. $\overline{yz}$					×	×

Крім мінімальних форм, функція, що розглядається, як це випливає з таблиці 2.23, має ряд тупикових ДНФ, в яких кількість букв буде більшою, ніж у мінімальних. Наприклад, тупиковою буде така ДНФ:

$$F = xy \vee yz \vee x\bar{z} \vee \overline{xy}.$$

У ній жодна проста імпліканта не є зайвою, але кількість імплікант не три, як це має місце в мінімальній ДНФ, а чотири.

Виходячи з розглянутого вище прикладу, наведемо решту кроків, необхідних для мінімізації логічних функцій.

4. За отриманою скороченою ДНФ будується імплікантна матриця.

5. В імплікантній матриці містяться набори простих імплікант, які накривають усі конституенти одиниці логічної функції, що подана в ДНФ і мінімізується.

6. Серед цих наборів знаходять один або кілька таких, які в сумі містять мінімальну кількість літер.

7. Об`єднують прості імпліканти цих наборів знаками диз`юнкції і одержують одну або декілька мінімальних ДНФ.

Приклад 2. Знайти мінімальну ДНФ логічної функції $F = F(x_1, x_2, x_3, x_4)$, яка дорівнює одиниці на наборах з номерами 1, 3, 5, 7, 14, 15 і нулю на решті наборів.

Розв'язання. Спочатку подамо функцію F у ДДНФ

$$F = \overset{1}{\bar{x}_1\bar{x}_2\bar{x}_3x_4} \vee \overset{2}{\bar{x}_1\bar{x}_2x_3x_4} \vee \overset{3}{\bar{x}_1x_2\bar{x}_3x_4} \vee \overset{4}{\bar{x}_1x_2x_3x_4} \vee \overset{5}{x_1x_2x_3\bar{x}_4} \vee \overset{6}{x_1x_2x_3x_4}.$$

Потім здійснимо операції неповного склеювання в такому порядку:

1. Виконаємо всі можливі неповні склеювання першого члена функції в ДДНФ з рештою членів, потім другого, третього й т.д.

2. Проведемо всі можливі операції поглинання конституент одиниці і результат подамо в табл. 2.24 у колонці "Імпліканта".

Таблиця 2.24. Перше склеювання конституент 1 у прикладі 2

Номер конституенти "1", яка склеюється		Імпліканта	Змінна, яка склеюється
1.	1–2	$\bar{x}_1\bar{x}_2x_4$	x_3
2.	1–3	$\bar{x}_1\bar{x}_3x_4$	x_2
3.	2–4	$\bar{x}_1x_3x_4$	x_2
4.	3–4	$\bar{x}_1x_2x_4$	x_3
5.	4–6	$x_2x_3x_4$	x_1
6.	5–6	$x_1x_2x_3$	x_4

З табл. 2.24 видно, що всі конституенти одиниці поглинаються імплікантами, отриманими після склеювання. У результаті отримуємо функцію:

$$F = \overset{1}{\bar{x}_1\bar{x}_2x_4} \vee \overset{2}{\bar{x}_1\bar{x}_3x_4} \vee \overset{3}{\bar{x}_1x_3x_4} \vee \overset{4}{\bar{x}_1x_2x_4} \vee \overset{5}{x_2x_3x_4} \vee \overset{6}{x_1x_2x_3}.$$

3. Для отриманої скороченої функції F знову проведемо всі можливі операції неповного склеювання й поглинання, а результати ведемо в табл. 2.25.

Таблиця 2.25. Друге склеювання конституент 1 у прикладі 2

Номер склеюваної конституенти "1"		Імплиіканта	Склеювана змінна
1.	1–4	$\bar{x}_1x_4$	x_2
2.	2–3	$\bar{x}_1x_4$	x_3

У результаті одержимо

$$F = F(x_1, x_2, x_3, x_4) = \bar{x}_1 x_4 \vee x_2 x_3 x_4 \vee x_1 x_2 x_3.$$

До цього виразу операції неповного склеювання й поглинання застосувати не можна, і тому він є дійсно скороченою ДНФ заданої логічної функції. Тобто цей вираз містить тільки прості імпліканти, серед яких можуть бути й надлишкові.

Побудуємо імплікантну матрицю для одержаної функції F (див. табл. 2.26).

Таблиця 2.26. Імплікантна матриця

Проста імпліканта	Конституента одиниці					
	1	2	3	4	5	6
	$\bar{x}_1\bar{x}_2\bar{x}_3x_4$	$\bar{x}_1\bar{x}_2x_3x_4$	$\bar{x}_1x_2\bar{x}_3x_4$	$\bar{x}_1x_2x_3x_4$	$x_1x_2x_3\bar{x}_4$	$x_1x_2x_3x_4$
1. $\bar{x}_1x_4$	×	×	×	×		
2. $x_2x_3x_4$				×		×
3. $x_1x_2x_3$					×	×

З табл. 2.26 випливає, що до мінімальної форми обов`язково має ввійти імпліканта $\bar{x}_1 x_4$, оскільки лише вона накриває хрестиками перший, другий, третій і четвертий стовпці імплікантної матриці. Крім того, обов`язково має бути вибрана імпліканта $x_1x_2x_3$, оскільки вона накриває п`ятий і шостий стовпці. При виборі цих двох імплікант всі стовпці табл. 5 залишаються перекритими, і тому імпліканта $x_2x_3x_4$ є зайвою. У цьому випадку логічна функція має єдину мінімальну ДНФ

$$F = F(x_1,\ x_2,\ x_3,\ x_4) = \bar{x}_1 x_4 \vee x_1 x_2 x_3.$$

Спочатку був символ.

Д. Гільберт

Лекція 17
МІНІМІЗАЦІЯ ЛОГІЧНИХ ФУНКЦІЙ У КНФ

Метод мінімізації логічних функцій у КНФ буде розглянутий більш коротко, ніж у попередній лекції, де мали місце мінімальні ДНФ, оскільки між цими двома методами спостерігається аналогія.

Мінімізація логічних функцій у КНФ використовує також теорему Квайна, але вже для ДКНФ.

Теорема Квайна для ДКНФ. ***Якщо в ДКНФ логічної функції F виконати всі операції неповного склеювання, а потім усі операції поглинання, то отримаємо скорочену КНФ цієї функції, тобто кон`юнкцію всіх її простих імпліцент.***

Доводиться за аналогією з доведенням, наданим у лекції 16 для ДДНФ.

Для мінімізації логічної функції F, поданої в КНФ, з допомогою теореми Квайна її спочатку треба перетворити на ДКНФ, а потім виконати такі кроки:

1. У ДКНФ для $F = F(x_1, x_2, ..., x_n)$ виконати всі операції неповного склеювання, а потім поглинання конституент нуля.
2. Виконати всі операції неповного склеювання й поглинання імпліцент з $(n-1)$ змінними, отриманих на першому кроці склеювання, потім – імпліцент з $(n-2)$ змінними й т.ін., поки ця процедура можлива. У результаті буде отримана скорочена КНФ.
3. З допомогою скороченої КНФ побудувати імпліцентну матрицю.
4. В імпліцентній матриці відшукати набори простих імпліцент, які накривають усі конституенти нуля логічної функції, що подана в ДКНФ і мінімізується.
5. Серед цих наборів знайти такі, які в сумі містять мінімальну кількість літер.
6. Одержати мінімальні КНФ, об'єднавши прості імпліценти набора з їх мінімальною кількістю знаками кон`юнкції. Серед них вибрати одну найбільш ефективну для реалізації.

Приклад. Знайти мінімальну КНФ логічної функції, що дорівнює нулю на наборах з номерами 0, 1, 2, 3, 7, 9, 12, 13, 15 і одиниці – на решті.

1. Знайдемо скорочену КНФ:

$$F(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \overset{1}{\vee} x_3 \vee x_4) \wedge (x_1 \vee x_2 \overset{2}{\vee} x_3 \vee \overline{x}_4) \wedge$$

$$\wedge\ (x_1 \vee x_2 \overset{3}{\vee} \overline{x}_3 \vee x_4) \wedge (x_1 \vee x_2 \overset{4}{\vee} \overline{x}_3 \vee \overline{x}_4) \wedge (x_1 \vee \overline{x}_2 \overset{5}{\vee} \overline{x}_3 \vee \overline{x}_4) \wedge$$

$$\wedge (\overline{x}_1 \vee x_2 \overset{6}{\vee} x_3 \vee \overline{x}_4) \wedge (\overline{x}_1 \vee \overline{x}_2 \overset{7}{\vee} x_3 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_2 \overset{8}{\vee} x_3 \vee \overline{x}_4) \wedge$$

$$\wedge (\overline{x}_1 \vee \overline{x}_2 \overset{9}{\vee} \overline{x}_3 \vee \overline{x}_4).$$

2. Виконаємо всі можливі операції неповного склеювання й поглинання:

$$\begin{array}{ll}
1-2 = x_1 \vee x_2 \vee x_3 & x_4 \\
1-3 = x_1 \vee x_2 \vee x_4 & x_3 \\
2-4 = x_1 \vee x_2 \vee \overline{x}_4 & x_3 \\
2-6 = x_2 \vee x_3 \vee \overline{x}_4 & x_1 \\
3-4 = x_1 \vee x_2 \vee \overline{x}_3 & x_4 \\
4-5 = x_1 \vee \overline{x}_3 \vee \overline{x}_4 & x_2 \\
5-9 = \overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4 & x_1 \\
6-8 = \overline{x}_1 \vee x_3 \vee \overline{x}_4 & x_2 \\
7-8 = \overline{x}_1 \vee \overline{x}_2 \vee x_3 & x_4 \\
8-9 = \overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_4 & x_3
\end{array}$$

У результаті отримуємо

$$F(x_1, x_2, x_3, x_4) = (x_1 \vee \overset{1}{x_2} \vee x_3) \wedge (x_1 \vee \overset{2}{x_2} \vee x_4) \wedge (x_1 \vee \overset{3}{x_2} \vee \overline{x}_4) \wedge$$

$$\wedge (x_2 \vee \overset{4}{x_3} \vee \overline{x}_4) \wedge (x_1 \vee \overset{5}{x_2} \vee \overline{x}_3) \wedge (x_1 \vee \overset{6}{\overline{x}_3} \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overset{7}{\overline{x}_3} \vee \overline{x}_4) \wedge$$

$$\wedge (\overline{x}_1 \vee \overset{8}{x_3} \vee \overline{x}_4) \wedge (\overline{x}_1 \vee \overset{9}{\overline{x}_2} \vee x_3) \wedge (\overline{x}_1 \vee \overset{10}{\overline{x}_2} \vee \overline{x}_4).$$

В одержаному виразі знову виконаємо всі операції неповного склеювання й поглинання

$$1-5=x_1 \vee x_2 \qquad x_3$$
$$2-3=x_1 \vee x_2 \qquad x_4$$

У кінцевому підсумку отримуємо

$$F(x_1,x_2,x_3,x_4)=\overset{1}{(x_1 \vee x_2)} \wedge \overset{2}{(x_2 \vee x_3 \vee \overline{x}_4)} \wedge \overset{3}{(x_1 \vee \overline{x}_3 \vee \overline{x}_4)} \wedge$$
$$\wedge \overset{4}{(\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4)} \wedge \overset{5}{(\overline{x}_1 \vee x_3 \vee \overline{x}_4)} \wedge \overset{6}{(\overline{x}_1 \vee \overline{x}_2 \vee x_3)} \wedge \overset{7}{(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_4)}.$$

До останнього виразу операції неповного склеювання й поглинання застосувати не можна, і тому він є скороченою КНФ.

3. Побудуємо імпліцентну матрицю (див. табл. 2.27). До мінімальної форми слід включити імпліценти $\overset{1}{x_1 \vee x_2}$ і $\overset{6}{\overline{x}_1 \vee \overline{x}_2 \vee x_3}$.

Унаслідок вибору цих імпліцент опиняються перекритими стовпці з номерами 1, 2, 3, 4, 7, 8. Решту рядків 5, 6, і 9 можна перекрити двома способами: імпліцентами $\overset{4}{(\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4)}$ і $\overset{2}{(x_2 \vee x_3 \vee \overline{x}_4)}$ або $\overset{4}{(\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4)}$ і $(\overline{x}_1 \vee x_3 \vee \overline{x}_4)$.

Таким чином, задана функція має дві мінімальні форми з однаковим кількістю літер:

$$F(x_1,x_2,x_3,x_4)=(x_1 \vee x_2)(\overline{x}_1 \vee \overline{x}_2 \vee x_3)(\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4)(x_2 \vee x_3 \vee \overline{x}_4),$$
$$F(x_1,x_2,x_3,x_4)=(x_1 \vee x_2)(\overline{x}_1 \vee \overline{x}_2 \vee x_3)(\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4)(\overline{x}_1 \vee x_3 \vee \overline{x}_4).$$

Слід зазначити, що кількість літер у мінімальних КНФ і ДНФ логічної функції різна. Тому при розв`язуванні задач мінімізації логічних функцій потрібно знайти як диз`юнктивні, так і кон`юнктивні мінімальні нормальні форми і вибрати серед них форму з найменшою кількістю літер.

Таблиця 2.27. Імпліцентна матриця

Проста імпліцента	Конституента нуля		
	1 $x_1 + x_2 + x_3 + x_4$	2 $x_1 + x_2 + x_3 + \overline{x}_4$	3 $x_1 + x_2 + \overline{x}_3 + x_4$
1. $x_1 + x_2$	X	X	X
2. $x_2 + x_3 + \overline{x}_4$		X	
3. $x_1 + \overline{x}_3 + \overline{x}_4$			
4. $\overline{x}_2 + \overline{x}_3 + \overline{x}_4$			
5. $\overline{x}_1 + x_3 + \overline{x}_4$			
6. $\overline{x}_1 + \overline{x}_2 + x_3$			
7. $\overline{x}_1 + \overline{x}_2 + \overline{x}_4$			

Продовження таблиці 2.27

Проста імпліцента	Конституента нуля		
	4 $x_1 + x_2 + \overline{x}_3 + \overline{x}_4$	5 $x_1 + \overline{x}_2 + \overline{x}_3 + \overline{x}_4$	6 $\overline{x}_1 + x_2 + x_3 + \overline{x}_4$
1. $x_1 + x_2$	X		
2. $x_2 + x_3 + \overline{x}_4$			X
3. $x_1 + \overline{x}_3 + \overline{x}_4$	X	X	
4. $\overline{x}_2 + \overline{x}_3 + \overline{x}_4$		X	
5. $\overline{x}_1 + x_3 + \overline{x}_4$			X
6. $\overline{x}_1 + \overline{x}_2 + x_3$			
7. $\overline{x}_1 + \overline{x}_2 + \overline{x}_4$			

Продовження таблиці 2.27

Проста імпліцента	Конституента нуля		
	7 $\overline{x}_1+\overline{x}_2+x_3+x_4$	8 $\overline{x}_1+\overline{x}_2+x_3+\overline{x}_4$	9 $\overline{x}_1+\overline{x}_2+\overline{x}_3+\overline{x}_4$
1. x_1+x_2			
2. $x_2+x_3+\overline{x}_4$			
3. $x_1+\overline{x}_3+\overline{x}_4$			
4. $\overline{x}_2+\overline{x}_3+\overline{x}_4$			X
5. $\overline{x}_1+x_3+\overline{x}_4$		X	
6. $\overline{x}_1+\overline{x}_2+x_3$	X	X	
7. $\overline{x}_1+\overline{x}_2+\overline{x}_4$		X	X

Математика – це один із видів мистецтв.

Н. Вінер

Лекція 18
ОДЕРЖАННЯ МІНІМАЛЬНИХ КНФ ЗА ДОПОМОГОЮ ДНФ

Логічні функції, подані у ДНФ, після мінімізації можна перетворити на КНФ, використовуючи доведені нижче властивості логічних функцій.

Припустимо, що дана логічна функція $F = F(x_1, x_2, ..., x_n)$, набуває на наборах $j_1, j_2, ..., j_m$ значення одиниці, а на наборах $i_1, i_2, ..., i_p$, де $p = 2^n - m$ значення нуля. Тоді має місце така теорема.

Теорема 1. ***Диз`юнкція всіх конституент одиниці $K_{i1} \vee K_{i2} \vee \vee ... \vee K_{ip}$, що не належать до ДДНФ логічної функції F, є запереченням цієї функції:***

$$K_{i1} \vee K_{i2} \vee ... \vee K_{ip} = \overline{F}.$$

Доведення. Сума всіх конституент 1 для n змінних створює функцію константу 1:

$$K_0 \vee K_1 \vee ... \vee K_{2^n - 1} = 1,$$

тобто функцію, яка дорівнює одиниці на всіх наборах значень змінних. Серед цих конституент виокремимо конституенти K_{j1}, K_{j2}, ..., K_{jm}, що утворюють функцію $F = K_{j1} \vee K_{j2} \vee ... \vee K_{jm}$. Тоді має місце співвідношення

$$F \vee K_{i1} \vee K_{i2} \vee ... \vee K_{ip} = 1.$$

З тотожності $x \vee \overline{x} = 1$ випливає, що і $F \vee \overline{F} = 1$, відповідно $\overline{F} = K_{i1} \vee K_{i2} \vee ... \vee K_{ip}$.

Теорему доведено.

Теорема 2. ***КНФ логічної функції F, одержана з мінімальної ДНФ функції $\overline{F}$ після її перетворення за допомогою формул де Моргана, також буде мінімальною.***

Доведення. Правило де Моргана не змінює кількість літер у логічному виразі, який є результатом перетворення, порівняно з початковим, що перетворюється.

Тому, якщо мінімальна ДНФ функції $\overline{F}$ містить K літер, то й одержана після перетворення за правилом де Моргана КНФ логічної функції F буде містити ту ж саму кількість літер K. Це число є мінімальним. Якщо це не так і існує інша КНФ логічної функції F з меншою кількістю літер, то, перетворивши її з КНФ на ДНФ за правилами де Моргана, одержимо ДНФ функції $\overline{F}$ з меншою кількістю літер, ніж це було визначено раніше у вихідній мінімальній. Унаслідок цього приходимо до суперечності.

Теорему доведено.

Виходячи з доведених теорем 1 і 2, запишемо алгоритм одержання мінімальної КНФ логічної функції F на основі мінімізації функції $\overline{F}$ у ДНФ.

1. Записують диз`юнкцію всіх конституент одиниці, які не належать до ДДНФ функції F.

2. Знаходять мінімальну ДНФ за відомим алгоритмом для функції $\overline{F}$.

3. Беруть заперечення від функції $\overline{F}$ і перетворюють її за правилами де Моргана в КНФ. Одержана КНФ буде мінімальною КНФ функції F.

Приклад 1. Знайти мінімальну КНФ логічної функції

$$F = F(A, B, C) = A\overline{C} \vee \overline{A}\overline{B}C.$$

Розв`язання.

1. Знайдемо ДДНФ функції F:

$$F = AB\overline{C} \vee A\overline{B}\overline{C} \vee \overline{A}\overline{B}C.$$

2. Одержимо ДДНФ функції $\overline{F}$:

$$\overline{F} = \overline{A}\overline{B}\overline{C} \vee \overline{A}B\overline{C} \vee \overline{A}BC \vee A\overline{B}C \vee ABC.$$

3. Знайдемо скорочену ДНФ функції $\overline{F}$, виконавши всі операції неповного склеювання й поглинання

$$\overline{F} = \overline{A}\overline{C} \vee \overline{A}B \vee BC \vee AC.$$

4. Складемо імплікантну матрицю для функції $\overline{F}$ (див. табл. 2.28).

Таблиця 2.28. Імплікантна матриця для приклада 1

Імпліканта	$\overline{A}\overline{B}\overline{C}$	$\overline{A}B\overline{C}$	$\overline{A}BC$	$A\overline{B}C$	ABC
1. $\overline{A}\overline{C}$	×	×			
2. $\overline{A}B$			×		
3. BC			×		
4. AC		×	×		×
				×	×

5. Знайдемо мінімальні форми функції $\overline{F}$ в ДНФ

$$\overline{F} = \overline{A}\overline{C} \vee BC \vee AC;$$
$$\overline{F} = \overline{A}\overline{C} \vee \overline{A}B \vee AC.$$

6. Узявши заперечення від обох частин останніх рівностей і застосовуючи формули де Моргана, одержимо дві мінімальні кон`юнктивні нормальні форми

$$F = (A \vee C)(\overline{B} \vee \overline{C})(\overline{A} \vee \overline{C});$$
$$F = (A \vee C)(A \vee \overline{B})(\overline{A} \vee \overline{C}).$$

Якщо Ви бажаєте навчитися плавати, то сміливо заходьте у воду, а якщо хочете навчитися розв`язувати задачі, то розв`язуйте їх!

Д. Пойа

Лекція 19
ТАБЛИЦІ ВЕЙЧА

Існують універсальні методи мінімізації логічних функцій, наприклад, розглянутий вище метод Квайна, який придатний для будь-якого числа змінних n. Він зручний для застосування на універсальних обчислювальних машинах в алгоритмічній формі в разі великої кількості змінних. Однак на практиці при проектуванні цифрових схем часто ставиться завдання мінімізації функцій для невеликого їх числа. З цією метою були розроблені методи мінімізації логічних функцій у наочній формі у вигляді спеціальних таблиць, що називаються також картами або діаграмами.

Розглянемо один з таких методів, яким будемо користуватися для мінімізації логічних функцій $f = f(x_1, x_2, ..., x_n)$, що містять не більше п`яти змінних, $n \le 5$, – ***метод таблиць Вейча***. Мінімізація в цьому випадку здійснюється для функції, записаної в аналітичній формі, – ДДНФ або ДКНФ.

Таблиця Вейча являє собою прямокутник, що вміщує 2^n клітинок, до яких заносяться одиниці при мінімізації логічних функцій, які задані в ДДНФ, або нулі у випадку мінімізації логічних функцій, поданих у ДКНФ. Якщо функція записана в ДНФ або КНФ, то її слід попередньо перетворити на ДДНФ або ДКНФ.

Щоб задати логічну функцію $f = f(x_1, x_2, ..., x_n)$, яка задана в ДДНФ, у вигляді таблиці Вейча, слід внести одиниці до клітинок, що відповідають конституентам одиниці, а решту клітинок залишити порожніми. Оскільки кількість конституент одиниці дорівнює числу клітинок у таблиці – 2^n, то будь-яку функцію від n змінних, число яких менше 5, подану в ДДНФ, можна навести у таблиці Вейча.

З цією метою спочатку треба виконати її розмітку. Для цього необхідно розмістити змінні x_i, $i = 1, 2, ..., n$, і їх заперечення $\overline{x}_i$ з боків прямокутника – зверху, знизу, зліва, справа, до того ж таким чином, щоб змінна без заперечення і ця сама змінна із запереченням порізно покривали 2^{n-1} різних клітинок кожна, а разом відповідно – 2^n. Це можливе в тому випадку, коли змінна x_i і її заперечення $\overline{x}_i$ знаходяться з одного боку таблиці. За такого розміщення дві різні змінні із запереченням чи без покривають сумісно 2^{n-2} клітинки, три – 2^{n-3} і т. д., поки n змінними не буде покриватися всього одна клітинка.

Такі покриття відповідають логічним добуткам n змінних – конституентам одиниці. Важливою властивістю цих конституент є те, що ті з них, які належать до сусідніх клітинок і до клітинок, що знаходяться з краю одних і тих же рядків і стовпців таблиці, відрізняються знаком заперечення лише в одній зі змінних. Це дозволяє здійснювати мінімізацію функції безпосередньо за таблицею в наочній формі.

Для цього в клітинках таблиці, що відповідають конституентам одиниці логічної функції, яка мінімізується, проставляються одиниці. Нижче в табл. 2.29 розглядається таблиця Вейча з трьома змінними і вісьма клітинками відповідно.

Якщо одиниці розташовані в сусідніх двох клітинках, наприклад, таких, що відповідають добуткам xyz і $x\overline{y}z$, то внаслідок того, що вони відрізняються знаком заперечення лише в одній змінній (у даному випадку y), відбувається склеювання за цією змінною. Результат склеювання для цього випадку $xyz + x\overline{y}z = xz(y + \overline{y}) = xz$. Дві змінні x і z у цьому випадку покривають сумісно разом дві клітинки.

Таблиця 2.29. Розміщення добутків змінних та їх заперечень у клітинках таблиці Вейча

	x		$\overline{x}$	
y	$xy\overline{z}$	xyz	$\overline{x}yz$	$\overline{x}y\overline{z}$
$\overline{y}$	$x\overline{y}\overline{z}$	$x\overline{y}z$	$\overline{x}\,\overline{y}\,z$	$\overline{x}\,\overline{y}\,\overline{z}$
	$\overline{z}$	z		$\overline{z}$

Якщо одиниці розташовані в чотирьох сусідніх клітинках, то це означає, що відбувається склеювання чотирьох сусідніх конституент за стовпцями й рядками. У результаті буде одержана одна змінна, яка покриє всі чотири сусідні клітинки.

Якщо в таблиці Вейча $n = 4$ і відповідно її покривають 4 змінні, вона повинна мати 16 клітинок. Процедура об'єднання клітинок у ній буде відбуватися аналогічно тому, як і в таблиці з вісьма клітинками, з тією відміною, що чотири змінні сумісно покривають одну клітинку, три – дві, дві – чотири і одна – вісім клітинок. Для таблиці з 32 клітинками п'ять змінних разом сумісно покривають одну клітинку, чотири – дві, три – чотири, дві – вісім і одна – шістнадцять клітинок.

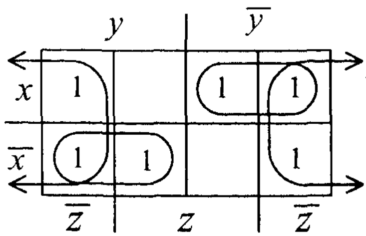
Проілюструємо викладений матеріал з допомогою прикладів.

Приклад 1. Знайти мінімальну ДНФ функції

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

Розв`язання. У табл. 2.30 чотири одиниці, що знаходяться у двох клітинках першого і останнього стовпців, накриваються змінною $\bar{z}$, а одиниці, які залишились, об'єднуються по дві в нижньому і верхньому рядках таблиці.

Таблиця 2.30 Об'єднання одиниць для приклада 1



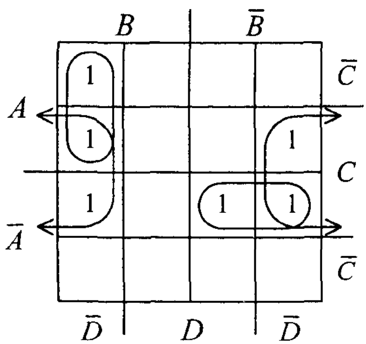
У результаті одержимо, що $f(x,\ y,\ z) = \bar{z} + x\bar{y} + \bar{x}y.$

Приклад 2. Знайти мінімальну ДНФ функції

$$f(A,\ B,\ C,\ D) = AB\bar{C}\bar{D} + ABC\bar{D} + \bar{A}BC\bar{D} + \bar{A}\bar{B}CD + A\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D}.$$

Розв'язання. У табл. 2.31 наведений спосіб найбільш раціонального об'єднання одиниць.

Таблиця 2.31. Об'єднання одиниць для приклада 2



При цьому мінімальна диз`юнктивна нормальна форма функції f матиме такий вигляд:

$$f(A,\ B,\ C,\ D) = C\overline{D} + AB\overline{D} + \overline{A}\overline{B}C.$$

Для мінімізації логічної функції $f = f\ (x_1,\ x_2,\ ...,\ x_n)$, поданої в ДКНФ, таблицю Вейча заповнюють не одиницями, а нулями. Вони заносяться в клітинки, що відповідають логічним сумам, на яких функція f дорівнює нулю, – конституентам нуля. У всьому іншому процедура мінімізації відбувається за аналогією з вищенаведеною для логічних функцій, поданих у ДНФ.

Приклад 3. Знайти мінімальні КНФ функції

$$f(A,\ B,\ C) = (A + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + B + C).$$

Розв'язання. Спочатку побудуємо для мінімізації наданої функції таблицю Вейча і розставимо в ній нулі відповідно до конституент нуля (див. табл. 2.32). Для одержання мінімальної кон'юнктивної нормальної форми цієї функції слід об`єднати нулі, які стоять поряд у табл. 2.32. Два окреслених суцільною лінією нулі покриваються логічною сумою $A + \overline{B}$, а нуль, який залишився, – конституентою нуля $\overline{A} + B + C$.

Таблиця 2.32. Об'єднання нулів для прикладу 3

	B		$\overline{B}$	
A			0	0
$\overline{A}$		0		
	$\overline{C}$	C		$\overline{C}$

Мінімальна кон`юнктивна нормальна форма вихідної функції, як це випливає з табл. 2.32, матиме вигляд

$$f(A,\ B,\ C) = (A + \overline{B})(\overline{A} + B + C).$$

Вихідну функцію приклада 3 можна мінімізувати й на основі ДДНФ, якщо для цього використати конституенти одиниці, які до неї належать. Очевидно, що їх буде п'ять.

Приклад 4. Знайти мінімальну ДНФ для функції $f(A, B, C)$, яка надана в прикладі 3.

Розв'язання. Для цього подамо вихідну функцію приклада 3 в ДДНФ з допомогою конституент одиниці

$$f(A,B,C) = AB\overline{C} + ABC + A\overline{B}C + \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C.$$

Мінімальну форму цієї функції знайдемо з допомогою табл. 2.33 і 2.34 у двох варіантах.

Таблиця 2.33. Об'єднання одиниць для приклада 4 (перший варіант)

	B		$\overline{B}$	
A	1	1	1	
$\overline{A}$			1	1
	$\overline{C}$	C	C	$\overline{C}$

Таблиця 2.34. Об'єднання одиниць для приклада 4 (другий варіант)

	B		$\overline{B}$	
A	1	1	1	
$\overline{A}$			1	1
	$\overline{C}$	C	C	$\overline{C}$

Цим таблицям відповідають дві мінімальні ДНФ функції

$$f(A,\ B,\ C) = AB + AC + \overline{A}\overline{B} \ \text{ і } \ f(A,\ B,\ C) = AB + \overline{B}C + \overline{A}\overline{B}.$$

Звернемо при цьому увагу на те, що мінімальна КНФ, яка була отримана раніше, містить менше літер, ніж кожна з цих двох.

Іноді зручно поєднувати в одній таблиці Вейча мінімізацію в ДНФ і мінімізацію в КНФ, наприклад, на основі табл. 2.33. Тоді при мінімізації в КНФ нулі записуються до порожніх клітинок цієї таблиці. Результатом цих дій є табл. 2.35. Результат мінімізації отримаємо в кон'юктивній формі як добуток диз'юнкцій інверсних значень змінних, що покривають клітинки з нулями. Він виходить з правила де Моргана, відповідно до якого $A+\bar{B}+C=\overline{\bar{A}B\bar{C}}$, $A+\bar{B}+\bar{C}=\overline{\bar{A}BC}$, $\bar{A}+B+C=\overline{A\bar{B}\bar{C}}$.

Таблиця 2.35. Об'єднана мінімізація за нулями та одиницями

	B		$\bar{B}$	
A	1	1	1	0
$\bar{A}$	0	0	1	1
	$\bar{C}$	C		$\bar{C}$

Відповідно до результуючої мінімальної функції записуються інверсні значення змінних, що покривають нулі

$$f(A,B,C)=(A+\bar{B})(\bar{A}+B+C).$$

Вона має такий самий вигляд, як і для табл. 2.32. Знайдіть самостійно відповідь на питання: "Чому?".

Спалити – не означає довести.

Д.Бруно

Лекція 20
МІНІМІЗАЦІЯ НЕ ПОВНІСТЮ ВИЗНАЧЕНИХ ЛОГІЧНИХ ФУНКЦІЙ

Означення 1. Логічна функція f, яка визначена *на всіх* наборах значень змінних, називається ***повністю*** визначеною.

Означення 2. Логічна функція f, яка визначена *не на всіх* наборах значень змінних, називається ***не повністю***, або ***частково***, визначеною.

Припустимо, що є не повністю визначена логічна функція f, яка не визначена на $p < 2^n$ наборах змінних $z_1, z_2, ..., z_n$. Тоді її можна доповнити 2^p способами до повністю визначеної логічної функції

$$\varphi = \varphi(z_1, z_2, ..., z_n).$$

Означення 3. Логічна функція $\varphi = \varphi(z_1, z_2, ..., z_n)$, значення якої збігаються зі значеннями функції $f = f(z_1, z_2, ..., z_n)$ на всіх наборах значень, на яких остання визначена, називається ***еквівалентною*** функції f.

Серед всіх 2^p функцій φ, еквівалентних f, є, очевидно, одна або декілька таких, що містять мінімальну кількість літер. Розглянемо їх пошук на прикладі.

Приклад 1. Знайти мінімальну диз`юнктивну нормальну форму логічної функції

$$f = f(A,B,C,D) = \bar{A}B\bar{C}\bar{D} \vee \bar{A}\bar{B}\bar{C}\bar{D} \vee AB\bar{C}\bar{D}.$$

При цьому задано, що функція не визначена на 4 наборах: 1110, 1011, 0011 і 0010. Цим наборам відповідають добутки змінних: $ABC\bar{D}$, $A\bar{B}CD$, $\bar{A}\bar{B}CD$ і $\bar{A}\bar{B}C\bar{D}$, які через невизначеність функції f на цих 5 наборах можуть дорівнювати на них як 1, так і 0.

Наведемо в табл. 2.36 діаграму Вейча для цієї функції.

Порожні $p = 4$ клітинки відповідають добуткам змінних, які можуть приймати значення як 1, так і 0. У цих клітинках одиниці й нулі можуть бути розміщені $2^p = 2^4 = 16$ способами.

Таблиця 2.36. Неповністю визначена функція

	B	B	$\bar{B}$	$\bar{B}$	
A	1	0	0	0	$\bar{C}$
A	0	0		0	C
$\bar{A}$		0			C
$\bar{A}$	1	0	0	1	$\bar{C}$
	$\bar{D}$	D	D	$\bar{D}$	

Виберемо такий розподіл одиниць і нулів, який мінімізує функцію f (див. табл. 2.37).

