Міністерство освіти і науки України
Сумський державний університет

М. М. Биков, В. Д. Черв’яков

ДИСКРЕТНИЙ АНАЛІЗ І ТЕОРІЯ АВТОМАТІВ

Навчальний посібник

Рекомендовано вченою радою Сумського державного університету

Суми
Сумський державний університет
2016

УДК 519.45(075.8)
ББК 22.145
Б81

Рецензенти:

В. В. Марасанов – доктор технічних наук, професор, завідувач кафедри технічної кібернетики Херсонського національного технічного університету;

А. С. Опанасюк – доктор фізико-математичних наук, професор, завідувач кафедри електроніки та комп’ютерної техніки Сумського державного університету;

К. Г. Гриценко – кандидат технічних наук, доцент, доцент кафедри економічної кібернетики Української академії банківської справи (м. Суми)

Рекомендовано до видання
вченою радою Сумського державного університету
як навчальний посібник
(протокол № 1 від 28 серпня 2015 року)

Биков М. М.

Б81 Дискретний аналіз і теорія автоматів : навчальний посібник / М. М. Биков, В. Д. Черв’яков. – Суми : Сумський державний університет, 2016. – 354 с.

ISBN 978-966-657-602-9

Викладено основні поняття й теоретичні положення дискретного аналізу, методи побудови математичних моделей цифрових та керуючих скінченних автоматів, функціонально орієнтованих на застосування в задачах розроблення технічних засобів автоматики та управління в комп’ютеризованих інформаційно-керуючих системах. Наведено приклади розв’язування типових задач, питання самоконтролю та зміст контрольної роботи, що сприяє кращому засвоєнню навчального матеріалу з однойменної навчальної дисципліни.

Для студентів, які навчаються за напрямом 050201 – системна інженерія.

УДК 519.45(075.8)
ББК 22.145

ISBN 978-966-657-602-9

ЗМІСТ

ПЕРЕДМОВА

Дискретний аналіз та його складова – теорія автоматів є важливою ланкою фундаментальної підготовки студентів вищих навчальних закладів, які навчаються за напрямом 050201 – Системна інженерія.

У сучасному розумінні дискретний аналіз являє собою "співдружність" взаємно проникнених математичних теорій, таких як теорія множин, математична логіка, теорія графів, теорія автоматів та ін., предметом дослідження яких є об'єкти, явища та процеси дискретного (скінченного) характеру. Під час вибору розділів навчального посібника автори керувалися:

– метою надання студентам знань із прикладної теорії цифрових і керуючих автоматів як теоретичної бази для успішного оволодіння матеріалами навчальних дисциплін професійної підготовки, пов'язаних із побудовою пристроїв обробки і передачі інформації, мікропроцесорних пристроїв та комп'ютеризованих систем управління технічними і технологічними об'єктами;

– передумовою, що студенти, які вивчають одноіменну навчальну дисципліну (або подібну за назвою та змістом), мають необхідні знання з вищої математики, методів обчислень, теорії інформації та кодування;

– доступністю навчального матеріалу, викладеного у цьому навчальному посібнику, для успішного засвоєння студентами як денної, так і заочної (зокрема дистанційної) форм навчання;

– особистим досвідом викладання однойменної та споріднених дисциплін у Вінницькому національному технічному та Сумському державному університетах.

У кожному з розділів відповідну математичну теорію викладено лише в її основах, за змістом і обсягом достатніх для засвоєння студентами. Виділенням у назві теорії автоматів (хоча вона є складовою дискретного аналізу) підкреслюється цільова спрямованість навчального посібника на створення теоретичної бази професійної підготовки фахівців із комп'ютеризованих систем управління та автоматизації.

Під час укладання навчального посібника використовувалися матеріали багатьох відомих на цей час літературних джерел з авторською систематизацією та переробленням відповідно до визначеної мети. Зокрема, використані деякі з відомих, на думку авторів вдалі практичні приклади та вправи.

Автори сподіваються, що навчальний посібник буде корисним як студентам вищих навчальних закладів, так і інженерно-технічним працівникам, навчальна або практична діяльність яких потребує знань і навичок із дискретного аналізу і теорії автоматів.

У навчальному посібнику вступ, розділи 1, 2, 3, предметний покажчик і додаток А складені кандидатом технічних наук, доцентом В. Д. Черв'яковим, розділи 4 і 5 – кандидатом технічних наук, професором М. М. Биковим, передмова і додаток Б складені авторами спільно. Загальна редакція виконана В. Д. Черв'яковим.

Автори вдячні за зауваження стосовно змісту навчального посібника рецензентам: Володимиру Васильовичу Марасанову – доктору технічних наук, професору, завідувачеві кафедри технічної кібернетики Херсонського національного технічного університету, Анатолію Сергійовичу Опанасюку – доктору фізико-математичних наук, професору, завідувачеві кафедри електроніки та комп'ютерної техніки Сумського державного університету, Костянтину Григоровичу Гриценку – кандидату технічних наук, доценту кафедри економічної кібернетики Української академії банківської справи (м.Суми).

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Скорочення

ЛФ	– логічна функція
КС	– комбінаційна схема
ЛЕ	– логічний елемент
ДНФ	– диз'юнктивна нормальна форма
КНФ	– кон'юнктивна нормальна форма
ДДНФ	– досконала диз'юнктивна нормальна форма
ДКНФ	– досконала кон'юнктивна нормальна форма
СДНФ	– скорочена диз'юнктивна нормальна форма
СКНФ	– скорочена кон'юнктивна нормальна форма
МДНФ	– мінімальна диз'юнктивна нормальна форма
МКНФ	– мінімальна кон'юнктивна нормальна форма
ЕОМ	– електронна обчислювальна машина
ГСА	– граф-схема алгоритму
ПК	– персональний комп'ютер
ЦА	– цифровий автомат

Позначення

$\varnothing$	– порожня множина
$a \in A$	– належність елемента a множині A
$A=\{x \mid P(x)\}$ або $A=\{x: P(x)\}$	– задання множини A описом $P(x)$ характеристичних властивостей елементів x
$A \subset B$	– строге включення множини A в множину B
$A \subseteq B$	– нестроге включення множини A в множину B
$A \times B$	– декартів добуток множин A і B
$\bar{x}$ або $\neg x$	– інверсія значення логічної змінної x

ВСТУП

Дискретний аналіз – це широка галузь знань із численних розділів математики, які надають споживачеві цих знань (інженеру, досліднику, студенту) методи побудови та аналізу математичних моделей процесів та явищ будь-якої природи, що мають дискретний характер, у сенсі часової зміни ситуацій чи станів досліджуваного об'єкта.

Бурхливий розвиток електроніки, обчислювальної техніки та інформаційних технологій, що спостерігається на сьогодні, привів до виникнення сучасної концепції керування різноманітними об'єктами за допомогою автоматизованих комп'ютерно-інтегрованих систем управління (інформаційно-управляючих систем). Вони будуються як ієрархічні системи, в яких апаратні ресурси та інформаційні потоки різних рівнів об'єднуються інформаційно-обчислювальними мережами. Більше того, в сучасних умовах набувають великого поширення інтелектуальні системи управління, здатні адаптуватися до змін зовнішнього середовища під час реалізації заданих функцій для досягнення мети управління. Апаратною базою таких систем є сучасні комп'ютери, мікропроцесори, програмовані логічні контролери, спеціалізовані автомати на програмованих інтегральних схемах, сенсори, регулятори та виконавчі пристрої дискретного принципу дії, мережеве обладнання та канали зв'язку.

Електронні обчислювальні машини (комп'ютери) призначені для обробки інформації, що подається у цифровій формі. Вони є найбільш поширеним видом цифрових автоматів. Для вивчення загальних принципів обробки цифрової інформації доцільно розглядати будь-який обчислювальний пристрій (комп'ютер, контролер) як абстрактний цифровий автомат дискретної дії (скінченний автомат), незалежно від його апаратної реалізації. Таким самим способом, тобто поданням математичною (структурно-логічною, функціональною) моделлю, доцільно вивчати принципи функціонування керуючих автоматів дискретної дії (автоматів управління), що виконують операції формування дискретних (типу “так – ні” або “вмикнути – вимкнути”) сигналів керування технічними об’єктами. Таким чином, під скінченним автоматом (цифровим, керуючим) в математиці розуміють не технічний пристрій певного призначення, а його математичну модель.

Створення нових комп’ютеризованих систем управління об’єктами організаційного, технічного й технологічного призначення, що є визначальною рисою об’єкта професійної діяльності фахівців із системної інженерії, потребує використання математичних моделей – абстракцій, які відображають істотні властивості досліджуваного об’єкта (наприклад, технічного пристрою) або процесу (технологічного, соціального, організаційно-економічного), що цікавлять дослідника (розробника, проектувальника). Методи дискретного аналізу є саме такими, що складають у сукупності математичний апарат для вирішення означених проблем, тобто побудови моделей обчислювальних і керуючих пристроїв дискретної дії (скінченних автоматів), потрібних для технічної реалізації комп’ютеризованих систем обробки інформації та управління. Теорія скінченних автоматів надає знання, необхідні для побудови пристроїв та алгоритмів обробки інформації, формування дискретних команд управління, грамотної експлуатації технічних засобів комп’ютеризованих систем обробки і передачі інформації та автоматизованого управління.

Актуальною проблемою сьогодення у сучасному житті суспільства є задовольнення зростаючої потреби у фахівцях із системної інженерії, які можуть глибоко й всебічно досліджувати керовані організаційно-економічні, технічні та технологічні системи, робити правильні висновки і прогнози, приймати оптимальні рішення щодо модернізації існуючих та створення нових комп’ютеризованих систем автоматики та управління. Вирішенню цієї проблеми сприяє внесення до навчального плану підготовки бакалавра із системної інженерії такої навчальної дисципліни, як “Дискретний аналіз і теорія автоматів”. Метою дисципліни є вивчення математичних основ теорії автоматів та їх структурних моделей. Дисципліна є складовою фундаментальної підготовки студентів, що надає базові знання та уміння для успішного вивчення матеріалів дисциплін професійної підготовки і виконання кваліфікаційної роботи бакалавра із системної інженерії.

Після засвоєння навчального матеріалу цієї дисципліни студент повинен знати на продуктивно-синтетичному рівні:

– основні елементи математичного базису теорії автоматів та методи їх використання під час побудови моделей структур, алгоритмів і програм обробки інформації в комп’ютеризованих системах управління;

– принципи побудови цифрових автоматів для здійснення обчислювальних операцій;

– принципи побудови скінченних автоматів для здійснення логічних функцій управління в об’єктах системотехніки.

На евристичному рівні даною дисципліною формуються такі уміння:

– визначати склад функцій управління, що повинні бути реалізовані автоматом у керованій системі, комплексів задач, які реалізуються в кожній підсистемі автомата за допомогою графічного й табличного уявлень;

– використовуючи визначення класів, поведінки об’єктів, структур даних та їх взаємозв’язки, виконувати функціональну декомпозицію автомата; будувати ефективні обчислювальні алгоритми для розрахункових задач;

– будувати ефективні алгоритми здійснення логічних операцій керування в об’єктах системотехніки.

Основними складовими цього навчального посібника, який є основною складовою дидактичного забезпечення навчального процесу з дисципліни “Дискретний аналіз і теорія автоматів” (або іншої за назвою та подібної за змістом), є теорія множин, математична логіка, теорія графів, теорія автоматів тощо. Усе це є теоретичною основою низки спеціальних інженерних дисциплін, що вивчаються за програмою підготовки фахівців із комп'ютеризованих систем управління та автоматизації. До професійно орієнтованих дисциплін відносять такі, як "Електроніка і мікросхемотехніка", "Інформаційні процеси в системах управління", "Комп'ютерні мережі", "Мікропроцесорні пристрої і системи", "Проектування пристроїв і систем автоматизації", "Інтелектуальні системи управління" та інши, що потребують від студента вихідних знань та вмінь із дискретного аналізу і теорії скінченних автоматів.

Теоретичний матеріал навчального посібника супроводжується практичними прикладами, переліками контрольних питань для самоперевірки знань, практичними завданнями (вправами) для самостійної роботи студента. Предметний покажчик дає читачеві можливість звернутися до попереднього тексту для нагадування означення того чи іншого математичного об'єкта. У додатках розміщені довідкові матеріали з методів обчислень і пропонований варіант контрольної роботи з дисципліни в цілому. Список рекомендованих літературних джерел, наведений у кінці посібника, повністю охоплює програмний матеріал навчальної дисципліни.

Розділ 1

ЕЛЕМЕНТИ ТЕОРІЇ МНОЖИН

Теорія множин є розділом математики, в якому вивчаються властивості множин незалежно від того, з яких елементів вони складаються. Теорія множин справедливо вважається основою сучасної математики. Поняття, аксіоми, теореми та закони теорії множин є такими, що використовуються як базові в інших математичних теоріях, зокрема у дискретному аналізі. Навчальний матеріал усіх розділів цього навчального посібника тією чи іншою мірою базується на основних положеннях теорії множин. Зміст даного розділу переслідує мету надання читачеві можливості одержання знань, необхідних для успішного вивчення матеріалів теоретичного і прикладного характеру, викладених у наступних розділах.

1.1 Поняття множини. Способи задання і структурні характеристики множин

Поняття “множина” та “елемент множини” є вихідними поняттями в математиці, їм не дається формалізованого означення. *Множиною* називають сукупність визначених помітних між собою об'єктів, об’єднаних за певною ознакою. Об’єкти, що утворюють множину, називають її *елементами*. Як приклади, можна назвати множини натуральних чисел, точок на площині тощо.

Характеристичною ознакою множини є те, що вона не містить однакових елементів. Належність елемента a множині A позначають символом $\in$: $a \in A$, а той факт, що елемент a не є елементом множини A, позначають $a \notin A$.

Залежно від кількості елементів множини поділяються на скінченні та нескінченні. Якщо множина містить скінченну кількість елементів, її називають *скінченною*. Інакше кажучи, існує натуральне число, яким визначається кількість елементів скінченної множини. Наприклад, множина вершин трикутника, множина студентів у групі тощо. Скінченну множину, яка не містить жодного елемента, називають *порожньою* і позначають символом $\varnothing$.

Нескінченною називається множина, яка містить нескінченну кількість елементів, тобто не існує натурального числа, що дорівнює кількості елементів даної множини. Нескінченні множини поділяються на *зліченні* та *незліченні*. Елементи зліченної множини можна поставити у відповідність ряду натуральних чисел. Прикладами зліченних є множини парних, непарних, раціональних чисел тощо. Елементи незліченної множини не можна підрахувати за допомогою натурального ряду. Наприклад, незліченними є множини дійсних та ірраціональних чисел, множина точок на прямій тощо.

Загальноприйнятими є позначення основних *числових множин*: N – натуральних чисел, Z – цілих чисел, R – дійсних чисел, Q – раціональних чисел, C – комплексних чисел.

Можливі такі *способи задання множин*:

– списком елементів у фігурних дужках. Наприклад, $A=\{x, y, z\}$, $B=\{2, 4, 6, 8, 10\}$;

– процедурою визначення повного набору елементів. Наприклад, $M=\{2^n, n\in N\}$ – множина чисел виду 2^n, $n\in N$;

– описом характеристичних властивостей елементів. Множина елементів x, які мають властивість $P(x)$, зазвичай позначається $\{x|\ P(x)\}$ або $\{x\colon P(x)\}$. Наприклад, множина натуральних чисел, що є степенями числа 2, може бути записана як $A=\{x|\ x=2^n, n\in N\}$.

Приклад 1.1. Нехай $A=\{1, 2, 3, 4, 5\}$. Позначимо через B множину, елементами якої є квадрати елементів множини A. Опис множини B можемо зробити двома способами:

$B=\{x^2|\ x\in A=\{1, 2, 3, 4, 5, 6\}\}$ або $B=\{1, 4, 9, 16, 25\}$.

Приклад 1.2. Зробимо опис множин точок площини, координати x і y яких задовольняють такі умови:

а) $A=\{(x,y)|\ y=x,\ x\in R\}$. Множина A нескінченна, зображається всіма точками (x,y) прямої $y = x$ на площині в декартових координатах x, y;

б) $A=\{(x,y)|\ y>x,\ x\in R\}\}$. Множина A нескінченна, зображається точками (x,y) площини над прямою $y = x$ в декартових координатах x, y (без точок на прямій);

в) $A=\{(x,y)|\ y<x,\ x\in R\}\}$. Множина A нескінченна, зображається точками (x,y) площини нижче прямої $y = x$ у декартових координатах x, y (без точок на прямій);

г) $A=\{(x,y)|\ y>x+1,\ x\in R\}\}$. Множина A нескінченна, зображається точками (x,y) площини над прямою $y=x+1$ у декартових координатах x, y (без точок на прямій);

д) $A=\{(x,y)|\ y\leq 4-x,\ x\in R\}\}$. Множина A нескінченна, зображається точками (x,y) площини нижче прямої $y=4-x$ у декартових координатах x, y і точками на цій прямій;

е) $A=\{(x,y)|\ |y|\leq 1-|x|,\ x\in R,\ |x|\leq 1\}$. Множина A нескінченна, зображається точками (x,y) площини квадрата, обмеженого прямими лініями $y=1-x$, $y=x-1$, $y=x+1$, $y=-x-1$ у декартових координатах x, y, з включенням точок сторін квадрата;

ж) $A=\{(x,y)|\ x^2+y^2\geq 1,\ x\in R\}$. Множина A нескінченна, зображається всіма точками (x,y) площини, за винятком внутрішньої частини круга радіусом 1, у декартових координатах x, y із включенням точок кола.

Приклад 1.3. Які з наведених визначень множин $A=\{1, 3, 5\}$, $B=\{4, 7, 7, 11\}$, $F=\{x|\ x\in A\}$, $D=\{A, B\}$ є коректними? Чи належить число 5 множині D?

Відповідь:

а) визначення множини A перерахуванням елементів коректне;

б) згідно з означенням множини її елементи повинні бути різні, тому в списку елементів не можна зазначати той самий елемент кілька разів. Коректне визначення множини B буде таким: $B=\{4, 7, 11\}$;

в) визначення множини F шляхом опису характеристичної властивості її елементів коректне;

г) визначення множини D коректне. Елементами множини D є множини A і B, тобто $D=\{\{1, 3, 5\}, \{4, 7, 11\}\}$. Однак $5 \notin D$, тому що цей елемент не перелічений у списку.

Множина A є *підмножиною* множини B, якщо кожний елемент x множини A належить і множині B; що позначається так: $A \subseteq B$ (або $B \supseteq A$). Таке відношення між множинами називають *нестрогим включенням*. Очевидно, будь-яка множина A містить порожню підмножину: $\varnothing \subseteq A$.

Якщо множина B містить, крім підмножини A, іншу непорожню підмножину, то має місце *строге включення* A в B, що позначається так: $A \subset B$ (або $B \supset A$). Будь-яка непорожня множина A має принаймні дві підмножини: $A \subseteq A$ і $\varnothing \subset A$.

Різницю понять належності елементів до множини і включення можна пояснити таким чином. Якщо $A=\{2, 4, 6, 8\}$, то $2 \in A$, $4 \in A$, але $\{2, 4\} \notin A$, у той час як $\{2, 4\} \subset A$. Якщо ж $A=\{\{2, 4\}, 6, 8\}$, то $2 \notin A$, $4 \notin A$, але $\{2, 4\} \in A$ і у той же час $\{2, 4\} \subset A$.

Приклад 1.4. Які з наведених тверджень правильні?

а) якщо $A \subset B$ і $B \subset D$, то $A \subset D$;

б) якщо $A \subseteq B$ і $B \subseteq A$, то $A=B$;

в) якщо $A \subseteq B$ і $B \subseteq D$, то $A \subseteq D$.

Відповідь: усі твердження правильні.

Приклад 1.5. Дана множина $D=\{7, 13, 25, 34, 101, 112\}$. Які з наведених нижче множин є підмножинами множини D:
а) $\{1, 7, 13\}$; б) $\{0, 1, 12\}$; в) $\{25, 112, 34\}$; г) $\{a, b, c, n\}$;
д) $\{7, 13, 25, 34, 101, 112\}$; е) $\varnothing$?

Відповідь: в) $\{25, 112, 34\}$; д) $\{7, 13, 25, 34, 101, 112\}$; е) $\varnothing$.

Множину $P(A)$ всіх підмножин, що складаються з елементів скінченної множини A, враховуючи включення $\varnothing \subset A$ і $A \subseteq A$, називають *булеаном* множини A. Наприклад, булеан множини $A=\{a, b, c, d\}$ має такий вигляд: $P(A)=\{\varnothing, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\}$. Булеан будь-якої n-елементної множини містить 2^n елементів (підмножин).

Потужність множини визначається як кількість її елементів. Наприклад, потужність множини $A=\{a, b, c, d\}$ дорівнює 4. Наведений вище булеан $P(A)$ містить $2^4=16$ елементів (підмножин). Отже, його потужність $|P(A)|=16$.

Множини A і B вважаються *рівними*, якщо будь-який елемент множини A є елементом множини B і навпаки. Рівність множин A і B позначають $A=B$, а їх нерівність – $A \neq B$.

З наведеного означення рівності множин випливає, що множина повністю визначається її елементами, і при заданні скінченної невпорядкованої множини *порядок*, в якому перелічені її елементи, не має значення.

Для будь-яких множин A, B, D виконуються умови: $A=A$, $B=B$, $D=D$; якщо $A=B$, то $B=A$, (з використанням символів випливу $A=B \Rightarrow B=A$ і $A=B \Leftrightarrow B=A$); якщо $A=B$ і $B=D$, то $A=D$.

Із рівності множин випливає рівність їх потужностей: якщо $A=B$, то $|A|=|B|$ ($A=B \Rightarrow |A|=|B|$), однак твердження $|A|=|B| \Rightarrow A=B$ у загальному випадку неправильне.

Скінченні множини з однаковою кількістю елементів, тобто з однаковою потужністю, називають *рівночисельними*, або *рівнопотужними*.

Оскільки всі зліченні множини мають однакову потужність, що дорівнює потужності $\aleph_0$ (читається "алеф-нуль") множини N натуральних чисел, то всі вони є рівночисельними. Потужність булеану множини натуральних чисел називають *потужністю континуума*. Потужність континуума вважають константою (з позначенням $\aleph$) та використовують для порівняння за потужністю нескінченних зліченних множин.

Приклад 1.6. Множина $\{\varnothing\}$ непорожня, тому що $\{\varnothing\}\neq\varnothing$. Вона є одноелементною, єдиним її елементом є порожня множина $\varnothing$.

Приклад 1.7. Визначити, які з наведених тверджень справедливі: а) $|\{\varnothing\}|=1$; б) $\{\{\varnothing\}\}\in\{\{\{\varnothing\}\}\}$; в) $|\{\{\varnothing\}\}|=2$; г) $x\in\{x\}$, д) $\{x\}\subseteq\{x\}$, е) $\{x\}\in\{x\}$, ж) $\{x\}\in\{\{x\}\}$.

Відповідь.

Справедливі: а) $|\{\varnothing\}|=1$ – потужність одноелементної множини $\{\varnothing\}$; б) $\{\{\varnothing\}\}\in\{\{\{\varnothing\}\}\}$, $\{\{\varnothing\}\}$ – елемент одноелементної множини $\{\{\{\varnothing\}\}\}$; г) $x\in\{x\}$; д) $\{x\}\subseteq\{x\}$;
ж) $\{x\}\in\{\{x\}\}$, множина $\{x\}$ є елементом множини $\{\{x\}\}$.

Несправедливі: в) $|\{\{\varnothing\}\}|=2$, $|\{\{\varnothing\}\}|$ – потужність одноелементної множини $\{\{\varnothing\}\}$; е) $\{x\}\in\{x\}$, множина $\{x\}$не містить елемента $\{x\}$.

Приклад 1.8. Визначте, скільки елементів містять такі множини: $\{x\}$, $\{\{x\}\}$; $\{x, \{x\}\}$; $\{\{x\}, x, \{\{x, \{x\}\}\}\}$.

Відповідь: множини $\{x\}$ та $\{\{x\}\}$ містять по 1 елементу, множина $\{x, \{x\}\}$ – 2 елементи, множина $\{\{x\}, x, \{\{x, \{x\}\}\}\}$ – 3 елементи.

Приклад 1.9. Визначити, які з наведених множин дорівнюють одна одній: а) $A=\{x|\ x=2y,\ y\in N\}$; б) $C=\{1, 2, 3\}$; в) $D=\{0, 2, –2, 3, –3, 4, –4, \dots\}$; г) $E=\{2x|\ x\in Z\}$.

Відповідь: $D=E$.

Приклад 1.10. Побудувати булеан для множин A, B, C, D, якщо: а) $A=\{\{\varnothing\}\}$; б) $B=\{1, 2, 3, 4\}$; в) $C=\{$день, ніч$\}$; г) $D=\{1, \{2, 3\}, 4\}$.

Відповідь.

$P(A)=\{\{\varnothing\}\}=A$.

$P(B)=\{\{\varnothing\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$, $\{3, 4\}$, $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{2, 3, 4\}$, $\{1, 2, 3, 4\}\}$.

$P(C)=\{\{\varnothing\}$, {день}, {ніч}, {день, ніч}}.

$P(D)=\{\{\varnothing\}$, $\{1\}$, $\{2, 3\}$, $\{4\}$, $\{1,\{2, 3\}\}$, $\{1, 4\}$, $\{\{2, 3\}, 4\}$, $\{1, \{2, 3\}, 4\}\}$.

Поняття множини повинне бути пов'язане з певними співвідношеннями між її елементами, тобто множина може бути певним чином структурованою. Одним із видів структурованості множин є упорядкованість. Множину M називають *упорядкованою*, коли в ній установлено *відношення порядку* $\prec$, що має такі властивості:

– $\forall a,b\in M$ або $a\prec b$ (a передує b), або $b\prec a$ (b передує a);

– $\forall a,b\in M$, якщо $a\prec b$ і $b\prec c$, то $a\prec c$.

Вибір відношення порядку називають *упорядкуванням* множини. Для впорядкування скінченної множини досить привласнити її елементам номери 1, ..., n, або просто записати їх у певному порядку. Простий приклад використання упорядкованих множин дає лінійна алгебра: елементи рядків і стовпців $n \times n$ – матриці у певній послідовності створюють n – вимірні впорядковані множини (кортежі, вектори). Інші приклади упорядкованих множин:

$A=\{x_{\prec} \mid x\in\{45, 1, 76, 4$; $\prec$ – зростання значення числа$\}=\{1, 4, 45, 76\}$;

$A=\{x_{\prec} \mid x\in$ {майстер, робітник, начальник цеху; $\prec$ – службова ієрархія} = {робітник, майстер, начальник цеху};

$A=\{x_{\prec} \mid x\in$ {генерал, майор, полковник; $\prec$ – ієрархія військових звань} ={майор, полковник, генерал}.

Будь-яка впорядкована скінченна множина має екстремальні елементи: мажоранту, міноранту, максимальний та мінімальний елементи, верхню та нижню грані. *Мажоранта* (*мі-*

норанта) деякої підмножини X упорядкованої множини M – це елемент $y \in M$ такий, що для будь-якого $x \in X$ справедливе $y \succeq x$ ($y \preceq x$). Якщо мажоранта (мінoранта) $y \in M$ належить підмножині X упорядкованої множини M, тоді вона називається *максимальним* (*мінімальним*) *елементом* множини $X \subseteq M$. Якщо множина мажорант (мінорант) множини $X \subseteq M$ має мінімальні (максимальні) елементи, то множину цих елементів називають *верхньою* (*нижньою*) *гранню* множини $X \subseteq M$, із умовним позначенням $supX$ ($infX$). Очевидно, якщо множина M є *строго впорядкованою* (будь-які y_i, $y_j \in M$ пов'язані відношенням порядку $y_i \prec y_j$ або $y_j \prec y_i$), то будь-яка множина $X \subseteq M$ має один мінімальний та один максимальний елементи, що одночасно є відповідно її верхньою та нижньою гранями.

Декартовим (прямим) добутком множин A і B називається третя множина $A \times B = \{(a,b) | a \in A, b \in B\}$, елементами якої є всі можливі упорядковані пари (a,b) елементів цих множин. У загальному випадку декартів добуток $A \times A \times \ldots \times A$ n однакових підмножників має назву "*множина-степінь*" і позначається A^n. Елементами множини A^n є впорядковані n-вимірні кортежі виду $(a_1, a_2, \ldots, a_n)$, де a_i, $i = 1, \ldots, n$ – елементи множини A, складені з використанням правила формування елементів добутку двох множин: $(\ldots((A \times A) \times A) \times \ldots \times A)$. Із цього правила, зокрема, випливає, що потужність множини-степеня $|A^n| = |A|^n$.

Елементи a_1, $a_2, \ldots, a_n$ n-вимірного вектора (кортежа) називають його *координатами* (в n-вимірному просторі). Прикладом використання добутку множин є множина $R \times R = R^2$, де R – множина дійсних чисел. Множина $\{(x,y) | x \in R, y \in R\}$ уявляє собою множину всіх точок площини (або двовимірних векторів) у декартових координатах x, y.

Приклад 1.11. Визначити добуток множин $X = \{x_1, x_2\}$, $Y = \{y_1, y_2, y_3\}$.

Відповідь: $X \times Y = Z = \{(x_1,y_1),\ (x_1,y_2),\ (x_1,y_3),\ (x_2,y_1),\ (x_2,y_2),\ (x_2,y_3)\}$.

Приклад 1.12. Який геометричний об'єкт описує множина $A=\{(x,y) \mid y^2+x^2=1,\ x\in R,\ y\in R\}$?

Відповідь: множина A уявляє собою множину всіх точок $(x,y)\in R^2$ кола радіусом 1 з центром у початку координат.

Контрольні питання

Поясніть такі поняття: множина, елемент множини, підмножина, скінченна і нескінченна множини, зліченна та незліченна множини, порожня множина, невпорядкована та впорядкована множини, булеан множини, декартів добуток множин.

Запишіть за допомогою позначень належності твердження, що елемент a належить, а елемент b не належить множині A.

Чи може множина містити два та більше однакових елементів?

Чи є множини натуральних, цілих, дійсних та комплексних чисел скінченними?

Наведіть приклади скінченних і нескінченних множин.

Якими способами можна задати множину?

Поясніть поняття строгого і нестрогого включень однієї множини в іншу.

Чи може порожня множина бути підмножиною непорожньої множини?

Наведіть приклад множини, елементами яких є множини.

Наведіть приклади зліченних і незліченних числових множин.

Який об'єкт теорії множин являє собою множину всіх підмножин деякої множини?

Які об'єкти є елементами булеану множини? Чи є порожня множина і сама множина (в цілому) елементами її булеану?

На прикладі 5-елементної множини поясніть "технологію" побудови її булеану.

Скільки елементів містить булеан n-елементної множини?

Яким чином визначається потужність множини?

За яких умов дві множини вважаються рівними?

Чи є рівність потужностей множин наслідком їх рівності?

Яким чином здійснюється процедура впорядкування множини? Наведіть приклад упорядкованої множини із лінійної алгебри.

Поясніть поняття екстремальних елементів упорядкованої скінченної множини: мажоранти, міноранти, максимального та мінімального елементів, верхньої та нижньої граней.

Поясніть поняття декартового добутку множин.

Нехай A і B різні множини. Поясніть, чому $A\times B\neq B\times A$?

Який об'єкт теорії множин має назву "множина-степінь"?

Які об'єкти є елементами множини-степеня A^n?

В якому співвідношенні знаходяться потужності множини-степеня A^n і самої множини A?