Таблиця 2.37. Доповнення функції одиницями

	B	B	$\bar{B}$	$\bar{B}$	
A	1	0	0	0	$\bar{C}$
A	0	0	0	0	C
$\bar{A}$	1	0	0	1	C
$\bar{A}$	1	0	0	1	$\bar{C}$
	$\bar{D}$	D	D	$\bar{D}$	

Результуюча функція матиме вигляд:

$$f = f(A,B,C,D) = \bar{A}\bar{D} \vee B\bar{C}\bar{D}.$$

Доповнимо тепер порожні клітинки лише нулями (див. табл. 2.38)

Таблиця 2.38. Доповнення функції нулями

	B	B	$\overline{B}$	$\overline{B}$	
A	1	0	0	0	$\overline{C}$
A	0	0	0	0	C
$\overline{A}$	0	0	0	0	C
$\overline{A}$	1	0	0	1	$\overline{C}$
	$\overline{D}$	D	D	$\overline{D}$	

У результаті мінімальна кон`юнктивна нормальна форма функції f матиме такий вигляд:

$$f = f(A,B,C,D) = \overline{C}\overline{D}(\overline{A} \vee B).$$

Як випливає з вищенаведеного, не повністю визначені функції дають можливість додатково зменшити кількість літер при мінімізації логічних функцій, а отже, дозволяють синтезувати більш економічні цифрові схеми. Тому синтез цифрових схем слід виконувати з урахуванням можливої їх невизначеності, що дозволить знаходити більш мінімальні схеми, ніж без такого урахування.

Фрази треба скорочувати до розмірів думки.

Із заповіту Прокруста

КОНТРОЛЬНІ ЗАВДАННЯ І ПИТАННЯ ДО ЧАСТИНИ II

Питання для самоконтролю

1. Що вам відомо про алгебру логіки і обчислення висловлень?

2. Що таке істинне і хибне, просте і складне висловлення? Наведіть приклади.

3. Дайте характеристику основних логічних операцій: константа нуля, константа одиниці, операції "НІ", "І", "АБО".

4. Дайте характеристику логічних операцій: "імплікація", "заборона", "рівнозначність", "нерівнозначність". Наведіть приклади.

5. Дайте характеристику логічних операцій: Шеффера, Пірса, "змінна". Наведіть приклади.

6. Що вам відомо про логічний закон, логічне протиріччя й твердження, яке логічно виконується? Наведіть приклади.

7. Які є основні закони алгебри логіки? Наведіть приклади.

8. Дайте визначення набору логічної функції. Що таке таблиця істинності? Як визначити кількість наборів від n аргументів функції?

9. Дайте визначення логічної функції та її аргументів. Як визначити кількість функцій від n аргументів? Доведіть відповідну формулу.

10. Які існують логічні функції двох аргументів? Визначить їх кількість і дайте назву. Наведіть таблиці істинності.

11. Як здійснюються доведення логічних тверджень і законів за допомогою таблиць істинності? Наведіть приклади.

12. Наведіть приклади доведення за допомогою таблиць істинності правила де Моргана, "імплікації", "рівнозначності".

13. Наведіть і доведіть співвідношення між двома аргументами, один із яких приймає значення 1 або 0.

14. Наведіть і доведіть співвідношення між двома аргументами x_1, x_2, коли $x_1 = x_2 = x$ і $x_1 = x$, а $x_2 = \overline{x}$.

15. Що вам відомо про булеву алгебру? Наведіть основні співвідношення в булевій алгебрі для операцій диз'юнкції та кон'юнкції.

16. Що таке спеціальні рівності булевої алгебри? Доведіть їх на прикладах.

17. Що вам відомо про диз'юнктивну нормальну форму (ДНФ)? Що таке елементарний добуток? Конституента одиниці? Наведіть основні теореми для конституенти одиниці та її наслідків.

18. Дайте визначення імпліканти і простої імпліканти. Наведіть приклади.

19. Що таке досконала диз'юнктивна нормальна форма (ДДНФ)? Який алгоритм одержання конституенти одиниці та ДДНФ? Наведіть приклади.

20. Дайте визначення скороченої ДНФ. Наведіть приклади.

21. Охарактеризуйте операції повного і неповного склеювання в ДНФ, операції поглинання й розгорнення. Наведіть приклади.

22. Сформулюйте теорему Квайна для ДДНФ. Доведіть її.

23. Що таке алгоритм мінімізації ДДНФ за Квайном? Наведіть приклади.

24. Що таке імплікантні матриці і як з їх допомогою можна одержати мінімальну ДНФ? Наведіть приклади.

25. Дайте визначення тупикових і мінімальних ДНФ. Наведіть приклади.

26. Дайте характеристику поняття про кон'юнктивну нормальну форму (КНФ). Що таке елементарна сума і конституента нуля? Наведіть основні теореми для КНФ.

27. Що таке досконала кон'юнктивна нормальна форма (ДКНФ)? Який існує алгоритм одержання конституенти нуля і ДКНФ? Наведіть приклади.

28. Дайте характеристику імпліценти і простої імпліценти. Наведіть приклади.

29. Дайте характеристику скороченої КНФ. Наведіть приклади.

30. У чому полягають операції повного і неповного склеювання в КНФ, операції поглинання і розгорнення? Наведіть приклади.

31. Сформулюйте теорему Квайна для ДКНФ. Доведіть її.

32. Дайте алгоритм мінімізації ДКНФ за Квайном. Наведіть приклади.

33. Що таке імпліцентна матриця для одержання мінімальної КНФ? Наведіть приклади.

34. Що таке тупикові й мінімальні КНФ? Наведіть приклади.

35. Як одержати мінімальні КНФ за допомогою диз'юнктивних форм? Наведіть приклади.

36. Наведіть алгоритм мінімізації булевих функцій, поданих в ДДНФ за допомогою карт Вейча. Приклади.

37. Як проводиться мінімізації булевих функцій, поданих в ДКНФ за допомогою карт Вейча. Наведіть приклади.

38. Як виконується мінімізація неповністю означених булевих функцій? Наведіть приклади.

Контрольні завдання[1]

1. Для функції, яка наведена в колонці 1 таблиці варіантів 2.39, задані номери наборів, на яких вона дорівнює одиниці. Необхідно мінімізувати цю функцію двома методами – Квайна і таблиць Вейча. Мінімізацію провести в ДНФ і КНФ. Результати порівняти.

2. Мінімізувати в ДНФ і КНФ за допомогою таблиць Вейча не повністю означену функцію, наведену в колонці 2 таблиці варіантів 2.39, оптимально її доповнивши.

3. Провести необхідні логічні перетворення функції *f*, яка задана в колонці 3 таблиці варіантів 2.39, і одержати ДДНФ і ДКНФ цієї функції. Провести її мінімізацію з допомогою таблиць Вейча.

[1]Контрольні завдання до частини I склала старший викладач кафедри електроніки і комп'ютерної техніки Сумського державного університету Протасова Т.О.

Таблиця 2.39. Варіанти завдань контрольних робіт до частини 3: Елементи математичної логіки

Варіант	1	2	3
1	3, 6, 7, 9, 10, 11, 15	111*00**01111010	$f = (A \sim B) \vee C \to A$
2	2, 3, 5, 7, 11, 14, 15	1011*1000110**01	$f = A \sim B \vee C \to B$
3	0, 1, 4, 10, 11, 12, 15	11000111*00011*0	$f = A \sim B \to C \vee A$
4	2, 3, 6, 8, 9, 13, 14	*101*011*0001001	$f = (A \sim B) \to C \vee B$
5	2, 3, 5, 7, 10, 11, 15	00100011**010011	$f = A \sim (B \mid C) \to B$
6	0, 3, 5, 7, 10, 14, 15	11*1000111*00011	$f = (A \to B) \wedge C \sim A$
7	0, 2, 5, 8, 13, 14, 15	*0111**001*0011*	$f = (A \sim B) \mid C \vee A$
8	0, 3, 5, 7, 8, 11, 12	0*01101100011**0	$f = (A \to B) \to C \vee A$
9	1, 2, 3, 7, 11, 12, 15	1*0001101111000*	$f = (A \to B) \sim C \mid B$
10	0, 1, 2, 3, 6, 8, 15	10011010111**000	$f = A \sim B \wedge C \mid A$
11	2, 3, 5, 6, 10, 12, 13	*101000001110*11	$f = A \to B \sim C \vee \bar{A}$
12	0, 1, 3, 5, 6, 7, 11	0110111*10000111	$f = (\bar{A} \sim B) \wedge (A \sim \bar{C})$
13	1, 2, 3, 4, 6, 7, 11	11100110100011**	$f = (C \vee \bar{A}) \to (A \sim B)$
14	0, 2, 3, 5, 6, 7, 11	**100011*0001101	$f = A \sim B \to C \vee \bar{A}$
15	1, 2, 3, 4, 6, 7, 15	10*1011*01110011	$f = A \sim C \vee B \mid A$

Продовження таблиці 2.39.

Варіант	1	2	3
16	0, 2, 3, 5, 6, 7, 15	010**10011110011	$f = A \sim C \mid B \to A$
17	2, 3, 4, 7, 8, 9, 12	0010**01*1011011	$f = A \sim (C \mid B \vee \overline{C})$
18	1, 4, 5, 9, 10, 11, 13	00110110000111**	$f = (A \mid B) \mid (A \mid C)$
19	0, 3, 6, 7, 8, 10, 12	1000110011001**0	$f = (A \to B) \sim C \wedge \overline{B}$
20	6, 9, 10, 11, 12, 14, 15	*1100110*111000*	$f = C \vee \overline{A} \to B \sim C$
21	7, 8, 10, 11, 13, 14, 15	000011011**00*10	$f = A \to B \sim (C \mid A)$
22	2, 7, 9, 10, 11, 14, 15	001110001*011*01	$f = (A \sim B) \wedge C \to \overline{B}$
23	3, 6, 10, 11, 13, 14, 15	001*0101110*1100	$f = (A \sim B) \vee \overline{A} \to C$
24	0, 1, 2, 3, 4, 7, 8	01*100110001110*	$f = A \to B \wedge C \sim A$
25	1, 3, 6, 7, 10, 11, 15	10111001**000110	$f = A \to B \mid (C \to A)$
26	1, 2, 3, 5, 8, 9, 11	1110001100**1010	$f = (A \sim B) \wedge (\overline{A} \to C)$
27	0, 1, 3, 9, 10, 11, 15	1000100011110**0	$f = (A \sim B) \sim C \vee \overline{A}$
28	3, 4, 5, 7, 10, 13, 15	01011*100011*011	$f = C \vee B \sim A \vee \overline{B}$
29	4, 5, 6, 11, 12, 13, 15	10110*100110011*	$f = A \to B \sim C \wedge \overline{B}$
30	1, 5, 6, 7, 11, 13, 15	111000**1100011*	$f = B \to C \sim A \vee \overline{B}$

Частина III

ЕЛЕМЕНТИ КОМБІНАТОРИКИ

Лекція 21
ЗАГАЛЬНА ХАРАКТЕРИСТИКА КОМБІНАТОРНИХ ЗАДАЧ

1. Основні положення

Під ***комбінаторикою*** звичайно розуміють розділ дискретної математики, присвячений розв'язанню задач про вибір та розміщення елементів скінченної множини згідно із заданими *правилами*. У результаті створюються необхідні комбінаторні *об'єкти* чи *конфігурації*. Характерними властивостями цих об'єктів є те, що вони відповідають деяким обмеженням щодо них, і тому завжди можна розпізнати дозволений комбінаторний об'єкт, який відповідає правилам його побудови, і недозволений, який не відповідає цим правилам.

З комбінаторикою мають справу хіміки при вивченні різних можливих типів зв'язків атомів у молекулах; біологи, наприклад, у процесі знаходження послідовностей амінокислот у білкових сполуках; кібернетики при розв'язанні задач кодування й побудові обчислювальних пристроїв, математики – при розв'язанні багатьох різних задач, особливо в теорії ймовірності. Також комбінаторику використовують у своїх моделях фізики, архітектори, економісти й представники багатьох інших наук.

2. Комбінаторні задачі

У комбінаториці є декілька задач, які вирішуються послідовно одна за одною. *Перша* з них спочатку формулює вимоги до класу комбінаторних конфігурацій, які потрібно побудувати. Доводиться, що хоча б одна така конфігурація існує, незважаючи на те, що побудувати таку конфігурацію може бути досить непросто. Тому інколи буває достатньо теоретичного доведення її існування.

Після розв'язання першої задачі комбінаторики розв'язується не менш важлива *друга* – задача *переліку* комбінаторних об'єктів, які відповідають вихідним правилам їх побудови. Саме на розв'язання цієї задачі спрямовані сьогодні зусилля багатьох учених. Є досить багато задач, які так чи інакше стосуються цієї загальної задачі. Наприклад, до неї належить питання про кількість різних способів, якими можна розмістити групу студентів з 30 чоловік на 30 чи більше місцях, або про кількість способів проведення матчів з футболу між 10 різними командами?

Далі на основі отриманих розв'язків конкретних задач з переліку комбінаторних об'єктів розв'язується *третя* задача комбінаторики – це її побудова. Наприклад, потрібно не лише підрахувати кількість можливих варіантів розподілу 30 студентів на 30 місцях, а й побудувати всі ці розподіли або деякі з них у вигляді їх комбінаторних конфігурацій. Також може виникнути потреба побудувати таблицю матчів між 10 футбольними командами, а не тільки знати їх кількість.

Четверта і остання задача комбінаторики – це задача про пошук серед комбінаторних конфігурацій такої, яка б приводила деяку функцію до *оптимуму*. Це на сьогодні досить нелегка для розв'язання загальна задача. Вона містить задачі *комбінаторної оптимізації*, наприклад, задачу комівояжера, яка на сьогодні ще не має остаточного розв'язання, хоча на її грунті сформульований клас складних переборних задач.

3. Правило Суми

В основі розв'язання багатьох задач комбінаторики лежать два простих правила – *правило Суми* та *правило Добутку*.

Правило Суми стверджує, що якщо є можливість вибрати елемент з деякої множини елементів A m способами, а елемент з множини B, яка не має спільних елементів з множиною A, – k способами, то вибрати елемент множини A або елемент множини B можна $m + k$ способами.

Це правило зручно продемонструвати з допомогою такої моделі. Якщо маємо дві урни і в одній з них знаходиться m куль, а в іншій k, то кількість способів, якими можна буде вийняти кулю з тієї чи іншої урни, дорівнюватиметься $m + k$. Дійсно, з першої урни кулю можна вийняти m способами, але якщо з першої урни кулю не виймати, то тоді з другої урни її можна вийняти k способами. Тому загальна кількість способів, якими можна вийняти *одну* кулю з *двох* урн, буде дорівнювати $m + k$.

У загальному випадку правило Суми може бути сформульоване таким чином.

Якщо треба виконати якусь дію n_1, n_2, або n_k способами, то кількість можливих способів реалізації цієї дії буде дорівнювати

$$N = n_1 + n_2 + ... + n_k.$$

Особливістю цього правила є те, що воно використовує сполучник *або*, який протиставляє різні дії одна одній.

Приклад 1. На денне чергування в студентському гуртожитку може піти або студент з кімнати 1, де проживають три студенти, або студент з кімнати 2, де проживають чотири студенти.

Скількома способами можна вибрати одного студента на денне чергування в гуртожитку?

Розв'язання. Загальна кількість способів, якими можна вибрати одного студента або з кімнати 1 або з кімнати 2 на денне чергування, згідно з правилом Суми буде $3 + 4 = 7$.

4. Правило Добутку

Правило Добутку використовується тоді, коли кожний елемент множини A може бути вибраний разом з елементом множини B. Відповідно до кожного способу вибору елемента множини A буде зіставлятися k способів вибору елемента множини B. Тоді загальна кількість способів сумісного вибору елементів множини A з елементами множини B, очевидно, дорівнюватиме $m \cdot k$.

Модель урн можна застосувати і для ілюстрації правила Добутку. У цьому випадку розглядаються дві урни, у першій з яких знаходиться m куль, а в другій k. Будемо вважати, що будь-якій кулі першої урни може відповідати будь-яка куля з другої урни. А оскільки в першій урні знаходиться m куль, то й кількість способів вибору куль з першої урни разом з різними кулями з другої урни буде дорівнювати $m \cdot k$.

У загальному вигляді правило Добутку буде мати такий вигляд.

Якщо треба виконати якусь дію, що може бути виконана k сумісними діями, перша з яких може бути виконана n_1 способами, друга – n_2 і т.д. до k – ої дії, яку можна виконати n_k способами, то основна дія може бути виконана M способами, де

$$M = n_1 \cdot n_2 \cdot \ldots \cdot n_k.$$

У цьому правилі важливу роль відіграє сполучник *і*, який об'єднує різні дії в одну.

Приклад 2. На денне чергування в студентському гуртожитку вибирається два студента – один студент з трьох, що проживають у кімнаті 1, і один студент з чотирьох, які проживають у кімнаті 2. Скільки існує можливих способів формування різних пар з двох студентів для чергування?

Розв'язання. Кількість способів чергувань двох студентів з різних кімнат відповідно до правила Добутку буде дорівнювати 12.

Існує вражаюча можливість оволодіти
предметом математично, не зрозумівши суті справи.

А. Ейнштейн

Лекція 22
ЕЛЕМЕНТАРНІ КОМБІНАТОРНІ КОНФІГУРАЦІЇ

1. Сполучення

Припустимо, що є множина, яка утворюється з n елементів. Кожна її підмножина з k елементів, має назву *сполучення* k елементів з n, де n, k – цілі додатні числа 0, 1, ...; $n \geq k$. Кількість усіх можливих сполучень k елементів з n називають *біноміальним* коефіцієнтом і позначають як C_n^k. Чому цей коефіцієнт одержав таку назву, буде з'ясовано пізніше.

Так, якщо елементами деякої множини будуть літери a, b, c, d, e, то з них можна отримати такі сполучення: abc, abd, abe, acd, ace, ade, bcd, bce, bde, cde, кількість яких, як це видно з їх переліку, $C_5^3 = 10$.

Різні сполучення відрізняються одне від одного лише складом елементів, які їх утворюють, а порядок їх розташування в сполученнях не відіграє ролі. Наприклад, послідовності abc, acb, bac, bca, cab, cba являють собою одне сполучення, що складається з 3 елементів a, b, c, порядок яких у ньому не важливий.

2. Розміщення

Якщо порядок у послідовностях з k різних елементів, що вибираються з n елементів, має значення, то кожна з таких послідовностей буде мати назву *розміщення* k елементів з n *без повторення*. Без повторення тому, що жоден елемент розміщення не може бути представлений у ньому більше ніж один разу. Якщо ж ця умова не виконується і елементи, що входять в розміщення, повторюються, то в такому разі отримують розміщення *з повтореннями*.

Наприклад, з $n = 3$ літер a, b, c можна отримати такі розміщення без повторення, кожне з яких складається з $k = 2$ елементів: ab, ba, ac, ca, bc, cb. Наведені вище шість послідовностей з трьох елементів a,b,c показують також розміщення без повторень.

3. Кількість розміщень

У багатьох комбінаторних задачах важливу роль виконує кількість розміщень. Тому виникає потреба в їх підрахунку.

Теорема 1. ***Кількість розміщень для k елементів з n, $n \geq k$,***

$$A_n^k = n(n-1)...(n-k+1).$$

Доведення. Перший елемент розміщення з n, можна обрати n способами, другий $n-1$ і т.д. до k-го елемента, який може бути обраний $n-k+1$ способами, оскільки в момент вибору k-го елемента залишилось $n-(k-1) = n-k+1$ елементів. У результаті кількість способів, якими можна обрати k елементів з n-елементної множини, а отже, і кількість розміщень

$$A_n^k = n(n-1)...(n-k+1).$$

Теорему доведено.

Розміщення, для якого $k = n$, має назву *перестановки.* Кількість перестановок

$$P_n = A_n^n = n! = n(n-1)...1.$$

Теорема 2.

$$A_n^k = P_k C_n^k = k!C_n^k.$$

Доведення. Кожному сполученню k елементів з n відповідає $P_k = k!$ перестановок, а всі можливі перестановки для всіх сполучень утворюють усі розміщення k елементів із n. Тому

$$A_n^k = P_k C_n^k = k!C_n^k.$$

Теорему доведено.

Приклад 1. Скільки шестизначних чисел, кратних п'яти, можна скласти із цифр 1, 2, 3, 4, 5 за умови, що в числі цифри не повторюються?

Розв'язання. Для того щоб шестизначне число, яке складається з наданих цифр, ділилось на 5, необхідно й достатньо, щоб цифра 5 в ньому стояла на останньому місці. Інші 5 цифр можуть стояти на місцях, які залишилися, у будь-якому порядку.

Отже, кількість шестизначних чисел, кратних 5, дорівнює кількості перестановок з 5 елементів, тобто

$$P_5 = 5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120\,.$$

Приклад 2. Студенти вивчають 14 предметів. Скільки існує способів, якими можна скласти розклад занять на суботу, якщо в цей день неділі повинно бути 5 різних занять?

Розв'язання. Різних способів складення розкладу, очевидно, буде стільки, скільки існує впорядкованих підмножин з п'яти елементів чотирнадцятиелементної множини. Кожна така упорядкована підмножина в цьому випадку буде зображати одне розміщення.

Отже, кількість способів складання розкладу занять дорівнює кількості розміщень 5 елементів з 14:

$$A_{14}^5 = 14 \cdot 13 \cdot 12 \cdot 11 \cdot 10 = 240240\,.$$

4. Кількість сполучень

Теорема 3. ***Кількість сполучень для k елементів із n***

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

Доведення. З теореми 2 випливає, що

$$C_n^k = \frac{A_n^k}{P_k} = \frac{A_n^k}{k!}\,,$$

а з теореми 1, що $A_n^k = n(n-1)...(n-k+1)$. Тому

$$C_n^k = \frac{n(n-1)...(n-k+1)}{k!} = \frac{n!}{k!(n-k)!}\,.$$

Теорему доведено.

Прийнято вважати, що 0! = 1, тому

$$C_n^0 = \frac{n!}{n!\cdot 0!} = \frac{n!}{n!\cdot 1} = 1,$$

$$C_0^0 = \frac{0!}{0!0!} = \frac{1}{1\cdot 1} = 1.$$

Приклад 3. Скільки комісій, які складаються із 7 членів, можна створити з 15 викладачів?

Розв'язання. Очевидно, стільки, скільки існує сполучень з 14 викладачів по 7

$$C_{14}^7 = \frac{14!}{7!7!} = \frac{14\cdot 13\cdot 12\cdot 11\cdot 10\cdot 9\cdot 8}{7\cdot 6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1} = 3432.$$

Математика – це наука про множини.

Імре Ружа

Лекція 23
ВЛАСТИВОСТІ БІНОМІАЛЬНИХ КОЕФІЦІЄНТІВ

1. Властивість симетрії

Теорема 1

$$C_n^k = C_n^{n-k}.$$

Доведення. Після того, як підставимо $n - k$ замість k у формулу, яка обчислює біноміальні коефіцієнти, одержимо

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

Теорему доведено.

Ця властивість відома як властивість *симетрії* біноміальних коефіцієнтів. Її особливість полягає в тому, що не має значення, як обирати елементи: з n по k або з n по $n - k$. У тому чи іншому випадку кількість підмножин буде однакова.

2. Властивості додавання

Теорема 2.

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

Доведення.

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

Теорему доведено.

Доведення даної теореми також можна отримати, якщо виходити з таких міркувань.

Число C_{n+1}^{k+1} визначає кількість тих підмножин з $k + 1$ елементів, які можуть бути обрані з множини, що містить $n + 1$ елементів.

Оберемо один з цих елементів, наприклад, x та розкладемо початкову (вихідну) множину з C_{n+1}^{k+1} підмножинами на два класи, один з яких містить у своїх підмножинах елемент x, а другий ні.

У першому класі елемент x присутній у всіх без винятку підмножинах. Це означає, що він просто приєднується до k елементів, які тепер обираються не з $n + 1$ елементів, як це було до розкладу на два класи, а з n (елемент x під час їх вибору відсутній). Тому кількість підмножин у першому класі дорівнює C_n^k.

Оскільки елемент x у другому класі відсутній, то підмножини, які до нього належать, не містять цього елементу, хоча й без нього до них належать $k + 1$ елементів, однак вони вибираються не з $n + 1$ елементів, а з n. Тому кількість підмножин у другому класі дорівнює C_n^{k+1}.

Відповідно сума підмножин першого та другого класу

$$C_n^k + C_n^{k+1} = C_{n+1}^{k+1},$$

що й доводить початкове твердження.

Теорема 3.

$$C_n^k = C_{n-1}^{k-1} + C_{n-2}^{k-1} + \ldots + C_{k-1}^{k-1} = \sum_{i=0}^{n-k} C_{k+i-1}^{k-1}.$$

Доведення. На основі теореми 2 запишемо рівності:

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1},\ C_{n-1}^k = C_{n-2}^k + C_{n-2}^{k-1},\ \ldots,\ C_{k+2}^k = C_{k+1}^k + C_{k+1}^{k-1},$$
$$C_{k+1}^k = C_k^k + C_k^{k-1}.$$

Оскільки $C_k^k = C_{k-1}^{k-1} = 1$, то, після підстановки виразу для C_{k+1}^k у вираз для C_{k+2}^k, отримаємо:

$$C_{k+2}^k = C_k^k + C_k^{k-1} + C_{k+1}^{k-1} = C_{k+1}^{k-1} + C_k^{k-1} + C_k^k = C_{k+1}^{k-1} + C_k^{k-1} + C_{k-1}^{k-1}.$$

Після того, як підставимо коефіцієнти для C_{k+2}^k у попередню йому рівність

$$C_{k+3}^{k} = C_{k+2}^{k} + C_{k+2}^{k-1},$$

одержимо, що

$$C_{k+3}^{k} = C_{k+2}^{k-1} + C_{k+1}^{k-1} + C_{k}^{k-1} + C_{k-1}^{k-1}.$$

Потім таку ж саму операцію виконаємо для решти коефіцієнтів безпосередньо до C_{n-1}^{k} і в результаті отримаємо необхідну рівність, яка доводить дану теорему.

Теорему доведено.

Теорема 4.

$$C_{n}^{k} = C_{n-1}^{k} + C_{n-2}^{k-1} + \ldots + C_{n-k+1}^{2} + C_{n-k}^{1} + C_{n-k-1}^{0} = \sum_{i=0}^{k} C_{n-k+i-1}^{i}.$$

Доведення. На основі рівності

$$C_{n}^{k} = C_{n-1}^{k} + C_{n-1}^{k-1} \text{ і } C_{n-k-1}^{0} = C_{n-k}^{0} = 1$$

маємо

$$C_{n-k}^{1} + C_{n-k}^{0} = C_{n-k+1}^{1},$$

потім

$$C_{n-k+1}^{2} + C_{n-k+1}^{1} = C_{n-k+2}^{2}$$

і так до одержання в кінцевому підсумку необхідного результату.

Теорему доведено.

3. Властивості винесення за дужки

Теорема 5. ***При k ≠ 0, n ≠ 0***

$$C_{n}^{k} = \frac{n}{k} C_{n-1}^{k-1}.$$

Доведення.

$$C_{n}^{k} = \frac{n!}{k!(n-k)!} = \frac{n}{k} \cdot \frac{(n-1)!}{(k-1)!(n-k)!}.$$

Оскільки відношення

$$\frac{(n-1)!}{(k-1)!(n-k)!} = C_{n-1}^{k-1},$$

то

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

Теорему доведено.

Наслідок 1.

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

Наслідок 2.

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

Теорема 6. ***При k ≠ 0, n ≠ 0***

$$\boldsymbol{C_n^k = \frac{n}{n-k} C_{n-1}^k}.$$

Доведення. Відповідно до теореми 5

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

Унаслідок теореми 1, згідно з якою

$$C_{n-1}^{n-k-1} = C_{n-1}^k,$$

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

Теорему доведено.

Лекція 24
ТРИКУТНИКИ ПАСКАЛЯ

1. Числові й символічні трикутники Паскаля

Теорема 2 з лекції 23 відома, як ***правило Паскаля***. Воно виражає згідно з правилами рекурсії один біноміальний коефіцієнт, який не дорівнює одиниці, через два інших. З допомогою цього правила можна побудувати трикутники Паскаля, як числовий (рис. 3.1а), так і символічний (рис. 3.1б).

При необмеженому продовженні трикутники Паскаля дозволяють знаходити біноміальні коефіцієнти з будь-яким n і k, що може в деяких випадках значно спростити процедуру їх обчислення.

					1							$n=0$
				1		1						$n=1$
			1		2		1					$n=2$
		1		3		3		1				$n=3$
	1		4		6		4		1			$n=4$
1		5		10		10		5		1		$n=5$

Рис. 3.1а. Числовий трикутник Паскаля

					C_0^0							$n=0$
				C_1^0		C_1^1						$n=1$
			C_2^0		C_2^1		C_2^2					$n=2$
		C_3^0		C_3^1		C_3^2		C_3^3				$n=3$
	C_4^0		C_4^1		C_4^2		C_4^3		C_4^4			$n=4$
C_5^0		C_5^1		C_5^2		C_5^3		C_5^4		C_5^5		$n=5$

Рис. 3.1б. Символічний трикутник Паскаля

2. Властивості трикутників Паскаля

У рядку n, $n = 0, 1, \ldots$, трикутника Паскаля стоять коефіцієнти C_n^k, до того ж кожен коефіцієнт, крім двох крайніх, які дорівнюють 1, дорівнює сумі двох коефіцієнтів з попереднього рядка, які стоять над ним (див. рис. 3.1а і 3.1б).

Сума біноміальних коефіцієнтів одного рядка дорівнює 2^n і, крім того, сума біноміальних коефіцієнтів, які стоять на непарних місцях, дорівнює їх сумі на парних. У свою чергу, кожна з цих сум дорівнює 2^{n-1}.

Числовий трикутник відповідно до властивості симетрії біноміальних коефіцієнтів строго симетричний відносно його середньої лінії. Тому при використанні його для обчислення біноміальних коефіцієнтів достатньо побудувати будь-яку одну його половину.

3. Таблична форма трикутників Паскаля

Іноді при обчисленні C_n^k більш зручно користуватися трикутником Паскаля, який поданий у табличній формі (див. табл. 3.1).

Таблиця 3.1. Трикутник Паскаля в табличній формі

N	Біноміальні коефіцієнти											
0	C_0^0						1					
1	C_1^0	C_1^1					1	1				
2	C_2^0	C_2^1	C_2^2				1	2	1			
3	C_3^0	C_3^1	C_3^2	C_3^3			1	3	3	1		
4	C_4^0	C_4^1	C_4^2	C_4^3	C_4^4		1	4	6	4	1	
5	C_5^0	C_5^1	C_5^2	C_5^3	C_5^4	C_5^5	1	5	10	10	5	1

Тут, як і раніше, усі коефіцієнти біноміального розкладу розташовані в одному рядку і в сумі утворюють величину 2^n.

У той же самий час правило Паскаля реалізується шляхом підсумовування двох біноміальних коефіцієнтів, які стоять поруч в одному рядку. Результуючий коефіцієнт при цьому знаходиться у сусідньому нижньому рядку під правим біноміальним коефіцієнтом, який підсумовується (див. табл. 3.1).

Наприклад,

C_3^1	C_3^2	
	C_4^2	

Це означає, що $C_4^2 = C_3^2 + C_3^1$ (правило Паскаля).

4. Знаходження біноміальних коефіцієнтів

Біноміальний коефіцієнт C_n^k можливо знайти в трикутнику Паскаля з допомогою теореми 4 (лекція 23). Для цього в трикутнику знаходять коефіцієнт C_n^k і потім підсумовують у верхньому рядку, паралельному правій стороні трикутника, біноміальний коефіцієнт, який стоїть над ним, і всі біноміальні коефіцієнти в цьому рядку, які знаходяться зліва від цього коефіцієнта (див. рис. 3.2). Отримана таким чином сума і буде біноміальним коефіцієнтом, який шукають.

					C_0^0						
				C_1^0		C_1^1					
			C_2^0		C_2^1		C_2^2				
		C_3^0		C_3^1		C_3^2		C_3^3			
	C_4^0		C_4^1		C_4^2		C_4^3		C_4^4		
C_5^0		C_5^1		C_5^2		C_5^3		C_5^4		C_5^5	

Рис. 3.2 Символічний трикутник Паскаля з поданням коефіцієнтів C_n^k у вигляді сумою коефіцієнтів верхнього рядка

Біноміальний коефіцієнт на основі вище зазначеної теореми 4 можна отримати й з допомогою трикутника Паскаля, який поданий у табличній формі. Для цього виділяється рядок, паралельний гіпотенузі прямокутного трикутника, який було одержано у табл. 3.1. Сума всіх коефіцієнтів у будь-якій початковій частині цього рядка дорівнює коефіцієнту паралельного знизу йому сусіднього рядка, який стоїть під останнім коефіцієнтом цієї частини, наприклад, як це показано на рис. 3.3.

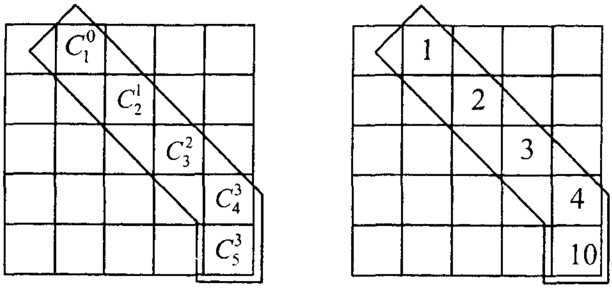


Рис. 3.3. Приклад обчислення біноміального коефіцієнта з допомогою трикутника Паскаля

Теорема 3 (лекція 23) реалізується в трикутнику Паскаля шляхом арифметичних операцій над сусідніми коефіцієнтами будь-якого рядка, розташованого паралельно лівій стороні трикутника. У цьому випадку коефіцієнт C_n^k знаходиться в сусідньому нижньому рядку і визначається як сума біноміального коефіцієнта, який стоїть над ним і всіх сусідніх справа коефіцієнтів верхнього рядка (рис. 3.1а, б).

Так, наприклад, коефіцієнт C_3^2 на рис. 3.1б утворюється коефіцієнтами C_2^1 и C_1^1 верхнього рядка. При цьому коефіцієнт C_2^1 стоїть над ним зверху, а коефіцієнт C_1^1 – справа від коефіцієнта C_2^1.

Відповідно коефіцієнт 3 у нижньому рядку рис. 3.1а дорівнює сумі двох коефіцієнтів – коефіцієнту 2 у верхньому рядку й сусідньому до нього справа коефіцієнту 1 у тому ж самому рядку.

Реалізація вищезгаданої теореми 3 в табличній формі трикутника Паскаля відбувається шляхом підсумовування всіх біноміальних коефіцієнтів, які стоять у одному й тому ж самому стовпчику, до коефіцієнта C_{n-1}^{k-1} включно. Коефіцієнт C_n^k, який шукають, знаходиться в сусідньому справа стовпчику в рядку, де розташований коефіцієнт C_n^{k-1} (див. табл. 3.1).