1.2 Операції над множинами

У багатьох розділах математики і, зокрема, в теорії ймовірностей розглядаються лише множини, що є підмножинами однієї й тієї самої множини, яку називають *універсальною множиною (універсумом)*. Для універсуму будемо використовувати позначення U. Над множинами, які є елементами деякого універсуму U, можна здійснювати певні операції, утворюючи з кількох множин іншу. Основними операціями над множинами є *бінарні* (коли з двох множин утворюється третя).

Означення 1.1. *Об'єднанням $A\cup B$ множин A* і *B* називається множина всіх елементів, що належать принаймні одній із множин A або B ("або" невиключне):

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

Означення 1.2. *Перетином $A\cap B$ множин A* і *B* називається множина всіх елементів, які належать як A, так і B:

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

Означення 1.3. *Різницею* $A \backslash B$ *множин* A і B називається множина всіх елементів, які належать A і не належать B:

$$A \backslash B = \{x |\ x \in A \text{ і } x \notin B\}.$$

Множину $A \backslash B$ називають також *доповненням* B до A.

Означення 1.4. Нехай $A \subseteq U$, де U – універсум. Доповненням A до універсуму називається множина

$$U \backslash A = \{x |\ x \in U \text{ і } x \notin A\} = \overline{A}$$

(верхнє підкреслення означає заперечення). Операція доповнення до універсуму є *унарною* (з однієї множини утворюється інша).

Означення 1.5. *Симетричною різницею* $A - B$ множин А і B називається множина всіх елементів, які належать A або B і не належать множині $A \cap B$:

$$A - B = \{x |\ (x \in A \text{ або } x \in B), \text{ але } x \notin A \cap B\}.$$

Приклад 1.13. Нехай $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, A – множина парних чисел із U, B – множина тих самих чисел із U, що кратні 3. Тоді $A = \{2, 4, 6, 8, 10\}$, $B = \{3, 6, 9\}$, $A \cup B = \{2, 3, 4, 6, 8, 9, 10\}$, $A \cap B = \{6\}$, $A \backslash B = \{2, 4, 8, 10\}$.

Схематичне зображення універсальної множини та її підмножин можна робити за допомогою *діаграм Венна* (Ейлера – Венна). На цих діаграмах універсальна множина зображається множиною точок прямокутника, а її підмножини – у вигляді кругів або іншими областями у цьому прямокутнику. Застосування діаграм Венна над множинами ілюструє рис.1.1, де заштрихованими ділянками прямокутника U відображені результати виконання операцій.

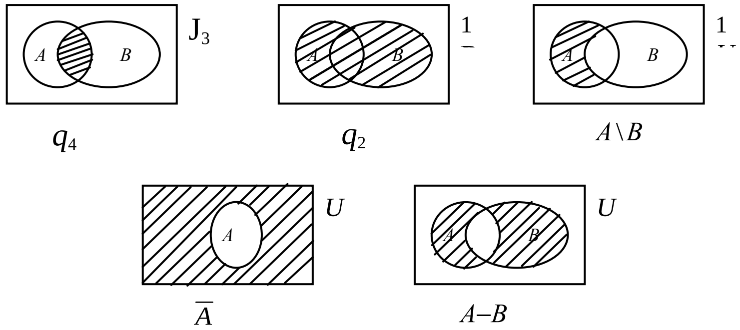


Рисунок 1.1 – Діаграми Венна операцій над множинами

За допомогою діаграм Венна неважко переконатися також у правильності таких тверджень:

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

$A-B=B-A$; $A-B=(A\cup B)\setminus(A\cap B)$; $(A-B)-C=A-(B-C)$;

$A\cap(B-C)=(A\cap B)-(A\cap C)$; $A-U=\overline{A}$; $A-\varnothing=A$.

Поєднання множини із заданими на ній операціями називають *алгеброю*. Множину в алгебрі називають *носієм*, а сукупність операцій – *сигнатурою*.

Алгебру Кантора < *P(U)*, $\cup$, $\cap$, $\overline{\ }$ >, носієм якої є булеан *P(U)* універсальної множини *U*, сигнатурою – операції об’єднання $\cup$, перетину $\cap$ та доповнення $\overline{\ }$, називають *алгеброю множин*. Множини *U* і $\varnothing$ у цій алгебрі є відповідно одиничним та нульовим елементами. За допомогою діаграм Венна можна переконатися, що операції над множинами володіють властивостями, які називають *законами алгебри множин* (табл. 1.1).

В алгебрі множин існує *принцип двоїстості*: для будь-якого твердження з використанням порожньої та універсальної множин і операцій $\cup$ та $\cap$ можна скласти двоїсте до нього твердження шляхом заміни $\varnothing$ на *U*, *U* на $\varnothing$, $\cup$ на $\cap$, $\cap$ на $\cup$, яке також буде правильним. Наприклад, двоїстими є твердження $A\cap(B\cup C) = (A\cap B)\cup(A\cap C)$ і $A\cup(B\cap C) =(A\cup B)\cap(A\cup C)$. Серед наведених у таблиці 1.1 пар тотожностей можна знайти інші, що пов'язані відношенням двоїстості.

Таблиця 1.1 – Закони алгебри множин

Назви законів	Тотожності
Закони ідемпотентності	$A\cup A =A$; $A\cap A =A$; $A\cup \overline{A} =U$; $A\cap \overline{A} = \varnothing$; $A\cup U =U$; $A\cap\varnothing = \varnothing$; $A\cap U =A$; $A\cup\varnothing =A$
Закони комутативності	$A\cup B=B\cup A$; $A\cap B =B\cap A$
Закони асоціативності	$A\cup(B\cup C) = (A\cup B)\cup C =A\cup B\cup C$; $A\cap(B\cap C) = (A\cap B)\cap C =A\cap B\cap C$
Закони дистрибутивності	$A\cap(B\cup C) = (A\cap B)\cup(A\cap C)$; $A\cup(B\cap C) = (A\cup B)\cap(A\cup C)$
Закони поглинання	$A\cup(A\cap B) =A$; $A\cap(A\cup B) =A$
Закони склеювання	$(A\cap B)\cup(\overline{A}\cap B) =B$; $(A\cup B)\cap(\overline{A}\cup B) =B$
Закони де Моргана	$\overline{A\cup B} = \overline{A}\cap \overline{B}$; $\overline{A\cap B} = \overline{A}\cup \overline{B}$
Закони Порецького	$A\cup(\overline{A}\cap B) =A\cup B$; $A\cap(\overline{A}\cup B) =A\cap B$
Закон подвійного заперечення	$\overline{\overline{A}} =A$

Узагальнення операцій над множинами:

об'єднання n множин $A_1\cup A_2\cup \ldots \cup A_n = \bigcup\limits_{i=1}^{n} A_i$;

перетин множин $A_1\cap A_2\cap \ldots \cap A_n = \bigcap\limits_{i=1}^{n} A_i$;

закони де Моргана $\overline{\bigcup\limits_{i=1}^{n} A_i} = \bigcap\limits_{i=1}^{n} \overline{A_i}$, $\overline{\bigcap\limits_{i=1}^{n} A_i} = \bigcup\limits_{i=1}^{n} \overline{A_i}$.

Базуючись на поняттях вищезазначених операцій над множинами, розглянемо поняття *розбиття множини*.

Означення 1.6. Система множин $\{X_1, X_2, \ldots, X_n\}$ називається розбиттям множини M, якщо вона задовольняє такі умови:

– будь-яка множина X_i, $i=\overline{1,n}$ є підмножиною M, тобто $X_i \subset M$;

– будь-які дві множини X_i та X_j, $i \neq j$, не перетинаються, тобто $X_i \cap X_j = \varnothing$;

– об'єднання всіх множин, що входять у розбиття, дає множину $M = \bigcup_{i=1}^{n} X_i$;

– для будь-якого i, $i=\overline{1,n}$, множина X_i є непорожньою, тобто $X_i \neq \varnothing$.

Отже, розбиття множини M – це така система непорожніх та різних її підмножин, коли кожен елемент множини M одночасно є елементом тільки однієї з цих підмножин. Наприклад, $\{\{1,2\},\{3\},\{4,5\}\}$ є розбиттям множини $\{1,2,3,4,5\}$.

Приклад 1.14. Задані: універсальна множина $U=\{0, 1, 2, 3, 4, 5, \ldots\}$, її підмножини $A=\{3x \mid x \in U\}$, $B=\{5x \mid x \in U\}$. Знайти $A \cap B$.

Відповідь:

$A=\{0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, \ldots\}=\{3x \mid x \in U\}$;

$B=\{0, 5, 10, 15, 20, 30, \ldots\}=\{5x \mid x \in U\}$;

$A \cap B=\{0, 15, 30, 45, \ldots\}=\{15x \mid x \in U\}$.

Приклад 1.15. Нехай універсальна множина U складається з чисел 1, 2, 3,...,9 і 33 букв українського алфавіту *а, б, в,...,я*. Якщо $A=\{1, 3, 5, а, в, д\}$ і $B=\{1, 2, 3, 4, 5, а, б, г, д, е\}$, знайти: а) $\overline{B}$, б) $A \cap B$, в) $A \cup B$, г) $A \cap \overline{B}$.

Відповідь: а) $\overline{B}=\{6, 7, 8, 9, в, ж, з, \ldots, я\}$; б) $A \cap B=\{1, 3, 5, а, д\}$; в) $A \cup B=\{1, 2, 3, 4, 5, а, б, в, г, д, е\}$; г) $A \cap \overline{B}=\{в\}$.

Приклад 1.16. Нехай U є множиною всіх точок (x,y) координатної площини, $A=\{(x,y)|\ y=|x|\}$, $B=\{(x,y)|\ y>|x|\}$, $L=\{(x,y)|\ x+y=2\}$, $M=\{(x,y)|\ x+y<2\}$. Вказати на площині такі множини: а) A, б) B, в) $A\cup B$, г) $\overline{B}$, д) L, е) $A\cap L$, ж) $B\cap M$, з) $\overline{M}$, к) $\overline{L}$.

Відповідь.

а) $A=\{(x,y)|\ y=|x|\}$ – множина точок (x,y) на графіку функції $y=\begin{cases} x\ при\ x\geq 0, \\ -x\ при\ x<0. \end{cases}$

б) $B=\{(x,\ y)|\ y>|x|\}$ – множина точок площини вище графіка функції $y=\begin{cases} x\ при\ x\geq 0, \\ -x\ при\ x<0 \end{cases}$ без точок на самому графіку.

в) $A\cup B$ – множина точок площини вище графіка функції $y=\begin{cases} x\ при\ x\geq 0, \\ -x\ при\ x<0 \end{cases}$ і точок на самому графіку.

г) $\overline{B}$ – множина точок площини нижче графіка функції $y=\begin{cases} x\ при\ x\geq 0, \\ -x\ при\ x<0 \end{cases}$ і точок на самому графіку.

д) $L=\{(x,\ y)|\ x+y=2\}$ – множина точок графіка функції $y=2-x$.

е) $A\cap L$ – точка (1, 1) перетину графіків функцій $y=2-x$ та $y=\begin{cases} x\ при\ x\geq 0, \\ -x\ при\ x<0 \end{cases}$.

ж) $B\cap M$ – множина точок площини, обмеженої графіками функцій

$$y=\begin{cases} x\ при\ x\geq 0, \\ -x\ при\ x<0 \end{cases} \text{ та } y=2-x \text{ при } x<1,$$

без точок на цих графіках.

з) $\overline{M}$ – множина точок площини вище графіка функції $y=2-x$ і точок на самому графіку.

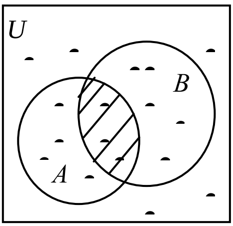
к) $\overline{L}$ – множина всіх точок (x,y) координатної площини, за винятком точок графіка функції $y = 2–x$.

Приклад 1.17. Нехай A і B – підмножини скінченної універсальної множини U. Кількість елементів деяких підмножин універсуму U вказана в стовпцях 2 – 5 таблиці. Поставимо завдання: визначити кількість елементів у підмножинах, вказаних у стовпцях 6 – 10 другого рядка таблиці, заповнити отриманими даними стовпці 6 – 10 третього рядка.

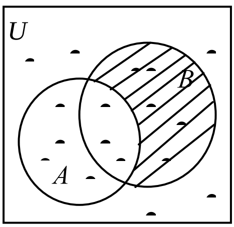
1	2	3	4	5	6	7	8	9	10
Множина	U	A	B	$A\cap B$	$\overline{A}\cap B$	$A\cap \overline{B}$	$A\cup B$	$\overline{A}$	$\overline{A}\cup B$
Кількість елементів	17	7	8	3	**5**	**4**	**12**	**10**	**13**

Розв’язування

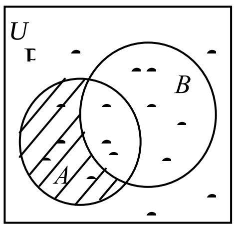
Побудуємо діаграми Венна для підмножин у стовпцях 6 – 10 (елементи множин на діаграмах позначені точками). Підрахувавши числа елементів в кожній із цих множин, отримаємо (див. табл.): $|\overline{A}\cap B| = 5$, $|A\cap \overline{B}|=4$, $|A\cup B|=12$, $|\overline{A}|=13$, $|\overline{A}\cup B|=13$.



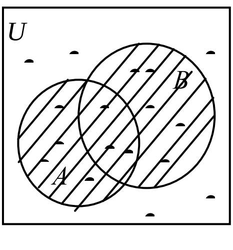
$A\cap B$

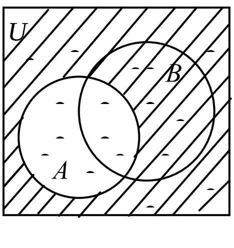


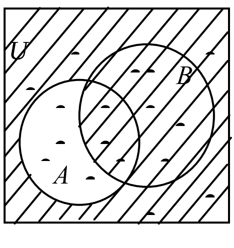
$\overline{A}\cap B=B\setminus A$



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







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

Рисунок 1.2 – Діаграми Венна до прикладу 1.17

Приклад 1.18. Розв’яжемо систему рівнянь

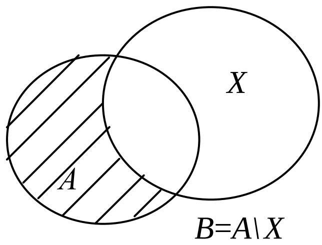


Рисунок 1.3 – До прикладу 1.18

$$\begin{cases} A \setminus X = B, \\ A \cup X = C, \end{cases}$$

де *A*, *B*, C — задані множини та $B \subset A \subset C$.

Відповідь: $X = C \setminus B = (C \setminus A) \cup (C \cap A)$, як показано на діаграмі Венна (рис. 1.3).

При створенні або спрощенні складних алгебраїчних виразів слід керуватись пріоритетом операцій. В алгебрі множин існує наступний пріоритет операцій: $\overline{A}$, $A \cap B$, $A \cup B$, $A \setminus B$, $A - B$. Поясненням приоритету операцій є наступний приклад, в якому парами дужок визначено послідовність виконання операцій для спрощення складного виразу:

$$Q = (E \setminus D \cup \overline{A} \cap B \cup C) \setminus F = (E \setminus (D \cup (((\overline{A}) \cap B) \cup C))) \setminus F.$$

Контрольні питання

Яка множина називається універсальною?

Запишіть відношення включення між універсальною множиною, її довільною підмножиною та порожньою множиною.

Наведіть перелік основних операцій над множинами. Проілюструйте їх за допомогою діаграм Венна.

Які властивості операцій над множинами відображають закони комутативності, асоціативності, дистрибутивності, де Моргана, поглинання, ідемпотентності, подвійного доповнення. Виразіть ці властивості відповідними тотожностями та проілюструйте за допомогою діаграм Венна.

Поясніть поняття розбиття множини. Проілюструйте розбиття множини за допомогою діаграми Венна.

У чому полягає принцип двоїстості?

Запишіть парами тотожності алгебри множин, які пов’язані відношенням двоїстості.

Запишіть формулу операції об’єднання n множин ($n>2$).

Запишіть формулу операції перетину n множин ($n>2$).