5. Історична довідка

Уперше опис біноміальних коефіцієнтів зустрічається в працях давньогрецьких авторів. У 12 віці докладний опис біноміальних коефіцієнтів дав індійський математик Бхаскара Ачарья.

Числовий трикутник, поданий на рис. 3.1а, уперше в Європі запропонував Паскаль у 1653 р. За свою форму цей трикутник отримав назву трикутника Паскаля, однак він був добре відомий задовго до нього, наприклад, китайському математику Чжу Ші-Цзе ще в 1303 році.

Заслугою Паскаля є також те, що він обчислив біноміальний коефіцієнт за допомогою явної формули, яка широко використовується й дотепер, і, крім того, він уперше дає доведення цієї формули на базі усвідомленого їм нового на той час фундаментального методу математичних доведень – методу математичної індукції.

Користь від знань – в їх практичному застосуванні.

Конфуцій

Лекція 25
ОБЧИСЛЕННЯ БІНОМІАЛЬНИХ КОЕФІЦІЄНТІВ

1. Загальні положення

Задачі на обчислення кількості сполучень k об'єктів з n, тобто значень біноміальних коефіцієнтів C_n^k, зустрічаються досить часто на практиці, наприклад, при кодуванні інформації. При цьому має значення можливість їх ефективної реалізації з допомогою спеціалізованих пристроїв. Тому поряд з уже добре відомими способами обчислення біноміальних коефіцієнтів, які розглядались у попередній лекції, розробляються й інші, у деяких випадках більш ефективні, способи обчислення біноміальних коефіцієнтів, які різняться між собою при їх реалізації надійністю, швидкодією, апаратурними затратами, складністю.

2. Обчислення з допомогою явних формул

Найбільш просто обчислити біноміальний коефіцієнт, як це вже розглядалося вище, можна з допомогою формул, які подаються в такому вигляді:

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

або

$$C_n^k = \frac{A_n^k}{k!} = \frac{n(n-1)...(n-k+1)}{k!}.$$

Наприклад, якщо $k = 2,\ n = 6$, то

$$C_6^4 = \frac{6!}{4!2!} = \frac{6 \cdot 5}{2} = 15.$$

3. Обчислення з допомогою рекурентних формул

Розглянутий вище метод при великих k і n потребує великої кількості операцій множення та ділення, що створює складності для проектування спеціалізованих обчислювальних пристроїв. Інколи для більш ефективної реалізації таких пристроїв може бути використана одна з рекурентних формул для біноміальних коефіцієнтів, наприклад:

$$C_n^k = \frac{n}{k} C_{n-1}^{k-1},\ k \neq 0.$$

Так, при обчисленні біноміального коефіцієнта C_{10}^4 буде отримана така послідовність:

$$C_{10}^4 = \frac{10}{4} C_9^3 = \frac{10}{4} \cdot \frac{9}{3} C_8^2 = \frac{10}{4} \cdot \frac{9}{3} \cdot \frac{8}{2} \cdot C_7^1 = \frac{10 \cdot 9 \cdot 8 \cdot 7}{4 \cdot 3 \cdot 2 \cdot 1} = 210.$$

Однак цей метод обчислення біноміальних коефіцієнтів, як і попередній, використовує операції множення й ділення, тому не завжди є достатньо ефективним при побудові спеціалізованих пристроїв. Тому можуть бути застосовані й інші методи їх обчислення. Деякі з них наведені нижче.

4. Обчислення біноміальних коефіцієнтів на основі побудови числових послідовностей

На основі трикутника Паскаля нами був розроблений метод обчислення біноміальних коефіцієнтів, який використовує тільки операції додавання між елементами числових послідовностей, що створюються біноміальними коефіцієнтами. Цей метод має значення при побудові спеціалізованих пристроїв, де виконування лише операцій додавання спрощує обчислення біноміальних коефіцієнтів, підвищує швидкодію і надійність.

Для реалізації цього методу спочатку на першому кроці його роботи з трикутника Паскаля береться перша послідовність натуральних чисел 1, 2, 3, 4 ... . Елементи наступної послідовності чисел будуються на другому кроці таким чином, що першим елементом є 1. Другий елемент після цього отримується в результаті додавання цієї 1 до другого елемента початкової послідовності – 2. У результаті буде отримане число 3. Третій елемент є результатом додавання другого елемента, тобто 3, нової послідовності до третього елемента початкової послідовності – 3, і так буде продовжуватися до того часу, поки не буде отримана послідовність 1, 3, 6, 10 ... з $k + 1$ елементів. На основі цієї послідовності створюється нова – третя послідовність $1, 4, 10, 20 \ldots$, потім ще одна, і так буде продовжуватися, поки не буде створена послідовність чисел на кроці

$n-k$. У ній останній її елемент буде мати значення біноміального коефіцієнта, який визначається.

З вищенаведеного випливає, що для того, щоб обчислити з допомогою розглянутих вище операцій біноміальний коефіцієнт C_n^k, потрібно здійснити такі кроки:

1. Побудувати відповідно до розглянутого вище правила $n-k$ числових послідовностей, починаючи з послідовності 1, 2, ..., $k+1$.

2. В останній $(n-k)$-й послідовності знайти $(k+1)$-е число, яке й буде значенням біноміального коефіцієнта C_n^k.

Приклад 1. Дано $k=4$, $n=10$. Треба знайти величину біноміального коефіцієнта C_{10}^4.

Рішення. З умови задачі випливає, що $n-k=6$. Тому потрібно збудувати 6 послідовностей, які мають по $k+1=5$ біноміальних коефіцієнтів: 1, 2, 3, 4, 5; 1, 3, 6, 10, 15; 1, 4, 10, 20, 35; 1, 5, 15, 35, 70; 1, 6, 21, 56, 126; 1, 7, 28, 84, 210. Коефіцієнт $C_{10}^4=210$, оскільки останній елемент послідовності з номером 6 є число 210.

5. Обчислення C_n^k на основі біноміальних квадратів і прямокутників

Біноміальні квадрати можна отримати з трикутника Паскаля. Щоб отримати біноміальний квадрат, треба повернути трикутник Паскаля вправо на 45° і доповнити його числами, які отримуються з допомогою встановлених вже раніше правил для побудови чисел цього трикутника, до форми квадрата. Побудований таким чином біноміальний квадрат зображено в табл. 3.2.

Таблиця 3.2. Біноміальний квадрат

k / $n-k$	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	1	1
1	10	9	8	7	6	5	4	3	2	1
2	55	45	36	28	21	15	10	6	3	1
3	220	165	120	84	56	35	20	10	4	1
4	715	495	330	210	126	70	35	15	5	1
5	2002	1287	792	462	252	126	56	21	6	1
6	5005	3003	1716	924	462	210	84	28	7	1
7	1144	6435	3432	1716	792	330	120	36	8	1
8	2431	1287	6435	3003	1287	495	165	45	9	1
9	4862	2431	1144	5005	2002	715	220	55	10	1

Однією з властивостей цього квадрата є те, що з його допомогою можна обчислювати будь-які біноміальні коефіцієнти C_n^k з величинами параметрів $k \le K$ і $n \le K + k$, де K – максимальне значення параметра k. Відповідно, кожна сторона біноміального квадрата має K + 1 стовпчиків і стільки ж рядків.

З метою знаходження величини біноміального коефіцієнта треба спочатку знайти значення різниці n – k, а потім вибрати відповідний отриманому значенню рядок. На перетині цього рядка й стовпчика, який відповідає числу k, буде знаходитись необхідне число.

Наприклад, треба обчислити біноміальний коефіцієнт з k = 5 і n = 7. Для цього спочатку обчислюємо різницю $n - k = 7 - 5 = 2$ і потім знаходимо перетин другого рядка з п'ятим стовпчиком. Там знаходиться число 21. Це означає, що біноміальний коефіцієнт C_7^5 має значення 21, що легко перевіряється з допомогою формули для біноміального коефіцієнта, яка подана в явному вигляді: $C_7^5 = \dfrac{71}{5121} = 21$.

Можна обчислити біноміальний коефіцієнт C_7^5 й інакше. Будемо виходити з того, що $C_7^5 = C_7^2$. Тоді $n - k = 7 - 2 = 5$, і відповідно отримаємо значення $C_7^5 = 21$ на перетині другого стовпчика і п'ятого рядка.

Достоїнством біноміального квадрата є те, що обчислення біноміальних коефіцієнтів з його допомогою не потребує, крім знаходження різниці $n - k$, інших арифметичних операцій. Це, у свою чергу, дозволяє отримати велику швидкодію пристроїв, які обчислюють величину біноміальних коефіцієнтів.

Якщо продовжити ту чи іншу сторону біноміального квадрата, то отримаємо біноміальний прямокутник.

Наприклад, у табл. 3.2 збільшимо кількість рядків з 9 до 18. Це вже буде біноміальний прямокутник. Його особливістю є те, що він дозволяє знаходити значення C_n^k для параметра n, який змінюється від 0 до 18.

Кращий спосіб вивчити будь-що – це відкрити самому.

Д. Пойа

Лекція 26
БІНОМ НЬЮТОНА

1. Загальні положення

Біном Ньютона – це досить відома формула в математиці. Вона подає вираз $(x-a)^n$, де n – ціле додатне число, в вигляді многочленна.

Теорема 1.

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

Доведення. Візьмемо добуток n двочленів

$$P=(x-a_1)(x-a_2)...(x-a_n)$$

і, якщо розкриємо в ньому дужки та згрупуємо члени з однаковими показниками степеня, одержимо многочлен, розташований за значеннями показників степені, які зменшуються

$$P = x^n - x^{n-1}(a_1+a_2+...+a_n) + x^{n-2}(a_1a_2+a_1a_3...+a_{n-1}a_n) -$$
$$-x^{n-3}(a_1a_2a_3+a_1a_2a_4+...+a_{n-2}a_{n-1}a_n)+...+(-1)^n(a_1a_2...a_n).$$

Неважко побачити, що в дужках виразу, який одержано вище, приведені всілякі сполучення k = 1, 2, …, n елементів множини $\{a_1, a_2, ..., a_n\}$ з n, кількість яких дорівнює C_n^k.

У випадку, коли $a_1 = a_2 = ... = a_n = a$,

$$P=(x-a)^n = x^n - C_n^1 a x^{n-1} + C_n^2 a^2 x^{n-2} - C_n^3 a^3 x^{n-3} + ... +$$
$$+(-1)^k C_n^k a^k x^{n-k} + ... + (-1)^n a^n.$$

У цій формулі k-й член буде мати вигляд:

$$T_k = (-1)^k C_n^k a^k x^{n-k}.$$

Тоді за умови, що

$$T_0 = (-1)^0 C_n^0 a^0 x^{n-0} = C_n^0 x^n = x^n$$

і

$$T_n = (-1)^n C_n^n a^n x^{n-n} = (-1)^n C_n^n a^n x^0 = (-1)^n a^n,$$

$$P = (x-a)^n = \sum_{k=0}^{n} (-1)^k C_n^k a^k x^{n-k}.$$

Теорему доведено.

Як бачимо з бінома Ньютона, кількості сполучень біля кожного з його членів відіграють важливу роль. Тому ці кількості були названі *біноміальними* коефіцієнтами.

Приклад 1. Знайти біном Ньютона для виразу $(x-a)^2$.

Розв'язання. Знайдемо добуток двох двочленів

$$(x-a_1)(x-a_2) = x^2 - (a_1 + a_2)x + a_1 a_2.$$

Припустимо, що $a_1 = a_2 = a$. Тоді

$$(x-a)(x-a) = (x-a)^2 = x^2 - 2ax + a^2.$$

Приклад 2. Знайти біном Ньютона для виразу $(x-a)^3$.

Розв'язання. Візьмемо добуток трьох двочленів:

$$(x-a_1)(x-a_2)(x-a_3) = x^3 - (a_1 + a_2 + a_3)x^2 +$$
$$+(a_1a_2 + a_1a_3 + a_2a_3)x - a_1a_2a_3.$$

Якщо $a_1 = a_2 = a_3 = a$, то

$$(x-a)(x-a)(x-a) = (x-a)^3 = x^3 - 3ax^2 + 3a^2x - a^3.$$

Під x та a можна розуміти будь-яке число, чи будь-який алгебраїчний вираз.

Приклад 3. Знайти розклад виразу $(3x-2a)^5$ на основі формули бінома Ньютона.

Розв'язання.

$$(3x-2a)^5=(3x)^5-5\cdot 2a(3x)^4+10(2a)^2(3x)^3-10(2a)^3(3x)^2+$$
$$+5(2a)^4 3x-(2a)^5=243x^5-810ax^4+1080a^2x^3-720a^3x^2+240a^4x-32a^5.$$

У випадку, якщо a є додатним числом, то

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

Приклад 4. Знайти біном Ньютона для виразу $(x+a)^3$, якщо a є додатним числом.

Розв'язання.

$$(x+a)^3=(x+a)(x+a)(x+a)=x^3+3ax^2+3a^2x+a^3.$$

Приклад 5. Обчислити з допомогою бінома Ньютона вираз $(10+2)^3$.

Розв'язання.

$$(10+2)^3=10^3+3\cdot 2\cdot 10^2+3\cdot 2^2\cdot 10+2^3=1728.$$

Приклад 6. Обчислити з допомогою бінома Ньютона вираз $(10+0{,}1)^3$.

Розв'язання.

$$(10+0{,}1)^3=10^3+3\cdot 0{,}1\cdot 10^2+3\cdot 0{,}1^2\cdot 10+0{,}1^3=1030{,}301.$$

2. Властивості бінома Ньютона

Спочатку зазначимо деякі очевидні властивості бінома Ньютона:

1. Кількість членів розкладу на 1 більше за показник n.

2. Показники степеня числа x зменшуються, а числа a збільшуються від члена до члена на 1.

3. Сума показників x і a в кожному члені дорівнює n.

Далі доведемо дві важливі властивості біноміальних коефіцієнтів бінома Ньютона у вигляді теорем 2 і 3.

Теорема 2.

$$\sum_{k=0}^{n} C_n^k = 2^n.$$

Доведення. Виходячи з того, що, при $x = a = 1$, з одного боку,

$$(x+a)^n = (1+1)^n = 2^n,$$

а з іншого,

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

то

$$\sum_{k=0}^{n} C_n^k = 2^n.$$

Теорему доведено.

Теорема 3. ***Сума біноміальних коефіцієнтів у бінома Ньютона, які займають парні місця, дорівнює сумі біноміальних коефіцієнтів, які займають непарні місця, і кожна з них дорівнює 2^{n-1}.***

Доведення. Припустимо, що для бінома Ньютона $x = 1$ і $a = -1$. Тоді

$$(x+a)^n = (1-1)^n = (0)^n = 0,$$

і оскільки

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

то і

$$\sum_{k=0}^{n} C_n^k a^k x^{n-k} = 0.$$

Доданки $C_n^k a^k x^{n-k}$ у наведеному вище виразі приймають у випадку, якщо k парне, додатні значення, які дорівнюють $C_n^{k'}$, а у випадку, якщо k непарне, від'ємне: $-\ C_n^{k''}$, де $k' = 0,2,4,...$; $k'' = 1,3,5,...$; k' –значення верхнього параметра коефіцієнтів, які стоять на парних місцях, а k'' – значення верхнього параметра коефіцієнтів, які стоять на непарних місцях. Тоді

$$\sum_{k=0}^{n} C_n^k a^k x^{n-k} = \sum_{k'=0}^{n'} C_n^{k'} - \sum_{k''=0}^{n''} C_n^{k''} = 0,$$

де n' – максимальне значення k';

n'' – максимальне значення k''.

З одержаної рівності випливає, що

$$\sum_{k'=0}^{n'} C_n^{k'} = \sum_{k''=0}^{n''} C_n^{k''}.$$

Оскільки з виразу, який наведений вище, виходить, що вказані в ньому суми дорівнюють одна одній, а з теореми 2 при $x = a = 1$, що

$$\sum_{k=0}^{n} C_n^k = \sum_{k'=0}^{n'} C_n^{k'} + \sum_{k''=0}^{n''} C_n^{k''} = 2^n,$$

то кожна з наведених сум повинна дорівнювати

$$\frac{2^n}{2} = 2^{n-1}.$$

Тобто

$$\sum_{k'=0}^{n'} C_n^{k'} = 2^{n-1}$$

і

$$\sum_{k''=0}^{n''} C_n^{k''} = 2^{n-1}.$$

Теорему доведено.

3. Історична довідка

Біном Ньютона для цілих додатних значень n знали до Ньютона (1643–1727) ще середньоазіатські математики, зокрема, Омар Хайям (1048–1131), а в Західній Європі – Паскаль (1623–1662). Цей вираз ще називають біноміальною формулою, або біноміальною теоремою.

Заслуга Ньютона полягає в тому, що він вперше узагальнив цю формулу для дробових та від'ємних показників n. Про це відкриття Ісаак Ньютон повідомив у 1676 р., але повноцінного доказу формули розкладу $(x-a)^n$ для нецілих та від'ємних n він не мав, хоча саме про це його просив Лейбніц. Доказ Ньютона базувався більше на прикладах і аналогіях, ніж на теорії. У 1774 році цю теорему намагався довести Леонард Эйлер і тільки в 1812 році її довів Карл Фрідріх Гаус з допомогою теорії нескінченних сум.

Як бачимо, розкладення виразу $(x+a)^n$, яке носить назву біному Ньютона для цілих, дробових, додатних і від'ємних n, має довгу історію, що свідчить про його велике значення в науці.

Доведення це є творча задача науки

В. Ф. Асмус

КОНТРОЛЬНІ ЗАВДАННЯ І ПИТАННЯ ДО ЧАСТИНИ III

Питання для самоконтролю

1. Дайте визначення комбінаторних задач і наведіть їх класифікацію.

2. Сформулюйте правило суми для комбінаторних задач.

3. Сформулюйте правило добутку для комбінаторних задач.

4. Що таке сполучення і біноміальні коефіцієнти?

5. Доведіть формулу для обчислення кількості сполучень k елементів з n.

6. Доведіть, що для $n, k \neq 0$ $C_{n+1}^{k+1} = C_n^{k+1} + C_n^k$.

7. Доведіть, що для $n, k \neq 0$ $C_n^k = \sum_{i=0}^{n-k} C_{k+i-1}^{k-1}$.

8. Доведіть, що для $n, k \neq 0$ $C_n^k = \sum_{i=0}^{k} C_{n-k+i-1}^{i}$.

9. Доведіть, що для $n, k \neq 0$ $C_n^k = \frac{n}{k} C_{n-1}^{k-1}$.

10. Доведіть, що для $n, k \neq 0$ $C_n^k = \frac{n}{n-k} C_{n-1}^{k}$.

11. Як виконується обчислення біноміальних коефіцієнтів з допомогою явних формул?

12. Як виконується обчислення біноміальних коефіцієнтів з допомогою рекурентних формул?

13. Як виконується обчислення біноміальних коефіцієнтів на основі побудови числових послідовностей?

14. Як виконується обчислення біноміальних коефіцієнтів на основі біноміальних квадратів і прямокутників?

15. В чому полягають властивість симетрії для біноміальних коефіцієнтів?

16. В чому полягають властивості додавання для біноміальних коефіцієнтів?

17. В чому полягають властивості винесення за дужки для біноміальних коефіцієнтів?

18. Дайте визначення розміщення й доведіть формулу обчислення числа розміщень k елементів з n.

19. Дайте визначення перестановки й доведіть формулу обчислення числа перестановок з n елементів.

20. Дайте визначення трикутника Паскаля й поясніть принцип його побудови.

21. Що таке числові й символічні трикутники Паскаля?

22. Які існують властивості трикутників Паскаля?

23. Побудуйте табличну форму трикутників Паскаля. У чому полягають переваги й недоліки такої побудови?

24. Як знайти біноміальні коефіцієнти з допомогою трикутників Паскаля?

25. Наведіть ознаки числового й символічного трикутника Паскаля.

26. Побудуйте трикутник Паскаля в табличній формі.

27. Наведіть формулу бінома Ньютона й доведіть її.

28. Доведіть на основі формули бінома, що сума біноміальних коефіцієнтів

$$\sum_{k=0}^{n} C_n^k = 2^n.$$

29. Доведіть, що сума біноміальних коефіцієнтів, які займають парні місця у формулі бінома Ньютона, дорівнює сумі біноміальних коефіцієнтів, які займають непарні місця.

30. Доведіть, що у формулі Ньютона сума біноміальних коефіцієнтів, які займають парні місця, і сума біноміальних коефіцієнтів, що займають непарні місця, дорівнює 2^{n-1}.

Контрольні завдання

1. Скількома різними способами можна розставити на майданчику 6 волейболістів? Відповідь: 720.

2. Скільки різних слів із чотирьох літер, що не повторюються, можна скласти, використовуючи український алфавіт, який налічує 32 літери? Відповідь: 863040.

3. Розв'яжіть рівняння $A_x^2 = 90$. Відповідь: $x = 10$.

4. З 10 студентів необхідно обрати 5 на студентську конференцію. Скількома способами це можна зробити? Відповідь: 252.

5. Скількома способами можна розмістити 4 шашки на 32 однокольорових клітинках? Відповідь: 35960.

6. Скількома способами можна розмістити на 32 однокольорових клітинках 4 білих і 4 чорних шашки? Відповідь: 736281000.

7. На профспілкову студентську конференцію необхідно обрати 4 делегати з 20 студентів. Скількома способами це можна зробити? Відповідь: 4845.

8. Один студент має 5 книг з математики, а інший 12 з фізики. Скільки існує варіантів для обміну по дві книги? Відповідь: 660.

9. В урні є 20 квитків. Скільки може бути варіантів вибору 2 квитка з 20 квитків? Відповідь: 190.

10. Шістнадцять студентів розподілилися на дві рівні групи, кожна з 8 студентів відповідно, для пошуку на місцевості об'єкта, який їх цікавив. Лише 4 з них були знайомі з місцевістю. Скількома способами вони можуть розподілитися таким чином, щоб у кожній групі було по два студенти, яким знайома місцевість? Відповідь: 5544.

11. Збори студентської групи, що складається з 20 чоловік (серед них дві дівчини), обрали делегацію із 6 чоловік на студентську конференцію. Скільки може існувати способів формування делегації, в якій присутні обидві дівчини? Відповідь: 3060.

12. Студентський комітет для допомоги першокурсникам надав групу студентів зі старших курсів, яка складається з 5 чоловік. Відбір проводили з 20 добровольців, серед яких було 5 студентів із шостого курсу і 4 з п'ятого. Яка кількість комплектацій групи можлива, якщо до неї обов'язково входить по одному студенту з шостого й п'ятого курсу? Відповідь: 3300.

13. Група студентів з 20 чоловік збирається мандрувати потягом. У касі є 12 квитків з місцями на нижній полиці й 8 – на верхній. При цьому 5 студентів бажають їхати знизу, а 4 зверху. Скількома способами їх можна розташувати у вагоні потягу, якщо порядок розміщення пасажирів як знизу, так і зверху не враховується? Відповідь: 330.

14. Вирішити попередню задачу за умови, що порядок розташування студентів як зверху, так й знизу враховується. Відповідь: 950400.

15. Та сама задача 13, але за умови, що враховується лише порядок розташування студентів знизу. Відповідь: 39600.

16. Ті самі умови задачі 13, але з вимогою, що враховується лише порядок розташування студентів зверху. Відповідь: 7920.

17. Скільки шестизначних чисел можна скласти з цифр $1,2,3,\ldots,9$, якщо кожне число повинно містити дві парні й чотири непарні різні цифри. Відповідь: 30.

18. Група студентів складається з 20 чоловік. Вони вирішили обмінятися на пам'ять фотокартками. Скільки всього було роздано фотокарток? Відповідь: 380.

19. Скільки різних спортивних команд можна скласти з 5 учасників, якщо взяти їх по троє, а скільки – якщо по 2? Відповідь: 10.

20. Десять студентів, які зустрілися після літніх канікул, обмінялися рукостисканнями. Скільки всього було зроблено рукостискань? Відповідь: 45.

21. Студенти вивчають 9 різних предметів. Першого вересня потрібно було провести 3 заняття. Скількома способами можна скласти розклад занять на перше вересня, щоб у цей день було 3 різних предмета? Відповідь: 504.

22. Для передачі сигналів вивішують одне під другим 3 різнокольорові полотнища. Скільки різних сигналів можна передати за наявності цих полотнищ? Відповідь: 6.

23. Знайдіть значення x з рівняння $C_{3x+1}^{3x-1} = 120$. Відповідь: $x = 5$.

24. Скільки існує різних п'ятизначних десяткових чисел з цифрами, які не повторюються? Відповідь: 30240.

25. Скількома способами можна призначити на чергування з охорони суспільного порядку групу з п'яти студентів і одного викладача, якщо маємо 15 студентів і 4 викладачів? Відповідь: 12012.

26. Скількома способами для 9 викладачів можна скласти розклад консультацій на 9 днів, якщо кожен викладач дає консультацію лише один раз? Відповідь: 362880.

27. Студентський комітет налічує 9 членів. Скількома способами можна утворити з них делегацію у складі 3 чоловік для поїздки до шефів? Відповідь: 84.

28. На основі формули бінома Ньютона обчисліть величину 11^3. Відповідь: 1331.

29. У виразі $(2x+2)^2=0$ з допомогою формули бінома Ньютона знайдіть *x*. Відповідь: x = –1.

30. Для виразу $(3x-2a)^2$ знайдіть з допомогою формули бінома Ньютона третій член його розкладу. Відповідь: $4a^2$.

Частина IV

СИСТЕМИ ЧИСЛЕННЯ

РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА СИСТЕМ ЧИСЛЕННЯ

Лекція 27
ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЧИСЛО І СИСТЕМИ ЧИСЛЕННЯ

1. Означення числа

Основою поняття числа є така абстракція як *кількість* елементів скінченної множини, що не має форми. Однак число більш загальне поняття, ніж кількість, оскільки воно, крім того, що містить у собі інформацію, ще й надає цій кількості специфічного кодового *зображення* чи *форми*. Його наявність дозволяє виконувати з кількістю різні арифметичні і логічні операції.

При цьому слід звернути особливу увагу на те, що одна й та ж сама кількість може зображатися по-різному. Так, десяткове число 123 може мати ще зображення, наприклад, у вигляді двійкового числа 1111011. Таких послідовностей, які кодують одну й ту ж саму кількість, але мають при цьому різний вигляд, може бути безмежно багато. Тому треба відрізняти кількість елементів від її зображення у вигляді числа. Від цього зображення значною мірою залежить ефективність виконання арифметичних і логічних операцій над кількостями.

Однак і зображенню числа можуть відповідати різні кількості. Наприклад, п'ятеричне число 123 кодує зовсім іншу кількість, ніж таке ж саме за виглядом десяткове число 123. У першому випадку п'ятеричне число 123 кодує в десятковій системі числення кількість 38, а в другому – 123. Більше того, послідовність знаків 123 може розглядатись навіть не як число, а як перестановка з трьох елементів, і тоді ця послідовність не кодує жодної кількості.

Як випливає з вищесказаного, під числом можна розуміти, з одного боку, кодове зображення кількості, а з іншого – саму кількість. Так поняття числа вживається і в даному підручнику, тобто воно розглядається або як кількість, або як її кодове зображення, або ж як те й інше поняття разом. Розуміння числа, що використовується в цьому підручнику, буде випливати з контексту. Воно залежить від того, який наголос робиться в матеріалі, що подається.

Відповідно до двох вищезазначених функцій числа – як носія кількості й одночасно як її зображення – існують дві пов'язані з ними теорії. Перша вивчає властивості чисел безвідносно до їх зображення.

Цим займається така непроста наука, як теорія *чисел*. Друга теорія знаходить способи ефективного формування чисел шляхом кодування кількостей. Це вже інша наука – теорія *кодування*.

Важливу роль у кодуванні кількостей відіграють особливі види чисел, які називаються *цифрами*. Ними називають такі специфічні числа, з яких складаються як завгодно великі інші числа. Порядок розміщення цифр у числах може бути різний. Найчастіше це буде лінійна форма, коли цифри йдуть послідовно одна за одною, але може бути розміщення цифр і на площині у вигляді матриць. У цих розміщеннях цифр виявляється спосіб кодування кількостей, від якого в багатьох випадках залежить ефективність їх використання на практиці.

2. Кількісні і порядкові функції чисел

Число, крім позначення кількості, також використовується й для позначення порядку, який складається між елементами множин. Таке число ще називається *номером*. Як правило, це буде ціле невід'ємне число. Таким чином, у загальному плані за своїми функціями число повинне давати можливість отримувати відповідь на питання як про кількість елементів у множинах, так і про порядок їх розміщення. У першому випадку відповідь буде мати вигляд – один, два, три, ..., а в другому – перший, другий, третій ... .

На першому з приведених вище питань базуються *кількісні* операції щодо елементів множин, а на другому – різні способи впорядкування цих елементів. Кількісні операції вивчає теорія *кардинальних* чисел, а порядкові – теорія *порядкових* чисел. Ці теорії дуже важливі для розуміння природи числа, яка до сьогодні, ще не зовсім визначена, і тому потребує подальших досліджень.

Способи кодування кількостей не залежать від того, що визначається – кількість чи порядок елементів у множині, тобто вигляд числа дозволяє визначити як їх кількість, так і номер елемента в ній по порядку. Наприклад, число 120 може визначати кількість – сто двадцять, а може й порядок – сто двадцятий. Однак самі властивості кількості й порядку, як уже зазначалося раніше, відмінні між собою. Відповідь щодо кількості елементів у тій чи іншій непорожній множині дає *кількісне* число, а щодо порядку розміщення того чи іншого елемента в ній – *порядкове* число – тобто його номер.

Тільки числа, які володіють зазначеними двома функціями, можуть вважатися достатньо ефективними при їх використанні на практиці, оскільки з допомогою цих функцій можна отримати відповідь на питання: скільки елементів містить множина і яке місце по порядку в ній посідає той чи інший елемент? В останньому випадку вирішується проблема *нумерації* чисел.

При цьому номерами, зазвичай, вважаються числа, які не тільки розв'язують питання щодо нумерації елементів, але й потребують для своєї побудови *мінімальної* кількості знаків. Тобто не кожне число можна вважати номером, а тільки те, яке містить у собі максимальну кількість інформації, а отже, не є надлишковим. Тому проблема нумерації стосується не лише питання визначення порядку, а й стиснення інформації.

3. Натуральні числа

Кількісне число характеризує деякий клас скінченних множин, тобто множин, в яких між їх елементами можна виявити взаємно-однозначну відповідність. Такими множинами, наприклад, можуть бути множина п'яти пальців руки і множина літер слова „книга", оскільки кожному пальцю може відповідати одна з літер слова „книга". Очевидно, що числу п'ять може відповідати значна кількість різних множин, і тому воно відображає те спільне, що мають ці множини між собою, – кількість елементів, яка дорівнює п'яти. Тобто число п'ять створює клас множин, кожна з яких містить рівно п'ять елементів.

Такі ж класи створюють і інші числа – 1, 2, ... Наведена нескінченна впорядкована дискретна послідовність чисел має назву *натурального* ряду чисел. Якщо конкретне число в ряду визначає кількість чисел у ньому, то це буде *кількісне* натуральне число, а якщо порядковий номер цього числа, то *порядкове* натуральне число. Ця послідовність дискретна, тому що між числами натурального ряду не існує проміжних елементів. Наприклад, між числами 2 і 3 немає такого елемента, як, наприклад, 2,5. У цьому ряду немає також найбільшого числа, оскільки, яке б натуральне число не взяли, за ним іде інше наступне натуральне число, що на одиницю більше від попереднього і яке відповідає попередній множині елементів із ще приєднаним до неї одним елементом.

4. Нуль і одиниця в ряду натуральних чисел

Слід окремо зазначити, що натуральний ряд починається з одиниці, а не з нуля. Назва нуль походить від латинського слова *порожній (nulus)*, тобто він відображає порожню множину. Нуль не має всіх властивостей натуральних чисел, і в той же самий час йому притаманні специфічні властивості, які відсутні у натуральних чисел. Наприклад, якщо будь-яке число помножити на нуль, то одержимо нуль. Для інших натуральних чисел це не так.

Хоча, зрештою, і одиниця, з якої починається натуральний ряд, має деякі властивості, відмінні від інших чисел натурального ряду. Так, якщо помножити одиницю на будь-яке число, то отримаємо те ж саме число. Щодо других чисел таке твердження не є справедливим. Тому не випадково визначний філософ і математик XVII століття Лейбніц вважав одиницю і нуль особливими числами. Одиниця була для нього божественним началом і початком усього існуючого, а нуль – небуттям усього. Але одиниця все ж таки належить до натурального ряду, а нуль ні. Це пов'язане з історією створення цього ряду, який до надання йому сучасного вигляду, як нескінченної послідовності цілих чисел, створювався не одну тисячу років.

При створенні цього ряду розглядались лише ті множини, які не були порожніми, і тому потреби в нулі, який би відображав кількість елементів у порожній множині, не було. Відповідно до цих непорожніх множин рахування теж починалося з одиниці, а не з нуля. Потреба в нулі виникла лише тоді, коли рівень абстрактного мислення людини дозволив усвідомити існування порожньої множини. Отже, з'явилась необхідність певним чином позначити кількість елементів у цій множині. Такою позначкою став нуль.

Множину, що утворюється шляхом приєднання до множини натуральних чисел нуля, називають ***множиною цілих невід'ємних чисел.*** Нуль у цій множині ставлять перед числом 1, тобто маємо таку впорядковану множину невід'ємних чисел – 0, 1, 2, 3, ... Множині цих чисел притаманні всі властивості натурального ряду чисел, тобто вона упорядкована, дискретна й нескінченна, але відрізняється від нього тим, що починається з нуля, а не одиниці.

5. Аксіома лічби

Зазначені вище дві теорії кардинальних і порядкових чисел певним чином використовуються, крім кодування чисел, ще і для *лічби* елементів у множинах, тобто для їх підрахунку шляхом послідовного перебирання.

Лічба відіграє особливу роль при кодуванні чисел, оскільки саме з її допомогою можна одночасно визначити як порядок розташування елементів у множинах, так і їх кількість. При цьому результат лічби не залежить ні від порядку, в якому вона відбувається, ні від системи нумерації предметів. Важливо лише, щоб під час лічби не був пропущений жодний з елементів і щоб кожний з них був полічений лише один раз. Тільки в такому разі може бути знайдений порядок кожного з елементів у множині і їх загальна кількість. Це визначення відоме як ***аксіома лічби***.

6. Поняття про системи числення

Число, визначене як кількість, є абстрактним поняттям і не залежить від способів його зображання – кодування, але сама техніка його кодування і виконання щодо нього арифметичних і логічних операцій істотною мірою зумовлена цими способами. Такі способи реалізуються з допомогою систем *числення*, чи, як ще інколи їх називають, систем *нумерацій*.

Поняття числа невід'ємне від поняття *системи числення*, які у своєму розвитку пройшли шлях від найпростіших непозиційних систем через лічбу на пальцях до теперішніх позиційних систем числення.

Системою числення, або нумерацією, називається сукупність правил і знаків, за допомогою яких можна відобразити (кодувати) будь-яке невід'ємне число.

До систем числення висуваються певні вимоги, серед яких найбільш важливими є вимоги *однозначного* кодування невід'ємних чисел 0, 1, ... з деякої їх скінченної множини – *діапазону Р* за *скінченне* число кроків і можливості виконання щодо чисел *арифметичних* і *логічних* операцій. Крім того, системи числення розв'язують задачу *нумерації*, тобто ефективного переходу від зображень чисел до номерів, які в даному випадку повинні мати *мінімальну* кількість цифр. Від вдалого чи невдалого вибору системи числення залежить ефективність розв'язання зазначених задач і її використання на практиці.

При розв'язанні проблеми кодування чисел може виникнути необхідність мати їх зображення, які вирішують деякі допоміжні питання, наприклад, питання підняття перешкодостійкості чисел. Але в цьому випадку вони можуть бути побудовані досить складно і тому недостатньо економно з точки зору кількості цифр у них. Тому часто на практиці можна зустрітися з задачею, що протилежна розглянутій вище, – нумерацією зображень чисел. При її вирішенні відбувається *стиснення* інформації, що дозволяє знайти рішення низки як теоретичних, так і практичних питань.

Під час побудови зображень чисел і їх нумерації системи числення в загальному вигляді вирішують задачу *кодування,* і тому теорію їх побудови слід віднести до теорії кодування. Однак особливість систем числення, яка полягає в потребі виконання ними арифметичних і логічних операцій, спричиняє необхідність розглядати їх також і з позиції теорії чисел.

Тобто теорія позиційних систем числення межує з теорією чисел і теорією кодування. У цьому полягає її особливість і специфіка, хоча значною мірою вона все ж таки стосується теорії кодування. З огляду на це теорія позиційних чисел повинна розглядатися як самостійна наука, яка має свій математичний апарат і свої лише їй притаманні завдання, такі як кодування чисел, та їх теоретичний аналіз.

Сьогодні інколи вважається, що на основні питання, які стосуються побудови систем числення, знайдені відповіді, тому що на практиці існуючі системи числення добре себе зарекомендували, звідки випливає, що нині потреба розвивати їх теорію відсутня. Але такому підходу властиві свої недоліки, оскільки розвиток теорії систем числення має значний, не використаний достатньою мірою потенціал.

Про це свідчить і те, що час від часу виникають нові системи числення з новими властивостями й можливостями, які можна використовувати в різних галузях науки й практики. Серед таких систем числення є досить складні, до яких можна віднести системи залишкових класів, факторіальні, поліадичні, фібоначчієві і багато інших.

Так, вони не настільки економні й прості з точки зору кількості елементів в їх структурах, як, наприклад, десяткова, але мають свої позитивні специфічні властивості, які слід використовувати на практиці. Це можливість генерації різних комбінаторних об'єктів, таких, наприклад, як перестановки або сполучення із самими різними обмеженнями щодо них, їх перешкодостійкість і інші більш специфічні властивості, які виявляються під час їх практичного застосування. Крім того, у деяких випадках з допомогою таких систем числення можна збільшити швидкодію обчислювальних пристроїв і систем до величин, які недосяжні при використанні звичайних традиційних систем.

Ці системи не є конкурентами традиційних систем числення, як інколи це вважається, хоча б тому, що останні за своєю універсальністю й простотою виконання арифметичних і логічних операцій недосяжні для будь-яких інших систем числення з більш складною структурою. Однак такі системи числення є ефективним доповненням до звичайних універсальних систем, оскільки здатні разом з останніми працювати над вирішенням одних і тих самих проблем, збільшуючи надійність чи швидкодію пристроїв і систем, які їх використовують. Тому не зовсім правильно протиставляти звичайні, прості, і більш нові, складні, системи числення. Будь-яка з них за певних умов зможе зарекомендувати себе з найкращого боку.

7. Історія виникнення систем числення

Історично першими виникли непозиційні системи числення. Вони ґрунтуються на кількісному підході до визначення числа, який для кодування тих чи інших кількостей застосовував особливі знаки – числа. Кожному такому знаку відповідав кількісний еквівалент. Наприклад, у так званій римській нумерації знаку X відповідала кількість елементів множини, яка дорівнювала 10.

У подальшому такими знаками – числами користувалися також і для одержання інших чисел. Так, якщо перед знаком X ставилась вертикальна риска, то отримували знак IX, який означав, що від десяти треба відняти одиницю і результуюча кількість буде дорівнювати 9. Знаки, подібні X, називаються вузловими. Вони широко використовувалися в первісних непозиційних системах числення. Слід ще раз зазначити, що серед цих знаків не було такого, який би відповідав нулю. Це свідчить про те, що нуль у той час ще не був сформований як число.

Кількість чисел, яку можна було одержати з допомогою непозиційного кодування, через його складність і відповідно велику кількість чисел, що потребували запам'ятовування, була обмежена кількома сотнями, і, крім того, щодо цих чисел досить важко було виконувати арифметичні й логічні операції. Тому в подальшому з розвитком науки виникла потреба в більш ефективних системах числення, які б мали прості правила кодування чисел, та легко виконували б щодо них арифметичні й логічні операції. Такі системи чисел були створені і отримали назву *позиційних*. Більш докладно ці системи числення будуть розглянуті нижче, тому що вони складають на сьогодні основу теорії систем числення взагалі.

Весь світ пройнятий кількістю.

А. И. Уайтхед

Лекція 28
ДЕСЯТКОВА І СПОРІДНЕНІ З НЕЮ ПОЗИЦІЙНІ СИСТЕМИ ЧИСЛЕННЯ

1. Позиційні числа

Для позиційних систем числення важливим є принцип ***позиційності***, який полягає в змінах кількісного значення цифр залежно від їх позицій. Ці позиції визначаються *розрядами* числа. Кожна цифра числа має свій розряд, в якому вона знаходиться і який *відрізняється від інших розрядів своїм номером.*

Принцип позиційності дозволяє легко з допомогою порівняно невеликої кількості цифр і позицій, які вони займають, зобразити будь-яке велике натуральне число. Це важлива властивість позиційних систем числення, яка широко використовується на практиці.

Кількість розрядів позиційного числа характеризує його *довжину*. У більшості випадків числа однієї і тієї ж самої системи числення мають однакову довжину. Тому вони легко розрізняються між собою Такі числа називають *рівномірними*. Однак ця довжина й відповідна їй кількість розрядів у загальному випадку для різних чисел однієї й тієї ж самої позиційної системи числення можуть бути різними. Такі числа з нерівномірною довжиною називаються *нерівномірними*. Це призводить до того, що виникає задача їх розпізнавання.

Ця задача розв'язується шляхом зображення чисел у вигляді *префіксних* кодів, тобто кодів, в яких кожне кодове слово не є початком іншого кодового слова цього коду. Так, кодові слова 000, 0010, 01100 створюють префіксний код, а слова коду 00, 001, 0101 не створюють його, оскільки слово 00 є початком слова цього коду – 001 і тому не може бути виділене як самостійне слово.

Для рівномірних чисел задача розпізнавання має значно простіший вигляд, оскільки її розв'язання відбувається шляхом підрахунку кількості цифр у числі. Якщо ця кількість буде дорівнювати довжині, яка для всіх чисел однакова, то воно вважається знайденим. Далі лише необхідно буде виокремити його з поміж інших чисел такої ж довжини. Так, рівномірні числа 00, 01, 10 і 11 легко розпізнаються, оскільки мають однакову довжину, яка складається з двох цифр. За наявності в числі в процесі його формування двох цифр воно вважається сформованим.

2. Структурні елементи позиційних систем числення

Будь-яка *система* числення, чи *нумерація*, у тому числі й позиційна, у загальному вигляді у своєму складі повинна мати скінченну множину невід'ємних чисел – *діапазон*, який вона кодує. До неї обов'язково в позиційних системах числення входить число 0 і далі числа натурального ряду, які починаються з 1. Крім того, в позиційну систему числення входить ще *нумераційна*, чи *числова*, функція, з допомогою якої ці числа нумеруються, а також *алфавіт* – скінченна множина цифр, з яких складаються кодові зображення чисел, і *обмеження* на значення цифр.

Для цифр алфавіту характерною ознакою є те, що вони починаються з нуля і далі йдуть як ряд натуральних чисел, тобто цифри алфавіту мають порядок 0, 1, ... Серед цих цифр не може хоча б одна бути пропущеною. Так, для найбільш поширеної на сьогодні і досить простої позиційної системи числення – десяткової алфавіт має 10 цифр – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, серед яких обов'язково буде нуль.

Число цих десяти цифр для десяткової системи числення, як і число цифр 2, 3, 4, 5, ... в алфавітах інших подібних їй двійкової, трійкової, четвіркової, п'ятеричної систем числення, має назву їх *основи*. Основа задає кількість цифр з урахуванням нуля, які має алфавіт тієї чи іншої системи числення. Тому максимальна цифра в алфавіті десяткової системи числення завжди буде на одиницю меншою від основи. Але це правило поширюється тільки на десяткову й подібні їй системи числення. У більш складних системах це може бути інакше.

Цікаво, що в непозиційних системах числення нуль був відсутній. Щоб зрозуміти значення нуля й відповідної йому порожньої множини, треба було винайти позиційні системи числення, а для цього повинні були пройти тисячі років. Позиційні системи числення і лічба на їх основі неможливі без нуля. Тому нуль є однією з фундаментальних основ сучасної позиційної лічби.

3. Десяткова система числення

Вважається, що першими позиційними системами числення, які одержали поширення на практиці, були п'ятерична і десяткова. Пізніше з'явилися двійкова, восьмерична, дванадцятерична та інші системи числення. Кожна з цих систем числення має однакову довжину чисел.

У десятковій системі числення кожне натуральне число з допомогою нумераційної функції можна подати у вигляді суми добутків цифр цього числа, починаючи з цифри нульового розряду, на степінь числа 10 з показниками степені 0, 1, ... Цим показникам відповідають номери розрядів числа при їх підрахунку справа. Наприклад, число 123 десяткової системи числення можна подати у вигляді виразу, який зображає таку нумераційну функцію:

$$1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0.$$

При цьому обмеження на цифри для кожного розряду в цій функції *для десяткової системи числення мають вигляд:* $0 \le i \le 9$, де i = 0,1, ..., 9. У системах числення з іншою основою обмеження на цифри, які використовуються в них, будуть іншими, але вони обов'язково існують. Наприклад, у найпростішій на сьогодні позиційній системі числення – двійковій, тобто в системі, основа якої дорівнює 2, цифри можуть приймати значення тільки 0 і 1. Тоді число 123 в двійковій системі буде мати вигляд – 1111011.

Діапазон десяткової системи числення дорівнює степеню числа 10, показник якого визначається кількістю розрядів десяткового числа. Діапазон чисел системи числення, якому належить десяткове число 123, очевидно, дорівнює 1000.

У позиційних системах числення кожний розряд має свою *вагу*. Ця вага для десяткової системи числення дорівнюватиме степеню 10 з показником степені, що дорівнює номеру розряду, вага якого визначається. Для вищенаведеного прикладу ваги нульового, першого й другого розрядів будуть дорівнювати величинам 10^0, 10^1, 10^2. Тобто в цьому десятковому числі вага нульового розряду дорівнює 1, а вага кожного наступного розряду збільшується в 10 разів і буде дорівнювати 10 і 100 відповідно для першого й другого розрядів.

Однак вага *розряду* в деяких системах числення втрачає свій первісний зміст, а натомість з'являється вага *цифри* цього розряду. Ця вага для кожної цифри розряду зображає її кількісне значення, яке може відрізнятися від значень інших цифр цього розряду. Однак для десяткової, двійкової й інших подібних систем числення вона дорівнює вазі розряду. Тому для цих систем числення можна говорити про вагу розряду, розуміючи при цьому вагу цифри, і навпаки, говорячи про вагу цифри, можна розуміти при цьому вагу розряду.

Крім ваги цифри розряду, будемо розглядати ще і її *сумарну* вагу в цьому розряді, яка має вигляд суми ваг усіх менших від цієї цифри можливих цифр цього розряду. Їх кількість з урахуванням 0 буде такою, що відповідає цифрі числа цього розряду.

Сумарна вага цифри 0 завжди буде дорівнювати 0, оскільки додатні цифри, що менші за 0, відсутні. Цифра 1 показує, що в розряді, до якого вона належить, може бути лише *одна* менша від неї цифра – 0, вага якої в даному випадку й буде сумарною вагою цифри 1. Цифра 2 буде показувати, що меншими від неї є *дві* цифри – 0 і 1, ваги яких при підсумовуванні й створюють сумарну вагу цифри 2. Дія цього самого правила поширюється і на решту цифр числа.

Так, наприклад, для цифри 2 десяткового числа 123, яка належить до першого розряду, сумарна вага буде складатися із суми ваг двох менших цифр – 0 і 1. Вага кожної з них дорівнює 10, тобто сумарна вага цифри 2 числа 123 буде дорівнювати 20. Очевидно, що сумарна вага цифри 3 нульового розряду числа 123 буде складатися з ваг трьох менших цифр – 0, 1, 2, які в цьому розряді дорівнюють 1. Тому вона буде дорівнювати 1+1+1=3. Сумарна вага цифри 1 другого розряду відповідно до вищезазначеного, буде складатися з ваги попередньої цифри 0, яка дорівнює 100.

Також у такому десятковому числі, як 202, вага цифри 2 нульового розряду дорівнює 1, цифри 0 першого розряду – 10, цифри 2 другого розряду – 100, а сумарні ваги цих цифр дорівнюватимуть сумам кількісних значень менших цифр, які належать до цих розрядів. Так, у нульовому розряді цифра 2 має сумарну вагу, яка складається з двох кількісних значень менших цифр – 0 і 1. Їх ваги, очевидно, дорівнюють 1, і тому сумарна вага цифри 2 буде дорівнювати 2. Відповідно в першому розряді цифра 0 має сумарну вагу, яка дорівнює 0, цифра 2 в другому розряді має сумарну вагу, яка складається з ваг цифр 0 і 1, тобто дорівнює 200.

4. Унітарна система числення

Серед систем числення перехідне місце між позиційними і непозиційними системами числення займає система числення з алфавітом цифр, який складається з однієї цифри – *нуля*. Це значить, що основа й діапазон чисел цієї системи дорівнює 1. Така система числення інколи називається ***унітарною***. Наявність в її алфавіті лише нуля, тобто однієї цифри, призводить до того, що, наприклад, число з трьох розрядів унітарної системи числення, буде мати вигляд:

$$0 \cdot 1^2 + 0 \cdot 1^1 + 0 \cdot 1^0 = 0.$$

Така система числення кодує лише одне початкове число натурального ряду – одиницю. При цьому вага 0 кожного розряду дорівнює 1, а його сумарна вага дорівнюватиме 0.

Здавалося б, що унітарна система числення не може мати прикладного значення через те, що кодує лише одне число – 1. Але це не так. Ця система використовувалась раніше і використовується сьогодні в сучасній цифровій техніці при побудові різних особливо надійних пристроїв автоматики й обчислювальної техніки, тому що вона може бути подана з допомогою так званого *число-імпульсного* коду. В цьому коді кожне число зображається з допомогою послідовності імпульсів, кількість яких дорівнює кількісному значенню цього числа.

Так, у вищенаведеному прикладі унітарна система числення зображує число 3. Тому воно в реальному пристрої кодується трьома імпульсами. Головний недолік використання такої системи числення – це значне зниження швидкодії пристроїв, які її використовують. Але там, де швидкодія не має особливого значення, унітарна система числення може бути досить ефективною завдяки високій надійності апаратури, яка її використовує.

5. Системи числення, споріднені з десятковою

У розглянутих прикладах систем числення в кожному розряді їх чисел вага розряду визначалась степенем основи. Це найбільш простий і уживаний спосіб побудови систем числення. Але зустрічаються й більш складні системи числення, деякі з котрих будуть розглянуті нижче в цьому й інших розділах. Основою цих систем, як і раніше, є натуральні числа.

Наприклад, можна побудувати систему числення, яка ґрунтується на простих числах, таких як 1, 3, 5, 7, ... Візьмемо в цій системі числення для нульового розряду одну основу – 1, для першого розряду дві основи – 1 і 3, для другого розряду три – 1, 3, 5 і т. д. З цього виходить, що алфавіт цифр для 0 розряду буде складатися з трьох цифр – 0, 1, 2; першого розряду – з п'яти цифр – 0, 1, 2, 3, 4; другого із семи – 0, 1, 2, 3, 4, 5, 6.

Ваги розрядів при цьому отримаємо як добутки взятих у кожному розряді основ. Тоді вага нульового розряду буде дорівнювати 1, вага першого розряду – 3, вага другого розряду – 15.

Відповідно до наведеного число 220 буде мати такий вигляд:

$$2 \cdot 5 \cdot 3 \cdot 1 + 2 \cdot 3 \cdot 1 + 0 \cdot 1.$$

У десятковій системі числення це число, очевидно, буде дорівнювати 36. Сумарна вага його цифри 0 у нульовому розряді буде дорівнювати 0, цифри 2 першого розряду – 6, цифри 2 другого розряду – 30. Тому кінцевий результат після підсумовування всіх сумарних ваг буде дорівнювати 36. Найбільше число в цій системі числення отримується шляхом підстановки в усі розряди числа максимальних цифр – 6, 4 і 2. У результаті отримаємо десяткове число 104. Відповідно, діапазон чисел, існуючих у цій системі числення, буде дорівнювати 105.

Отримана система числення хоча й близька до десяткової, але все ж таки має суттєві відмінності. Основ у неї не одна, а декілька. Також алфавіт цифр змінюється від розряду до розряду. Відповідно, і їх ваги змінюються не за степеневим законом, як це має місце в десятковій системі числення, а за більш складним. Ще більше відмінностей мають узагальнені позиційні системи числення, початкова теорія яких викладена в наступних розділах.

Природа – це те, що підлягає обчисленню.
Шпенглер

Лекція 29
ПОЗИЦІЙНЕ КОДУВАННЯ ЧИСЕЛ

1. Розбиття множин

Позиційні системи числення, які розглядались у попередньому розділі, мають широке застосування на практиці, але вони відображають лише частину можливих їх структур. Щоб в узагальненому вигляді уявити всі ці структури, потрібно зрозуміти ідею, на якій ґрунтується їх побудова.

Такою ідеєю є введення послідовного *розбиття* деякої скінченної вихідної множини Q з числом елементів, більшим від одиниці на непорожні підмножини. Ця ж сама ідея є підґрунтям будь-якого іншого кодування інформації. Слід зазначити, що кодування чисел має свою специфіку, і тому можна говорити про відповідну цьому кодуванню теорію позиційних систем числення.

Розбиття на підмножини, які мають в них місце, відбуваються певними *кроками*, які в подальшому в системах числення розглядаються як відповідні цим крокам *розряди* чисел, а кількості елементів в підмножинах – як *ваги* їх цифр. У загальному випадку на кожному кроці розбивається більше, ніж одна отримана на попередньому кроці розбиття підмножина. Тільки на першому кроці розбивається лише одна вихідна множина Q. З розбиття цієї множини й починаються подальші розбиття отриманих на цьому кроці підмножин на нові підмножини до того часу, поки кожній одержаній під час розбиття підмножині не буде належати один елемент. Таке розбиття вихідної множини чи якоїсь з отриманих при цьому підмножин будемо називати *повним*.

Так, вихідна множина, до якої належить 100 елементів, при побудові десяткових чисел має повне розбиття, коли на першому кроці розбивається на 10 підмножин з 10 елементами в кожній, а на другому кожна з отриманих підмножин розбивається ще на 10 підмножин, тепер уже з одним елементом і на цьому розбиття підмножин закінчуються. Тобто дане розбиття буде повним і кожне з отриманих на їх основі число буде складатися з двох розрядів, у кожному з яких є дві цифри від 00 до 99.

Але в загальному випадку ніщо не забороняє ці ж самі 100 елементів розбивати на підмножини з різною кількістю елементів, і тоді при повному їх розбитті можна отримувати більш складні системи числення, ніж десяткова. Через свою складність подібні системи менш поширені на практиці, але все ж таки вони мають позитивні якості, які можуть з успіхом бути використані в сучасній обчислювальній техніці. Тому їх слід розглядати не тільки із теоретичних міркувань, а й з практичних теж.

2. Правило Суми

Щоб довести важливі властивості позиційного кодування чисел, необхідно сформулювати правило розбиття множин на підмножини, яке назвемо *правилом Суми.*

Якщо деяку скінченну множину елементів розбити на декілька підмножин, що не перетинаються між собою, то сума їх елементів дорівнюватиме числу елементів, які були в цій множині до її розбиття.

Це правило через його очевидність не визиває сумніву і в неявному вигляді, як аксіома, широко використовується на практиці. Але для цього навчального курсу був сенс сформулювати його таким чином, щоб потім свідомо використовувати його для вивчення способів побудови позиційних чисел. У більш загальному вигляді воно застосовується разом з правилом Добутку в комбінаториці, а також має пряму причетність до правила Діріхле.

Правило Суми відносно розбиття скінченних множин на підмножини є загальним. Воно поширюється в тому числі й на порожні множини, але більш важливим є випадок, коли при розбитті скінченної множини на підмножини порожні множини будуть відсутні. Далі завжди, коли йтиме мова про розбиття множин на підмножини, будемо вважати, що й множини і їх підмножини не є порожніми.

З наведеного правила випливає, якщо скінченна множина Q розкладається на декілька підмножин, то кожній з них повинно належати число елементів *менше*, ніж їх було в цій множині до її розбиття. Інакше сума елементів у цих підмножинах буде перевищувати число елементів у вихідній множині, що призведе до порушення цього правила.

Якщо ж продовжити розбиття отриманих підмножин на число, більше ніж одна підмножина, далі, то кількість елементів, які належать цій підмножині, буде відповідно до наведеного вище правила Суми зменшуватися і через певне число кроків виявляться лише *одноелементні* підмножини.

На цьому процедура розбиття вихідної множини на підмножини може або закінчитися, або, у разі необхідності, продовжитися, але вже у вигляді розбиття кожної з отриманих одноелементних підмножин на одну таку ж саму підмножину. Цій підмножині згідно з правилом Суми повинно належати стільки ж елементів, скільки й підмножині, яка розбивалась, тобто один елемент. Як наслідок, розбиття вихідної множини й після появи одноелементних підмножин може бути продовжене як завгодно довго.

3. Кодування чисел на першому кроці розбиття

Одноелементні підмножини можуть бути отримані вже на першому кроці розбиття, коли розбивається безпосередньо множина Q. У цьому випадку, крім цих підмножин, можуть бути отримані й підмножини з більшим числом елементів. Це відповідно до правила Суми можливо у випадку, якщо кількість підмножин, на які розбивається вихідна множина Q, буде меншою, ніж кількість елементів, які в ній містяться.

Візьмемо одну з одноелементних підмножин і вилучимо її зі списку множин першого кроку розкладів. Тоді згідно з правилом Суми загальна сума елементів, які залишилися в підмножинах першого кроку, буде на одиницю меншою від числа елементів вихідної множини. Відповідно й кількість елементів цієї множини зменшиться на 1.

Значення цієї загальної суми елементів підмножин полягає в тому, що вона може бути використана, по-перше, для кодування вилученої одноелементної підмножини, а отже, і відповідного елемента вихідної множини, і, по-друге, для встановлення номера (порядкового числа) цього елемента щодо інших елементів даної множини.

З наведеного вище випливає, що отриманий номер елемента і, відповідно, вилученої одноелементної підмножини буде дорівнювати числу елементів вихідної множини, зменшеному на одиницю. Це, очевидно, буде найбільше значення серед можливих номерів, які можуть упорядковувати елементи вихідної множини. Після кодування вилученої одноелементної підмножини даний номер закріплюється за нею й за відповідним елементом вихідної множини.

Цей номер, якщо до нього додати одиницю, буде відображати кількість елементів у вихідній множині. Це означає, що загальна сума елементів у підмножинах першого розбиття після вилучення з них однієї одноелементної підмножини не тільки стає найбільшим номером елемента, який кодується, а й дозволяє визначити *кількість* елементів у вихідній множині.

Однак на першому кроці розбиття може бути отримана не одна одноелементна підмножина, а дві й більше, і навіть може трапитися таке, що всі підмножини будуть одноелементними. Розглянемо цей хоча й надзвичайний, але корисний для аналізу позиційного кодування випадок.

Якщо всі підмножини, на які була розбита вихідна множина під час першого кроку розбиття, будуть одноелементними, то тоді серед них обирається якась одна підмножина і відповідно до розглянутого вище правила кодується номером, який на одиницю буде меншим від суми елементів у цій множині. Після цього вона вилучається з розгляду.

З підмножин, що залишилися після першого кодування однієї з них, вибирається нова одноелементна підмножина і знову кодується номером, який тепер уже буде на одиницю меншим від попереднього. Далі ця підмножина вилучається з загальної їх множини. Таким чином буде встановлений номер ще одного елемента вихідної множини і одночасно, якщо додати до цього числа 1, визначена кількість елементів, що залишились у вихідній множині після першого кодування.

Подібне кодування із встановленням порядкових номерів і вилученням одноелементних підмножин буде продовжуватися до того часу, поки воно не дійде до останньої одноелементної підмножини. У цьому випадку вже не залишиться одноелементних підмножин, сумарна кількість елементів яких визначала б номер останньої підмножини, крім порожньої множини, яка присутня всюди, у тому числі і в ряду одноелементних підмножин.

Відомо, що в ній відсутні елементи, тобто їх кількість дорівнює 0. Тому остання одноелементна підмножина, яка ще не має свого кодового номеру, може кодуватися в даному випадку тільки *нулем*. У цьому кодуванні виявляється особлива роль 0, без якого неможливі позиційні системи числення. Розглянемо більш докладно цей випадок.

Наявність 0 вказує на те, що попереду нього є 0 множин і тому потреба в їх кодуванні відсутня. Він є першим порядковим числом чи номером, з якого починається перелік підмножин. Тобто 0 створює базу переліку елементів вихідної множини при позиційному кодуванні. При цьому важливо також і те, що нуль, хоча й указує на порядок, але одночасно кодує одноелементну множину.

Після кодування початкової чи базової підмножини нулем наступна за нею підмножина буде кодуватися числом, яке буде дорівнювати кількості елементів у підмножині, кодованій 0. Оскільки кількість елементів у цій одноелементній підмножині дорівнює 1, то і її порядковий номер буде дорівнювати 1. Номер третьої одноелементної підмножини визначається вже сумарною кількістю елементів у двох одноелементних підмножинах, кодованих 0 і 1, які йдуть попереду підмножини, що кодується, тобто він буде дорівнювати 2.

Те ж саме стосується будь-якого великого номера, що кодує одноелементну підмножину. Цей номер також визначається кількістю одноелементних підмножин, які йдуть в їх упорядкованому ряді попереду одноелементної множини, що кодується.

Наприклад, якщо попереду одноелементної підмножини, порядковий номер якої визначається, іде 120 одноелементних підмножин, номера яких уже визначені, це означає, що номер цієї підмножини, а отже, і елемента вихідної множини, буде дорівнювати 120 і, відповідно, номера всіх підмножин, які передують їй, будуть змінюватися від 0 до 119. При цьому сумарна кількість елементів у всіх одноелементних підмножинах до 119 підмножини включно буде дорівнювати 120.

Таким чином, у наведеному прикладі маємо 120 порядкових номерів 0, 1, ..., 119, які відображають номера одноелементних підмножин, і, відповідно, задається порядок розташування 120 елементів у вихідній множині.

4. Кодування чисел на двох кроках розбиття

З викладеного вище матеріалу випливає, що розбиття вихідної множини на одноелементні підмножини тільки на першому кроці може призвести до необхідності запам'ятовувати велику кількість номерів. У результаті виникає потреба в пам'яті з такою ж великою ємністю, яку не завжди можна легко реалізувати.

Ця потреба є основним недоліком кодування елементів вихідної множини тільки на одному кроці. Але зазначений недолік можна усунути, якщо збільшити число кроків розбиття вихідної множини, і тим самим обмежити в кожному розбитті кількість підмножин, які нумеруються досить невеликим числом. Отримані при кодуванні цих підмножин номери потребують для свого зберігання значно меншої ємності пам'яті, і тому їх можна досить легко запам'ятати, а потім уже використовувати для побудови кодових зображень чисел.

Припустимо, що в розглянутому вище прикладі вихідній множині належить не 120, а 140 елементів і при цьому вона на першому кроці розбивається, як і раніше, на те ж саме число підмножин – 120. Тоді згідно з правилом Суми хоча б одній із цих підмножин повинно належати елементів більше ніж один. Візьмемо одну таку підмножину, яка кодується 0, і будемо вважати, що їй належить 21 елемент. Будемо також вважати, що при цьому кількість елементів, що належить кожній підмножині з номерами 1, 2, ...,119, буде дорівнювати одиниці. Відповідно, номерами їх елементів будуть числа 22, 23, ..., 140.

У цьому випадку неможливо пронумерувати всі елементи вихідної множини протягом одного кроку. Для того, щоб нумерація елементів була можливою, виконуються подальші кроки розбиття отриманих на першому кроці підмножин на нові підмножини до того моменту, поки всі вони не стануть одноелементними, тобто поки не буде здійснене повне розбиття вихідної множини.

Відповідне до цього розбиття позиційне число буде мати в результаті стільки розрядів, скільки було виконано кроків до появи одноелементної підмножини. Тому з допомогою розбиття, яке виконується шляхом більше ніж одного кроку, можна зобразити будь-яке велике число. Хоча, правда, це потрібно буде відшкодувати додатковими розрядами числа, кількість яких, очевидно, буде поступово збільшуватися зі зростанням кількості чисел, які зображаються з їх допомогою порівняно з повним розбиттям вихідної множини на одному кроці на одноелементні підмножини. Однак це зростання розрядів буде не таким стрімким, як зростання кількості номерів при повному розбитті, де кодування виконується за один крок, тобто при непозиційному кодуванні.

При кодуванні, яке відбувається більше ніж за один крок, кожну одноелементну підмножину, яка залишилася після першого розкладення, можна або розбити на другому кроці ще на одну таку ж саму одноелементну підмножину, або не розбивати. В останньому випадку, коли одноелементну підмножину отримують один раз, номери, які кодують одноелементні підмножини, створюють числа довжиною в один розряд. Тому в кінцевому підсумку ці числа сумісно з числами з більшою кількістю розрядів будують множину чисел з різною довжиною, тобто вони будуть належати до *нерівномірних* кодів. Крім того, їх кількість буде дорівнювати кількості елементів у вихідній множині.

Це випливає з того, що відповідно до правила Суми всі одноелементні підмножини кожного розбиття, створені на другому кроці, будуть мати сумарну кількість елементів, яка дорівнює кількості елементів у підмножинах, з яких вони були отримані при розбитті на першому кроці. Але ці кількості спільно з одноелементними підмножинами першого кроку розбиття на основі того ж самого правила Суми дорівнюють кількості елементів вихідної множини. Тому такою ж самою буде кількість буде чисел з одним або двома розрядами, побудованих після повного розбиття на першому і другому кроці.

Якщо ж на другому кроці розбивати не тільки отримані раніше не одноелементні підмножини, а й одноелементні, що побудовані на першому кроці розбиття, а також якщо в кінцевому підсумку всі отримані підмножини стануть одноелементними, то на основі цього розбиття можна створити числа з двома розрядами. У такому випадку отримані числа будуть належати до *рівномірних* чисел. Їх кількість, як і кількість нерівномірних чисел, буде дорівнювати кількості елементів вихідної множини. Доведення, що це дійсно так, ґрунтується на тих самих міркуваннях, що й для нерівномірних чисел.

Щоб краще зрозуміти викладений вище матеріал, розглянемо такий приклад. Припустимо, що вихідна множина з 12 елементів розбивається на підмножини впродовж двох кроків. На першому з них розбиття відбувається на дев'ять непорожніх підмножин, серед яких є вісім одноелементних, а на другому розбивається одна підмножина, до якої належать чотири елементи. Домовимося, що на першому кроці ця підмножина з чотирма елементами кодується 0, а решта вісім одноелементних підмножин кодуються в довільному порядку номерами 1, 2, ...

Тоді на першому кроці отримаємо послідовність номерів 0,1, ..., 8, які кодують усі без винятку отримані після першого розбиття підмножини, а на другому кроці отримана чотирьохелементна підмножина ще раз розбивається на підмножини, які тепер уже будуть одноелементними. Згідно з правилом Суми кількість цих одноелементних підмножин буде дорівнювати чотирьом. У результаті отримаємо повне розбиття вихідної множини.

Якщо на другому кроці кодування одноелементних підмножин, які з'являються після розбиття підмножини з чотирма елементами, буде виконуватися номерами 0, 1, 2, 3, то отримаємо такі зображення 12 елементів вихідної множини – 00, 01, 02, 03, 1, 2, 3, ..., 8 у вигляді чисел. Очевидно, що чотири числа з них будуть мати два розряди, а решта – один розряд. Тобто матиме місце нерівномірне кодування чисел.

Усі ці зображення чи числа створюють префіксний код, тому що жодне з них не є префіксом іншого. У результаті їх легко розпізнати в створеній ними послідовності, наприклад, такій, як 10022802103. Дійсно, спочатку однозначно розпізнається 1 як окреме число, яке має у своєму складі одну цифру, потім число 00, за ним 2 і так далі до числа 03.

У випадку рівномірного кодування вісім одноелементних підмножин, отриманих на першому кроці кодування, ще раз розбиваються на одноелементні підмножини на другому кроці і кодуються 0. Їх кодові зображення разом із зображеннями чотирьох одноелементних підмножин, які були одержані на першому кроці розбиття, будуть мати вигляд – 00, 01, 02, 03, 10, 20, 30, ..., 80. Тобто, мінімальне число 00 відображає 0, а максимальне 80 – суму двох чисел 8 + 4 = 12.

Ці зображення чисел ще краще, ніж у разі нерівномірного кодування, відокремлюються одне від одного в їх послідовності 1000202080002010003, яка була зроблена на основі попередньої, оскільки для цього досить порахувати по дві цифри й отримати в результаті цієї процедури окремі числа. Крім того, вони будуть більш перешкодостійкими. Так, наприклад, якщо після двох 2, які йдуть підряд, буде йти будь-яка із цифр – 1, 2, ..., 8, то це вже свідчить про помилку. Але в такій послідовності цифр їх кількість буде значно більшою, ніж у попередній для нерівномірних чисел, що взагалі є недоліком рівномірного кодування стосовно нерівномірного.