Запишіть формули, за якими використовуються закони де Моргана для n множин ($n>2$).

Поясніть пріоритет операцій алгебри множин при побудові або спрощенні складних виразів.

1.3 Відношення

Поняття відношення застосовуються у математиці для характеристики зв’язків між об’єктами. З деякими видами відношень ми вже зустрічалися, це відношення належності, включення, рівності, порядку, двоїстості.

Унарні (одномісні) відношення віддзеркалюють певну ознаку (властивість) елементів множини: всі елементи x множини X, що мають ознаку $P(x)$, утворюють підмножину $A \subseteq X$, яку називають *відношенням на множині* X.

Відношення між парами об'єктів називаються *бінарними* (двомісними). Розглянемо дві множини X та Y, елементи яких складають упорядковані пари (x,y). Якщо спосіб складання упорядкованих пар визначений, то між множинами X і Y існує бінарне відношення. Елемент x називають першою, а елемент y – другою координатами впорядкованої пари, або двовимірного кортежа (x,y).

Відношення може бути *n-арним*, яким пов’язуються елементи n множин (приклад – множина-степінь A^n). Як бінарне відношення породжує двовимірні кортежі, так і n-арне породжує n-вимірні кортежі (вектори). Наприклад, декартів добуток $R^3=R\times R\times R$ визначає 3-арне відношення, що породжує множину тривимірних кортежей виду (x,y,z), де x, y, z – дійсні числа, в тривимірному лінійному (декартовому) просторі.

Основну увагу ми будемо приділяти бінарним відношенням як таким, що є предметом дискретного аналізу.

Означення 1.7. *Відношенням* Q множин A і B називається довільна підмножина $Q \subseteq A \times B$ декартового добутку цих множин.

В іншій термінології Q є відношенням на добутку $A \times B$.

Якщо впорядкована пара $(a,b) \in Q$, то говорять, що a і b знаходяться у *відношенні* Q, з позначним записом bQa.

Якщо $Q \subseteq A \times A$, то відношення Q називають *бінарним відношенням на множині* A або на добутку A^2. В подальшому замість терміна “бінарне відношення” будемо застосовувати термін “відношення”.

Приклад 1.19. Нехай $A=\{1,2,3\}$, $B=\{a,b\}$. Тоді декартів добуток $A \times B=\{(1,a), (1,b), (2,a), (2,b), (3,a), (3,b)\}$. Множина $Q \subset A \times B$, наприклад, $Q=\{(1,a), (2,b), (3,a)\}$ є відношенням на $A \times B$.

Приклад 1.20. Якщо R – множина дійсних чисел, то множина $Q=\{(x,y) \in R^2\colon x^2+y^2=16\}$ є бінарним вілношенням на R^2.

Означення 1.8. *Областю відправлення* відношення $Q \subseteq A \times B$ називається множина A, елементи якої беруть участь у відношенні Q як перші елементи упорядкованих пар $(a,b) \in Q$.

Означення 1.9. *Областю прибуття* відношення $Q \subseteq A \times B$ називається множина B, елементи якої беруть участь у відношенні Q як другі елементи впорядкованих пар $(a,b) \in Q$.

Означення 1.10. Відношенням, *оберненим* до відношення $Q \subseteq A \times B$, називається таке відношення $Q^{-1} \subseteq B \times A$, що $Q^{-1}=\{(b,a)\colon (a,b) \in Q\}$.

Іншими словами, $(b,a) \in Q^{-1}$ лише тоді, коли $(a,b) \in Q$ або, що рівносильно, $aQ^{-1}b$ лише тоді, коли bQa.

Приклад 1.21. Нехай $Q=\{(a_1,b_2), (a_1,b_3), (a_2,b_4), (a_3,b_5)\}$. Тоді $Q^{-1}=\{(b_2,a_1), (b_3,a_1), (b_4,a_2), (b_5,a_3)\}$.

Задання відношення може бути здійснене наступними способами.

1. Перерахуванням упорядкованих пар, як у прикладах 1.19 – 1.21.

2. *Матрицею відношення*, кожен елемент a_{ij} якої дорівнює одиниці, якщо y_jQx_i, і нулю у протилежному випадку. Наприклад, відношення $Q=\{(a_1,b_2), (a_1,b_3), (a_2,b_4), (a_3,b_5), (a_4,b_3)\}$ на добутку $A\times B$ подається такою матрицею:

(A, B, Q)	b_1	b_2	b_3	b_4	b_5
a_1	0	1	1	0	0
a_2	0	0	0	1	0
a_3	0	0	0	0	1
a_4	0	0	1	0	0

3. *Стрілковою діаграмою* (графом). Елементи x, y множин X, Y зображаються точками на площині. Якщо y_jQx_i, точки x_i і y_j з’єднуються лініями (не обов’язково прямими) зі стрілками, спрямованими від x_i до y_j.

Зробивши таку операцію для всіх пар, що належать відношенню, одержимо його стрілкову діаграму.

Як приклад, відношення $Q=\{(a,1), (a,2), (b,1), (c,2), (c,4), (c,9), (e,16)\}$ на добутку $X\times Y$ у вигляді стрілкової діаграми подано на рис. 1.4.

Якщо ми маємо два заданих відношення, то з них можна побудувати третє, на підставі означення 1.11.

Означення 1.11. Нехай $Q\subseteq A\times B$ – відношення на $A\times B$, а $T\subseteq B\times D$ – відношення на $B\times D$. *Композицією* відношень Q і T називається відношення $M\subseteq A\times D$, визначене таким чином: $M=\{(a, d)$: існує такий елемент b із B, що $(a,b)\in Q$ і $(b, d)\in T\}$. Операцію композиції позначають так: $M=T\circ Q$.

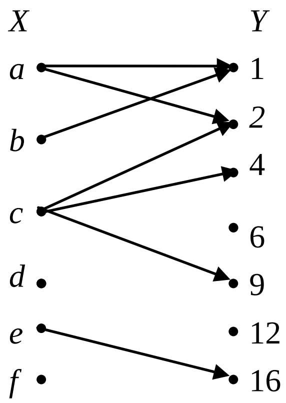


Рисунок 1.4 – Подання відношення напрямленим графом

Приклад 1.22. Нехай $A=\{1, 2, 3\}$, $B=\{a, b\}$, $D=\{я, ти, він, вона\}$, відношення $Q\subseteq A\times B$ і $T\subseteq B\times D$ задані у вигляді $Q=\{(1,a), (1,b), (3,b)\}$, $T=\{(a,я), (a,ти), (b,він), (b,вона)\}$. Тоді

$$T\circ Q=\{(1,я), (1,ти), (1,він), (1,вона), (3,він), (3,вона)\}.$$

Операція композиції відношень у загальному випадку некомутативна: $T\circ Q\neq Q\circ T$, однак володіє властивістю асоціативності, як це випливає з наступної теореми.

Теорема 1.1. Композиція відношень асоціативна, тобто якщо A, B, D, E – множини, $Q\subseteq A\times B$, $T\subseteq B\times D$ і $M\subseteq D\times E$ – відповідні відношення, то $M\circ(T\circ Q)=(M\circ T)\circ Q$.

Доведення. Рівність множин $M\circ(T\circ Q)=(M\circ T)\circ Q$ має місце, якщо $M\circ(T\circ Q)\subseteq(M\circ T)\circ Q$ і $(M\circ T)\circ Q\subseteq M\circ(T\circ Q)$. Доведемо перше з цих нестрогих включень. Покажемо, що $M\circ(T\circ Q)\subseteq(M\circ T)\circ Q$. Нехай $(a, e)\in M\circ(T\circ Q)$. Тоді існує такий елемент d із D, що $(a,d)\in T\circ Q$ і $(d,e)\in M$. Оскільки $(a,d)\in T\circ Q$, існує таке $b\in B$, що $(a,b)\in Q$ і $(b,d)\in T$. Оскільки $(b,d)\in T$ і $(d,e)\in M$, то $(b,e)\in M\circ T$. Оскільки $(b,e)\in M\circ T$ і $(a,b)\in Q$, то $(a,e)\in(M\circ T)\circ Q$. Таким чином, $M\circ(T\circ Q)\subseteq(M\circ T)\circ Q$. Включення $(M\circ T)\circ Q\subseteq M\circ(T\circ Q)$ доводиться аналогічно. Надамо таку можливість читачеві.

Відношення на множині, тобто виду $Q \subseteq A^2$, може володіти спеціальними властивостями, які випливають із наведених нижче означень.

Означення 1.12. Відношення $Q \subseteq A^2$ на множині A називається *рефлексивним*, якщо $(a,a) \in Q$ для всіх $a \in A$, що беруть участь у відношенні Q.

Це означає, що кожний елемент $a \in A$ перебуває у відношенні до самого себе (і, можливо, до інших елементів множини A). Матриця такого відношення буде мати діагональ, заповнену одиницями, а на стрілковій діаграмі кожна точка $a \in A$ буде мати петлю (a,a).

Означення 1.13. Відношення $Q \subseteq A^2$ називається *антирефлексивним*, якщо для всіх $(a,b) \in Q$ має місце нерівність $a \neq b$ (при цьому, очевидно, $a = b \Rightarrow (a,b) \notin Q$).

Матриця такого відношення має діагональ, заповнену нулями, і жодна точка стрілкової діаграми не буде мати петлі.

Означення 1.14. Відношення $Q \subseteq A^2$ називається *симетричним*, якщо для всіх пар $(a,b) \in Q$ при $a \neq b$ існують пари $(b,a) \in Q$.

Означення 1.15. Відношення $Q \subseteq A^2$ називається *антисиметричним*, якщо для всіх $a \in A$ і $b \in A$ із $(a,b) \in Q$ і $(b,a) \in Q$ виходить $a = b$.

Означення 1.16. Відношення $Q \subseteq A^2$ називається *транзитивним*, якщо для всіх $a \in A$, $b \in A$ і $c \in A$ при $a \neq b \neq c$ з того, що $(a,b) \in Q$ і $(b,c) \in Q$, виходить $(a,c) \in Q$.

Приклад 1.23. Нехай $A = \{1, 2, 3, 4, 5, 6\}$, $Q = \{(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (1,2), (1,4), (2,1), (2,4), (3,5), (5,3), (4,1), (4,2)\}$. Відношення $Q \subseteq A^2$ рефлексивне, оскільки для кожного (без винятку) $a \in A$ існує пара $(a,a) \in Q$. Для кожної пари $(a,b) \in Q$, де $a \neq b$, існує пара $(b,a) \in Q$, а це означає, що відношення $Q \subseteq A^2$ симетричне. Простим перебором можем переконатися, що для кожного сполучення пар $(a,b) \in Q$ і $(b,c) \in Q$ у множині Q існує пара $(a,c)\in Q$, а це означає, що відношення $Q\subseteq A^2$ транзитивне. Відношення $Q\subseteq A^2$ не є антисиметричним, оскільки не для всіх сполучень двох пар $(a,b)\in Q$ і $(b,a)\in Q$ має місце $a=b$. Зокрема, $(1,2)\in Q$ і $(2,1)\in Q$, але $1\neq 2$.

Означення 1.17. Відношення $Q\subseteq A^2$ н*азивається відношенням еквівалентності*, якщо воно є рефлексивним, симетричним і транзитивним.

Відношення Q на множині A, тобто $Q\subseteq A^2$, формується за деяким зазначеним принципом (правилом). Відношення $Q\subseteq A^2$ з прикладу 1.23 є відношенням еквівалентності на множині A, оскільки воно відповідає означенню 1.17. Класи еквівалентності у відношенні Q визначені для кожного $a\in A$ заданням відношень $Q=\{\{x\colon xQa_i,\ x\in A,\ a_i\in A\}\colon i=\overline{1,6}\}$. Для елемента $a_1=1$ маємо клас еквівалентності $[a_1]=\{1, 2, 4\}$. Аналогічно для елементів a_2 – a_6: $[a_2]=\{2,\ 1,\ 4\}$, $[a_3]=\{3,\ 5\}$, $[a_4]=\{4,\ 1,\ 2\}$, $[a_5]=\{5,\ 3\}$, $[a_6]=\{6\}$. Кількість різних класів еквівалентності дорівнює 3, це класи $\{1, 2, 4\}$, $\{3, 5\}$, $\{6\}$.

Таким чином, заданням відношення еквівалентності $Q\subseteq A^2$ на множині A здійснюється *розбиття* цієї множини на підмножини, елементи яких еквівалентні один одному за певною ознакою і не еквівалентні елементам інших підмножин (див. п.1.2). Розбиття множин на класи еквівалентності використовується в задачах управління реляційними базами даних.

Можливі три окремі випадки бінарних відношень на множині: тотожне, повне і порожнє. Якщо відношення $Q\subseteq A^2$ містить лише пари $(a,a)\in Q$ і ніяких інших, воно називається *тотожним*. Якщо $Q=A^2$, то відношення Q називається *повним*. Якщо $Q=\varnothing$, таке відношення називається *порожнім*. Надаємо читачеві можливість самостійно виявити особливості матриць і стрілкових діаграм таких відношень.

Поняття відношення і декартового добутку скінченних упорядкованих множин відіграють велику роль в теорії множин і математичних теоріях прикладного характеру. Розглянуті тут відношення на множинах мають узагальнений характер у тому сенсі, що в практичних застосуваннях зазвичай об'єктами дослідження є спеціальні форми відношень, такі як відповідності та відображення.

Контрольні питання

Які математичні об'єкти називають відношеннями? Які види відношень є унарними (одномісними), бінарними (двомісними), n-арними (n-місними)?

Якими способами можна здійснювати задання відношень двох множин? Опишіть технології задання відношень цими способами.

Поясніть поняття композиції відношень. Задайте композицію двох бінарних відношень за допомогою стрілкової діаграми.

Чи володіє операція композиції відношень властивостями комутативності та асоціативності?

Сформулюйте означення властивостей відношень: рефлексивності, антирефлексивності, симетричності, антисиметричності, транзитивності.

Які особливості мають матриці та стрілкові діаграми рефлексивного, антирефлексивного та симетричного відношень?

Яким чином пов'язані поняття класів еквівалентності .та розбиття множини?

Якими особливостями відрізняються матриці та стрілкові діаграми тотожного, повного та порожнього відношень?

1.4 Відповідності та відображення

Одним із видів відношень, дуже важливим в аспекті практичного застосування, є відповідність.

Означення 1.18. *Відповідністю* q називається трийка множин (X, Y, Q), в якій X – *область відправлення* відповідності, Y – *область прибуття* відповідності, Q – множина, яку називають *графіком відповідності*.

Серед елементів множин X і Y можуть бути такі, що не входять ні до жодної з пар (x,y) відповідності q. Тому з кожною відповідністю пов'язані ще дві множини: область визначення та область значень.

Означення 1.19. *Область визначення* відповідності – це елементи множини X (позначення Пр1Q), які беруть участь у відповідності (X, Y, Q).

Означення 1.20. О*бласть значень* відповідності – це елементи множини Y (позначення Пр2Q), які беруть участь у відповідності (X, Y, Q).

Позначення Пр1Q та Пр2Q областей визначення та значень відповідності мають сенс проекцій множини Q на множини X та Y відповідно. Тому очевидно, що область визначення є підмножиною області відправлення, а область значень є підмножиною області прибуття: Пр1$Q \subseteq X$, Пр2$Q \subseteq Y$.

Якщо $(x,y) \in Q$, то говорять, що елемент y відповідає елементу x (позначення $x \xrightarrow{Q} y$ або yQx). Графічну ілюстрацію відношення відповідності наведено на рис. 1.5.

Оскільки графіком Q відповідності q визначаються упорядковані пари (x,y), то очевидно, що останні є елементами декартового добутку множин $X \times Y$, тобто $Q \subseteq (X \times Y)$. Тому часто графік Q відповідності q також називають відповідністю. Прикладом відповідності є декартів добуток $A \times B$ множин A і B, якщо відповідністю охоплені всі можливі пари (a, b): $Q = A \times B = \{(a, b) | a \in A, b \in B\}$.

Приклад 1.24. Нехай $X=\{1, 2\}$, $Y=\{10, 20\}$. Добуток цих множин $X \times Y = \{(1, 10), (1, 20), (2, 10), (2, 20)\}$.

Маємо можливість отримати такі відповідності: $Q1=\{(1, 10)\}$, Пр1$Q1=\{1\} \subseteq X$, Пр2$Q1=\{10\} \subseteq Y$; $Q2=\{(1, 10), (1, 20)\}$, Пр1$Q2=\{1\} \subseteq X$, Пр2$Q2=\{10, 20\}=Y$.

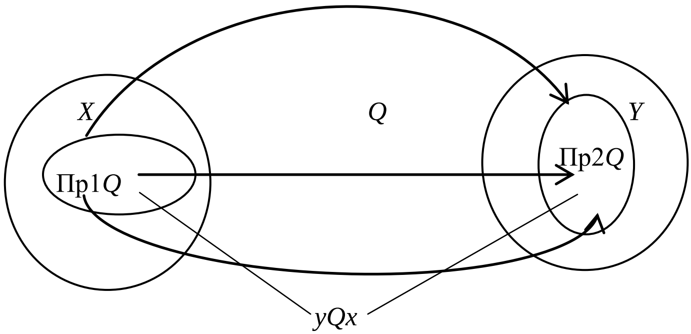


Рисунок 1.5 – Ілюстрація відповідності $q=(X, Y, Q)$

Приклад 1.25. Знайти області визначення і значень відповідності $Q=\{(a, 1), (a, 2), (b, 1), (c, 2), (c, 4), (c, 9), (d, 16)\}$.

Відповідь. Область визначення відповідності Пр1$Q=\{a, b, c, d\}$, а область значень Пр2$Q=\{1, 2, 4, 9, 16\}$.

Для задання відповідності можуть бути використані будь-які способи задання відношень: переліченням упорядкованих пар, пов'язаних відношенням відповідності, як у прикладах 1.24, 1.25; графіком Q відповідності (X, Y, Q) у вигляді матриці, кожен елемент a_{ij} якої дорівнює 1, якщо y_jQx_i, і 0 – у протилежному разі.

Наприклад, графік відповідності $Q=\{(a, 1), (a, 2), (b, 1), (c, 2), (c, 4), (c, 9), (e, 16)\}$ подається показаною нижче матрицею та стрілковою діаграмою (рис. 1.6).

(X, Y, Q)	1	2	4	9	16
a	1	1	0	0	0
b	1	0	0	0	0
c	0	1	1	1	0
e	0	0	0	0	1

Відмінність стрілкової діаграми відповідності (рис. 1.6) від стрілкової діаграми відношення (рис. 1.4) полягає у відсутно-

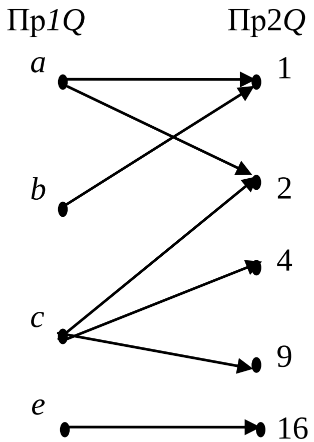


Рисунок 1.6– Подання відповідності стрілковою діаграмою

сті позначень (точками) елементів $x_i \in X$ та $y_j \in Y$, що не беруть участі у відповідності q.

Якщо області визначення та значень відповідності є підмножинами однієї множини, то має місце відповідність на множині. Очевидно, що у такому разі відповідність $Q \subseteq A^2$ є підмножиною (в загальному випадку нестрогою) декартового добутку A^2.

Приклад 1.26. Побудуємо матрицю відповідності $Q=\{(1,2), (3,1)\}$ на множині $A=\{1, 2, 3\}$.

Маємо окремий випадок відповідності виду $Q \subseteq A^2$, областями визначення і значень якої є множини Пр1$Q=\{1, 3\} \subset A$ та Пр2$Q=\{1, 2\} \subset A$. Матриця відповідності має такий вигляд:

(A, A, Q)	1	2
1	0	1
3	1	0

Відповідності на множинах володіють такими ж властивостями, як і бінарні відношення, тобто відповідність $Q \subseteq A^2$ є:

рефлексивною, якщо $(a,a) \in Q$ (або aQa) для всіх $a \in A$, що беруть участь у відповідності;

антирефлексивною, якщо для всіх $(a,b) \in Q$ має місце нерівність $a \neq b$;

симетричною, якщо для всіх пар $(a,b) \in Q$ при $a \neq b$ існують пари $(b,a) \in Q$;

антисиметричною, якщо для всіх $a \in A$ і $b \in A$ із $(a,b) \in Q$ і $(b,a) \in Q$ випливає $a = b$;

транзитивною, якщо для всіх $a \in A$, $b \in A$ і $c \in A$ з того що $(a,b) \in Q$ і $(b,c) \in Q$ випливає $(a,c) \in Q$.

Читачеві надається можливість самостійно виявити особливості стрілкових діаграм, якими задаються відповідності, що володіють такими властивостями.

Для кожної відповідності $q = (X, Y, Q)$, $Q \subseteq X \times Y$ існує *обернена відповідність*, яку одержують, якщо відповідність q розглядають у зворотному напрямку, тобто визначають елементи $x \in X$, що відповідають елементам $y \in Y$. Відповідність, обернена до відповідності q, позначається так: $q^{-1} = (Y, X, Q^{-1})$, де $Q^{-1} \subseteq Y \times X$, причому $(Q^{-1})^{-1} = Q$. Зрозуміло, що перехід від Q до Q^{-1} здійснюється взаємною перестановкою координат кожної впорядкованої пари. Необхідно зазначити, що при переході від Q до Q^{-1} область визначення стає областю значень і навпаки. Матриця графіка відповідності Q^{-1} – це транспонована матриця графіка Q.

Означення 1.21. Якщо матриця графіка Q прямої відповідності (X, Y, Q) збігається зі своєю транспонованою відносно головної діагоналі матрицею, то відповідність (X, Y, Q) називається *взаємно однозначною*. При цьому транспонована матриця є матрицею графіка Q^{-1} відповідності (Y, X, Q^{-1}), оберненої до прямої відповідності (X, Y, Q).

Означення 1.22. *Композицією відповідностей* (як і будь-яких відношень) називається послідовне застосування двох відповідностей, тобто композиція $p \circ q$ відповідностей q і p – це операція над трьома множинами X, Y, Z, на яких визначені дві відповідності: $q = (X, Y, Q)$, $Q \subseteq X \times Y$; $p = (Y, Z, P)$, $P \subseteq Y \times Z$, причому область значень першої відповідності (q) збігається з областю визначення другої відповідності (p): $\text{Пр}2Q = \text{Пр}1P$.

Приклад 1.27. Множини $X = \{x_1, \ldots, x_5\}$, $Y = \{y_1, \ldots, y_4\}$ і

$Z=\{z_1, z_2, z_3\}$ пов'язані відповідностями $A=\{(x_1, y_1), (x_2, y_1), (x_2, y_2), (x_2, y_4), (x_3, y_3), (x_3, y_4), (x_4, y_4), (x_5, y_2), (x_5, y_3), (x_5, y_4)\}$ і $B=\{(y_1, z_2), (y_2, z_1), (y_2, z_2), (y_3, z_3), (y_4, z_3)\}$. Композиція відповідності A з відповідністю B має вигляд: $C=B\circ A=\{(x_1, z_2), (x_2, z_2), (x_2, z_1), (x_2, z_3), (x_3, z_3), (x_4, z_3), (x_5, z_1), (x_5, z_2), (x_5, z_3)\}$.

Для опису відповідностей між множинами використовують поняття *відображення* однієї множину в іншу як окремий випадок відповідності. Для задання відображення потрібно вказати: множину, яка відображається, тобто *область визначення* відображення; множину, в яку відображається область визначення, тобто *область значень* відображення; закон відповідності, за яким для елементів першої множини вибираються елементи із другої множини.

Означення 1.23. *Відображенням* множини *A в множину B* називається відповідність φ: $A\to B$ (за іншою позначенням $A\xrightarrow{\varphi}B$), згідно з якою кожному елементу множини A ставиться у відповідність один і не більше ніж один елемент множини B. Елемент $b\in B$ називається *образом* елемента $a\in A$, а останній, у свою чергу, називається *прообразом* елемента $b\in B$.

Запис $\varphi:A\to B$ означає, що множина B містить образи всіх элементів множини A (позначення образу $b\varphi a$), тобто відображення є *повним*. Згідно з означенням 1.23 кожному елементу $b\in B$ (образу відповідних елементів $a\in A$) може відповідати один або більше елементів $a\in A$ (прообразів відповідного елемента $b\in B$), а кожному елементу $a\in A$ може відповідати не більше ніж один елемент $b\in B$, тобто відображення є *однозначним*. Цим клас відображень виділяється із загального класу відповідностей як окремий випадок. Відзначимо, що так само, як і відповідність загального виду, відображення не передбачає обов'язкової участі у відповідності $A\xrightarrow{\varphi}B$ всіх елементів $b\in B$, а однозначність відображення не означає, що для будь-яких пар $(a_1,b_1)\in\varphi$ та $(a_2,u_2)\in\varphi$ із $a_1 = a_2$ випливає $b_1=b_2$.

Кількість можливих відображень множини A в множину B визначити нескладно. Припустимо, що потужності множин A і B дорівнюють n і m відповідно. Побудувавши матрицю відображення (A, B, F) розмірністю $n\times m$, дійдемо до висновку, що кількість можливих відображень дорівнює m^n.

Означення 1.24. Відображення множини A в множину B називається *сюр'єкцією*, якщо для кожного $b\in B$ існує елемент $a\in A$ такий, що має місце відповідність $b\varphi a$.

Якщо відображення φ: $A\rightarrow B$ є сюр'єкцією, то на його стрілковій діаграмі в кожну точку, що позначає елемент $b\in B$, входить принаймні одна стрілка і всі $b\in B$ беруть участь у відображенні. Очевидно, що сюр'єкція множини A в множину B можлива тільки в тому випадку, якщо потужності цих множин перебувають у співвідношенні $|A|\geq|B|$.

Означення 1.25. Відображення множини A в множину B називається *ін'єкцією*, якщо різним елементам $a\in A$ відповідають різні елементи $b\in B$, тобто для $a_i, a_j\in A$, $a_i\neq a_j$ мають місце відповідності $b_i\varphi a_i$ та $b_j\varphi a_j$ такі, що $b_i\neq b_j$.

Якщо відображення φ: $A\rightarrow B$ є ін'єкцією, то на його стрілковій діаграмі в кожну точку, що позначає елемент $b\in B$, що бере участь у відображенні, входить в точності одна стрілка, а із кожної точки, що позначає елемент $a\in A$, виходить в точності одна стрілка. Очевидно, що ін'єкція множини A в множину B можлива тільки в тому випадку, якщо потужності цих множин знаходяться у співвідношенні $|A|\leq|B|$.

Означення 1.26. Якщо відображення φ: $A\rightarrow B$ є одночасно сюр'єктивним та ін'єктивним, то воно називається *взаємно-однозначним* відображенням множини *A на множину B*, або *бієкцією A на B*.

Для бієкції повинна виконуватися рівність $|A|=|B|$.

Означення 1.27. Повне однозначне відображення f: $X\rightarrow Y$ (або $X \xrightarrow{f} Y$) називається функціональним, або *функцією*.

Для функції можна записати таке визначення: $f=\{(x,y)\in X\times Y|\ y=f(x)\}$, тобто функцією є множина, елементами якої є пари (x,y), які беруть участь у функціональному відображенні. Значення $y=f(x)$ для будь-якої пари $(x,y)\in f$ називається функцією від x. Елемент $x\in X$ називають *аргументом* функції $y=f(x)$. Символічне позначення функціональної залежності y від x може бути й одним з таких: $x\xrightarrow{f}y$, $x\to y$, yfx.

Якщо область визначення функції f містить тільки один елемент, то її називають *функцією-константою*.

Для задання функції використовуються такі способи:

– переліченням всіх пар (x,y) елементів множин X і Y;

– формулою $y=f(x)$, $x\in X$, $y\in Y$, якщо X та Y – множини чисел (натуральних, цілих, дійсних, комплексних тощо);

– у вигляді графіка $(x,y)\in f$, $x\in X$, $y\in Y$, якщо X та Y – множини дійсних чисел.

Кількість аргументів функції (їх називають ще незалежними змінними) може бути більшою від одиниці. Наприклад, функція двох змінних $f=f(x, y)$, де $x\in X$, $y\in Y$, має областю визначення декартів добуток $X\times Y$. Формально таку функцію можна означити так: $f=\{(x,y,z)\in X\times Y\times Z|\ z=f(x, y)\}$. Аналогічно означаються функції більшого числа змінних. Додавання, віднімання, множення і ділення є двомісними (бінарними) функціями на R (множині дійсних чисел), тобто функціями виду f: $R^2\to R$, або $z=f(x,y)$, де $x, y, z\in R$.

Оскільки функції належать до класу відображень, то їм притаманні всі властивості останніх. Зокрема, функції можуть бути сюр'єктивними, ін'єктивними та бієктивними, а в останньому випадку функціональна залежність є *взаємно однозначною*, тобто функція f^{-1}: $Y\to X$, обернена до функції f: $X\to Y$, буде також однозначною.

Композиція функцій f: $X\to Y$ і g: $Y\to Z$ визначає множину значень $z\in Z$, що відповідають за послідовно застосованими законами f і g елементам $x\in X$: $z=(g\circ f)x=g(f(x))$. Функцію f, отриману з функцій g, q, …, u підстановкою їх одне в одне з перейменуванням аргументів, називають *суперпозицією функцій* g, q, …, u. Наприклад, суперпозицією є функція $p=f(z)$, де $z=g(y)$, $y=q(x)$, тобто $p=f(g(q(x)))$. Вираз, що описує суперпозицію з використання символів аргументів та знаків математичних операцій, називається *формулою*.

Більш загальним видом відображення, ніж функція, є *функціонал*, який встановлює залежність між множиною чисел та множиною функцій (іншими словами – числові оцінки функцій). Приклад функціонала:

$$I = \int_a^b f(x)dx.$$

За допомогою функціоналів визначають, зокрема, цільові функції (мінімізації, максимізації функціонала) в задачах оптимального управління динамічними системами.

Ще більш загальним видом відображення є *оператор*, ним встановлюється така відповідність між двома множинами функцій, що кожній функції з однієї множини відповідає певна функція з іншої. Наприклад, інтеграл Лапласа

$$\int_0^\infty f(t)e^{-pt}dt = F(p)$$

є оператором (з умовним позначенням $f(t) \risingdotseq F(p)$, який здійснює інтегральне перетворення множини функцій $f(t)$ дійсної перемінної t, $0 < t < \infty$, у множину функцій $F(p)$ комплексної змінної p. Інтегральні перетворення Лапласа широко використовуються в теорії автоматичного керування.

Контрольні питання

Поясніть поняття відповідності множин. Які підмножини визначаються поняттями областей відправлення, прибуття, визначення та значень відповідності?

Яким чином пов’язані між собою: області значень і прибуття відповідності; упорядковані пари (x, y) і елементи декартового добутку $X \times Y$; упорядковані пари (x, y) і графік відповідності?

У чому полягає відмінність понять “відношення” і “відповідність”?

Якими способами можна здійснювати задання відповідності множин?

У чому полягає відмінність стрілкових діаграм відповідності та відношення?

За яких умов має місце відповідність на множині?

Якими властивостями володіють відповідності на множинах?

Наведіть приклад рефлексивної (антирефлексивної, симетричної, антисиметричної, транзитивної) відповідності, зробіть її задання описом складу елементів, матрицею, стрілковою діаграмою.

Яку відповідність називають оберненою? Як її отримують із прямої відповідності?

За яких умов відповідність є взаємно однозначною?

Поясніть поняття композиції відповідностей.

Поясніть поняття “відображення множини в множину”.

За яких умов відповідність є відображенням?