5. Позиційні числа з довільним розбиттям

Якщо після другого кроку розбиття нові підмножини містять більше ніж один елемент, то повинна бути виконана процедура їх третього, четвертого, а потім подальших кроків розбиття, які будуть здійснюватися до того моменту, поки всі отримані в результаті розбиття підмножини повністю не стануть одноелементними. Як тільки це станеться, відповідні цьому розбиттю числа слід вважати сформованими.

При цьому ці числа можуть належати як до рівномірних кодів, так і до нерівномірних. Усе залежить від розбиття одноелементних множин. Якщо вони після того, як були отримані, далі розбиваються на самих себе спільно з неодноелементними множинами, тоді ці числа можуть стати рівномірними. А якщо розбиття одноелементних множин після їх отримання припиняються, а розбиття неодноелементних підмножин ще продовжуються, тоді в кінцевому результаті будуть сформовані нерівномірні числа.

Для сформованих тим чи іншим способом чисел, з точки зору їх кількісного значення, не дуже важливо, до яких кодів вони належать – рівномірних чи нерівномірних. Для них важливим є лише те, щоб їх кількісні значення були одними й тими ж самими. Все інше стосується лише форм зображення цих кількісних значень.

Однак ці форми в багатьох випадках відіграють неабияку роль під час виконання різних практичних завдань. Тому на практиці питанню вибору форми зображення чисел, тобто способу їх кодування в позиційних системах числення, приділяється велика увага. А для цього необхідно дослідити всі можливі способи такого кодування чисел, навіть у межах одного й того ж самого розбиття вихідної множини, тобто їх *структури*.

Кількість одноелементних підмножин при повному розбитті вихідної множини повинна дорівнювати кількості елементів у вихідній множині. Це випливає з того, що на кожному кроці розбиття всіх наявних підмножин включно з одноелементними підмножинами нові підмножини відповідно до правила Суми містять результуючу кількість елементів, яка буде дорівнювати сумарній кількості елементів усіх підмножин, які були розбиті. Ці розбиті підмножини, у свою чергу, будуть мати загальну кількість елементів, що також буде дорівнювати сумі попередніх їм підмножин, з яких вони були отримані, і так буде продовжуватися до вихідної множини.

Таким чином, можна стверджувати, що кількість одноелементних підмножин, яка підрахована спільно з уже отриманими раніше такими ж підмножинами на більш ранніх кроках розбиття, буде також дорівнювати кількості елементів вихідної множини.

Більше того, з розглянутих вище співвідношень випливає, що на кожному кроці розбиття сумарна кількість елементів у підмножинах, які розкладаються, разом з одноелементними підмножинами буде дорівнювати кількості елементів вихідної множини.

Математика – це один із способів водити себе за ніс.

Ейнштейн

Лекція 30
ЧИСЛОВА ФУНКЦІЯ

Отримані вище розбиття одних підмножин на інші, які продовжуються до одноелементних, і відношення належності між ними дозволяють побудувати довільні позиційні системи числення. Елементи таких систем будуть утворені в процесі розбиття підмножини, а зв'язки між ними створюють структури цих систем. Якщо при цьому ще задати кількість елементів у кожній із підмножин і відповідним чином ввести їх нумерацію для кожного розбиття, то можна вважати, що основні елементи системи числення є в наявності.

Тепер залишилось лише отримати числову функцію системи числення, яка б дозволяла за зображеннями чисел знаходити самі ці числа (номера). Задача побудови такої числової функції і обмежень на її параметри в багатьох спеціальних випадках є досить складною, але в загальному випадку вона досить просто розв'язується таким чином:

Для того, щоб знайти за зображенням числа в позиційній системі числення номер елемента вихідної множини, треба на кожному кроці розбиття в одному з них знайти підмножину, номер якої дорівнює цифрі відповідного даному кроку розряду числа, і потім у цьому розбитті знайти сумарну кількість елементів у всіх його підмножинах, номера яких будуть меншими від цієї цифри, а далі знайти загальну суму всіх сум, отриманих таким чином на кожному кроці розбиття. Ця загальна сума й буде порядковим номером зображеного числа.

Доведемо можливість цього рішення. Дійсно, якщо число правильне, то на першому кроці розбиття серед всіх підмножин, пронумерованих номерами 0,1, ..., знайдеться підмножина з номером, який буде співпадати з відповідною цій підмножині першою цифрою числа. Якщо ця підмножина одноелементна, то якими б не були всі інші підмножини з меншими номерами – одноелементними чи ні, її номер буде визначатися сумарною кількістю елементів у цих підмножинах.

Оскільки цей номер дорівнює відповідній цифрі числа, то й сумарна вага цієї цифри буде визначатися також сумарною кількістю. Але підмножина, яка розглядається, є одноелементною. Тому й число в даному випадку формується цією однією цифрою, а її сумарна вага відображає кількісне значення цього числа.

Якщо підмножина, що розглядається, не одноелементна, то тоді її номер і відповідна цифра числа визначає лише частину його кількісного значення. Інша частина, якої бракує, може бути знайдена при розбитті цієї неодноелементної підмножини на наступному кроці розбиття. У цей час з'являються нові підмножини з номерами 0, 1, ...

Далі виконуються послідовні порівняння цифри сусіднього меншого розряду числа з отриманими номерами. Унаслідок таких порівнянь буде знайдена підмножина з номером, який відповідає цій цифрі. Сума елементів всіх інших підмножин, номера яких менші від номера знайденої підмножини, створює сумарну вагу відповідної цифри.

Потім до неї додається отримана раніше сумарна вага попередньої цифри старшого розряду, і, якщо підмножина, номер якої збігається з номером цифри, що визначається, буде одноелементною, на цьому кількісне значення числа, яке визначає відповідний йому номер елемента вихідної множини, вважається знайденим, інакше цикл визначення величини цифри числа повторюється, але вже для цифри наступного меншого розряду числа. Він буде продовжуватися до того моменту, поки при подальшому розбитті не отримають одноелементну підмножину, а, що це рано чи пізно станеться, випливає, як було показано вище, з правила Суми.

Наслідок 1. *Серед чисел позиційної системи числення обов'язково є число, яке дорівнює нулю.*

Це випливає з того, що кожне розбиття обов'язково має підмножину, з якої починається і яка повинна нумеруватися нулем. Серед таких підмножин при подальшому розбитті обов'язково на якомусь його кроці з'явиться одноелементна підмножина і тоді сформується число, цифри якого складаються лише з нулів.

Кожна підмножина, яка отримана в процесі розбиття і кодується нулем, не має підмножин з меншими номерами і тому містить сумарну кількість елементів в них, яка дорівнює *нулю*. Відповідно й *сумарні* ваги цифр числа, яке відображає нуль, будуть такими ж самими, звідки випливає, що кількісне значення цього числа дорівнює *нулю*.

Наслідок 2. *Найбільше число позиційної системи числення на одиницю менше від кількості всіх можливих чисел, які належать їй.*

Це твердження ґрунтується на тому, що діапазону чисел, які представляє позиційна система числення, обов'язково належить 0.

Відповідно до наслідку 2 можна зробити висновок, що якщо діапазон чисел складається з одного числа, то це число може бути тільки 0. Тобто 1 кодується 0.

Таким чином, з огляду на викладене вище числова функція для кодування числа використовує три ідеї:

1. Введення на кожному кроці розбиття *номерів*, які нумерують їх підмножини в порядку 0, 1, ... При цьому номер 0 є обов'язковим елементом, з якого починається кодування підмножин, а інші номери повинні йти тільки в зростаючому порядку натуральних чисел без винятків.

2. Знаходження *сумарної* ваги кожної цифри числа. Її знаходять як суму елементів підмножин відповідного їй розбиття, номера яких менші, ніж надана цифра числа.

3. Знаходження кількісного значення числа як *суми* сумарних ваг усіх його цифр.

Ці ідеї прості й ефективні. Більш досконалі ідеї порівняно з ними сьогодні відсутні. Тому позиційне кодування чисел витіснило всі інші непозиційні методи кодування, як менш досконалі. Людство йшло до цих ідей досить довго й почало застосовувати їх не так уже й давно – менше ніж 1000 років назад – у вигляді позиційних систем числення (таких, наприклад, як десяткова).

Викладені вище ідеї, що пов'язані з числовою функцією, можна подати з допомогою відповідних формул. Однак вони досить громіздкі й будуть обтяжувати текст, що недоцільно для підручника, який адресований студентам технічних спеціальностей. Важливо познайомити їх лише з основними ідеями. Більш докладно ці питання викладаються в спеціальній літературі, з якою в разі необхідності можна ознайомитися. Крім того, студент, знайомий з цими ідеями, може й сам отримати необхідні формули, демонструючи тим самим свої здібності до наукової праці. На основі цих формул можна навіть розробити нові системи числення, які ще невідомі науковому загалу.

Число – знак досконалого обмеження.

Шпенглер

Лекція 31
ПОЗИЦІЙНІ СИСТЕМИ ЧИСЛЕННЯ

1. Кодування підмножин

Позиційні системи числення, у тому числі й ті з них, які розглядалися раніше, характеризуються деревовидною структурою, яка утворюється в процесі послідовного *розбиття* деякої скінченної множини елементів $Q=\{q_1,\ q_2,\ldots,q_P\}$ за кроками $j=1,2,\ldots,n$ на підмножини (класи еквівалентності) у загальному випадку з різною кількістю елементів до створення в кожній з них *одного* елемента.

Підмножини кожного розбиття l на кроці j кодуються з допомогою послідовності номерів $x_{jl}=0,1,\ldots,\ N_{jl}$. Це кодування підмножин обов'язково починається з нуля, а далі йде в зростаючому порядку. При цьому кожне розбиття має свою послідовність кодуючих його підмножини номерів – 0, 1, ..., яка відрізняється від інших послідовностей тільки їх кількістю.

Якщо на кожному кроці розбиття, починаючи з першого, вибирати одну підмножину, до якої належить елемент вихідної множини, що кодується, то отримаємо послідовність підмножин

$$X=x_1,x_2,\ldots,x_j,\ldots,x_n,$$

у кінці якої в разі повного розбиття вихідної підмножини буде стояти одноелементна підмножина.

Ця послідовність номерів кодує заданий елемент вихідної множини однозначно. Жодна інша послідовність номерів при отриманому розбитті не зможе його кодувати.

У результаті з допомогою послідовності номерів X буде отримана вся інформація про місце цього елемента серед елементів вихідної множини. Поряд з цим буде визначена сумарна *кількість* елементів у вихідній множині.

2. Позиційні числа

Послідовність номерів $X=x_1,\ x_2,\ \ldots,\ x_j,\ldots,x_n$ можна з допомогою рівності $j=n-\gamma,\ \ \gamma=1,\ 2,\ \ldots,\ n$, подати у вигляді

$$X=x_{n-1}x_{n-2}\ldots x_i\ldots x_0\,.$$

Ця послідовність номерів має назву позиційного *числа,* її номери x_i називаються ще *цифрами,* а індекси цих цифр $i = 0, 1, \ldots, n-1$ – *розрядами.* Кожна з цифр належить відповідній їй упорядкованій у зростаючому порядку множині – *алфавіту* цифр.

Кількість алфавітів цифр у системі числення загального виду буде дорівнювати кількості впорядкованих в ній (системі) нумерацій підмножин і різних розбиттів одних підмножин на інші відповідно.

Щоб звести алфавіти цифр у позиційній системі числення до узагальненого алфавіту, слід знайти в ній розбиття з найбільшою кількістю підмножин і відповідно найбільшою кількістю номерів, які ці підмножини кодують. Ці номери створюють узагальнений алфавіт і в подальшому будуть розглядатися як цифри позиційних чисел у даній системі числення.

Число цифр у алфавітах системи числення, функція чи будь-яка інша умова її побудови називається ***основою***. У загальному випадку такою основою будуть усі алфавіти системи числення, які отримані в процесі розбиття вихідної множини на підмножини на всіх його кроках.

На відміну від десяткової системи числення й споріднених з нею позиційних систем кількість розрядів в узагальнених числах, тобто числах, цифри яких належать різним алфавітам, може бути різною, що уможливлює нерівномірність їх довжин. При цьому ці числа створюють префіксний код. Щоб уникнути цієї нерівномірності довжин, потрібно лише дописати до молодших розрядів, менших за довжиною чисел, нулі в такій кількості, щоб їх (чисел) довжина дорівнювала числу з найбільшою довжиною.

Кількість елементів у підмножині, яка з'являється після розбиття однієї з множин і кодується тим чи іншим номером, як і раніше, будемо називати ***вагою*** однойменної цьому номеру цифри числа, а суму значень ваг всіх підмножин даного розбиття з меншими номерами, починаючи з нуля, її ***сумарною*** вагою.

Тобто, якщо маємо цифру 5 в одному з розрядів числа, то це означає, що до підмножини, яка кодується відповідним номером 5, у розбитті ідуть п'ять підмножин, у загальному випадку з різною кількістю елементів, з номерами 0, 1, 2, 3, 4. Сумарна кількість елементів у цих підмножинах створює сумарну вагу цифри 5, а кількість елементів у підмножині з номером 5 буде вагою цієї цифри.

Очевидно, що в позиційних системах числення при кодуванні того чи іншого розбиття, перед підмножиною, яка кодується нулем, є лише пуста підмножина з нульовою кількістю елементів. Тому сумарна вага нуля в будь-якому числі завжди дорівнює нулю, але при цьому вага безпосередньо самого нуля буде більшою за нуль. Інші ж цифри чисел мають на відміну від нуля сумарну вагу, яка не дорівнює нулю. Але ваги цих цифр завжди будуть меншими або, у крайньому разі, дорівнювати вазі відповідної їм цифри сусіднього старшого розряду числа, до якого вони безпосередньо належать.

З усього розглянутого вище випливає, що головним чинником у побудові узагальненої позиційної системи числення є кількість підмножин у розбитті вихідної множини на підмножини, узятому за всіма кроками. Якщо це розбиття відбувається на кожному кроці на однакову кількість підмножин, наприклад, на 10, то в результаті отримують систему числення з найпростішою структурою. І навпаки, коли відбувається розбиття на підмножини з неоднаковою кількістю підмножин, отримують узагальнену систему числення. У ній буде не один алфавіт цифр, а декілька.

Саме ці алфавіти й утворюють основу системи числення, оскільки з них вибирають цифри, що формують числа, а це для системи числення є головним. Тому опис алфавітів у вигляді формул з обмеженнями щодо кількості цифр у їх змінних або таблиць з обмеженнями щодо кількості в них цифр має велике значення для побудови систем числення.

Побудована система числення дозволяє створювати зображення чисел і, у разі необхідності, виконує з допомогою числової функції їх нумерацію. Створюються ці зображення конкретними цифрами, що співпадають з номерами, які кодують отримані в разі повного розбиття підмножини. Найчастіше їх записують у вигляді лінійної послідовності цифр.

3. Приклад системи числення з довільним розбиттям

Розглянемо приклад побудови однієї з узагальнених позиційних систем числення, для якої розбиття вихідної множини в загальному випадку здійснюється на підмножини з нерівною кількістю елементів. Припустимо, що розбивається вихідна множина {1, 2, 3, 4, 5}, наприклад, на дві підмножини {1, 2} і {3, 4, 5}. Це буде першим кроком розбиття. Кожну з отриманих підмножин розіб'ємо далі на другому кроці таким чином: підмножину {1, 2} на підмножини {1} і {2}, а підмножину {3, 4, 5} на підмножини {3} і {4, 5}. Потім підмножину {4, 5} розіб'ємо на підмножини {4} і {5}, і на цьому розбиття підмножин закінчується, оскільки всі отримані в його процесі підмножини мають по одному елементу.

У вищенаведеному прикладі вихідна множина, очевидно, повинна кодуватися нулем. Далі одна з одержаних при розбитті підмножин, наприклад, $\{1, 2\}$ кодується нулем, а друга $\{3, 4, 5\}$ – одиницею. Потім у розбитті підмножини $\{1, 2\}$ її підмножину $\{1\}$ кодуємо нулем, а підмножину $\{2\}$ – одиницею. Аналогічно кодуються підмножини $\{3\}$ і $\{4, 5\}$ при розбитті підмножини $\{3, 4, 5\}$, тобто нулем і одиницею, і підмножини $\{4\}$ і $\{5\}$ при розбитті підмножини $\{4, 5\}$. У результаті, якщо не враховувати нуль, який кодує початкову множину, отримаємо п'ять чисел 00, 01, 10, 110, 111.

Очевидно, що ці числа мають різну довжину і що кожне з них не є початком іншого числа, тобто ці числа мають властивість префіксності і разом створюють префіксний код. Їх можна в разі необхідності перетворити на рівномірні числа, якщо дописати нулі справа таким чином, щоб загальна кількість цифр у всіх п'яти числах дорівнювала кількості чисел найбільш довгого числа, тобто трьом. Це будуть такі числа – 000, 010, 100, 110, 111.

У такому разі можна говорити про довжину кожного числа, яка дорівнює трьом розрядам, хоча інформаційними розрядами для перших трьох чисел будуть лише два розряди. Надлишкові розряди можуть бути використані для пошуку помилок у зображеннях цифр і тому в багатьох випадках є корисними. Наприклад, якщо під час операцій над числами, які розглядаються, буде отримане число 001, це означає, що в їх процесі була зроблена помилка. Крім того, з їх допомогою можна отримувати більш просту цифрову апаратуру при зображенні чисел і виконанні над ними арифметичних і логічних операцій.

Знайдемо ваги цифр одного з отриманих у розглянутому вище прикладі числа, наприклад, 110. Очевидно, що вага цифри 0, яка стоїть в нульовому розряді, буде дорівнювати одиниці, а сумарна вага – нулю. Вага цифри 1 першого розряду буде дорівнювати двом, а сумарна – одиниці. У другому розряді вага 1 дорівнює трьом, а сумарна – двом. Третій розряд – вихідний, тобто той, з якого розпочалося розкладення. У ньому стоїть єдина цифра – 0. Очевидно, що його вага буде дорівнювати п'яти, а сумарна – нулю. Узагалі його можна не писати, тому що в цьому розряді інших цифр, крім 0, бути не може.

Сума сумарних ваг 0, 2, 1 усіх розрядів дає кількість, яка відповідає кодовому зображенню числа – 110. Ця кількість, очевидно, дорівнює трьом. Аналогічно можна знайти кількісні еквіваленти для решти кодових зображень чисел. Так, число 00 зображує кількість, яка дорівнює 0, 01 – 1, 10 – 2, 111 – 4. Таким чином, з допомогою послідовного розбиття вихідної множини на підмножини за вищенаведеним правилом можна отримати всі числа з їх діапазону, який має п'ять чисел, починаючи з 0 і закінчуючи 4.

Як видно з вищенаведених прикладів, для виконання нумерації кількостей у системах числення, крім їх кодових зображень – чисел, потрібна ще *нумераційна,* або *числова* функція. Ця функція реалізує операцію підсумовування сумарних ваг цифр усіх розрядів числа. Вона має стільки змінних, скільки розрядів має те чи інше число, що належить до діапазону даної системи числення.

У вищенаведеному прикладі таких змінних для трьох чисел буде дві, а для решти двох чисел – три. Кожна з цих змінних належить до того чи іншого розряду нумераційної функції і приймає значення відповідної їй цифри числа. Ця цифра має свою сумарну вагу, з допомогою якої далі в процесі її підсумовування з сумарними вагами цифр інших розрядів знаходять кількісне значення числа, тобто його номер.

Цілі числа створив Бог, а все інше – діло рук людських.

Кронекер

Лекція 32
КЛАСИФІКАЦІЯ ПОЗИЦІЙНИХ СИСТЕМ ЧИСЛЕННЯ

З метою ефективного використання систем числення в теорії і на практиці важливо їх класифікувати. Будемо робити це, використовуючи наданий вище матеріал. У першу чергу розглянемо системи числення, які генерують числа однакової довжини.

Системи числення з рівною довжиною чисел належать до класу *рівномірних* кодів. Однакову довжину числа мають лише тоді, коли підмножини, які отримують в системах числення на кожному кроці розбиття вихідної множини, містять однакове число елементів.

Такі системи числення назвемо ***однорідними***. Їх також ще називають ***природними***, або ***степеневими***. Як уже зазначалось, характерною ознакою таких систем є однаковість їх чисел за довжиною. Крім цього, другою не менш важливою особливістю цих систем є те, що вага розрядів в них змінюється згідно зі степеневим законом. До цих систем числення належать двійкова, десяткова, п'ятерична і безліч подібних інших. За їх основи беруть числа 2, 10, 5 і т. д.

Розроблення більш складних, ніж однорідні, позиційних систем числення почалося в основному в другій половині 20-го століття після того, як з'явилася цифрова обчислювальна техніка. Такі системи назвемо ***неоднорідними***. Вони використовувались здебільшого при побудові спеціалізованих обчислювачів, систем зв'язку та керування, кодуючих та декодуючих пристроїв з метою підвищення їх ефективності.

Найпростішими неоднорідними системами числення є системи, в яких кількість елементів в усіх підмножинах, отриманих на попередньому кроці розбиття, буде однаковою. При цьому встановлюється функціональний зв'язок між номером кроку розбиття й числом підмножин у розбитті на цьому кроці.

Ваги цифр, які належать до одного розряду числа, у цьому випадку *рівні* між собою, однак вони на відміну від однорідних систем числення *змінюються* від розряду до розряду не за степеневим законом, як це має місце для однорідних систем числення, а за більш складним. Числа для неоднорідних систем числення з такими обмеженнями мають, як і для однорідних, рівну довжину. Прикладом таких систем числення є факторіальні, а в більш загальному випадку системи зі змішаною основою, чи поліадичні.

У результаті подальшого розвитку таких і їм подібних позиційних систем числення і отримання нових їх структур виникла задача дослідження загальних принципів їх побудови, рішення якої дозволило б виконати більш детальну класифікацію позиційних систем числення, одержати нові їх класи та розробити ефективні системи числення. Як правило, ці системи числення мають числа з рівною довжиною чи кількістю розрядів, що спричинене однаковою кількістю елементів у підмножинах на тому чи іншому кроці розбиття.

Однак узагалі підмножини, які виникають на конкретному кроці розбиття, можуть містити різну кількість елементів, і тому довжина чисел у побудованих таким чином системах числення буде різна. Причому, жодне число в цьому випадку не може бути початком чи префіксом іншого. Це випливає з того, що остання цифра кожного числа кодує підмножину, яка містить *один* елемент, і тому подальше розбиття цих підмножин на менше число елементів стає неможливим. А оскільки це так, продовження числа в інформаційному плані відсутнє, і, отже, воно не може бути початком жодного іншого числа.

Системи числення з різною довжиною чисел далі будуть називатися ***структурними*** системами числення. Їх характерною ознакою буде різна довжина чисел, які до них належать. Вони, з точки зору теорії кодування, належать до *нерівномірних,* або *префіксних,* кодів і тому мають більш складну структуру, ніж звичайні рівномірні коди.

У структурних системах числення ваги цифр чисел залежать як від номера розряду, де вони стоять, так і від попередніх цифр у числі. Прикладом структурних систем числення є *біноміальні*, які будуть розглядатися нижче.

Системи числення, основами яких обрані різні комбінаторні співвідношення, будемо називати ***комбінаторними.*** Прикладом комбінаторних систем числення є факторіальні системи, які згадувались вище, (вони ґрунтуються на факторіалах), поліадичні, основами яких є різні числа, кількість яких більша ніж одне, і біноміальні, що використовують як свою основу функції для обчислення біноміальних коефіцієнтів.

Якщо кількість підмножин у розбиттях є випадковою, тобто підмножини виникають поза будь-яких правил, то числа в таких системах числення матимуть у загальному випадку різну довжину і відповідно системи числення будуть належати до структурних систем. Назвемо такі системи числення ***табличними***, оскільки задати їх можна лише за допомогою таблиць. Для таких систем числення будь-які закономірності їх побудови відсутні.

На рис. 4.1 у вигляді блок-схеми наведена класифікація позиційних систем числення.

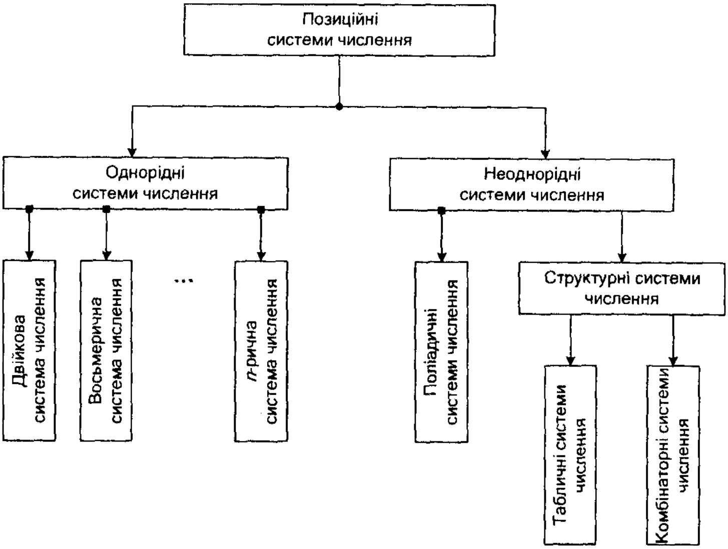


Рис. 4.1. Класифікація позиційних систем числення

Таким чином, позиційні системи числення розподіляються на два великих класи – ***однорідні*** (з рівною довжиною чисел і основою в вигляді натурального числа) і ***неоднорідні*** (з рівною та нерівною довжиною чисел і більш складною основою, ніж натуральні числа). Однорідні системи числення відповідно до числа, яке взяте за їх основу, у свою чергу, поділяються на двійкові, трійкові, десяткові й т.д. Неоднорідні поділяються на системи зі *змішаною* основою, або поліадичні, і *структурні* – з числовою або функціональною основою. Останні, у свою чергу, поділяться на *комбінаторні* і *табличні*.

Всі позиційні системи числення без винятку можуть бути подані у вигляді дерев розбиття, вершини яких відображають кількість елементів у підмножинах, що розбиваються, а гілки – номери, що кодують підмножини, які виникають після розбиття. При цьому послідовності цих номерів утворюють числа позиційних систем числення. Номери в даному випадку є цифрами чисел.

Лекція 33
СТРУКТУРИ ПОЗИЦІЙНИХ СИСТЕМ ЧИСЛЕННЯ

Розглянемо більш докладно структури різних систем числення. Для кожної з таких структур необхідно задати числову (нумераційну) функцію, з допомогою якої виконується нумерація чисел, і обмеження на цифри, які в ній використовуються. На рис. 4.2, 4.3, 4.4 наведені приклади систем числення, серед яких найбільш простою є двійкова. Остання зображена на рис. 4.2. Її діапазон $P = 8$.

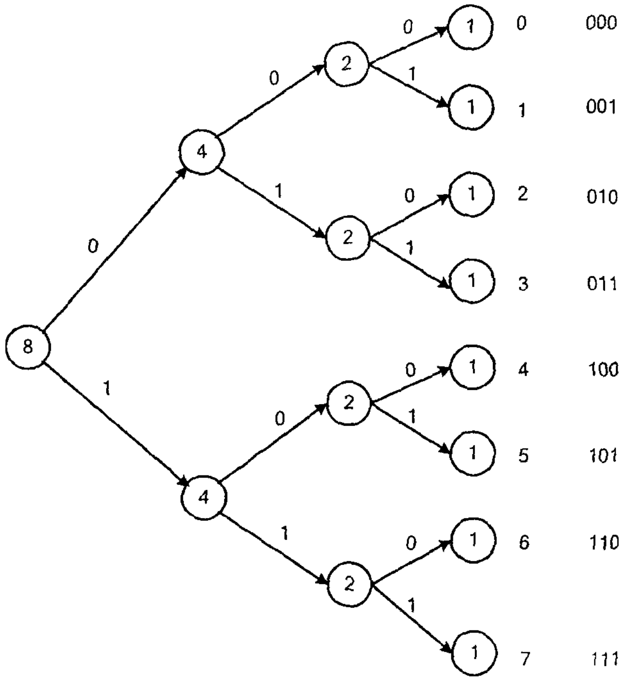


Рис. 4.2. Дерево розбиття двійкової системи числення

Вона утворюється степеневою функцією $P = 2^n$, розбиття на кожному кроці в якої відбувається рівно на два класи, що кодуються номерами 0 і 1. Усі двійкові числа при цьому мають однакову довжину, а алфавіт A, який їх породжує, містить дві цифри – 0 і 1 відповідно.

Числова функція, яка задає кількісний еквівалент двійкового числа, має вигляд

$$A_{\langle 2\rangle} = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \ldots + x_0 2^0 .$$

Обмеження щодо цифр при цьому будуть такі:

$$0 \le x_i \le 1,$$
$$i = 0,\ 1,\ \ldots,\ n-1.$$

Звернемо увагу на те, що таке ж саме дерево розбиття як на рис. 4.2 можуть мати й інші коди, наприклад, код Грея. Однак будь-яке інше кодування класів еквівалентності, яке відмінне від наведеного на рис. 4.2, це вже більш складне кодування, яке формує слова, а не числа. Для чисел необхідне лише кодування множин розбиття номерами у вищезазначеному порядку, починаючи з 0. У цьому полягає відміна звичайних кодів від кодів, що формуються системами числення, тобто є системою числення не будь-який код, але кожний код ґрунтується на системі числення. Нумераційна, або числова функція системи числення з її обмеженнями – це тільки математичний запис її структури, на базі якої можна отримати необмежену кількість різних кодів, комбінації яких є словами, а не числами.

Структура дерева двійкової системи числення дає приклад структур однорідних (природних, степеневих) систем числення. Структури неоднорідних систем числення складаються зі структур систем числення зі змішаною числовою основою, числа яких мають рівну довжину, і структур систем числення з функціональною основою та з нерівною довжиною чисел.

Прикладом систем числення зі змішаною числовою основою, що широко застосовуються на практиці, є факторіальні системи числення з числовою (нумераційною) функцією

$$F = x_n n! + x_{n-1}(n-1)! + \ldots + x_1 1! + x_0 0!$$

і обмеженнями

$$0 \le x_i \le i,$$
$$i = 0,\ 1, \ldots, n.$$

Діапазон цих систем числення $P = n!$.

Структура з $n = 3$ кроками розбиття факторіальної системи числення для діапазону із $P = n! = 3! = 6$ чисел подана на рис. 4.3.

Інформаційними розрядами цієї системи числення є тільки перший і другий, а нульовий розряд надлишковий.

Числова функція й обмеження для цієї системи числення мають вигляд:

$$F = x_2 2! + x_1 1! + x_0 0!,$$
$$0 \le x_i \le i,\ i = 0, 1, 2.$$

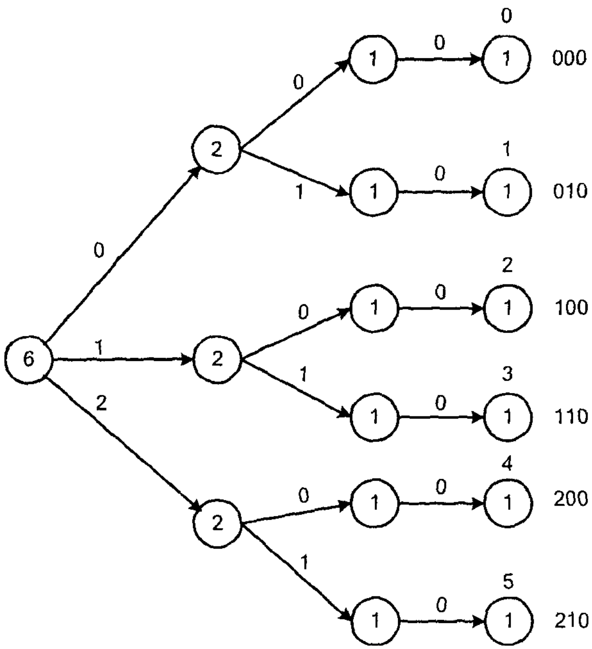


Рис. 4.3 Дерево розбиття факторіальної системи числення

Корисною особливістю факторіальних систем числення є те, що вони здатні генерувати перестановки. Це пояснюється тим, що структурою множин перестановок є факторіальна система числення. Крім перестановок, ця система числення може генерувати й інші комбінаторні об'єкти, так чи інакше пов'язані з перестановками.

На рис. 4.3 наведена структура системи числення з нерівною довжиною чисел і двійковим алфавітом.

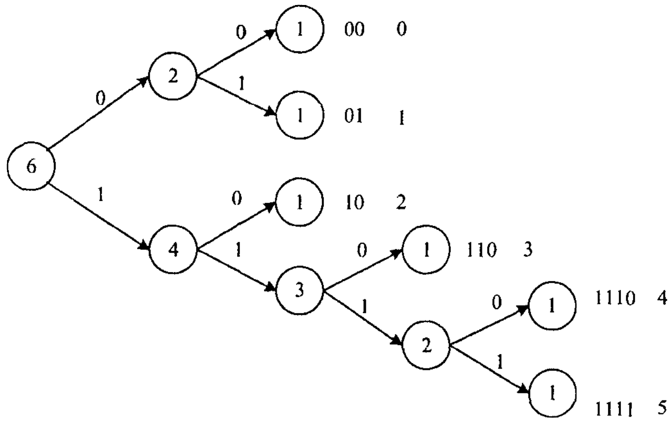


Рис. 4.3. Структурна система числення з нерівною довжиною чисел

Множина чисел 00, 01, 10, 110, 1110, 1111 цієї системи числення являє собою префіксний код, оскільки жодне з наведених на рис. 4.3 чисел не є початком іншого. При цьому кожному числу відповідає десятковий номер 0, 1, 2, 3, 4, 5. Ці номери можна знайти, якщо застосувати правило їх обчислення з допомогою числової функції. Оскільки для цієї системи числення не задані правила побудови, то її слід віднести до табличної системи.

Після писемності самим великим відкриттям було використання людством десяткової системи числення

М. В. Остроградський

Лекція 34
ІСТОРІЯ ПОЗИЦІЙНИХ СИСТЕМ ЧИСЛЕННЯ

Позиційні системи числення з'явилися в Європі відносно недавно – у XII столітті. Вони були принесені з Індії арабськими завойовниками. Але історія їх виникнення губиться в далекому минулому, і на сьогодні існує небагато достовірних відомостей про них. Упевнено можна стверджувати лише те, що позиційні системи числення спричинили революцію в мисленні та практичній діяльності людини і те, що створювалися вони людством протягом тисячоліть.

Найбільш ранні форми подання чисел на основі позиційності ґрунтувалися на групуванні предметів у класи. На основі такого підходу вавилонські математики запропонували шістдесятеричну систему числення, уже добре відому ще до нашої ери.

Система з основою 20 була відома в Центральній Америці індійцям Майя близько 2000 років тому. Вони вже в той час використовували у своїх записах нуль.

Греки для обчислень широко застосовували лічильну дошку – абаку, в якій були накреслені рядки й стовпчики, що відповідали десятковій системі числення.

Десяткова система числення, яку використовують нині, вірогідно, виникла в Індії. Там же з'явилися нуль і його нинішня форма запису. У перших варіантах десяткової системи старший розряд розташовувався справа, а менший зліва, і лише через значний проміжок часу запис цих розрядів був змінений на зворотній. Після 700 року н.е. ця система числення потрапила до арабів у Персію (Іран). Вирішальне значення для поширення десяткової системи числення мала книга з арифметики Леонардо Пізано (Фібоначі), яка вийшла в 1202 році.

Спочатку десяткова система числення застосовувалася лише до цілих чисел. Десяткові дроби з'явилися в X столітті в трактаті з арифметики, який було написано в листі в Дамаску математиком на ім'я «аль Уклідісі» (послідовник Евкліду), а потім математиком аль-Коші, який помер у 1429 році.

Особливу історію має двійкова система числення. Багато первісних племен використовували елементи двійкового числення, групуючи при підрахунку предмети по два. Чисто двійкова система числення з'явилася вперше в 1605 році в неопублікованих роботах американського вченого Томаса Херіота.

Перший аналіз двійкової системи числення з'явився в праці іспанського священика Хуана де Кармюеля Лобковиця в 1670 році, потім її було досліджено в статті Лейбніца в 1703 році.

Після цього двійкова система числення стала відомою й застосовувалася при аналізі деяких ігор та головоломок.

Арчибальд Р. К. зібрав більше 20 праць з цієї теми, що були написані в той час.

Антон Глейзер у своїй книзі 1981 року «History of Binary and other Nondecimal Numeration» навів вичерпну інформацію щодо двійкової системи числення.

Інформація про дискусії з питань двійкової системи числення, які відбулися на початку 20 століття, наведена в статті Е. Уільяма Філіпса «Двійкові обчислення».

Слід зазначити, що в перших ЕОМ, які були створені в США на початку 40-х років, використовувалась десяткова арифметика і лише пізніше, починаючи з 1946 року, почала застосовуватися двійкова.

Крім десяткової й двійкової систем числення, неодноразово пропонувались системи числення з іншими основами. Однак припущення, що будь-яке природнє число, яке більше від одиниці, може бути основою системи числення, уперше довів Блез Паскаль. Він також запропонував перейти до дванадцятеричної системи числення.

Эрхард Вайгель, починаючи з 1673 року, у своїх працях пропонував використовувати четверичну систему числення. Її пропонував використовувати і Джошуа в праці Duodecimal Arithmetic у 1687 році.

Король Швеції Карл XII у 1717 році захопився восьмеричною (вісімковою) арифметикою й збирався ввести її у Швеції. Однак його загибель в одній із битв перешкодила це зробити. Ця система була також запропонована близько 1750 року в Америці Хью Джонсом, професором коледжу «Уільям і Мері», й описана Тейлором А. Б.

Через 100 років американець, швед за національністю, Джон Ністр запропонував шістнадцятеричну систему числення.

У подальшому з'явилися екзотичні системи числення з дробовими, ірраціональними й від'ємними основами. Останню з них уперше описав Віторіо Грюнвальд у 1885 році. Потім були запропоновані системи числення з комплексними основами, наприклад, $2i$, $\sqrt{2}\ i$, $i-1$, а також урівноважена трійкова система числення, в якій використовуються цифри 0, +1, –1.

Позиційні системи числення з від'ємними цифрами були запропоновані Дж. Колсоном у 1726 році й Дж. Леслі в 1817 році.

Відносно недавно з'явилися неоднорідні системи числення зі складними комбінаторними основами. На сьогодні відомі лише три класи таких систем числення: факторіальні, фібоначчиєві й біноміальні.

Безперечно, кількість таких систем не обмежується наведеними трьома класами. Тому ймовірність появи нових цікавих у теоретичному й практичному плані позиційних систем числення досить велика.

Математика показує дорогу до вершини гори,
але вона не спроможна зробити її більш низькою.

Б. Макмілан

РОЗДІЛ 2. ОДНОРІДНІ СИСТЕМИ ЧИСЛЕННЯ

Лекція 35
ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОДНОРІДНИХ СИСТЕМ ЧИСЛЕННЯ

Однорідні (природні) системи числення (ОСЧ) – це клас позиційних систем числення, найбільш поширений на практиці, оскільки шістнадцятерична, десяткова, восьмерична, двійкова системи числення, що належать до нього, найчастіше використовуються в усній і машинній лічбі.

Як цифри в однорідних системах числення використовується звичайно арабська символіка 0, 1, …, 9, до якої в разі необхідності додаються букви латинського алфавіту, наприклад, A, B, C, D, E, F для шістнадцятеричної системи числення. Цифра A в цьому випадку відповідає десятковому числу 10, B – 11, C – 12, D – 13, E – 14, F – 15.

Сукупність цифр, що використовуються, якщо їх подавати в тій чи іншій однорідній системі числення, утворює її *алфавіт* A, а число цифр q в алфавіті A створює *основу* цієї системи числення.

Кількість розрядів n, які використовують для запису числа, має назву *довжини* числа, або його *розрядності*.

Однорідне число $F_{\langle q\rangle}$ кодується за допомогою послідовності знаків

$$x_{n-1}\, x_{n-2} \ldots x_i \ldots x_0 \text{, де } x_i \in A .$$

Кількісний еквівалент однорідного числа $F_{\langle q\rangle}$ отримують за допомогою числової функції

$$F_{\langle q\rangle} = x_{n-1}q^{n-1} + x_{n-2}q^{n-2} + \ldots + x_0 q^0 ,$$

де

$$0 \le x_i \le q-1, \quad i = 0, 1, \ldots, n-1.$$

Відношення $$S_i = \frac{q^i}{q^0}$$

в даному випадку створює вагу розряду ,

де $i = 0, 1, \ldots, n-1$ – номер розряду, q^i – *вага* цифри в цьому розряді.

Приклад 1.

$$A_{\langle 10\rangle} = 123 = 1\cdot 10^2 + 2\cdot 10^1 + 3\cdot 10^0 = 1\cdot 100 + 2\cdot 10 + 3\cdot 1 = 123_{\langle 10\rangle}.$$

Приклад 2. $A_{\langle 5\rangle} = 123_{\langle 5\rangle} = 1\cdot 5^2 + 2\cdot 5^1 + 3\cdot 5^0 = 38_{\langle 10\rangle}$.

Особливе значення для ЕОМ має двійкова система числення, оскільки вона легко реалізується з допомогою елементів з двома стійкими станами.

Приклад 3.

$$A_{\langle 2\rangle} = 1011 = 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 8 + 2 + 1 = 11_{\langle 10\rangle}.$$

Діапазон чисел, які зображаються в однорідній позиційній системі числення, задається *степеневою* функцією

$$P = q^n.$$

Тому однорідні системи числення ще мають назву *степеневих*.

Мінімальне число, яке задають у однорідній системі числення $F_{\min} = 0$, а максимальне $F_{\max} = P - 1 = q^n - 1$.

Основна ідея, якою пройнята вся математика, - це ідея рівності

Г. Спенсер

Лекція 36
ОПЕРАЦІЇ ДОДАВАННЯ І ВІДНІМАННЯ В ОДНОРІДНИХ СИСТЕМАХ ЧИСЛЕННЯ

1. Додавання

Операція додавання чисел у однорідних системах числення відбувається згідно з таблицями додавання, які задаються за основами систем числення q. При цьому 1 переносу в старший розряд у цих таблицях не враховується. Так для двійкової, восьмеричної, десяткової й шістнадцятеричної систем числення це будуть відповідно табл. 4.1, 4.2, 4.3, 4.4.

Таблиця 4.1. Додавання для двійкової системи числення

	0	1
0	0	1
1	1	0

Таблиця 4.2. Додавання для восьмеричної системи числення

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

Таблиця 4.3. Додавання для десяткової системи числення

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

Таблиця 4.4. Додавання для шістнадцятеричної системи числення

	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*
0	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*
1	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*	0
2	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*	0	1
3	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*	0	1	2
4	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*	0	1	2	3
5	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*	0	1	2	3	4
6	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*	0	1	2	3	4	5
7	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*	0	1	2	3	4	5	6
8	8	9	*A*	*B*	*C*	*D*	*E*	*F*	0	1	2	3	4	5	6	7
9	9	*A*	*B*	*C*	*D*	*E*	*F*	0	1	2	3	4	5	6	7	8
A	*A*	*B*	*C*	*D*	*E*	*F*	0	1	2	3	4	5	6	7	8	9
B	*B*	*C*	*D*	*E*	*F*	0	1	2	3	4	5	6	7	8	9	*A*
C	*C*	*D*	*E*	*F*	0	1	2	3	4	5	6	7	8	9	*A*	*B*
D	*D*	*E*	*F*	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*
E	*E*	*F*	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*
F	*F*	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*

Приклад 1. Скласти два числа $A_{\langle 2\rangle}=1101$ і $B_{\langle 2\rangle}=1011$ у двійковій системі числення.

Розв'язання.

$$\begin{array}{r} 1101_{\langle 2\rangle} \\ +\;\; \\ 1011_{\langle 2\rangle} \\ \hline 11000_{\langle 2\rangle} \end{array}$$

Перевірка.

$1101_{\langle 2\rangle}=1\cdot 2^3+1\cdot 2^2+0\cdot 2^1+1\cdot 2^0=13_{\langle 10\rangle}$;

$1011_{\langle 2\rangle}=1\cdot 2^3+0\cdot 2^2+1\cdot 2^1+1\cdot 2^0=11_{\langle 10\rangle}$;

$11000_{\langle 2\rangle}=1\cdot 2^4+1\cdot 2^3+0\cdot 2^2+0\cdot 2^1+0\cdot 2^0=24_{\langle 10\rangle}$;

$13_{\langle 10\rangle}+11_{\langle 10\rangle}=24_{\langle 10\rangle}$.

Приклад 2. Скласти два числа $A_{\langle 8\rangle}=517$ і $B_{\langle 8\rangle}=243$ у восьмеричній системі числення.

Розв'язання.

$$\begin{array}{r} 243_{\langle 8\rangle} \\ +\;517_{\langle 8\rangle} \\ \hline 762_{\langle 8\rangle} \end{array}$$

Перевірка.

$243_{\langle 8\rangle}=2\cdot 8^2+4\cdot 8^1+3\cdot 8^0=163_{\langle 10\rangle}$;

$517_{\langle 8\rangle}=5\cdot 8^2+1\cdot 8^1+7\cdot 8^0=335_{\langle 10\rangle}$;

$762_{\langle 8\rangle}=7\cdot 8^2+6\cdot 8^1+2\cdot 8^0=498_{\langle 10\rangle}$;

$163_{\langle 10\rangle}+335_{\langle 10\rangle}=498_{\langle 10\rangle}$.

Приклад 3. Скласти два числа $A_{\langle 16\rangle}=A1B$ і $B_{\langle 16\rangle}=11F$ у шістнадцятеричній системі числення.

Розв'язання.

$$\begin{array}{r} A1B_{\langle 16\rangle} \\ +\;11F_{\langle 16\rangle} \\ \hline B3A_{\langle 16\rangle} \end{array}$$

Перевірка.

$A1B_{\langle 16\rangle}=10\cdot 16^2+1\cdot 16^1+11\cdot 16^0=2560+16+11=2587_{\langle 10\rangle}$;

$11F_{\langle 16\rangle}=1\cdot 16^2+1\cdot 16^1+15\cdot 16^0=256+16+15=287_{\langle 10\rangle}$;

$B3A_{\langle 16\rangle}=11\cdot 16^2+3\cdot 16^1+10\cdot 16^0=2816+48+10=2874_{\langle 10\rangle}$;

$2587_{\langle 10\rangle}+287_{\langle 10\rangle}=2874_{\langle 10\rangle}$.

2. Віднімання

Для операції віднімання застосовуються відповідні таблиці, які так само, як і для додавання, задаються з допомогою основ систем числення q (див. табл. 4.5, 4.6, 4.7, 4.8). У нихураховується 1 займу зі старшого розряду..

Таблиця 4.5. Віднімання у двійковій системі числення

	0	1
0	0	1
1	1	0

Таблиця 4.6. Віднімання у восьмеричній системі числення

	0	1	2	3	4	5	6	7
0	0	7	6	5	4	3	2	1
1	1	0	7	6	5	4	3	2
2	2	1	0	7	6	5	4	3
3	3	2	1	0	7	6	5	4
4	4	3	2	1	0	7	6	5
5	5	4	3	2	1	0	7	6
6	6	5	4	3	2	1	0	7
7	7	6	5	4	3	2	1	0

Таблиця 4.7. Віднімання в десяткової системі числення

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

Таблиця 4.8. Віднімання в шістнадцятеричній системі числення

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1
1	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2
2	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3
3	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4
4	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5
5	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6
6	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9
9	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A
A	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B
B	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C
C	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D
D	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E
E	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

Приклад 4. Відняти від числа $A_{\langle 2\rangle}=1101$ число $B_{\langle 2\rangle}=1011$.

Розв'язання.

$$\begin{array}{r} 1101_{\langle 2\rangle} \\ -\ 1011_{\langle 2\rangle} \\ \hline 0010_{\langle 2\rangle} \end{array}$$

Перевірка.

$1101_{\langle 2\rangle}=1\cdot 2^3+1\cdot 2^2+0\cdot 2^1+1\cdot 2^0=13_{\langle 10\rangle}$;

$1011_{\langle 2\rangle}=1\cdot 2^3+0\cdot 2^2+1\cdot 2^1+1\cdot 2^0=11_{\langle 10\rangle}$;

$0010_{\langle 2\rangle}=0\cdot 2^3+0\cdot 2^2+1\cdot 2^1+0\cdot 2^0=2_{\langle 10\rangle}$;

$13_{\langle 10\rangle}-11_{\langle 10\rangle}=2_{\langle 10\rangle}$.

Приклад 5. Відняти від числа $A_{\langle 8\rangle}=517$ число $B_{\langle 8\rangle}=243$.

Розв'язання.

$$\begin{array}{r} - \begin{array}{l} 517_{\langle 8\rangle} \\ 243_{\langle 8\rangle} \end{array} \\ \hline 254_{\langle 8\rangle} \end{array}$$

Перевірка.

$517_{\langle 8\rangle} = 5\cdot 8^2 + 1\cdot 8^1 + 7\cdot 8^0 = 335_{\langle 10\rangle}$;

$243_{\langle 8\rangle} = 2\cdot 8^2 + 4\cdot 8^1 + 3\cdot 8^0 = 163_{\langle 10\rangle}$;

$254_{\langle 8\rangle} = 2\cdot 8^2 + 5\cdot 8^1 + 4\cdot 8^0 = 172_{\langle 10\rangle}$;

$335_{\langle 10\rangle} - 163_{\langle 10\rangle} = 172_{\langle 10\rangle}$.

Приклад 6. Відняти від числа $A1B_{\langle 16\rangle}$ число $11F_{\langle 16\rangle}$.

Розв'язання.

$$\begin{array}{r} - \begin{array}{l} A1B_{\langle 16\rangle} \\ 11F_{\langle 16\rangle} \end{array} \\ \hline 8FC_{\langle 16\rangle} \end{array}$$

Перевірка.

$A1B_{\langle 16\rangle} = 2587_{\langle 10\rangle}$;

$11F_{\langle 16\rangle} = 287_{\langle 10\rangle}$;

$2587_{\langle 10\rangle} - 287_{\langle 10\rangle} = 2300_{\langle 10\rangle}$;

$8FC_{\langle 16\rangle} = 9\cdot 16^2 + 15\cdot 16^1 + 12\cdot 16^0 = 2048 + 240 + 12 = 2300_{\langle 10\rangle}$.

Вирішити задачу - значить звести її до більш простої.

У. Сойер

Лекція 37
ОПЕРАЦІЇ МНОЖЕННЯ І ДІЛЕННЯ В ОДНОРІДНИХ СИСТЕМАХ ЧИСЛЕННЯ

1 Множення

Операція множення виконується на базі таблиць множення й додавання за основою q (див. табл. 4.9, 4.10, 4.11, 4.12).

Ці таблиці будуються шляхом множення того, що множать (зліва) на множник (справа) у десятковій системі числення, і потім результат множення (добуток) подається в іншій системі числення шляхом розкладання за степенями відповідно до числової функції для ОСЧ.

Наприклад, для десяткової системи числення після множення 5 на 4 отримуємо результат 20 (табл. 4.11). Якщо подати цей результат у восьмеричній системі числення, отримаємо

$$20_{\langle 10\rangle} = 2\cdot 8^1 + 4\cdot 8^0 = 24_{\langle 8\rangle}.$$

Тому в табл. 4.10 у клітинці, яка стоїть на перетині рядка, що відповідає цифрі 5, і стовпчика, що відповідає цифрі 4, маємо результат перемноження чисел 5 на 4 у восьмеричній системі числення, який дорівнює $24_{\langle 8\rangle}$.

Аналогічно в табл. 4.12 для шістнадцятеричної системи числення на перетині рядка «5» і стовпчика «4» маємо число

$$14_{\langle 16\rangle} = 1\cdot 16^1 + 4\cdot 16^0 = 20_{\langle 10\rangle},$$

а на перетині того ж самого рядка «5» і стовпчика «9» число

$$2D_{\langle 16\rangle} = 2\cdot 16^1 + D\cdot 16^0 = 2\cdot 16^1 + 13\cdot 16^0 = 45_{\langle 10\rangle}.$$

Таблиця 4.9. Множення для двійкової системи числення

	0	1
0	0	0
1	0	1

Таблиця 4.10. Множення для восьмеричної системи числення

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

Таблиця 4.11. Множення для десяткової системи числення

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	10	12	14	16	18
3	0	3	6	9	12	15	18	21	24	27
4	0	4	8	12	16	20	24	28	32	36
5	0	5	10	15	20	25	30	35	40	45
6	0	6	12	18	24	30	36	42	48	54
7	0	7	14	21	28	35	42	49	56	63
8	0	8	16	24	32	40	48	56	64	72
9	0	9	18	27	36	45	54	63	72	81

Таблиця 4.12. Множення для шістнадцятеричної системи числення

	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*
2	0	2	4	6	8	*A*	*C*	*E*	10	12	14	16	18	1*A*	1*C*	1*E*
3	0	3	6	9	*C*	*F*	12	15	18	1*B*	1*E*	21	24	27	2*A*	2*D*
4	0	4	8	*C*	10	14	18	1*C*	20	24	28	2*C*	30	34	38	3*C*
5	0	5	*A*	*F*	14	19	1*E*	23	28	2*D*	32	37	3*C*	41	46	4*B*
6	0	6	*C*	12	18	1*E*	24	2*A*	30	36	3*C*	42	48	4*E*	54	5*A*
7	0	7	*E*	15	1*C*	23	2*A*	31	38	3*F*	46	4*D*	54	5*B*	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1*B*	24	2*D*	36	3*F*	48	51	5*A*	63	6*C*	75	7*E*	87
A	0	*A*	14	1*E*	28	32	3*C*	46	50	5*A*	64	6*E*	78	82	8*C*	96
B	0	*B*	16	21	2*C*	37	42	4*D*	58	63	6*E*	79	84	8*F*	9*A*	*A*5
C	0	*C*	18	24	30	3*C*	48	54	60	6*C*	78	84	90	9*C*	*A*8	*B*4
D	0	*D*	1*A*	27	34	41	4*E*	5*B*	68	75	82	8*F*	9*C*	*A*9	*B*6	*C*3
E	0	*E*	1*C*	2*A*	38	46	54	62	70	7*E*	8*C*	9*A*	*A*8	*B*6	*C*4	*D*2
F	0	*F*	1*E*	2*D*	3*C*	4*B*	5*A*	69	78	87	96	*A*5	*B*4	*C*3	*D*2	*E*1

Якщо користуватись вищенаведеними таблицями множсння, можна легко виконати множення числа у відповідних системах числення.

Приклад 1. Перемножити числа $A_{\langle 2\rangle}=1101$ і $B_{\langle 2\rangle}=1011$ у двійковій системі числення.

Розв'язання.

$$
\begin{array}{r}
1101_{\langle 2\rangle} \\
\times \quad 1011_{\langle 2\rangle} \\
\hline
1101 \\
1101 \\
0000 \\
1101 \\
\hline
10001111_{\langle 2\rangle}
\end{array}
$$

Перевірка.

$1101_{\langle 2\rangle}=1\cdot 2^3+1\cdot 2^2+0\cdot 2^1+1\cdot 2^0=13_{\langle 10\rangle}$;

$1011_{\langle 2\rangle}=1\cdot 2^3+0\cdot 2^2+1\cdot 2^1+1\cdot 2^0=11_{\langle 10\rangle}$;

$13 \times 11 = 143_{\langle 10 \rangle}$;

$10001111 = 1 \cdot 2^7 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$
$= 128 + 8 + 4 + 2 + 1 = 143_{\langle 10 \rangle}$.

Приклад 2. Помножити число $A_{\langle 8 \rangle} = 517$ на число $B_{\langle 8 \rangle} = 243$ у восьмеричній системі числення.

Розв'язання.

$$
\begin{array}{r}
243_{\langle 8 \rangle} \\
\times \quad 517_{\langle 8 \rangle} \\
\hline
1755 \\
2474 \\
1236 \\
\hline
152515_{\langle 8 \rangle}
\end{array}
$$

Перевірка.

$517_{\langle 8 \rangle} = 5 \cdot 8^2 + 1 \cdot 8^1 + 7 \cdot 8^0 = 335_{\langle 10 \rangle}$;

$243_{\langle 8 \rangle} = 2 \cdot 8^2 + 4 \cdot 8^1 + 3 \cdot 8^0 = 163_{\langle 10 \rangle}$;

$335 \times 163 = 54605_{\langle 10 \rangle}$;

$152515_{\langle 8 \rangle} = 1 \cdot 8^5 + 5 \cdot 8^4 + 2 \cdot 8^3 + 5 \cdot 8^2 + 1 \cdot 8^1 + 5 \cdot 8^0 =$
$= 32768 + 20480 + 1024 + 320 + 8 + 5 = 54605_{\langle 10 \rangle}$.

Приклад 3. Помножити число $A1B_{\langle 16 \rangle}$ на число $11F_{\langle 16 \rangle}$ у шістнадцятеричній системі числення.

Розв'язання.

$$
\begin{array}{r}
A1B_{\langle 16 \rangle} \\
\times \quad 11F_{\langle 16 \rangle} \\
\hline
9795 \\
A1B \\
A1B \\
\hline
B5445_{\langle 16 \rangle}
\end{array}
$$

Перевірка.

$A1B_{\langle 16\rangle} = 2587_{\langle 10\rangle}$;

$11F_{\langle 16\rangle} = 287_{\langle 10\rangle}$;

$2587_{\langle 10\rangle} \times 287_{\langle 10\rangle} = 742469_{\langle 10\rangle}$;

$$B5445_{\langle 16\rangle} = 11\cdot 16^4 + 5\cdot 16^3 + 4\cdot 16^2 + 4\cdot 16^1 + 5\cdot 16^0 =$$
$$= 720896 + 20480 + 1024 + 64 + 5 = 742469_{\langle 10\rangle}.$$

2. Ділення

Операція ділення виконується на основі таблиць множення й віднімання за основою q за звичайними правилами до отримання залишку меншого дільника.

Приклад 4. Поділити число $1101_{\langle 2\rangle}$ на число $1011_{\langle 2\rangle}$ у двійковій системі числення.

Розв'язання.

$$\begin{array}{r|l} -1101 & 1011 \\ \cline{2-2} \underline{1011} & 1 \\ 10 & \end{array}$$

Перевірка.

Частка від ділення, очевидно, дорівнює одиниці, а залишок $10_{\langle 2\rangle} = 2_{\langle 10\rangle}$. Ділення числа $13_{\langle 10\rangle} = 1101_{\langle 2\rangle}$ на число $11_{\langle 10\rangle} = 1011_{\langle 2\rangle}$ у десятковій системі числення дасть той самий результат.

$$\begin{array}{r|l} -13 & 11 \\ \cline{2-2} \underline{11} & 1 \\ 2 & \end{array}$$

Приклад 5. Поділити число $517_{\langle 8\rangle}$ на число $243_{\langle 8\rangle}$ у восьмеричній системі числення.

Розв'язання.

$$\begin{array}{r|l} -517 & 243 \\ \cline{2-2} \underline{506} & 2 \\ 11 & \end{array}$$

Перевірка.

$517_{\langle 8\rangle} = 5\cdot 8^2 + 1\cdot 8^1 + 7\cdot 8^0 = 335_{\langle 10\rangle}$;

$243_{\langle 8\rangle} = 2\cdot 8^2 + 4\cdot 8^1 + 3\cdot 8^0 = 163_{\langle 10\rangle}$;

$11_{\langle 8\rangle} = 9_{\langle 10\rangle}$;

$2_{\langle 8\rangle} = 2_{\langle 10\rangle}$;

$$\begin{array}{r|l} -335_{\langle 10\rangle} & 163_{\langle 10\rangle} \\ \cline{2-2} \underline{326} & 2 \\ 9 & \end{array}$$

Приклад 6. Поділити число $A1B_{\langle 16\rangle}$ на число $11F_{\langle 16\rangle}$ у шістнадцятеричній системі числення.

Розв'язання.

$$\begin{array}{r|l} -A1B_{\langle 16\rangle} & 11F_{\langle 16\rangle} \\ \cline{2-2} \underline{A17} & 9 \\ 4 & \end{array}$$

Перевірка.

$A1B_{\langle 16\rangle} = 2587_{\langle 10\rangle}$;

$11F_{\langle 16\rangle} = 287_{\langle 10\rangle}$;

$$\begin{array}{r|l} -2587_{\langle 10\rangle} & 287_{\langle 10\rangle} \\ \cline{2-2} \underline{2583} & 9 \\ 4 & \end{array}$$

Математика – це мистецтво давати одну й ту саму назву різним речам.

Пуанкаре

Лекція 38
ПЕРЕВЕДЕННЯ ЧИСЕЛ

Переведення, або перетворення, чисел з однієї системи числення до іншої широко використовується на практиці, оскільки різні цифрові пристрої з різною ефективністю перетворюють інформацію залежно від систем числення, які в них застосовуються. Найчастіше уживані сьогодні пристрої, які використовують двійкову систему числення, але існує багато таких, що працюють у інших системах числення, з яких найбільш поширені вісімкова (восьмерична), шістнадцятерична, десяткова. Тому існує потреба в переведенні числа однієї системи числення до іншої, а для цього необхідні відповідні алгоритми. Нижче будуть розглянуті найбільш уживані з них на практиці.

1. Табличний метод переведення

Необхідно скласти таблицю, яка містить числа однієї системи числення й відповідні їм числа іншої. Процедура переведення складається з того, що віднаходиться рядок, в якому стоїть початкове число, що переводять, і в цьому ж самому рядку відбувається пошук числа, що міститься в системі числення, до якої переводиться початкове число (див. табл. 4.13). Недоліком такого переведення є громіздкість таблиць при великій кількості чисел, які переводять, а перевагою – велика швидкість перетворення. У перспективі зі зростанням ємності запам'ятовуючих пристроїв таке переведення може виявитися достатньо ефективним і практичним. Сьогодні воно використовується в окремих спеціальних випадках, особливо для швидкодіючих спеціалізованих цифрових пристроїв.

Таблиця 4.13. Переведення десяткових чисел у двійкові

Десяткові числа	Двійкові числа	Десяткові числа	Двійкові числа
00	0000	08	1000
01	0001	09	1001
02	0010	10	1010
03	0011	11	1011
04	0100	12	1100
05	0101	13	1101
06	0110	14	1110
07	0111	15	1111

2. Переведення чисел у систему числення з кратною основою

Цей метод ефективний, якщо основи систем числення, які підлягають перетворенню, кратні між собою. Наприклад, якщо необхідно перевести десяткове число в п'ятеричну систему числення чи навпаки. У цьому випадку основа 10 десяткової системи числення *націло ділиться на основу 5 п'ятеричної системи числення.*

Наприклад, число 17 з десяткової системи числення у п'ятеричну можна перевести таким чином:

$$17_{\langle 10\rangle} = 1\cdot 10^1 + 7\cdot 10^0 = 2\cdot 5^1 + 1\cdot 5^1 + 2\cdot 5^0 = 3\cdot 5^1 + 2\cdot 5^0 = 32_{\langle 5\rangle}.$$

3. Переведення чисел шляхом підбору степенів

Суть цього методу полягає в тому, що число, яке переводять, подається як сума добутків різних степенів нової основи на можливі цифри числа, яке отримують. Вони обираються таким чином, щоб ця сума була якомога ближчою до числа, що переводиться, але не більшою за нього. При цьому степені нового числа можна обрати з відповідної таблиці. Коли сума добутків дорівнюватиме вихідному числу, то новий запис числа буде отриманий.

Наприклад, необхідно перевести десяткове число $123_{\langle 10\rangle}$ у двійкове. Найбільшим степенем числа 2, який не більший ніж число 123, буде число $2^6 = 64 < 123$, оскільки степінь $2^7 = 128 > 123$. Наступним степенем у сумі з попереднім, який не більший ніж число 123, буде $2^5 = 32$ (32 + 64 = 98 < 123), потім степінь $2^4 = 16$ (16 + 32 + + 64 = 112 < 123) і $2^3 = 8$ (8 + 16 + 32 + 64 =120 < 123). Степінь $2^2 = 4$ необхідно проминути, тому що 4 + 8 + 16 + 32 + 64 = 124 > 123. Проте, залишається степінь $2^1 = 2$, оскільки 2 + 8 + 16 + 32 + 64 = 122 < 123. У кінці додамо $2^0 = 1$ і отримаємо, що 1 + 2 + 8 +16 + 32 + 64 = 123.

Отже, двійковим числом, яке дорівнює кількісному еквіваленту числа $123_{\langle 10\rangle}$, буде число

$$1111011_{\langle 2\rangle} = 1\cdot 2^6 + 1\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 123_{\langle 10\rangle}.$$

Аналогічно можна виконати переведення десяткового числа в шістнадцятеричну систему числення, наприклад, число $377_{\langle 10\rangle}$:

$$377_{\langle 10\rangle} = 1\cdot 16^2 + 7\cdot 16^1 + 9\cdot 16^0 = 256 + 112 + 9 = 179_{\langle 16\rangle}.$$

Відмінність переведення в шістнадцятеричну систему числення від переведення у двійкову, розглянутого вище, полягає в тому, що при переведенні у двійкову систему числення в процесі отримання суми степенів числа 2 їх множення відбувалося на 1 чи 0 (у випадку множення на 0 степінь виключався із суми степенів нового числа). У даному ж випадку кожний новий степінь числа 16 треба послідовно множити на числа 0, 1, …, 15 до того моменту, поки серед отримуваних добутків не виявиться той найбільший, за якого сума здобутих раніше добутків з даним ще не перевищує початкового числа, що переводять.

4. Переведення чисел з основою, одна з яких є степенем іншої

У випадку, переведення чисел із системи числення з основою q, що є степенем основи p системи числення, в яку відбувається переведення, тобто $q = p^m$, кожна цифра числа, що переводять, може бути подана m цифрами в системі числення з основою p, і в результаті буде отримане необхідне число.

Приклад 1. Дано восьмеричне число $12156_{\langle 8\rangle}$ з $q = 8$. Потрібно перевести його у двійкову систему числення, тобто з $p = 2$.

Розв'язання. Очевидно, що $8 = 2^3$.
Тому $12156_{\langle 8\rangle} = 010001101110_{\langle 2\rangle}$.

У разі зворотного переведення із системи числення з основою p у систему числення з основою q, $p < q$, цифри числа, яке переводять, починаючи з молодших розрядів, об'єднують за m розрядами й далі записують з допомогою однієї цифри в q-ричній системі числення.

Приклад 2. Дано двійкове число $111001111111_{\langle 2\rangle}$. Потрібно перетворити його в шістнадцятеричне.

Розв'язання. Якщо виходити зі співвідношення $2^4 = 16$, отримаємо:

$$1110\ 0111\ 1111_{\langle 2\rangle} = E7F_{\langle 16\rangle}.$$

5. Переведення чисел у систему числення на базі проміжного перетворення цифр

Основою даного методу переведення є числова функція для позиційних чисел. Відповідно до цієї функції кожна цифра початкового числа й основа q переводиться в систему числення з основою p, а потім виконуються всі арифметичні операції, які передбачені цією функцією.

Приклад 3. Перевести десяткове число $121_{\langle 10\rangle}$ з основою $q = 10$ у двійкове з $p = 2$.

Розв'язання.

1. Переведемо всі цифри числа й основу $q = 10$ у двійкову систему числення

$$1_{\langle 10\rangle} = 0001_{\langle 2\rangle},\ 2_{\langle 10\rangle} = 0010_{\langle 2\rangle},\ 10_{\langle 10\rangle} = 1010_{\langle 2\rangle}.$$

2. Обчислимо

$$(1010)^2 = 1010 \times 1010 = 1100100 \text{ и } 2 \times 10 = 0010 \times 1010 = 10100.$$

3. Підставимо отримані значення в числову функцію для числа $121_{\langle 10\rangle}$ і отримаємо вираз:

$$121_{\langle 10\rangle} = 1 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0 = 0001 \times 1100100 + 0010 \times 1010 + 0001 \times 0001 =$$

$$= 1100100 + 10100 + 0001 = 1111001_{\langle 2\rangle}.$$

Приклад 4. Перевести двійкове число $1001110_{\langle 2\rangle}$ у десяткове.

Розв'язання. Після запису числової функції числа, що перетворюється у двійкову форму, і після того, як виконали в десятковому вигляді всі зазначені в ній арифметичні операції, отримаємо десяткове число, яке шукали:

$$1001110_{\langle 2\rangle} = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 =$$

$$= 64 + 8 + 4 + 2 = 78_{\langle 10\rangle}.$$

6. Переведення чисел за допомогою використання проміжної системи числення

Суть цієї дії полягає в тому, щоб з метою спрощення чи прискорення переведення застосовувати проміжну систему числення.