Поясніть поняття образу і прообразу відображення.

В якому співвідносяться кількості образів і прообразів відоборaження?

Чому відображення на множині є однозначним?

За яких умов відображення на множині є: сюр’єкцією, ін’єкцією, біекцією?

За яких умов відображення на множині є взаємно однозначним?

Яке відображення називається функцією?

Якими способами можна здійснювати задання функції?

За яких умов функція є взаємно однозначною?

Поясніть процедури формування композиції та суперпозиції функцій.

Чи є функціонали та оператори видами відображення множини на множину? Наведіть їх приклади.

1.5 Вправи

1. Наведіть приклади множин, елементами яких є: а) люди, б) літери алфавіту, в) числа, г) геометричні фігури.

2. Опишіть і зобразіть на рисунку множини точок, координати x і y яких задовольняють такі умови: а) $y=x$; б) $y>x$; в) $y<x$; г) $x+y\leq 10$; д) $x^2+y^2\geq 1$.

3. Нехай $A=\{1, 2, 3, 4, 5\}$. Позначимо через B множину, елементами якої є квадрати елементів A. Задайте множину B всіма відомими Вам способами.

4. Опишіть множину $A=\{4, 8, 12, 16, 20, 24, 28\}$ за допомогою характеристичної властивості її елементів.

5. Наведіть декілька прикладів універсальних множин та їх підмножин, використовуючи для цього множини об’єктів і понять.

6. Визначте, які з наведених тверджень справедливі:

а) $|\{\varnothing\}|=1$; б) $\{\{\varnothing\}\}\in\{\{\{\varnothing\}\}\}$; в) $|\{\{\varnothing\}\}|=2$; г) $x\in\{x\}$, д) $\{x\}\subseteq\{x\}$, е) $\{x\}\in\{x\}$, ж) $\{x\}\in\{\{x\}\}$.

7. Визначте, скільки елементів містять такі множини: $\{x\}$, $\{\{x\}\}$, $\{x, \{x\}\}$, $\{\{x\}, x, \{\{x, \{x\}\}\}\}$.

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

9. Чи справедлива рівність $\{\{1, 2\}, \{2, 3\}\}=\{1, 2, 3\}$?

10 Наведіть множину розв’язків нерівності $(x–2/(8–x)\geq 0$.

11. Які з наведених тверджень правильні?

а) якщо $A\subset B$ і $B\subset C$, то $A\subset C$;

б) якщо $A\subseteq B$ і $B\subseteq A$, то $A=B$;

в) якщо $A\subseteq B$ і $B\subseteq C$, то $A\subseteq C$.

12. Які з наведених множин є підмножинами множини $D=\{7, 13, 25, 34, 101,112\}$?

а) $\{1, 7, 13\}$; б) $\{0, 1, 12\}$; в) $\{25, 112, 34\}$; г) $\{a, b, x, n\}$;

д) $\{7, 13, 25, 34, 101, 112\}$; е) $\varnothing$.

13. Доведіть, що дві множини рівні тоді і лише тоді, коли результати їх перетину та об'єднання збігаються.

14. Визначте, які з наведених множин дорівнюють одна одній: а) $A=\{x|\ x=2y,\ y\in N\}$; б) $C=\{1, 2, 3\}$;

в) $D=\{0, 2, -2, 3, -3, 4, -4, \dots\}$; г) $E=\{2x|\ x\in Z\}$.

15. Побудуйте булеан для множини A, якщо:

а) $A=\{\{\varnothing\}\}$; б) $A=\{1, 2, 3, 4\}$;

в) $A=\{$кіт, собака$\}$; г) $A=\{1, \{2, 3\}, 4\}$.

16. Нехай універсум U складається з чисел 1, 2, 3, ..., 9 і 33 літер українського алфавіту *а, б, в, ..., я*. Якщо $A=\{1, 3, 5,$ *а, в, д*$\}$ і $B=\{1, 2, 3, 4, 5,$ *а, б, г, д, е*$\}$, знайдіть: а) $\overline{B}$, б) $A\cap B$, в) $A\cup B$, г) $A\cap\overline{B}$.

17. Дано: $U=\{0, 1, 2, 3, 4, 5, \dots\}$, $A=\{3x|\ x\in U\}$, $B=\{5x|\ x\in U\}$. Визначте $A\cap B$.

18. Нехай U є множиною всіх точок (x, y) координатної площини, $A=\{(x,y)|\ y=|x|\}$, $B=\{(x,y)|\ y>|x|\}$, $L=\{(x,y)|\ x+y=2\}$, $M=\{(x,y)|\ x+y<2\}$. Зобразіть точками на площині такі множини: а) A, б) B, в) $A\cup B$, г) $\overline{B}$, д) L, е) $A\cap L$, ж) $B\cap M$, з) $\overline{M}$, к) $\overline{L}$.

19. На універсумі $U=\{0, 1, 2, 3, 4, 5, 6\}$ задані множини $A=\{1, 2, 3\}$, $B=\{3, 4, 5\}$. Визначте множини $A\cup B$, $A\cap B$, $A\setminus B$, $A-B$, $\overline{A}$, $\overline{B}$.

20. Доведіть за допомогою діаграм Венна такі тотожності:

$(A\setminus B)\cap(B\setminus A)=\varnothing$;

$A\cap(B\setminus A)=\varnothing$;

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

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

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

21. Чи випливає з $A=B\cup C$ те, що $A\setminus B=C$?

22.)Чи випливає з $A\setminus B=C$ те, що $A=B\cup C$?

23. Зобразіть на діаграмі Венна множини A, B, і C такі, що $A \subset B$ і $B \subset C$. Чи правильно у такому випадку включення $A \subset C$?

24. Розв'яжіть систему $\begin{cases} A \cap X = B, \\ A \cup X = C \end{cases}$, якщо $B \subseteq A \subseteq C$, $B \neq \varnothing$.

25. Задайте графік відповідності $A=\{(a,1), (a,3), (b,1), (b,2), (c,2), (c,4), (c,7), (d,15)\}$ матрицею та стрілковою діаграмою.

26. Множини $X=\{x_1, \ldots, x_4\}$, $Y=\{y_1, \ldots, y_6\}$ та $Z=\{z_1, z_2, z_3\}$ пов'язані відповідностями $A=\{(x_1,y_1), (x_2,y_1), (x_2,y_2), (x_2,y_4), (x_3,y_3), (x_3,y_4), (x_4,y_4), (x_4,y_5), (x_4,y_6)\}$ та $B=\{(y_1, z_2), (y_2, z_1), (y_2, z_2), (y_3, z_3), (y_4, z_3)\}$. Визначте композицію $C=A \circ B$.

27. Знайти області визначення та значень відповідностей: $\{(a,1), (a,2), (c,1), (c,2), (c,4), (d,3), (d,5)\}$; $\{(x,y) \in R^2 \mid y=x^2\}$;

$\{(a,1), (a,2), (a,3), (a,4), (a,5)\}$.

28. Нехай $A=\{1, 2, 3, 4\}$, $B=\{5, 6, 7, 8\}$, $C=\{11, 22, 33\}$.

Відповідності $R \subseteq A \times B$ і $S \subseteq B \times C$ задані таким чином:

$R=\{(1,6), (1,8), (2,5), (3,6)\}$,

$S=\{(5,11), (6,11), (7,22), (8,23)\}$.

Чи є вони рефлексивними, антирефлексивними, симетричними, антисиметричними?

Визначте відповідності:

R^{-1} і S^{-1}, $S \circ R$, $S \circ S^{-1}$ і $S^{-1} \circ S$, $R^{-1} \circ S^{-1}$.

Чи є вони рефлексивними, антирефлексивними, симетричними, антисиметричними, транзитивними?

29. Задано множину $A=\{1, 2, 3\}$. Дослідіть відповідності $R=\{(1,1), (1,4), (2,2), (2,5), (3,1), (3,3), (3,4)\} \subset A^2$ і $S=\{(1,2), (1,3), (2,1), (2,2), (3,1)\} \subset A^2$ на рефлексівність, антирефлексівність, симетрічність, антисиметрічність, транзитивність. Чи убачаються перелічені властивості в матрицях та стрілкових діаграмах цих відповідностей?

30. Які з наступних функцій f: $X \to Y$ на множині дійсних чисел є сюр'єктивними, ін'єктивними, бієктивними, якщо $x \in [-2, 4]$, $y \in [-1, 15]$?

а) $f(x,y) = \sin x$; б) $f(x,y) = 2^x - 1$; в) $f(x,y) = 15 - x^2$, г) $f(x,y) = 1$.

Розділ 2

ЕЛЕМЕНТИ МАТЕМАТИЧНОЇ ЛОГІКИ

Математична логіка – це наука, предметом дослідження якої є методи логічного виведення. При цьому розглядаються форми, а не сенс суджень. Формалізація процедур логічного виведення базується на поняттях істинності та хибності посилань і висновків, причому формалізоване оцінювання істинності та хибності здійснюється за допомогою символів 1 та 0 відповідно, тобто в алфавіті алгебри логіки (алгебри Буля). У цьому розділі розглядаються основні поняття об'єктів алгебри логіки, її закони та методи приведення логічних функцій до виду, придатного для розв'язування задач побудови моделей обчислювальних і керувальних пристроїв дискретної дії.

2.1 Прості та складні висловлення

Знання про явища в навколишньому світі відображаються нами в категоріях *об'єктів мислення*, таких як предмети, події та відношення між ними. Будь-який з цих об'єктів володіє комплексом певних ознак, тобто особливостей, властивих даному об'єкту. Для кожної з конкретних наук деякі ознаки об'єкта мислення будуть суттєвими, а інши – несуттєвими. Кожному об'єкту мислення людина ставить у відповідність поняття (термін, слово), що є характеристичною рисою об'єкта, яка відрізняє його від інших об'єктів. Отже, *поняття* – це форма мислення, що відображає об'єкти мислення в їх суттєвих ознаках. Між поняттями також можуть існувати певні відношення. Сукупності понять і відношень між ними як об'єкти мислення називають *судженнями*.

Моделлю судження в математичній логіці є *висловлення*, яким визначаються суттєві ознаки об'єкта мислення, групи об'єктів та зв'язків між об'єктами. Зрозуміло, що термін "висловлення" також є поняттям, тому в математичній логіці йому дається формалізоване означення.

Означення 2.1. *Висловленням* називається будь-яке розповідне речення (твердження, судження), про яке в даний момент можна сказати, істинне воно чи хибне.

Згідно із цим означенням єдиною ознакою висловлення як об'єкта математичної логіки є його характеристика істинності – хибності. Якщо речення не є розповідним, його істинність визначити неможливо. Тому речення, наприклад, "Ура!" або "Навіщо вивчати математичну логіку?" не є висловленнями. Розповідні речення також не завжди є висловленнями. Наприклад, речення "Незабаром буде тепло" або "Математична логіка – захоплива наука" не є висловленнями, бо неможливо встановити їх істинність або хибність. У той самий час речення A = "Україна розміщена у південній півкулі Землі" або B = "Знання математичної логіки необхідне фахівцю з інформаційних технологій" є висловленнями, оскільки очевидною є хибність першого з них та істинність – другого. Більше того, характеристика хибності – істинності цих висловлень не залежить ні від яких обставин. Такі висловлення називають відповідно *тотожно хибним* і *тотожно істинним*, їм присвоюються символічні "значення" $A=0$ і $B=1$. Характеристика істинності (хибності) висловлення може бути залежною від певних обставин. Прикладом може служити висловлення C="Значення першого і другого коренів квадратного рівняння дорівнюють 0,005 і 0,15 відповідно". Істинність (C=1) або хибність (C =0) цього висловлення залежить від значень коефіцієнтів рівняння. Висловлень, які одночасно є істинними й хибними, не існує.

У математичних моделях електронних пристроїв дискретної дії символи 1 і 0 відповідають двом (наприклад, високому та низькому) рівням електричних сигналів на входах і виходах пристрою.

Висловлення різного змісту не можна вважати однаковими (рівними). Однак два висловлення *називають еквівалентними*, якщо вони одночасно істинні або хибні. Еквівалентність висловлень позначається знаком ~ або =. Наприклад, запис $A\sim B$ (або $A=B$) означає, що A та B істинні або хибні одночасно.

У наведених вище прикладах висловлення є *простими*, оскільки вони не можуть бути поділені на частини, які також можна вважати висловленнями. Із простих (атомарних) висловлень за допомогою логічних зв'язків можна створювати *складні* (молекулярні) висловлення, які по суті є *формулами*. Зазвичай елементи складного висловлення, тобто прості висловлення, позначаються малими літерами (x, y, z, …), а складні висловлення – великими літерами (X, Y, Z, …). Істинність складного висловлення залежить від істинності його складових. Наприклад, складним буде висловлення C=“У тесті з теорії інформації правильною є відповідь за №3 (просте висловлення c_1), і студент Петренко це знає (просте висловлення c_2)”. Істинність складного висловлення C, безумовно, залежить від істинності простих висловлень c_1 і c_2. Отже, складне висловлення є *логічною функцією* простих висловлень – *логічних змінних* (аргументів логічної функції). Семантичні зв'язки (і, або, не, отже), що з'єднують логічні змінні у складних висловленнях, є знаками логічних дій, тобто *логічних операцій*.

Контрольні питання

Що розуміється під поняттям “об'єкт мислення”?

Яким чином відображаються об'єкти мислення в їх суттєвих ознаках?

Яку логічну конструкцію являє собою судження?

Які ознаки об'єкта мислення відображаються висловленням про нього?

Які математичні об'єкти є предметом дослідження в математичній логіці?

Наведіть коректне означення терміна “висловлення”.

Чи є характеристика істинності/хибності єдиною ознакою висловлення як об'єкта математичної логіки?

Які вимоги повинне задовольняти висловлення, щоб було можливим визначення його істинності?

Яке висловлення називають тотожно істинним?

Яке висловлення називають тотожно хибним?

Якими символами позначають істинність та хибність висловлень? Чи є ці символи числами?

Чи може висловлення бути одночасно істинним і хибним?

За яких умов два висловлення будуть еквівалентними? Що означає запис $A \sim B$?

Які висловлення називають простими?

Які висловлення називають складними?

Чи залежить істинність складного висловлення від істинності простих висловлень, з яких воно складене? Наведіть приклади.

Чи можна вважати складне висловлення логічною функцією простих висловлень?

Чи можна вважати прості висловлення аргументами логічної функції?

Наведіть приклади семантичних зв'язків, що використовуються як логічні операції для побудови складних висловлень.

Які мовні конструкції є аргументами логічної функції – складного висловлення?

2.2 Логічні операції

Побудова складних висловлень з простих здійснюється за допомогою *логічних операцій*. Розглянемо всі можливі логічні операції над висловленнями, що дозволить визначити закони алгебри логіки і дати формалізоване означення логічної функції.

1. Логічна операція "*Константа нуль*". Результатом цієї операції над висловленням A є висловлення, яке має значення $f(A)=0$ незалежно від істинності або хибності висловлення A.

2. Логічна операція "*Константа одиниця* ". Результатом

цієї операції над висловленням A є висловлення, яке має значення $f(A)=1$ незалежно від істинності чи хибності висловлення A.

3. Логічна операція "*Не*" (*інверсія, заперечення*). Результатом цієї операції над висловленням A є висловлення $f(A)=\overline{A}$ або $f(A)=\neg A$ ("$\bar{\ }$" і "$\neg$" – символи заперечення), яке істинне лише тоді, коли A хибне, і хибне, коли A істинне (читається "Не A").

4. Логічна операція "*Змінна А*". Результатом цієї операції над висловленням A є висловлення $f(A)=A$, яке істинне лише тоді, коли A істинне, і хибне, коли A хибне. (читається "Збереження A", "Повторення A", "Передавання A").

Перелічені чотири логічні операції є такими, що здійснюються над одним висловленням (необов'язково простим). Інших операцій такого роду в алгебрі логіки не існує.

Розглянемо повний перелік логічних операцій над двома висловленнями.