Приклад 5. Перевести десяткове число $121_{(10)}$ у двійкову систему числення з використанням як проміжної вісімкової системи числення.

Розв'язання. Перетворимо з допомогою одного з відомих методів десяткове число $121_{\langle 10\rangle}$ у вісімкову систему числення $121_{\langle 10\rangle}=171_{\langle 8\rangle}$.

Потім з вісімкової системи числення перейдемо до двійкової $171_{\langle 8\rangle}=1111001_{\langle 2\rangle}$.

7. Переведення чисел у систему числення шляхом ділення на її основу

Алгоритм переведення чисел із системи числення з основою q у систему числення з основою p є універсальним і найбільш широко використовується на практиці.

Він містить такі кроки:

1. Розділити число, яке переводять, у системі числення з основою q на основу p за правилом системи числення з основою q.

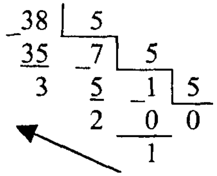
2. Перевірити, чи не дорівнює частка нулю. Якщо не дорівнює, то прийняти її за нове число й повернутися до пункту 1.

3. Якщо частка дорівнює нулю, то виписати всі отримані залишки від ділення в порядку, зворотному їх отриманню.

4. Отриманий запис є записом числа в системі числення з основою p.

Приклад 6. Перевести число $38_{\langle 10\rangle}$ у п'ятеричну систему числення

Розв'язання.



Відповідь. $38_{\langle 10\rangle}=123_{\langle 5\rangle}$.

Приклад 7. Перевести число $11_{\langle 10\rangle}$ у двійкову систему числення.

Розв'язання

```
_11 | 2
 10   _5 | 2
  1    4   _2 | 2
       1    2   _1 | 2
            0    0   0
                 1
```

Відповідь. $11_{\langle 10\rangle} = 1011_{\langle 2\rangle}$.

Приклад 8. Перевести число $1101110_{\langle 2\rangle}$ у десяткову систему числення шляхом ділення на основу 10.

Розв'язання. Попередньо переведемо число 10 у двійкову систему числення $10_{\langle 10\rangle} = 1010_{\langle 2\rangle}$.

Потім, якщо використовувати алгоритм перетворення, треба виконати згідно з ним ділення за правилами двійкової системи числення:

```
_1101110 | 1010
 1010    |_1011 | 1010
 _0111     1010 |_0001 | 1010
  0000     0001   0000 | 0000
  _1111           0001
   1010
   _1010
    1010
    0000
```

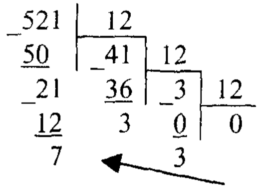
Залишки, які отримано у двійковій системі числення, виписують послідовно один за одним у порядку, зворотному до порядку їх отримання, і потім переводять у десяткові цифри:

$$0000_{\langle 2\rangle} = 0_{\langle 10\rangle},\ 0001_{\langle 2\rangle} = 1_{\langle 10\rangle},\ 0001_{\langle 10\rangle} = 1,\ 0000_{\langle 2\rangle} = 0_{\langle 10\rangle}.$$

Послідовність цих цифр утворює десяткове число, яке шукали: $0110_{\langle 10\rangle} = 1101110_{\langle 2\rangle}$.

Приклад 9. Перевести вісімкове число $521_{\langle 8\rangle}$ у десяткову систему числення.

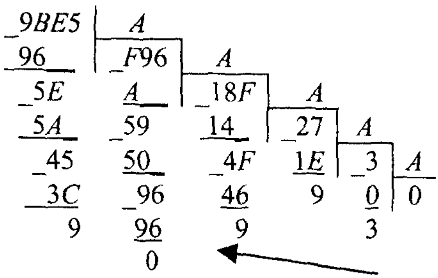
Розв'язання. Оскільки $10_{\langle 10\rangle}=12_{\langle 8\rangle}$, то, якщо виконувати ділення десяткового числа 521 за правилами вісімкової системи числення, отримаємо



Відповідь. $521_{\langle 8\rangle}=337_{\langle 10\rangle}$.

Приклад 10. Перевести шістнадцятеричне число $9BE5_{\langle 16\rangle}$ у десяткову систему числення й виконати перевірку розв'язка.

Розв'язання. Якщо врахувати, що в шістнадцятеричній системі числення $10_{\langle 10\rangle}=A$, то ділимо число $9BE5_{\langle 16\rangle}$ на A за правилами шістнадцятеричної системи числення, користуючись при цьому відповідними таблицями множення й віднімання.



Відповідь. $9BE5_{\langle 16\rangle}=39909_{\langle 10\rangle}$.

Перевірка

```
_39909 | 16
 32__  |_2494 | 16
 _79     16__ |_155 | 16
  64_    _89    144 | _9 | 16
 _150     80_    11    0 | 0
  144     _94          9
   _9      80
    64     14
     5
```

Відповідь. $39909_{\langle 10\rangle} = 9BE5_{\langle 16\rangle}$.

Необхідність введення додаткових символів у шістнадцятеричній системі числення може спричинити при переведенні деякі труднощі, оскільки необхідно весь час звертатись до таблиць віднімання й множення в цій системі.

Ці труднощі можна зменшити, якщо замість символів A, B, C, D, E, F у шістнадцятеричній системі числення скористатися десятковими цифрами 10, 11, 12, 13, 14, 15 відповідно. При цьому треба виокремити їх з обох боків у числах крапками. Тоді шістнадцятеричне число $9BE5_{\langle 16\rangle} = 9.11.14.5_{\langle 16\rangle}$, а переведення його до десяткової системи числення буде мати такий вигляд:

```
_9.11.14.5 | 10
 9.6_      |_15.9.6 | 10
 _5.14       10__   |_1.8.15 | 10
  5.10_      _5.9     1.4    |_2.7 | 10
   _4.5       5.0_   _4.15    1.14 | _3 | 10
    3.12      _9.6    4.6_      9     0 | 0
       9       9.6      9             3
                 0
```

Звертання до таблиць множення й віднімання в цьому випадку можна уникнути, якщо виконувати безпосереднє переведення із шістнадцятеричної системи числення у десяткову систему під час кожної операції множення й віднімання, що може виявитись простішою задачею, ніж пошуки відповідей у таблицях.

Наприклад, при знаходженні частки від ділення 9.11 на 10 слід перевести 9.11 із *шістнадцятеричної* системи числення в десяткову $9 \cdot 16^1 + 11 \cdot 16^0 = 155_{\langle 10 \rangle}$.

Тоді ділення 155 на 10 дасть ціле значення 15, яке й ставимо як першу цифру частки. Потім після множення 15 на 10 відбувається переведення числа $150_{\langle 10 \rangle}$ у шістнадцятеричну систему числення ($150_{\langle 10 \rangle} = 9.6_{\langle 16 \rangle}$), і тепер уже це число $9.6_{\langle 16 \rangle}$ віднімається від того, що ділили. Аналогічно можна отримати й цифри інших часток.

Приклад 11. Перевести шістнадцятеричне число $7B4_{\langle 16 \rangle}$ у десяткову систему числення.

Розв'язання. Подамо число $7B4_{\langle 16 \rangle}$ у вигляді числа 7.11.4 і розділимо його на $A = 10$:

```
_7.11.  | 10
 7.8    |_12.5 | 10
 _3.4     10   |_1.3 | 10
  3.2     _2.5    10 |_1 | 10
    2      1.14    9   0 | 0
              7        1
```

Відповідь. $7B4_{\langle 16 \rangle} = 1972_{\langle 10 \rangle}$.

Перевірка.

```
_1972 | 16
 16   |_123 | 16
 _37    112 | 7
  32     11
  _52
   48
    4
```

$7.11.4_{\langle 16 \rangle} = 7B4_{\langle 16 \rangle}$.

РОЗДІЛ 3. НЕОДНОРОДНІ СИСТЕМИ ЧИСЛЕННЯ

Лекція 39
ФАКТОРІАЛЬНІ СИСТЕМИ ЧИСЛЕННЯ

1. Загальні положення

Факторіальні системи числення належать до систем зі *змішаною* основою.

Зазвичай під факторіальною системою числення розуміють вираз, який має вигляд:

$$F_{<\phi>} = X_n \cdot n! + X_{n-1} \cdot (n-1)! + \ldots + X_l \cdot l! + \ldots + X_1 \cdot 1! + X_0 \cdot 0!,$$

де $$l = 0,1,\ldots,n; \quad 0 \le X_l \le l.$$

Він має назву *нумераційної,* або *числової,* функції. Максимальне число у факторіальній системі має вигляд:

$$F_{<\text{ф}>\max} = (n+1)! - 1.$$

Це випливає з нижченаведених перетворень числової функції, коли $X_l = l$. У цьому разі

$$F = F_{\max} = (n+1-1) \cdot n! + ((n-1)+1-1) \cdot (n-1)! + \ldots$$
$$\ldots + (l+1-1) \cdot l! + \ldots + (1+1-1) \cdot 1! + (0+1-1) \cdot 0! =$$
$$= (n+1)! - n! + n! - (n-1)! + \ldots + (l+1)! - l! + \ldots + 2! - 1! + 1 - 1! =$$
$$= (n+1)! - 1.$$

Мінімальне число 00...0...0 у факторіальній системі числення $F_{\min} = 0$.

Дійсно, якщо всі розряди $X_l = 0$, то

$$F = F_{\min} = 0 \cdot n! + 0 \cdot (n-1)! + \ldots + 0 \cdot l! + \ldots + 0 \cdot 1! + 0 \cdot 0! = 0.$$

Діапазон факторіальних чисел

$$P = F_{\max} + 1.$$

При його знаходженні враховується, крім максимального числа, ще й нуль.

2. Арифметичні операції

При виконанні арифметичних операцій додавання й віднімання у факторіальній системі числення в нульовому розряді використовують правила унітарної (одиничної) системи числення, у першому – двійкової, у другому – трійкової й т.д. Операції множення й ділення виконуються з допомогою операцій додавання й віднімання за загальними правилами для однорідних систем числення.

Приклад 1. Виконати операції додавання й віднімання факторіальних чисел $A_{<ф>} = 23110$ і $B_{<ф>} = 12200$.

Розв'язання. Застосовуючи вираз для числової факторіальної функції, отримаємо максимальне число:

$$F_{max} = 4\cdot 4! + 3\cdot 3! + 2\cdot 2! + 1\cdot 1! + 0\cdot 0! = 119.$$

Додавання й віднімання чисел виконується таким чином:

$$\begin{array}{r} 23110 \\ +\;12200 \\ \hline 42010 \end{array} \qquad \begin{array}{r} 23110 \\ -\;12200 \\ \hline 10210 \end{array}$$

Перевірка.

$23110_{<ф>} = 2\cdot 4! + 3\cdot 3! + 1\cdot 2! + 1\cdot 1! + 0\cdot 0! = 69_{<10>}$;

$12200_{<ф>} = 1\cdot 4! + 2\cdot 3! + 2\cdot 2! + 0\cdot 1! + 0\cdot 0! = 40_{<10>}$;

$42010_{<ф>} = 4\cdot 4! + 2\cdot 3! + 0\cdot 2! + 1\cdot 1! + 0\cdot 0! = 69_{<10>} + 40_{<10>} = 109_{<10>}$;

$10210_{<ф>} = 1\cdot 4! + 0\cdot 3! + 2\cdot 2! + 1\cdot 1! + 0\cdot 0! = 69_{<10>} - 40_{<10>} = 29_{<10>}$.

3. Перетворення факторіальних чисел в однорідні числа

Перетворення факторіального числа в однорідне виконується шляхом підстановки факторіального числа в числову (нумераційну) функцію для факторіальних систем числення. При цьому виконуються всі вказані в цій функції операції множення і додавання.

Приклад 2. Перетворити факторіальне число $F_{<\phi>} = 232200$ у десяткову систему числення.

Розв'язання.

$$F_{<\phi>} = 2 \cdot 5! + 3 \cdot 4! + 2 \cdot 3! + 2 \cdot 2! + 0 \cdot 1! + 0 \cdot 0! = 240 + 72 + 12 + 4 = 328_{<10>}.$$

4. Виявлення помилок

У факторіальній системі числення можуть бути одержані помилкові числа. Ознакою помилок буде порушення обмежень щодо величин цифр X_l, де $0 \le X_l \le l$.

Наприклад, факторіальне число 23210 є правильним, а число 23220 – помилковим, оскільки в першому розряді цього числа порушене обмеження щодо величини його цифри, яка повинна бути не більшою за 1, тобто дорівнювати 0 або 1.

5. Перетворення однорідних чисел у факторіальні

Перетворення числа із однорідної системи числення у факторіальну відбувається в такій послідовності.

Першим кроком буде ділення числа, яке перетворюється, на 1. Залишок у цьому випадку буде створювати цифру нульового розряду. Очевидно, що вона дорівнює нулю, а частка – числу, яке перетворюється. Якщо це число й відповідно знайдена частка дорівнюють 0 або 1, то тоді в перший розряд відповідного числа записується 0 або 1 і перетворення закінчується. Якщо перетворюване число і відповідна частка після першого кроку ділення буде більшою за 1, то наступним (другим) кроком буде ділення її на двійку, і тоді отриманий залишок від ділення буде записаний як цифра першого розряду факторіального числа. Потім аналізується величина отриманої під час діленні на двійку частки. Якщо вона менше від 3, то в другий розряд факторіального числа записується її значення, а якщо дорівнює або більша за 3, то виконується ділення цієї частки на 3. Далі під час наступних кроків виконуються щодо залишку й частки такі ж самі операції, як і під час другого кроку. Відмінність полягає в тому, що під час четвертого кроку відповідно виконується ділення на 4, на п'ятому на 5, і так продовжується до того часу, поки частка не стане меншою за свого дільника. Потім ця частка виписується і справа наліво за нею виписуються всі отримані раніше залишки. Їх послідовність створює число, яке потрібно було знайти.

Приклад 3. Перетворити десяткове число $D = 69_{<10>}$ у факторіальне $F_{<\phi>}$.

Розв'язання. Процес перетворення однорідного числа $69_{<10>}$ відображений на рис. 4.5.

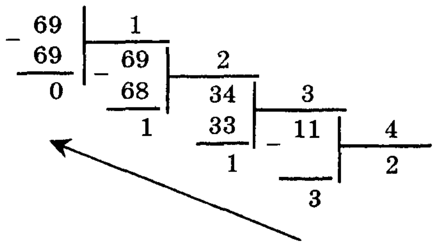


Рис 4.5. Перетворення $69_{<10>}$ *в* $23110_{<\phi>}$

Відповідь. $F_{<\phi>} = 23110_{<\phi>}$.

6. Побудова перестановок на основі факторіальних чисел

Факторіальні системи числення дають можливість для побудови широкого класу комбінаторних конфігурацій, серед яких особливе значення мають перестановки.

Розглянемо алгоритм їх побудови. Для того, щоб знайти відповідність між числом у факторіальній системі числення й перестановкою, необхідно цифру, яка стоїть у n-му розряді факторіального числа залишити без змін і вважати її *першим* елементом перестановки. Наступну цифру $(n-1)$-го розряду факторіального числа необхідно порівняти з першим елементом перестановки і, якщо вона буде дорівнювати йому або буде більшою від нього, то треба збільшити цю цифру на 1, а якщо ні, то залишити її без змін. В обох випадках буде отриманий *другий* елемент перестановки.

Далі цифру $(n-2)$-го розряду факторіального числа порівнюють спочатку з меншим за величиною елементом серед двох елементів раніше сформованої частини перестановки і якщо вона дорівнює йому або більше його, то збільшують її на 1, а якщо ні, залишають без змін. У цьому випадку третій елемент перестановки буде сформований. Якщо ж факторіальна цифра була збільшена на 1, виконують порівняння збільшеної на 1 цифри $(n-2)$-го розряду з елементом перестановки, що залишився, і потім, якщо вона дорівнює йому або більша за нього, знову збільшують її на 1. А якщо ні, то залишають її без змін. Отримана таким чином збільшена цифра факторіального числа буде третім елементом перестановки.

Аналогічно виконують порівняння цифри $(n-3)$-го розряду й далі всіх цифр факторіального числа, які ще не порівнювались, аж до нульового розряду числа з елементами перестановки, яка будується.

Тобто в загальному випадку спочатку виконують порівняння цифри факторіального числа з найменшим елементом серед уже знайдених елементів перестановки. Якщо ця цифра дорівнює цьому найменшому елементу або більша за нього, то тоді вона збільшується на одиницю. А якщо ні, вона стає черговим елементом перестановки. Збільшена ж на одиницю цифра факторіального числа далі порівнюється з найменшим елементом сформованої частини перестановки, до якої не належить елемент, щодо якого вже відбулося порівняння, і далі цикл повторюється до того часу, поки не буде сформований елемент перестановки. Далі вибирають наступну цифру факторіального числа і з її допомогою за вищенаведеним правилом знаходять новий елемент перестановки, і так буде продовжуватися до останньої цифри факторіального числа.

Приклад 4. Дане факторіальне число $F_{<ф>} = 1200_{<ф>}$. Треба знайти перестановку $B_{<пер>}$, відповідну до цього числа.

Розв'язання. Цифра 1 факторіального числа $1200_{<ф>}$ буде першим елементом перестановки, цифра 2 більша від цієї 1 і тому відповідно збільшується на 1. Це означає, що другий елемент перестановки буде дорівнювати 3. Далі цифра 0 факторіального числа менша за цифри 1 і 3. Отже, вона залишається без змін. Тобто третім елементом перестановки буде 0. Остання цифра факторіального числа є 0. Вона дорівнює лише третьому елементу перестановки і менша за перше й друге. Тому збільшуємо цифру 0 на 1 і отримуємо 1. Ця цифра 1 дорівнює першому елементу перестановки. Тому збільшуємо її ще раз на 1 і отримуємо цифру 2. Ця цифра менша від другого елемента перестановки, і тому більше немає потреби збільшувати її на 1, тобто останній елемент перестановки буде дорівнювати 2. У результаті будемо мати перестановку $B_{<пер>} = 1302_{<пер>}$. Таким чином, факторіальному числу 1200 відповідає перестановка 1302.

7. Перехід від перестановок до факторіальних чисел

Алгоритм переходу від перестановки до факторіального числа містить такі кроки: за цифру n-го розряду факторіального числа береться старший n-ий елемент перестановки. Наступна цифра факторіального числа повинна дорівнювати $(n-1)$-му елементу перестановки, якщо цей елемент менший від n-го елемента перестановки, або дорівнювати зменшеному на 1 його значенню, якщо вона більша цього елемента. Очевидно, що дорівнювати йому вона не може. Відповідно від l-го елемента перестановки віднімається стільки одиниць, скільки в перестановці буде менших попередніх йому елементів. Якщо таких елементів не буде, то цифра l факторіального числа буде дорівнювати l-му елементу перестановки.

Приклад 5. Дана перестановка $B_{<пер>} = 045321$. Треба знайти відповідне їй число $F_{<ф>}$ у факторіальній системі числення.

Розв'язання. Спираючись на вищерозглянутий алгоритм перетворення перестановок у факторіальні числа, отримуємо:

$$B_{<пер>} = 045321_{<пер>} \to F_{<ф>} = 033210_{<ф>}.$$

8. Нумерація перестановок

За допомогою факторіальних чисел розв'язується також задача перетворення перестановок у відповідні їм числа однорідних систем числення, тобто нумерації перестановок. Для цього потрібно за наданою перестановкою знайти відповідне факторіальне число і потім за допомогою числової факторіальної функції одержати номер перестановки.

Приклад 6. Знайти номер перестановки $B_{<пер>} = 045321$.

Розв'язання. Згідно з прикладом 5 маємо $045321_{<пер>} \leftrightarrow 033210_{<ф>}$. Застосувавши факторіальну числову функцію до числа $033210_{<ф>}$, отримуємо:

$$033210_{<\phi>} = 0 \cdot 5! + 3 \cdot 4! + 3 \cdot 3! + 2 \cdot 2! + 1 \cdot 1! + 0 \cdot 0! = 95_{<10>}.$$

Тобто перестановці 045321 відповідає десяткове число 95.

Перестановки нині досить широко використовуються в різних задачах, наприклад, у задачах захисту даних, задачах абстрактної алгебри, комбінаторній оптимізації. Тому алгоритми їх породження і нумерації, у тому числі і на основі факторіальних чисел, можуть знайти практичне застосування. Особливістю використання факторіальних систем числення при вирішенні таких і подібних їм задач є те, що вони дозволяють будувати спеціалізовані обчислювальні пристрої, які дозволяють збільшити швидкодію й надійність алгоритмів, які реалізуються.

Не все, що можна порахувати, пораховане,
і не все, що пораховане, можна порахувати.

А. Ейнштейн

Лекція 40
БІНОМІАЛЬНІ СИСТЕМИ ЧИСЛЕННЯ

1. Загальні положення

Нижче розглядаються комбінаторні системи числення з біноміальною основою й двійковим алфавітом {1, 0} – двійкові *біноміальні системи числення* (ще є й багатозначний варіант біноміальних систем числення). Їх діапазон $P = C_n^k$.

Корисними властивостями біноміальних систем числення є: перешкодостійкість їх чисел, яка використовується під час передавання, зберігання та оброблення інформації, а також здатність перебирати, генерувати й нумерувати комбінаторні коди. Це призводить до можливості побудови на основі біноміальних чисел перешкодостійких цифрових пристроїв і систем, які обробляють, передають, стискають і захищають інформацію.

Біноміальні числа мають дві форми, в яких вони можуть бути зображені, – *лінійну* і *матричну*. Для лінійної форми характерне розташування цифр у числах у вигляді їх послідовностей, а для матричної – у вигляді матриць. Спочатку розглянемо лінійні біноміальні системи числення, а вже потім на їх основі матричні, як більш складні. Математична модель як лінійних, так і матричних систем числення при цьому залишається однією і тією ж самою – змінюється лише форма подання чисел. Розглянемо більш докладно цю модель.

Кількісний еквівалент числа n-розрядної двійкової k-біноміальної системи числення $A_i = (a_{j-1}, a_{j-2}, ..., a_0)$, $i = 0, 1, ..., P-1$, визначається числовою функцією:

$$A_i = a_{j-1} C_{n-1}^{k-q_j} + ... + a_l C_{n-j+l}^{k-q_{l+1}} + ... + a_0 C_{n-j}^{k-q_1}$$

при дотриманні однієї з двох систем обмежень:

$$\begin{cases} q_0 = k \\ k \le j \le n-1 \\ a_0 = 1 \end{cases}$$

та

$$\begin{cases} n-k=j-q_0 \\ 0 \le q_0 \le k-1 \\ a_0=0, \end{cases}$$

де q_0 – кількість одиниць в біноміальному числі;

j – кількість розрядів біноміального числа (довжина);

$l = 0, 1, ..., j-1$ – порядковий номер розряду;

q_{l+1} – сума одиничних значень цифр від $(j-1)$-го розряду до $(l+1)$-го включно,

$$q_{l+1} = \sum_{\gamma=l+1}^{j-1} a_\gamma ,$$

$$q_j = a_j = 0 .$$

Максимальне число в біноміальній системі числення відповідно до виразу для біноміальної числової функції має вигляд:

$$1C_{n-1}^{k-q_j} + 1C_{n-2}^{k-q_j-1} + ... + 1C_{n-j}^{k-q_1} = C_n^k - 1 = P - 1 ,$$

а мінімальне дорівнює нулю.

Кількість біноміальних чисел, які містять у кінці одиницю

$$N_1 = \sum_{i=0}^{n-k-1} C_{n-2-i}^{n-k-1-i} = C_{n-1}^{n-k-1} = C_{n-1}^k ,$$

а які містять нуль,

$$N_0 = \sum_{i=0}^{k-1} C_{n-2-i}^{k-1-i} = C_{n-1}^{k-1} .$$

Їх сумарна кількість

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

дає загальну кількість біноміальних чисел і відповідно їх діапазон для біноміальної системи числення. Тобто $P = C_n^k$.

У табл. 4.14 для $n = 6$ і $k = 4$ наведені біноміальні числа і їх кількісні еквіваленти, формування яких відбувається за таким алгоритмом:

1. Формується початкове біноміальне число, яке складається з $(n - k)$ нулів.

2. У молодший розряд цього числа записується 1 і знаходиться число одиниць у ньому.

3. Якщо число одиниць менше за k, то справа від отриманої 1 записується 0 і далі виконується перехід до пункту 2.

4. Пункт 2 і 3 повторюються до тих пір, поки кількість одиниць у біноміальному числі не стане дорівнювати k.

5. Якщо k одиниць займають у числі k старших розрядів, то подальше формування біноміальних чисел припиняється, а якщо ні, то в молодший розряд цього числа, який містить 0, записується 1, а у всі інші менші відносно до нього розряди з одиницями – 0.

6. Якщо при цьому кількість одиниць у числі не стала дорівнювати k, то справа від меншого розряду, який містить 1, записуються нулі до тих пір, поки їх загальна кількість у числі не стане дорівнювати $(n - k)$.

7. Повернення до пункту 2.

Таблиця 4.14. Біноміальні числа і їх кількісні еквіваленти

Біноміальне число	Кількісний еквівалент
00	$0 \cdot C_5^4 + 0 \cdot C_4^4$
010	$0 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3$
0110	$0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2$
01110	$0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 0 \cdot C_1^1$
01111	$0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 1 \cdot C_1^1$
100	$1 \cdot C_5^4 + 0 \cdot C_4^3 + 0 \cdot C_3^3$
1010	$1 \cdot C_5^4 + 0 \cdot C_4^3 + 1 \cdot C_3^3 + 0 \cdot C_2^2$
10110	$1 \cdot C_5^4 + 0 \cdot C_4^3 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 0 \cdot C_1^1$
10111	$1 \cdot C_5^4 + 0 \cdot C_4^3 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 1 \cdot C_1^1$
1100	$1 \cdot C_5^4 + 1 \cdot C_4^3 + 0 \cdot C_3^2 + 0 \cdot C_2^2$

Продовження таблиці 4.14.

Біноміальне число	Кількісний еквівалент
11010	$1\cdot C_5^4+1\cdot C_4^3+0\cdot C_3^2+1\cdot C_2^2+0\cdot C_1^1$
11011	$1\cdot C_5^4+1\cdot C_4^3+0\cdot C_3^2+1\cdot C_2^2+1\cdot C_1^1$
11100	$1\cdot C_5^4+1\cdot C_4^3+1\cdot C_3^2+0\cdot C_2^1+0\cdot C_1^1$
11101	$1\cdot C_5^4+1\cdot C_4^3+1\cdot C_3^2+0\cdot C_2^1+1\cdot C_1^1$
1111	$1\cdot C_5^4+1\cdot C_4^3+1\cdot C_3^2+1\cdot C_2^1$

2. Виявлення помилок

Особливою властивістю біноміального рівномірного коду є його здатність знаходити помилки при обробці інформації. Це дозволяє організовувати одночасний контроль у каналах оброблення й передавання інформації, до яких належать біноміальні цифрові пристрої.

Ознаками помилки в біноміальному числі є перевищення в ньому кількості одиниць величини k, або кількості нулів, які стоять до останньої 1, величини $(n-k-1)$.

Для виявлення помилок за допомогою біноміальних комбінацій необхідно поповнити їх нулями до отримання рівномірного $(n-1)$-розрядного біноміального числа, як це видно з табл. 4.15.

Таблиця 4.15. Нерівномірні й рівномірні біноміальні числа

№	Біноміальний код	Біноміальний рівномірний код	№	Біноміальний код	Біноміальний рівномірний код
0	00	00000	8	10111	10111
1	010	01000	9	1100	11000
2	0110	01100	10	11010	11010
3	01110	01110	11	11011	11011
4	01111	01111	12	11100	11100
5	100	10000	13	11101	11101
6	1010	10100	14	1111	11110
7	10110	10110			

3. Побудова рівноважних кодів

Корисною властивістю біноміальних чисел є їх здатність отримувати ті комбінаторні об'єкти, які для своєї побудови використовують біноміальні коефіцієнти. До таких об'єктів належать *рівноважні* коди, або, як ще їх називають, коди с *постійною* вагою, які найшли досить широке застосування на практиці, частково в системах передавання інформації.

У табл. 4.16 наведений перехід від біноміальних чисел до комбінацій коду з постійною вагою, який здійснюється шляхом дописування до числа одиниць, якщо воно містить $(n - k)$ нулів, чи нулів, якщо в ньому міститься k одиниць, до того моменту, поки довжина числа не стане дорівнювати n.

Таблиця 4.16. Біноміальні числа з відповідними їм комбінаціями з постійною вагою

№	Біноміальний код	Код з постійною вагою	№	Біноміальний код	Код з постійною вагою
0	00	001111	8	10111	101110
1	010	010111	9	1100	110011
2	0110	011011	10	11010	110101
3	01110	011101	11	11011	110110
4	01111	011110	12	11100	111001
5	100	100111	13	11101	111010
6	1010	101011	14	1111	111100
7	10110	101101			

Біноміальне число є номером комбінації з постійною вагою, тобто є її стиснутим відображенням. Якщо існує потреба зобразити її номером у однорідній системі числення, то тоді необхідно використати числову біноміальну функцію.

4. Матричні біноміальні числа

Розглянуті вище біноміальні числа були лінійними, оскільки вибудовувались у вигляді ланцюжка цифр, які були розташовані послідовно в рядку одна за одною. Але біноміальні числа дають можливість будувати й більш складні структури їх запису, які відрізняються від лінійної форми.

Однією з важливих властивостей біноміальних систем числення є можливість подання їх чисел у вигляді спеціальних біноміальних (0, 1)-матриць (БМ), які містять k стовпчиків і $n-k$ рядків з елементами 0 і 1. Загальна кількість елементів у матриці дорівнює:

$$N = nk - k^2 = k(n-k).$$

У стовпчику матриці знаходиться не більше однієї 1. При цьому загальна кількість одиниць у матриці не перевищує значення k, а *загальна кількість нулів* – N. *Одиниці в матриці в кількості від 1 до* k розташовані в одному чи декількох стовпчиках так, що перша з них знаходиться в крайньому лівому, а остання – у будь-якому наступному стовпчику. При цьому між стовпчиками з одиницями відсутні стовпчики без них. Якщо у ($n-k$)-му рядку є послідовність одиниць, то вона завжди розташована в її початковій частині, починаючи зі старшого розряду так, що між одиницями відсутні нулі. Серед елементів будь-якої діагоналі матриці, яка починається зверху і спрямована зліва направо, лише один елемент може дорівнювати 1. У першому стовпчику БМ за винятком БМ нульового числа обов'язково міститься 1. Одиниці в матриці розташовані так, що одиниця кожного наступного стовпчика знаходиться або в тому самому рядку, що й в попередньому стовпці, або в одному з верхніх рядків. Якщо в першому стовпці БМ 1 відсутня, то у решті стовпчиків одиниць також не буде.

Кожна БМ зі своїм розташуванням одиниць і нулів відповідає одному біноміальному числу. Кількість одиниць у біноміальній матриці й лінійному біноміальному числі, яке їй відповідає, дорівнюють одна одній. Логічне підсумовування елементів діагоналей біноміальної матриці, які йдуть направо й униз, утворює цифри біноміального числа.

Біноміальні матриці дозволяють уникнути переносів при виконанні підсумовування й віднімання з біноміальних чисел 1 і мають підвищену здатність до знаходження та виправлення помилок.

На рис. 4.6 для $n=6$ і $k=3$ наведені всі можливі $P=C_n^k$ БМ з десятковими (зверху) і біноміальними (знизу) номерами. Алгоритми побудови матричних біноміальних чисел на основі їх лінійних форм і, навпаки, побудови лінійних біноміальних чисел на основі матричних легко знайти на рис. 4.6, що рекомендується зробити самостійно як контрольне завдання.

0
$$\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix}$$
0 0 0 0 0

1
$$\begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}$$
0 0 1 0 0

2
$$\begin{bmatrix}1&1&0\\0&0&0\\0&0&0\end{bmatrix}$$
0 0 1 1 0

3
$$\begin{bmatrix}1&1&1\\0&0&0\\0&0&0\end{bmatrix}$$
0 0 1 1 1

4
$$\begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix}$$
0 1 0 0 0

5
$$\begin{bmatrix}0&1&0\\1&0&0\\0&0&0\end{bmatrix}$$
0 1 0 1 0

6
$$\begin{bmatrix}0&1&1\\1&0&0\\0&0&0\end{bmatrix}$$
0 1 0 1 1

7
$$\begin{bmatrix}0&0&0\\1&1&0\\0&0&0\end{bmatrix}$$
0 1 1 0 0

8
$$\begin{bmatrix}0&0&1\\1&1&0\\0&0&0\end{bmatrix}$$
0 1 1 0 1

9
$$\begin{bmatrix}0&0&0\\1&1&1\\0&0&0\end{bmatrix}$$
0 1 1 1 0

10
$$\begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix}$$
1 0 0 0 0

11
$$\begin{bmatrix}0&1&0\\0&0&0\\1&0&0\end{bmatrix}$$
1 0 0 1 0

12
$$\begin{bmatrix}0&1&1\\0&0&0\\1&0&0\end{bmatrix}$$
1 0 0 1 1

13
$$\begin{bmatrix}0&0&0\\0&1&0\\1&0&0\end{bmatrix}$$
1 0 1 0 0

14
$$\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}$$
1 0 1 0 1

15
$$\begin{bmatrix}0&0&0\\0&1&1\\1&0&0\end{bmatrix}$$
1 0 1 1 0

16
$$\begin{bmatrix}0&0&0\\0&0&0\\1&1&0\end{bmatrix}$$
1 1 0 0 0

17
$$\begin{bmatrix}0&0&1\\0&0&0\\1&1&0\end{bmatrix}$$
1 1 0 0 1

18
$$\begin{bmatrix}0&0&0\\0&0&1\\1&1&0\end{bmatrix}$$
1 1 0 1 0

19
$$\begin{bmatrix}0&0&0\\0&0&0\\1&1&1\end{bmatrix}$$
1 1 1 0 0

Рис. 4.6. Біноміальні (0, 1) матриці

КОНТРОЛЬНІ ЗАВДАННЯ І ПИТАННЯ ДО ЧАСТИНИ IV

Питання для самоконтролю

Початкові поняття про системи числення

1. Охарактеризуйте кількісні й порядкові властивості чисел.

2. Дайте характеристику непозиційним системам числення.

3. Що таке позиційні системи числення?

4. Які ознаки позиційних систем числення відрізняють їх від непозиційних?

5. Що вам відомо про історію розвитку систем числення? У чому їх значення для науки й техніки.

Загальна характеристика систем числення

6. Що таке числова функція позиційних систем числення?

7. Дайте визначення структури позиційних систем числення.