1. Логічна операція "*І*" (*кон'юнкція*, *добуток*, *логічне множення*). Результатом цієї операції над висловленнями A і B є складне висловлення $f(A,B)=A\wedge B$ (в інших позначеннях $A{\cdot}B$, AB, $A\&B$), яке істинне лише тоді, коли A і B одночасно істинні, і хибне, коли хоча б одне з них хибне (читається "A і B").

2. Логічна операція "*Або*" (*диз'юнкція*, *сума*, *логічне додавання*, *невиключне або*). Результатом цієї операції над висловленнями A і B є складне висловлення $f(A,B)=A\vee B$ (в іншому позначенні $A+B$), яке є хибним лише тоді, коли A і B одночасно хибні, і істинне, коли хоча б одне з них істинне (читається "A або B").

3. Логічна операція "*Якщо – то*" (*імплікація*). Результатом цієї операції над висловленнями A і B є складне висловлення $f(A,B)=A\rightarrow B$, яке є хибним лише тоді, коли A істинне, а B хибне, і істинне в інших випадках (читається "Якщо A, то B").

4. Логічна операція “*Заборона В*” (*заперечення імплікації* $A \to B$). Результатом цієї операції над висловленнями A і B є складне висловлення $f(A,B)=\overline{A \to B}$ (в іншому позначенні $A \Delta B$), яке є істинним лише тоді, коли A істинне, а B хибне, і хибне в інших випадках (читається “Невірно, що якщо A, то B”).

5. Логічна операція “*Заборона А*” (*заперечення імплікації* $B \to A$). Результатом цієї операції над висловленнями A і B є складне висловлення $f(A,B)=\overline{B \to A}=B \Delta A$, яке є істинним лише тоді, коли B істинне, а A хибне, і хибним у всіх інших випадках (читається “Невірно, що якщо B, то A”).

6. Логічна операція “*Рівнозначність*” (*еквівалентність*). Результатом цієї операції над висловленнями A і B є висловлення $f(A,B)=A \sim B$ (в іншому позначенні $A=B$), яке є істинним лише тоді, коли A і B одночасно істинні або хибні, і хибним в інших випадках (читається “A рівнозначне B”).

7. Логічна операція “*Нерівнозначність*” (*сума за модулем два, виключне або*). Результатом цієї операції над висловленнями A і B є висловлення $f(A,B)=A \oplus B$ (в іншому позначенні $A \neq B$), яке є істинним лише тоді, коли одне з висловлень A і B істинне, а інше – хибне, і хибним в інших випадках (читається “A нерівнозначне B”, “Або A, або B”).

8. Логічна операція “*Стрілка Пірса*” (*операція Пірса, функція Вебба*). Результатом цієї операції над висловленнями A і B є складне висловлення $f(A,B)=A \downarrow B=\overline{A \vee B}$, яке є істинним лише тоді, коли A і B одночасно хибні, і хибним в інших випадках (читається “Не A і не B”).

9. Логічна операція “*Штрих Шеффера*” (*операція Шеффера, функція Шеффера*). Результатом цієї операції над висловленнями A і B є складне висловлення $f(A,B)=A|B=\overline{A \wedge B}$, яке є хибним лише тоді, коли A і B одночасно істинні, та істинним, коли хоча б одне з них хибне (читається “Невірно, що A і B”).

Логічні операції в комп’ютерах виконуються над двійковими числами, порозрядно. Наведемо приклад виконання логічних операцій над двома двійковими числами.

Приклад 2.1. Логічні операції над двійковими числами

І	Або	Нерівнозначність	Не
01011010	0101010	0101010	X=01011010
&	+	$\oplus$	
11110000	1111000	1111000	
=	=	=	
01010000	1111010	1010010	$\overline{X}$ =10100101

Контрольні питання

Який математичний апарат застосовують для побудови складних висловлень?

Наведіть перелік логічних операцій.

Які математичні об’єкти одержуються в результаті виконання логічних операцій “*Константа нуль*” та “*Константа одиниця*” над висловленням *А*?

Які математичні об’єкти одержуються в результаті виконання логічних операцій “*Змінна А*” та “*Не*” над висловленням *А*?

Які математичні об’єкти одержуються в результаті виконання логічних операцій “*І*” та “*Або*” над висловленнями *А* і *В*? Які інші назви мають ці операції?

Які математичні об’єкти одержуються в результаті виконання логічних операцій “*Якщо – То*” і “*Заборона В*” над висловленнями *А* і *В*? Які інші назви мають ці операції?

Які математичні об’єкти одержуються в результаті виконання логічних операцій “*Рівнозначність*” і “*Нерівнозначність*” над висловленнями *А* і *В*? Які інші назви мають ці операції?

Які математичні об’єкти одержуються в результаті виконання логічних операцій “*Стрілка Пірса*” і “*Штрих Шеффе-*

ра" над висловленнями *А* і *В*? Які інші назви мають ці операції?

У чому полягає відмінність операції "*Стрілка Пірса*" від операції "*Штрих Шеффера*" над висловленнями *А* і *В*?

2.3 Алгебра логіки

Алгебра логіки є однією з багатьох алгебраїчних структур. Її називають *алгеброю Буля* (за авторством) або булевою алгеброю, хоча алгебра Буля не єдина у класі булевих алгебр. Алгебру логіки називають також *алгеброю висловлень*, оскільки операндами (логічними змінними, над якими здійснюються алгебраїчні операції) в цій алгебрі є висловлення.

З опису логічних операцій неважко помітити, що алгебра логіки подається структурою $<M, \wedge, \vee, \bar{\ }>$, де $M=\{0, 1\}$ – носій алгебри (двохелементна множина), елементи 0 та 1 якого використовуються відповідно як значення (числові еквіваленти) хибності та істинності висловлень (логічних змінних), а базовий набір $(\wedge, \vee, \bar{\ })$ операцій алгебри достатній для подання формулами будь-яких логічних операцій над логічними змінними.

Правила логічного виведення в алгебрі логіки базуються на її *законах* (табл. 2.1). Закони алгебри логіки являють собою тотожні рівності між аналітичними формулами, якими відображаються логічні зв'язки між логічними змінними (операндами), з використанням символів логічних операцій та дужок. Тотожність цих рівностей означає, що вони справедливі при будь-яких комбінаціях значень 0 і 1 істинності логічних змінних.

Таблиця 2.1 – Закони алгебри логіки

Назви законів	Тотожності
Закони ідемпотентності	$A+A=A$; $A\cdot A=A$

Закони комутативності	$A+B=B+A;\ A\cdot B=B\cdot A$
Закони асоціативності	$A+(B+C)=(A+B)+C=A+B+C;$ $A\cdot(B\cdot C)=(A\cdot B)\cdot C=A\cdot B\cdot C$
Закони дистрибутивності	$A\cdot(B+C)=(A\cdot B)+(A\cdot C);$ $A+(B\cdot C)=(A+B)\cdot(A+C)$
Закони поглинання	$A+(A\cdot B)=A;\ A\cdot(A+B)=A$
Закони повного склеювання	$(A\cdot B)+(\overline{A}\cdot B)=B;$ $(A+B)\cdot(\overline{A}+B)=B$
Закони неповного склеювання	$AB+A\overline{B}=A+AB+A\overline{B};$ $(A+B)(A+\overline{B})=A(A+B)(A+\overline{B})$
Закони де Моргана	$\overline{A+B}=\overline{A}\cdot\overline{B};$ $\overline{A\cdot B}=\overline{A}+\overline{B}$
Закон подвійного заперечення	$\overline{\overline{A}}=A$
Закони суперечності	$\overline{A\cdot\overline{A}}=1;$ $\overline{A+\overline{A}}=0$
Властивості одиниці	$A+1=1;\ A+\overline{A}=1;\ A\cdot 1=A$
Властивості нуля	$A\cdot 0=0;\ A\cdot\overline{A}=0;\ A+0=A$

В алгебрі логіки (табл. 2.1) так само, як і в алгебрі множин (п. 1.2, табл. 1.1), існує *принцип двоїстості*. Сутність його полягає в тому, що для будь-якого твердження з використанням символів 1 і 0 та операцій диз'юнкції й кон'юнкції можна скласти двоїсте до нього твердження шляхом заміни 0 на 1, 1 на 0, + на $\cdot$, $\cdot$ на +, яке також буде вірним. Наприклад, двоїстими є твердження алгебри логіки $A\cdot(B+C)=(A\cdot B)+(A\cdot C)$ і $A+(B\cdot C)=(A+B)\cdot(A+C)$. У таблиці 2.1 можна знайти інші пари тотожностей, що пов'язані відношенням двоїстості.

У вірності формул будь-якого із законів алгебри логіки можна переконатися за допомогою *таблиці істинності*. Колонки таблиці істинності поділяються на дві групи, перша з яких містить всі можливі *набори* (комбінації) значень істинності логічних змінних, а друга – значення істинності результатів логічних операцій та формул. Як приклад побудуємо таблицю істинності одного із законів де Моргана (табл. 2.2).

Таблиця 2.2 – Таблиця істинності закону $\overline{A \cdot B} = \overline{A} + \overline{B}$

№	A	B	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$	$A \cdot B$	$\overline{A \cdot B}$
0	0	0	1	1	**1**	0	**1**
1	0	1	1	0	**1**	0	**1**
2	1	0	0	1	**1**	0	**1**
3	1	1	0	0	**0**	1	**0**

Контрольні питання

Напишіть структурне подання алгебри множин. Що являють собою елементи цієї структури?

Наведіть перелік операцій алгебри логіки та їх умовні позначення (базові та варіативні).

Напишіть тотожні рівності, що відображають закони алгебри логіки: ідемпотентності, комутативності, асоціативності, дистрибутивності, поглинання, повного і неповного склеювання, де Моргана, суперечності, подвійного заперечення.

Керуючись правилами побудови таблиць істинності та її прикладом (табл. 2.2), побудуйте таблиці істинності законів алгебри логіки: ідемпотентності, комутативності, асоціативності, дистрибутивності, поглинання, повного і неповного склеювання, де Моргана, суперечності, подвійного заперечення, властивостей одиниці та нуля.

2.4 Логічні функції

Розглядаючи формули алгебри логіки (табл. 2.1) та приклад таблиці істинності (табл. 2.2), неважко помітити, що ліві та праві частини тотожностей по суті являють собою функції логічних змінних, якими є прості висловлення, причому ці функції набувають значення 0 або 1 у функціональній залежності від значень істинності аргументів. У подальшому значення істинності логічних змінних і функцій цих змінних будемо називати просто їх значеннями. Тому можемо ввести формалізоване означення логічної функції (ЛФ).

Означення 2.2. *Функція* $f(x_1, x_2, ..., x_n)$ называється *логічною* (*перемикальною, булєвою*), якщо вона, так само як її аргументи (логічні змінні) x_i, $i=\overline{1,n}$, може набувати тільки два значення: 0 та 1.

Означення 2.3. Упорядкована множина значень аргументів логічної функції називається *набором* і позначається кортежем $(a_1, a_2, ..., a_i, ..., a_n)$, $a_i \in \{0, 1\}$, $i=\overline{1,n}$.

Областю визначення логічної функції n змінних є сукупність усіх можливих упорядкованих наборів. Набір являє собою число у двійковій системі числення, якому відповідає певне число в десятковій системі числення. Ці числа можна розглядати як *номер набору* у відповідній системі числення. У такий спосіб установлюється порядкова нумерація наборів для будь-якої ЛФ, як це показано в таблиці 2.2. Таким чином, будь-яким набором $(a_1,a_2,...,a_i,...,a_n)$ задається його власний порядковий номер у двійковій та десятковій системах числення, і в той самий час в його i-му розряді ($i = 1, 2, \ldots, n$) знаходиться значення 0 або 1 змінної x_i даного набору.

Беручи до уваги поняття набору, логічній функції можна дати також наступне означення.

Означення 2.4. *Логічною функцією* (функцією алгебри логіки) називається функція $f(x_1, x_2, ..., x_n)$, що набуває значення 0 або 1 на наборах логічних змінних $x_1, x_2, ..., x_n$.

Оскільки кожний із n аргументів логічної функції може набувати двох можливих значення, то очевидно, що *областю визначення* логічної функції n змінних є сукупність $m=2^n$ можливих наборів.

Оскільки логічна функція n аргументів може набувати лише двох з можливих значень (0 і 1) на кожному з m наборів, то кількість усіх різних логічних функцій n аргументів дорівнює $k=2^m$. Значення величин m і k обчислюються за формулами комбінаторики (додаток А).

Означення 2.5. Якщо логічна функція n змінних визначена на всіх 2^n наборах, її називають *повністю визначеною*, в протилежному разі – *неповністю визначеною*.

Для наборів, на яких ЛФ не визначена, її значення можна взяти будь-яким із множини {0, 1} і таким чином привести її до повністю визначеної.

Логічну функцію можна задати її вербальним описом. Наприклад: "Функція трьох аргументів набуває значення 1, якщо два аргументи або всі три дорівнюють 1, а в інших випадках вона набуває значення 0".

Для задання логічної функції можна застосувати табличний спосіб, коли функція подається своєю *таблицею істинності*. Прикладом може служити таблиця 2.2 істинності функції $F=\overline{A\cdot B}=\overline{A}+\overline{B}$. При більшій ніж 2 кількості аргументів табличний спосіб незручний через великий розмір таблиці. Логічна функція може бути задана також у вигляді формули, наприклад: $f(x_1,x_2,x_3)=x_1\vee\overline{x_2}\vee x_3$.

Логічні функції одного і двох аргументів називають *елементарними*. Їх можна задати як формулами, так і таблицями істинності.

Для логічної функції одного аргумента існують два набори, кількість таких функцій – чотири, згідно з переліком:

$f(x)=0$, має назву "Константа нуль";

$f(x)=x$, має назву "Змінна x" (повторення x);

$f(x)=\overline{x}$, має назву "Інверсія (заперечення) x";

$f(x)=1$, має назву “Константа одиниця”.

Цей перелік по суті збігається з переліком логічних операцій над простим висловленням, наведеним у п. 2.2. Опис цих функцій подано таблицею 2.3.

Таблиця 2.3 – Таблиця істинності логічних функцій $f(x)$

Номер набору	x	$f(x)=0$	$f(x)=x$	$f(x)=\bar{x}$	$f(x)=1$
0	0	**0**	**0**	**1**	**1**
1	1	**0**	**1**	**0**	**1**

Для ЛФ двох аргументів існують чотири набори, кількість таких функцій – шістнадцять. Відповідність цих функцій переліку операцій алгебри логіки, наведеному в п. 2.2, зручніше буде пояснити на основі таблиці 2.4.

Таблиця 2.4 – Таблиця істинності логічних функцій $f(x_1,x_2)$

№	x_1	x_2	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**

У 16 клітинках першого рядка таблиці 2.4 розміщені номери $i=0, \ldots, 15$ функцій $f_i(x_1,x_2)$, а в тих самих за номерами клітинках інших рядків – виділені жирним шрифтом значення функцій $f_i(x_1,x_2)$ для відповідних наборів. Неважко помітити, що немає жодної пари функцій з однаковими значеннями на всіх наборах, тобто всі функції різні. Наведемо назви цих функцій у відповідності до логічних операцій, які вони реалізують:

$f_0(x_1,x_2)=0$, має назву “Константа нуль”;

$f_1(x_1,x_2)=x_1\wedge x_2$, має назву “Кон’юнкція” (функція “І”);

$f_2(x_1,x_2)=\overline{x_1 \rightarrow x_2}$, має назву “Заборона x_2”;

$f_3(x_1,x_2)=x_1$, має назву “Змінна x_1”;

$f_4(x_1,x_2)=\overline{x_2 \rightarrow x_1}$, має назву “Заборона x_1”;

$f_5(x_1,x_2)=x_2$, має назву “Змінна x_2”;

$f_6(x_1,x_2)=x_1\oplus x_2$, має назву “Сума за модулем 2” (логічна нерівнозначність, виключне “Або”);

$f_7(x_1,x_2)=x_1\vee x_2$, має назву “Диз’юнкція” (функція “Або”);

$f_8(x_1,x_2)=x_1\downarrow x_2$, має назву “Функція (стрілка) Пірса” (функція “Або–Не”);

$f_9(x_1,x_2)=x_1\sim x_2$, має назву “Рівнозначність”;

$f_{10}(x_1,x_2)=\overline{x_2}$, має назву “Інверсія x_2” (функція “Не”);

$f_{11}(x_1,x_2)=x_2\rightarrow x_1$, має назву “Імплікація від x_2 до x_1” (функція “Якщо–то”);

$f_{12}(x_1,x_2)=\overline{x_1}$, має назву “Інверсія x_1” (функція “Не”);

$f_{13}(x_1,x_2)=x_1\rightarrow x_2$, має назву “Імплікація від x_1 до x_2” (функція “Якщо–то”);

$f_{14}(x_1,x_2)=x_1|x_2$, має назву “Функція (штрих) Шеффера” (функція “І–Не”).