8. Дайте визначення розрядів позиційних чисел і їх ваги.

9. Що таке основа, діапазон і алфавіт позиційних систем числення?

10. Дайте характеристику рівномірних й нерівномірних позиційних чисел.

Класифікація позиційних систем числення

11. Що таке однорідні системи числення?

12. Що таке неоднорідні системи числення?

13. Охарактеризуйте структурні системи числення.

14. Дайте характеристику комбінаторним системам числення.

15. Що таке табличні системи числення?

Однорідні системи числення

16. Що таке алфавіт і основа в однорідних системах числення?

17. Що таке числова функція однорідних систем числення?

18. Як виконується операція додавання в однорідних системах числення?

19. Охарактеризуйте особливості додавання, віднімання, множення й ділення у двійковій системі числення.

20. У чому виявляються особливості додавання, віднімання, множення й ділення у восьмеричній (вісімковій) системі числення?

21. Що являють собою операції додавання, віднімання, множення й ділення в шістнадцятеричній системі числення?

Переведення однорідних чисел

22. Що таке табличний метод переведення чисел з однієї системи числення в іншу?

23. Як відбувається переведення чисел у систему числення з кратною основою?

24. Як відбувається переведення чисел шляхом підбору степеней?

25. Як відбувається переведення чисел у систему з основою, що є степенем початкової?

26. Як відбувається переведення чисел у систему числення на базі проміжного перетворення цифр?

27. Як відбувається переведення чисел на основі використання проміжної системи числення?

28. Як відбувається переведення чисел у систему числення шляхом ділення її на основу?

29. Як відбувається переведення чисел з двійкової системи числення у восьмеричну й навпаки?

30. Як відбувається переведення чисел з двійкової системи числення в шістнадцятеричну й навпаки?

Неоднорідні системи числення

31. Що таке факторіальні системи числення?

32. Як відбувається переведення чисел з факторіальної системи числення?

33. Як відбувається переведення чисел у факторіальну систему числення?

34. Як відбувається перехід від факторіального числа до перестановки?

35. Як відбувається перехід від перестановки до факторіального числа?

36. Які системи числення називаються біноміальними?

37. Як відбувається переведення чисел з біноміальної системи числення в однорідну?

38. Як відбувається переведення чисел у біноміальну систему числення з однорідної?

39. Як відбувається перехід від біноміального числа до сполучення?

40. Як відбувається перехід від сполучення до біноміального числа?

41. Дайте означення матричних біноміальних чисел і охарактеризуйте спосіб їх побудови.

Контрольні завдання

1. Згідно із заданим нижче в першій колонці табл. 4.17 номеру варіанту перевести числа, розташовані в другій і третій колонці цієї ж самої таблиці при рахунку зліва направо з десяткової системи числення у двійкову шляхом ділення на основу; виконати додавання й віднімання отриманих двійкових чисел; перевести результат в десяткову систему числення, а також у вісімкову й шістнадцятеричну. Провести перевірку.

2. Відповідно до вашого варіанта, поданого в табл. 4.17, перетворити всіма вищенаведеними способами число із шістнадцятеричної системи числення в десяткову й навпаки. Дане число розташоване в четвертій колонці в разі рахування зліва направо.

Таблиця 4.17. Варіанти чисел для переводу

1	108	21	A81C
2	109	47	B93A
3	119	93	CA48
4	191	53	DB59
5	112	84	EC61
6	113	37	FD72
7	184	95	1EA3
8	125	18	2FB4
9	146	56	37CA
10	117	28	48DB
11	178	45	59EC
12	107	98	6AFD
13	196	49	7B1E
14	165	48	8C2F
15	124	53	9D3A
16	193	71	AE45
17	182	88	BF56
18	101	75	C96B
19	129	56	DA77
20	120	79	E38B
21	181	73	FC93
22	137	49	5DA1
23	132	51	6EB2
24	197	83	7FC3
25	141	52	8AD4
26	144	36	9BE5
27	147	57	A4F6
28	151	61	B31A
29	156	43	C23B
30	195	72	F421

Додатки

Додаток 1
МАТЕМАТИЧНА ІНДУКЦІЯ

1. Загальні положення

У різних сферах своєї діяльності (науці, побуті чи виробництві) людина у своїх логічних висновках застосовує дедуктивний або індуктивний підходи, які ґрунтуються на поняттях *дедукції* й *індукції*. Дедукція (лат. *deduction* – висновок) являє собою перехід від загального до окремого, а індукція (лат. *inductio* – наведення) – перехід від окремого до загального. Загальним для цих підходів є те, що вони доводять істинність чи хибність деяких тверджень, які до цього сприймались як *гіпотези*, тобто як передбачувані припущення.

Ці твердження поділяються на *загальні* й *окремі*. Твердження, що всі натуральні парні числа діляться на 2, є загальним, а твердження, що ціле число $x = 6$ ділиться на 2, є окремим. Важливість дедукції полягає в тому, що вона дозволяє на основі загальних тверджень доводити окремі твердження. Дійсно, якщо парні числа діляться на 2, а число 6 парне, то звідси випливає, що число 6 повинне ділитися на 2.

Індукція на відміну від дедукції виходить з окремих тверджень і тому не завжди може переходити від них до загальних тверджень. Але все ж таки в багатьох випадках на її основі робляться близькі до достовірних вірогідні припущення. Наприклад, коли за деякими зразками визначаються властивості золота, то ці властивості людина поширює на все інше золото, яке є в світі. Такий підхід застосовується не лише щодо золота, але й щодо інших хімічних елементів, і поки що він себе виправдовував. Хоча повної гарантії такий підхід усе ж таки не дає, і тому він є неповним. Це ж саме стосується й багатьох законів природи. Їх людина не доводить, а виявляє. Але, крім такої неповної індукції, у математиці окремо використовується ще *повна* індукція.

2. Неповна (звичайна) індукція

Звичайна індукція являє собою індуктивний підхід до науки взагалі, а не тільки до математики і є *неповною* індукцією, оскільки не дає можливості одержувати завжди достовірну інформацію про властивості, які досліджуються.

Ця індукція хоча й широко використовується в математиці, але не завжди дає вірний і кінцевий висновок про істинність або хибність того чи іншого математичного твердження, оскільки вона створює його на основі певних окремих результатів, одержаних при дослідженні тієї чи іншої математичної властивості. Це твердження можна вважати дійсним лише в разі дослідження всіх без винятку можливих окремих результатів, що буває тільки тоді, коли їх кількість є *скінченною*. У випадку, коли кількість можливих результатів *нескінченна*, отримати висновок про істинність або хибність твердження щодо тієї чи іншої властивості за допомогою звичайної індукції *неможливо* в принципі.

Нехай маємо тричлен $x^2 + x + 41$. Якщо підставити в нього замість x нуль, то одержимо просте число 41; якщо одиницю, то – 43; якщо 2, 3, 4, 5, 6, 7, 8, 9, 10, то – числа 47, 53, 61, 71, 83, 87, 113, 131, 151 відповідно. Усі вони прості числа. Здається, можна припустити, що в разі підстановки в цей тричлен будь-якого цілого додатного числа завжди в результаті будемо одержувати просте число. Однак це не так. Уже в разі $x = 40$ зазначений тричлен ділиться на 41, а за умови $x = 41$ $x^2 + x + 41 = 41^2 + 41 + 41$, тобто тричлен ділиться на 41. У цьому небезпека звичайної індукції, оскільки вона не гарантує позитивного результату в будь-якому випадку.

3. Повна індукція

Повна, або цілковита, індукція – це ***математична*** індукція, яка дозволяє робити достовірні узагальнення на основі неповної індукції. Тобто для доведення теореми з допомогою математичної індукції потрібно, щоб результат цієї теореми з деякою вірогідністю вже був встановлений раніше у вигляді *гіпотези*. Потрібно лише далі для неї довести, що цей результат є достовірно *загальним* для задач даного типу. Тобто доведенню тієї чи іншої теореми, яка відображає рішення відповідної математичної задачі, повинна передувати гіпотеза, яку й потрібно довести або відкинути. Це означає, що математична індукція ґрунтується на *звичайній* індукції.

Математична індукція на протилежність звичайній гарантує стовідсоткову вірогідність, тобто достовірність одержаних з її допомогою результатів. Цей метод стосується тільки теорем, що відображають загальні властивості натуральних чисел 1, 2,..., та інших розділів математики, які спираються на натуральні числа. Наприклад, до таких розділів належить арифметика цілих чисел та теорія раціональних чисел, що ґрунтується на ній. Існують також розділи математики, які можуть бути інтерпретовані в термінах арифметики, наприклад, евклідова геометрія. Відповідно, у цих розділах також може бути використаний метод математичної індукції. Математична індукція, яка ще має назву ***індукція за побудовою***, використовується також для доведення *логічних* формул.

Основою метода математичної індукції є її принцип, що поширюється на будь-які твердження $P(n)$, які стосуються чисел $n = 1, 2, \ldots$. Сформулюємо цей принцип у вигляді теореми.

Теорема 1. ***Будь-яке твердження $P(n)$ дійсне для будь-якого n у випадку, якщо воно дійсне для $n = 1$, та із істинності цього твердження для будь-якого довільного $n = k$ випливає його істинність для $n = k + 1$.***

Ця теорема розпадається на дві леми, перша з яких (**лема 1**) вимагає, щоб твердження $P(n)$ було справедливим для $n = 1$, а друга (**лема 2**) – для $n = k + 1$, за умов, що твердження $P(n)$ справедливе для $n = k$. Лише в цьому випадку твердження $P(n)$ буде справедливе для будь-якого n.

Доведення. Припустимо, що умови лем 1 і 2 виконуються. Тоді твердження $P(n)$ відповідно до леми 1 дійсне для $n = k = 1$. Відповідно до другої леми $P(n)$ дійсне для $n = k + 1 = 1 + 1 = 2$. Але якщо $P(n)$ дійсне для $n = k = 2$, то відповідно до тієї самої другої леми воно буде дійсне і для $n = k + 1 = 2 + 1 = 3$ і далі для $n = 4$ тощо необмежено для всіх можливих n.

Теорему доведено.

Із теореми випливає метод математичної індукції, що складається із виконання нижченаведених пунктів, які стосуються одержання твердження $P(n)$:

1. Висувається нова гіпотеза у вигляді твердження $P(n)$ про деяку математичну властивість, що може бути як істинною, так і хибною.

2. Здійснюється перевірка гіпотези $P(n)$ для $n = 1$. Якщо $P(n = 1)$ підтверджується, то відбувається перехід до наступного пункту 3. А якщо ні, то гіпотеза, що перевіряється, вважається неправильною й виконується перехід до пункту 1.

3. Здійснюється доведення гіпотези $P(n)$ для $n = k + 1$ за умови припущення, що $P(n = k)$ істинне.

4. Якщо $P(n = k + 1)$ істинне, то доведення гіпотези $P(n)$ для будь-якого натурального n одержане. Якщо $P(n = k + 1)$ хибне, то здійснюється перехід до пункту 1.

Перший крок наведеного алгоритму являє собою звичайну (неповну) індукцію, яка реалізує індукційний підхід до науки в цілому.

Другий і третій кроки алгоритму є наслідком дії принципу математичної індукції та реалізують його практично. Принцип математичної індукції гарантує за умови, якщо другий і третій кроки методу виконані, що твердження $P(n)$ дійсне. При цьому другий крок грунтується на лемі 1 і є *основою* (базою) метода математичної індукції, а третій – на лемі 2, яка визначає *індукційний крок* (перехід) методу.

На рис.1.1 наведена блок-схема реалізації метода математичної індукції. Вона показує, що метод математичної індукції містить два основних блоки – блок, що реалізує основу індукції, і блок, що реалізує індукційний перехід.

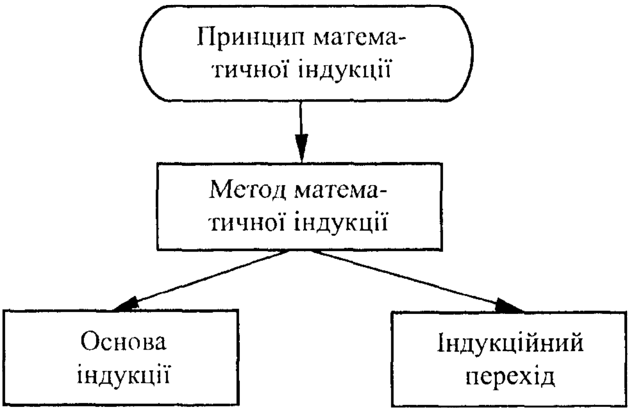


Рис. 1.1. Блок-схема, яка реалізує математичну індукцію

Слід ще раз звернути увагу на те, що основою метода математичної індукції є принцип математичної індукції, який твердить, що якщо доведені леми 1 і 2 для твердження $P(n)$, то це твердження істинне для будь-якого $n = 1, 2, \dots$.Тому завданням метода є *доведення* лем 1 і 2 для $P(n)$.

4. Приклади доведення методом математичної індукції

Приклад 1. Знайти вираз для суми перших n непарних чисел натурального ряду

$$P(n) = 1 + 3 + ... + (2n - 1).$$

Розв'язання. На основі аналізу перших початкових чисел натурального ряду за допомогою неповної індукції отримуємо гіпотезу, що $P(n) = n^2$.

Застосувавши метод математичної індукції, доведемо дану гіпотезу.

Відповідно до першого кроку метода, який доводить лему 1, одержимо, що $P(n = 1) = n^2 = 1^2$.

Лема 1 доведена.

Перейдемо до другого кроку – доведення леми 2.

Для цього припустимо відповідно до леми 2, що гіпотеза $P(n)$ правильна для $n = k$. Тоді

$$P(n = k) = 1 + 3 + ... + (2\,k - 1) = k^2.$$

Доведемо тепер, що ця гіпотеза правильна й для

$$P(n = k + 1) = 1 + 3 + ... + (2k - 1) + (2\,(k + 1) - 1) = (k + 1)^2.$$

Подамо для цього $P(n = k + 1)$ таким чином:

$$P(n = k + 1) = P(n = k) + (2\,(k + 1) - 1).$$

Тоді, оскільки $P(n = k) = k^2$,

$$P(n = k + 1) = k^2 + 2\,k + 1 = (k + 1)^2.$$

Результат співпадає з потрібним.

Гіпотеза підтверджена.

Приклад 2. Довести, що сума n перших чисел натурального ряду

$$P(n) = 1 + 2 + ... + n = \frac{n(n+1)}{2}.$$

Розв'язання. У даному випадку неповну індукцію для висунення гіпотези $P(n)$ виконувати не потрібно, оскільки вона задана в умові прикладу.

Наведемо доведення леми 1 для даного випадку:

$$P(n=1)=\frac{1(1+1)}{2}=1.$$

Таким чином, основа повної індукції одержана.

Доведення леми 2 має такий вигляд. Нехай

$$P(n=k)=1+2+...+k=\frac{k(k+1)}{2}.$$

Тоді відповідно до леми 2 потрібно одержати вираз

$$P(n=k+1)=1+2+...+k+(k+1)=\frac{(k+1)(k+2)}{2}.$$

Доведемо це. Дійсно,

$$P(n=k+1)=P(n=k)+(k+1)=\frac{k(k+1)}{2}+(k+1)=$$

$$=\frac{k^2+3k+2}{2}=\frac{(k+1)(k+2)}{2}.$$

Гіпотеза підтверджена.

Приклад 3. Довести, що сума квадратів перших чисел натурального ряду

$$P(n)=1^2+2^2+...+n^2=\frac{n(n+1)(2n+1)}{6}.$$

Розв'язання. У цьому прикладі, як і в прикладі 2, твердження $P(n)$ задане у вигляді гіпотези, яка може перетворитися в істинне твердження тільки після доведень лем 1 і 2.

Лема 1 доводиться шляхом підстановки 1 замість n:

$$P(n=1)=\frac{1(1+1)(2\cdot 1+1)}{6}=\frac{6}{6}=1=1^2.$$

Для доведення леми 2 припустимо, як і в попередніх прикладах, що твердження $P(n)$ у випадку $n = k$ справедливе, тобто для цього прикладу

$$P(n=k)=1^2+2^2+...+k^2=\frac{k(k+1)(2k+1)}{6}.$$

Доведемо, що справедливим буде також твердження, що

$$P(n = k + 1) = 1^2 + 2^2 + ... + k^2 + (k + 1)^2 = P(n = k) + (k + 1)^2.$$

Для цього спочатку доведемо, що

$$P(n = k + 1) - P(n = k) = (k + 1)^2.$$

Дійсно,

$$P(n = k + 1) - P(n = k) =$$

$$=\frac{(k+1)(k+2)(2k+3)}{6}-\frac{k(k+1)(2k+1)}{6}=\frac{k+1}{6}\left[(k+2)(2k+3)-k(2k+1)\right]=$$

$$=\frac{k+1}{6}\left[2k^2+3k+4k+6-2k^2-k\right]=\frac{k+1}{6}\left[6k+6\right]=(k+1)^2.$$

З доведеного далі випливає, що $P(n = k + 1) = P(n = k) + (k + 1)^2$. Гіпотеза підтверджена.

Слід особливо підкреслити, що для правильного використання метода математичної індукції обов'язково потрібно доводити обидві леми 1 і 2, оскільки лема 1 є основою для проведення індукційних кроків у методі, що розглядається, а лема 2 дозволяє виконувати правильний перехід від випадку з $n = k$ до випадку з $n = k + 1$, який іде за ним. Якщо лема 1 не доведена, то тоді відсутня основа для проведення індукційних кроків. У результаті за допомогою леми 2 можна довести помилкову гіпотезу $P(n)$.

5. Вправи для самостійної роботи

1. Довести, що сума квадратів n перших непарних чисел натурального ряду

$$P(n)=1^2+3^2+...+(2n-1)^2=\frac{n(2n-1)(2n+1)}{3}.$$

2. Довести, що сума кубів n перших чисел натурального ряду

$$P(n)=1^3+2^3+...+n^3=\left(\frac{n(n+1)}{2}\right)^2.$$

3. Довести, що

$$P(n)=1\cdot 2+2\cdot 3+...+(n-1)n=\frac{(n-1)n(n+1)}{3}.$$

4. Довести, що

$$P(n)=1\cdot 2\cdot 3+2\cdot 3\cdot 4+...+n(n+1)(n+2)=\frac{n(n+1)(n+2)(n+3)}{4}.$$

5. Довести, що

$P(n) = 1\cdot 1!+2\cdot 2!+...+n\cdot n! = (n+1)!-1$, де $n! = 1\cdot 2\cdot ...\cdot n$.

6. Із $2n$ чисел 1, 2, ..., $2n$ довільно обрані $(n + 1)$ чисел. Довести твердження $P(n)$, що серед обраних чисел знайдуться хоча б два числа, з яких одне ділиться на друге.

7. Довести, що n різних прямих, проведених на площині через одну точку, ділять площину на $2n$ частин.

6. Історична довідка

Першим відомим ученим, який застосував метод математичної індукції в сучасному вигляді в 1575 році, був італійський вчений Франческо Мауроліко. У XVII столітті цей метод був удосконалений П'єром де Ферма, який назвав його методом нескінченного спуску. Він також використовувався Паскалем, зокрема, з метою доведення формули для біноміальних коефіцієнтів.

До цього часу з допомогою метода математичної індукції було виконано безліч різних доведень математичних теорем, і їх кількість, безперечно, у майбутньому буде зростати.

Додаток 2
ЕТИМОЛОГО-ТЕРМіНОЛОГІЧНИЙ СЛОВНИК[1]

Основні означення

Термін	Етимологія	Значення
1	2	3
Абсолютний	лат. *absolutus* – необмежений, безумовний	безвідносний, досконалий, повний
Алгебра	араб. *аль джебр* – відновлення	частина математики, наука про загальні операції над числами, многочленами, векторами, матрицями тощо
Аналогія	гр. *ἀναλογία* – відповідність, співрозмірність	подібність у будь - якому відношенні
Асоціативний	лат. *associatio* – приєднання, об'єднання	сполучний
Бієкція	лат. *bis* – двічі; *iacio* - кидати	взаємно однозначна відповідність множин
Бінарний	лат. *binarius* – подвійний	двійковий
Вектор	лат. *vector* – той, що веде, несе	величина, що характеризується не лише числовим значенням, але й напрямком
Графік	гр. *γραφικός* – накреслений	зображення функції на координатній площині
Диз'юнкція	лат. *disiunctio* – відокремлення, розділення	логічна операція поєднання кількох висловлювань за допомогою логічного сполучника "АБО"
Дистрибутивний	лат. *distributio* – розділення, розподіл	розподільний

1	2	3
Діаграма	гр. *διάγραμμα* – рисунок	креслення, що наочно показує співвідношення між різними величинами, у вигляді відрізків або геометричних фігур
Дуальний	лат. *dualis* – двоїстий	двоїстий, притаманний дуалізму
Еквівалентність	лат. *aequus* – рівний; *valens* – той, що має силу, значення	рівноцінність, відповідність до чого-небудь у будь-якому відношенні
Елемент	лат. *elementum* – першоречовина, стихія	складова частина будь-чого
Ідемпотентність	лат. *idem* – той самий; *potens* – сильний, здатний	рівносильність, рівнозначність
Ізоморфний	гр. *ἰσομόρφος* – рівновидний, рівноформний	взаємно однозначний у відображенні двох множин при збереженні їх структурних властивостей
Ідентичний	лат. *identicus* – тотожний	тотожний, однаковий
Імпліканта	лат. *implicans* – вплетений, оплетений, переплетений	логічна функція, що не дорівнює одиниці на тих наборах, де дорівнює нулю дана функція
Імпліцента	лат. *implicens* – вплетений	логічна функція, що не дорівнює нулю на тих наборах, де дорівнює одиниці дана функція
Імплікація	лат. *implicatio* – вплетення, переплетення, оплетення	логічна функція, що створює складне вислов-лювання з двох простих за допомогою логічної зв'язки "якщо..., то..."
Інверсія	лат. *inversio* – перевертання, перестановка	логічна функція, що приймає значення, протилежне аргументу
Індукція	лат. *inductio* – виведення	логічний прийом, що полягає в переході від окремих випадків до загального висновку
Ін'єкція	лат. *iniectio* – вкидування	відображення в

1	2	3
Інтуїція	лат. *intueor* – уважно приглядатися	безпосереднє осягнення істини без логічного обґрунтування на основі попереднього досвіду
Коефіцієнт	лат. *coefficiens* – сприяючий	постійний множник при змінній (невідомій)
Композиція	лат. *compositio* – складання	структура, побудова, послідовне використання
Компонент	лат. *componens* – складова	складова частина будь-чого
Комутативний	лат. *commuto* – переставляти, змінювати	переставний
Константа	лат. *constans* – постійний	стала величина
Конституента	лат. *constituens* – установлюючий	логічна функція, що встановлює одиницю (або нуль) лише на одному своєму наборі
Кон'юнкція	лат. *coniunctio* – з'єднання, зв'язок	логічна операція "І"
Координата	лат. *co-(cum)* – разом, з *ordinatus* – упорядкований	величина, що визначає положення точки (вектора) у будь-якому просторі
Кортеж	фр. *cortége* – урочистий хід, виїзд	упорядкований набір елементів
Логіка	гр. *λογικός* – розумний	наука про закони й форми мислення
Матриця	лат. *matrix* – джерело, початок	таблиця розташованих у вигляді прямокутника будь-яких математичних об'єктів
Модуль	лат. *modulus* – міра	абсолютна величина числа, основа системи числення
Нейтральний	лат. *neutralius* – той, що не належить ні до того, ні до іншого	той, що не приєднається до жодної з протилежних сторін
Оператор	лат. *operator* – той, що діє	відображення між елементами множин
Операція	лат. *operatio* – дія	будь-яка дія, у тому числі математична
Проекція	лат. *proiectio* – кидання вперед	зображення будь-якого об'єкта на площині або прямій
Раціональний (про числа)	лат. *rationalis* – розумний	цілі і дрібні числа, а також нуль

1	2	3
Символ	гр. *συμβολον* – знак, ознака	умовне позначення будь-якої величини
Символ	гр. *συμβολον* – знак, ознака	умовне позначення будь-якої величини
Симетричний	гр. *συμμετρία* – співрозмірність, належна пропорція	співрозмірний, маючий властивість симетрії
Система	гр. *συστημα* – склад, упорядковане ціле	цілісне утворення, множина закономірно зв'язаних між собою елементів у деяку єдність
Сюр'єкція	лат. *superiectio* – кидання зверху	відображення **на**
Суперпозиція	лат. *superpositio* – спорудження	накладання
Тривіальний	лат. *trivialis* – звичайний	звичний, неоригінальний
Теорема	гр. *θεωρημα* – видовище, вчення	твердження, істинність якого доводиться
Універсальний	лат. *universalis* – загальний	загальний, різносторонній
Універсум	лат. *universum* – всесвіт	загальна множина, для якої всі інші є підмножинами
Формула	лат. *formula* – правило	математичне співвідношення
Функціонал	лат. *functio* виконання	число, що є функцією від функції
Функція	лат. *functio* – виконання	математична залежність однієї змінної від іншої

Додаток 3
ГРЕЦЬКИЙ І ЛАТИНСЬКИЙ АЛФАВІТИ

Грецький алфавіт	
Позначення літери	Назва літери
Α α	альфа
Β β	бета
Γ γ	гамма
Δ δ	дельта
Ε ε	епсілон
Ζ ζ	дзета
Η η	ета
θ θ	тета
Ι ι	йота
Κ κ	капа
Λ λ	лямбда
Μ μ	мю
Ν ν	ню
Ξ ξ	ксі
Ο ο	омікрон
Π π	пі
Ρ ρ	ро
Σ σ ς	сігма
Τ τ	тау
Υ υ	іпсілон
Φ φ	фі
Χ χ	хі
Ψ ψ	псі
Ω ω	омега

Латинський алфавіт	
Позначення літери	Назва літери
A a	а
B b	бе
C c	це
D d	де
E e	е
F f	еф
G g	ге
H h	аш
I i	і
J j	йот
K k	ка
L *l*	ель
M m	ем
N n	ен
O o	о
P p	пе
Q q	ку
R r	ер
S s	ес
T t	те
U u	у
V v	ве
X x	ікс
Y y	ігрек

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Андерсон Дж. Дискретная математика и комбинаторика: Пер. с англ. – М.: Вильямс, 2003. – 960 с.
2. Бардачов Ю.М., Соколова Н.А., Ходаков В.Е. Дискретна математика: Підручник. – К.: Вища школа, 2002. – 287 с.
3. Бондаренко М.Ф., Білоус Н.В., Руткас А.Г. Комп'ютерна дискретна математика: Підручник. – Х.: Компанія СМІТ, 2004. – 480 с.
4. Борисенко О.А. Лекції з дискретної математики (множини і логіка): Навч. посіб. – Суми: Університетська книга, 2002. – 176 с.
5. Борисенко А.А. Введение в теорию биномиального счета: Монография. – Сумы: ИТД «Университетская книга», 2004. – 88 с.
6. Борисенко А.А. Биномиальный счет: Монография. – Сумы: ИТД «Университетская книга», 2004. – 170 с.
7. Бородин А.И. Из истории арифметики. – К.: Вища школа, 1986. – 95 с.
8. Бородин А.И. Число и мистика. – Донецк: Донбас, 1972. – 143 с.
9. Бородін І.О. Історія розвитку поняття про число і системи числення. – К.: Радянська школа, 1978. – 104 с.
10. Вавилов Е.Н., Егоров Б.М., Ланцев В.С., Тоценко В.Г. Синтез схем на пороговых элементах. – М.: Советское радио, 1970. – 368 с.
11. Вавилов Е.Н., Портной Г.П. Синтез схем электронных цифровых машин. – М.: Советское радио, 1963. – 440 с.
12. Вивальнюк Л.М., Григоренко В.К., Левіщенко С.С. Числові системи: Навч. посіб. – К.: Вища школа, 1988. – 272 с.
13. Вирченко Н.А. Математика в афоризмах, цитатах, высказываниях. – К.: Вища школа, 1983. – 280 с.
14. Гаврилов Г.П., Сапоженко А.А. Сборник задач по дискретной математике. – М.: Наука, 1977. – 367 с.
15. Глушков В.М. Введение в кибернетику. – К.: Изд-во АН УССР, 1964. – 324 с.
16. Горбатов В.А. Основы дискретной математики: Учеб. пособ. – М.: Высшая школа, 1986. – 311 с.
17. Ежов И.И., Скороход А.В., Ядренко М.И. Элементы комбинаторики: Пер. с укр. – М.: Наука, 1977. – 80 с.
18. Калбертсон Дж.Т. Математика и логика цифровых устройств: Пер. с англ. – М.: Просвещение, 1965. – 267 с.
19. Касаткин В.М. Новое о системах счисления. – К.: Вища школа, 1982. – 96 с.
20. Кац М., Улам С. Математика и логика. Ретроспектива и перспективы: Пер. с англ. – М.: Мир, 1971. – 250 с.
21. Кемени Дж., Снелл Дж., Томпсон Дж. Введение в конечную математику: Пер. с англ. – М.: Мир, 1965. – 486 с.
22. Кнут Д. Искусство программирования: Учебн. пособ.: Пер. с англ. – 3-е изд. – М.: Вильямс, 2000. – Т. 1. Основные алгоритмы. – 720 с.
23. Коршунов Ю.М. Математические основы кибернетики: Учеб. пособ. – М.: Энергия, 1972. – 376 с.
24. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергия, 1980. – 344 с.

25. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергоатомиздат, 1988. – 480 с.
26. Кухар В.М., Білий Б.М. Теоретичні основи початкового курсу математики: Навч. посіб. – К.: Вища школа, 1980. – 360 с.
27. Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов. – М.: Наука, 1975. – 240 с.
28. Лапа В.Г. Математические основы кибернетики. – К.: Вища школа, 1971. – 418 с.
29. Математична хрестоматія. Алгебра і початки аналізу / За ред. М.І. Кованцова. – К.: Радянська школа, 1977. – 215 с.
30. Нікольский Ю.В., Пасічник В.В., Щербина Ю.М. Дискретна математика: Підручник. – Л.: Магнолія Плюс, 2006.
31. Новиков Ф.А. Дискретная математика для программистов: Учебник. – СПб.: Питер, 2001. – 304 с.
32. Основи дискретної математики / Ю.В. Капітонова, С.Л. Кривий, О.А. Летичевський та ін. – К.: Наукова думка, 2002. – 580 с.
33. Пойа Д. Математика и правдоподобие рассуждения. / Пер. с англ. – 2-е изд. – М.: Наука, 1975. – 464 с.
34. Пойа Д. Математическое открытие: Пер. с англ. – 2-е изд. – М.: Наука, 1976. – 448 с.
35. Поспелов Д.А. Логические методы анализа и синтеза схем. – М.: Энергия, 1974. – 368 с.
36. Поспелов Д.А. Арифметические основы вычислительных машин дискретного действия: Учеб. пособ. – М.: Высш. школа, 1970. – 308 с.
37. Процай В.Ф. Новикова У.В. Комбінаторика і теорія ймовірностей у школі: Навч. посіб. - Х.: Каравела, 1997. – 192 с.
38. Роберт Р. Столл. Множества. Логика. Аксиоматические теории: Пер. с англ. – М.: Просвещение, 1968. – 231 с.
39. Сигорский В.П. Математический аппарат инженера. – К.: Техніка, 1975. – 766 с.
40. Соминский И.С. Метод математической индукции. – М.: Наука, 1974. – 63 с.
41. Спекторський І.Я. Дискретна математика: Навч. посіб. – 2-ге вид. – К.: Політехніка, 2004 – 220 с.
42. Успенский В.А. Треугольник Паскаля. – М.: Наука, 1979. – 48 с.
43. Халамайзер А.Я. Комбинаторика и бином Ньютона: Учеб пособ. – М.: Просвещение, 1980. – 32 с.
44. Феферман С. Числовые системы. Основания алгебры и анализа. / Пер. с англ. – М.: Наука, 1971. – 440 с.
45. Фомин С.В. Системы счисления. – М.: Наука, 1980. – 48 с.
46. Шоломов Л.А. Основы теории дискретных, логических и вычислительных устройств. – М.: Наука, 1980. – 400 с.

Навчальне видання

Борисенко Олексій Андрійович

Дискретна математика

Підручник

Художник А.К. Соколов

Директор видавництва Р.В. Кочубей
Головний редактор В.І. Кочубей
Технічний редактор І.Є. Бражник
Комп'ютерний набір О.А. Борисенко,
І.Є. Бражник, І.А. Заярна
Дизайн обкладинки і макет В.Б. Гайдабрус
Комп'ютерна верстка О.А. Борисенко

ТОВ «ВТД «Університетська книга»
40030, м. Суми, вул. Кірова, 27, 5-й пов.
Тел.: (0542) 27-51-43
E-mail: *publish@book.sumy.ua*

Відділ реалізації
Тел./факс: (0542) 21-26-12, 21-11-25
E-mail: info@book.sumy.ua

Підписано до друку 29.11.2007.
Формат 60x90 $^{1}/_{16}$. Папір офсетний. Гарнітура Times New Roman
Друк офсетний. Ум. друк. арк. 16,0. Обл.-вид. арк. 15,1.
Тираж 1000 прим. Замовлення № 5129

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготівників і розповсюджувачів
видавничої продукції ДК № 489 від 18.06.2001

Надруковано відповідно до якості наданих діапозитивів
у друкарні «Принт-Лідер»
Україна, 61070, м. Харків, вул. Рудика, 8

SUMMARY

Consistently elementary aspects of sets theory, logic, combinatorial calculus and number systems are stated in the consisting of five chapters textbook. The material of the five chapters is given in series in the form of lecturers with a great number of examples and exercises. There are tests at the end of the chapters.

In the first chapter of the textbook the basic definitions of set theory and operations on sets are under review. Algebra of sets is given.

In the second chapter the logic operations and functions, which have meaning for synthesizing digital circuits, are considered. The methods of logic functions presentation in disjunction and conjunction forms and the methods of their minimization are given.

In the third chapter the elements of combinatorial calculus and elementary combinatorial configurations, such as permutations, arrangements and combinations, are considered. It is placed high emphasis on Binomial theorem.

In the fourth chapter the questions, connected with theory and practice of positional count and number systems, are under review. It is placed high emphasis on untraditional systems of count – factorial and binomial number systems, including matrix ones.

In the fifth final chapter the additional data on the method of mathematical induction, etymology-terminological dictionary, Greek and Roman alphabets are given in the form of supplements.

The textbook is sent, first of all, higher education students, which learn electronics, but it can be applied for learning informatics, automation and other technical specialties, where discrete mathematics is one of the basic subjects.