$f_{15}(x_1,x_2)=1$, має назву “Константа одиниця”.

З числа ЛФ двох змінних f_0 – f_{15} (табл. 2.4) можна виділити *базисні*, за допомогою яких можна одержати будь-які інші ЛФ. У той самий час базисні ЛФ неможливо одержати за допомогою інших. До базисних належать такі елементарні ЛФ: f_0, f_1, f_3, f_5, f_7, f_{10}, f_{12}, f_{15}.

За допомогою елементарних ЛФ можна будувати складні ЛФ від довільної скінченної кількості аргументів. Будь-яка складна ЛФ може бути поданою у вигляді формули, що зазначає послідовність елементарних логічних операцій над логічними змінними. Пріоритет виконання операцій у формулах такий: спочатку виконуються операції всередині дужок, після чого – операції заперечення, потім кон’юнкції, а далі – диз’юнкції та всі інші операції в порядку запису зліва направо.

Логічні вирази, тобто формули, що містять операції диз’юнкції та кон’юнкції, можна перетворювати (розкривати дужки, виносити загальний множник, переставляти місцями члени і т.п.) за правилами алгебри чисел, вважаючи формально диз'юнкцію операцією додавання, а кон'юнкцію – операцією множення. Але потрібно чітко пам'ятати, що в алгебрі логіки на відміну від алгебри чисел знаки додавання і множення означають логічні зв'язки "Або" та "І".

Формулу будь-якої ЛФ можна, використовуючи операції алгебри логіки та опис елементарних функцій, перетворити до такого виду, щоб усі логічні зв'язки між логічними змінними визначалися операціями диз'юнкції, кон'юнкції та інверсії. Покажемо це для ЛФ f_2, f_4, f_6, f_8, f_9, f_{11}, f_{13}, f_{14} двох змінних (табл. 2.4), які не є базисними:

$$f_2(x_1, x_2) = \overline{x_1 \to x_2} = \overline{\overline{x_1} \vee x_2} = x_1 \wedge \overline{x_2};$$

$$f_4(x_1, x_2) = \overline{x_2 \to x_1} = \overline{\overline{x_2} \vee x_1} = x_2 \wedge \overline{x_1};$$

$$f_6(x_1, x_2) = x_1 \oplus x_2 = \overline{x_1 \sim x_2} = \overline{(\overline{x_1} \vee x_2)(x_1 \vee \overline{x_2})} = (x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2);$$

$$f_8(x_1, x_2) = x_1 \downarrow x_2 = \overline{x_1 \vee x_2} = \overline{x_1} \wedge \overline{x_2};$$

$$f_9(x_1, x_2) = x_1 \sim x_2 = x_1 \wedge x_2 \vee \overline{x_1} \wedge \overline{x_2};$$

$$f_{11}(x_1, x_2) = x_2 \to x_1 = \overline{x_2} \vee x_1;$$

$$f_{13}(x_1, x_2) = x_1 \to x_2 = \overline{x_1} \vee x_2;$$

$$f_{14}(x_1, x_2) = x_1 | x_2 = \overline{x_1 \wedge x_2} = \overline{x_1} \vee \overline{x_2}.$$

Розглянемо особливі випадки, коли мають місце такі співвідношення між аргументи ЛФ: один із двох аргументів є константою 1 або 0; обидва аргументи дорівнюють один одному ($x_1 = x_2 = x$); один з аргументів є інверсією іншого ($x_1 = x$, $x_2 = \overline{x}$). У таких випадках мають місце такі тотожності:

$x \vee 1 = 1$, $x \wedge 1 = x$, $x \sim 1 = x$, $x \oplus 1 = \overline{x}$,

$x \to 1 = 1$, $1 \to x = x$, $x | 1 = \overline{x}$, $x \downarrow 1 = 0$;

$x \vee 0 = x$, $x \wedge 0 = 0$, $x \sim 0 = \overline{x}$, $x \oplus 0 = x$,

$x \to 0 = \overline{x}$, $0 \to x = 1$, $x | 0 = 1$, $x \downarrow 0 = \overline{x}$;

$x \vee x = x$, $x \wedge x = x$, $x \sim x = 1$, $x \oplus x = 0$, $x \to x = 1$, $x | x = \overline{x}$, $x \downarrow x = \overline{x}$;

$$x \vee \overline{x} = 1,\ x \wedge \overline{x} = 0,\ x \sim \overline{x} = 0,\ x \oplus \overline{x} = 1,$$
$$x \rightarrow \overline{x} = \overline{x},\ \overline{x} \rightarrow x = x,\ x | \overline{x} = 1,\ x \downarrow \overline{x} = 0.$$

Доведення правильності наведених вище перетворень формул логічних функцій легко здійснюється за допомогою таблиць істинності (аналогічно табл. 2.2), за збіжністю значень лівої та правої частин рівності, що доводиться.

Розглянути так само логічні функції трьох аргументів неможливо, оскільки їх кількість становить $2^8 = 256$. Проте це не має принципового значення, бо аргументи будь-якої ЛФ також можуть бути логічними функціями. Це означає, що створювати складні ЛФ можна шляхом підстановки інших ЛФ як аргументів.

Означення 2.6. Функція $F(f_1,\ldots, f_n)$, що одержується із функції $F(x_1,\ldots, x_n)$ шляхом підстановки функцій $f_1(x_1,\ldots, x_n),\ldots,$ $f_n(x_1,\ldots, x_n)$ замість аргументів $x_1,\ldots, x_n$, називається *суперпозицією* функцій $f_1,\ldots, f_n$.

Приклад 2.2. Задана функція $F(x_1, x_2) = x_1 \vee x_2$, аргументи x_1 та x_2 якої являють собою, відповідно, функції $f_1 = x_1 \wedge \overline{x_3}$, $f_2 = x_2 \vee x_4$. Потрібно одержати суперпозицію функцій f_1, f_2.

Розв'язання

Підставимо у формулу функції $F(x_1, x_2)$ замість аргументів x_1, x_2 функції f_1, f_2 та одержимо суперпозицію

$$F(f_1, f_2) = f_1 \vee f_2 = G(x_1, x_2, x_3, x_4) = x_1 \wedge \overline{x_3} \vee x_2 \vee x_4.$$

Таким чином, знаючи властивості логічних функцій та принцип суперпозиції, можна з простих логічних висловлень утворювати складні. Необхідно відзначити, що при виконанні операції підстановки функцій $f_1,\ldots, f_n$ замість аргументів у формулу функції $F(x_1,\ldots, x_n)$ не виключається можливість заміни не всіх аргументів. Крім того, серед аргументів функцій $f_1,\ldots, f_n$ можуть бути відсутніми деякі зі змінних $x_1,\ldots, x_n$. Тоді, після перейменування аргументів, буде одержана функція $G(y_1,\ldots, y_m)$ з числом аргументів $m<n$. Саме такі випадки віддзеркалює наступний приклад.

Приклад 2.3. Задана функція $F(x_1, x_2, x_3, x_4) = (x_1 \vee x_2) \wedge \overline{x_3} \vee x_4$, аргументи x_1, x_2, x_3, x_4 якої являють собою відповідно функції $f_1 = x_1 \wedge \overline{x_2}$, $f_2 = x_1 \oplus x_2$, $f_3 = x_1$, $f_4 = x_3$.

Потрібно одержати суперпозицію функцій f_1, f_2, f_3, f_4.

Розв'язування.

Підставимо у формулу функції $F(x_1, x_2, x_3, x_4)$ замість аргументів x_1, x_2, x_3, x_4 функції f_1, f_2, f_3, f_4 та одержимо суперпозицію

$$F(f_1, f_2, f_3, f_4) = (f_1 \vee f_2) \wedge \neg f_3) \vee f_4 = (x_1 \wedge \overline{x_2} \vee (x_1 \oplus x_2)) \wedge \overline{x_1} \vee x_3 =$$
$$= x_1 \wedge \overline{x_2} \wedge \overline{x_1} \vee (x_1 \oplus x_2) \wedge \overline{x_1} \vee x_3.$$

Враховуючи, що

$$x_1 \oplus x_2 = \neg (x_1 \sim x_2) = \neg ((\overline{x_1} \vee x_2) \wedge (x_1 \vee \overline{x_2})) = \neg (\overline{x_1} \vee x_2) \vee \neg (x_1 \vee \overline{x_2}) =$$
$$= (x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2),$$

одержимо суперпозицію

$$G(x_1, x_2, x_3) = x_1 \wedge \overline{x_2} \wedge \overline{x_1} \vee ((x_1 \wedge \overline{x_2}) \vee (\overline{x_1} \wedge x_2)) \wedge \overline{x_1} \vee x_3 =$$
$$= x_1 \wedge \overline{x_2} \wedge \overline{x_1} \vee (x_1 \wedge \overline{x_2}) \wedge \overline{x_1} \vee (\overline{x_1} \wedge x_2) \wedge \overline{x_1} \vee x_3 =$$
$$= \overline{x_1} \wedge (x_1 \wedge \overline{x_2} \vee x_1 \wedge \overline{x_2} \vee \overline{x_1} \wedge x_2) \vee x_3 =$$
$$= \overline{x_1} \wedge (x_1 \wedge \overline{x_2}) \vee \overline{x_1} \wedge x_2) \vee x_3 = 0 \wedge \overline{x_2} \vee \overline{x_1} \wedge x_2 \vee x_3.$$

Остаточно одержимо:

$$F(f_1, f_2, f_3, f_4) = G(x_1, x_2, x_3) = \overline{x_1} \wedge x_2 \vee x_3.$$

Результати наведених вище перетворень формул ЛФ та використання суперпозиції є такими, що всі логічні зв'язки проміж логічними змінними визначаються операціями диз'юнкції, кон'юнкції та інверсії. Це означає, що відповідні цим операціям елементарні функції $f_1(x_1, x_2) = x_1 \wedge x_2$ (функція

“І”), $f_{\text{Або}}(x_1, x_2) = x_1 \vee x_2$ (функція “Або”), $f_{\text{Не}}(x) = \overline{x}$ (функція “Не”) створюють систему логічних функцій, яка володіє властивістю функціональної повноти. Наведемо формальне означення цього поняття.

Означення 2.7. Набір елементарних ЛФ, за допомогою якого можна подати будь-яку ЛФ, використовуючи закони алгебри логіки, суперпозиції та підстановки, називається *функціонально повним*.

Система всіх розглянутих вище елементарних ЛФ є функціонально повною, однак надлишковою. Меншою за складом є система базисних фнкцій f_0, f_1, f_3, f_5, f_7, f_{10}, f_{12}, f_{15} (табл. 2.4), але й ця система є надлишковою. Найменшою надлишковістю володіє згадана вище функціонально повна система “І–Або–Не”:

$$\begin{cases} f_{\text{I}}(x_1, x_2) = x_1 \wedge x_2, \\ f_{\text{Або}}(x_1, x_2) = x_1 \vee x_2, \\ f_{\text{Не}}(x) = \overline{x}. \end{cases} \tag{2.1}$$

Система (2.1) надлишкова, оскільки з неї можна вилучити або диз’юнкцію, або кон’юнкцію без втрати повноти. Вилучену функцію можна дістати з інших двох, що залишилися. Наприклад, виключивши із (2.1) кон’юнкцію $f_{\text{I}}(x_1,x_2)$, останню одержують, застосувавши заперечення у функції $f_{\text{Або}}(x_1, x_2)$ до окремих доданків і до всього виразу у правій частині:

$$f_{\text{I}}\overline{(x_1, x_2)} = \overline{\overline{x_1} + \overline{x_2}} = \overline{\overline{x_1}} \cdot \overline{\overline{x_2}} = x_1 \cdot x_2.$$

Отже, через вилучення із системи (2.1) кон’юнкції $f_{\text{I}}(x_1, x_2)$ набір функцій $f_{\text{Або}}(x_1, x_2)$ та $f_{\text{Не}}(x)$ не втрачає функціональної повноти. Однак, якщо із набору $f_{\text{Або}}(x_1, x_2)$ та $f_{\text{Не}}\overline{(x)}$ вилучити одну з цих функцій, то одержаний набір у складі однієї функції вже не буде функціонально повним.

Означення 2.8. Якщо з функціонально повного набору елементарних ЛФ не можна вилучити жодної функції без втрати його повноти, то такий набір називається *базисом*.

Базисами логічних функцій є:

$\{\wedge,\ ^{-}\}$ – кон'юнктивний базис;

$\{\vee,\ ^{-}\}$ – диз'юнктивний базис;

$\{\rightarrow\}$ – імплікативний базис, де $x_1 \rightarrow x_2 = f_{13}(x_1, x_2) = \overline{x_1} \vee x_2$;

$\{\downarrow\}$ – базис Вебба (Пірса), де $x_1 \downarrow x_2 = f_8(x_1, x_2) = \overline{x_1 \vee x_2}$;

$\{|\}$ – базис Шеффера, де $x_1 | x_2 = f_{14}(x_1, x_2) = \overline{x_1 \wedge x_2}$.

Для кожної складної ЛФ можна одержати еквівалентну до неї функцію в будь-якому з перелічених базисів. Наприклад, $\overline{x}_1 \vee x_2 \wedge x_3 = x_1 \rightarrow (\overline{x_2 \rightarrow \overline{x}_3})$.

Відзначимо, що функціонально повна система (2.1) "І–Або–Не" формально не є базисом. Проте вона широко застосовується при аналітичному поданні й перетвореннях логічних функцій та при технічній реалізації останніх у пристроях дискретної дії (дискретних пристроях). Тому набір функцій (2.1) умовно відносять до числа базисів.

Завершуючи розгляд властивостей логічних функцій та їх перетворень, уведемо ще три специфічні поняття.

Означення 2.9. Логічна функція називається *інверсною* по відношенню до іншої, якщо вона одержується із останньої шляхом інверсії її значень на всіх наборах.

Кожна з функцій f_0 – f_{15} має інверсну (табл. 2.4). Наприклад, $f_0 = \neg f_{15}$, $f_1 = \neg f_8$, $f_2 = \neg f_{15}$.

Означення 2.10. Дві логічні функції з однаковою кількістю аргументів *рівносильні* одна одній, якщо вони набувають на всіх наборах однакові значення.

Означення 2.11. Логічна змінна x_i є *дійсною*, якщо значення логічної функції $f(x_1,\ x_2,\ ...,\ x_n)$ змінюється при зміні значення x_i. У протилежному разі ця змінна для даної функції є *фіктивною*, тобто не є її аргументом.

Поняття рівносильності ЛФ, дійсності та фіктивності їх аргументів необхідні для розуміння того, що в практичних застосуваннях може бути необхідним здійснення аналізу ЛФ з невизначеним числом аргументів, для якої потрібно скласти аналітичний вираз, маючи на увазі, що деякі з аргументів, включених з надлишковістю до списку аргументів ЛФ, можуть дійсно бути аргументами даної функції. Цей факт з'ясовується у підсумку проведеного аналізу ЛФ.

Контрольні питання

Наведіть означення логічної функції.

Наведіть означення набору логічної функції.

Наведіть означення області визначення ЛФ.

Чи є набір логічної функції кортежем?

Чи є набір логічної функції числом?

В якій системі числення подається число, яким позначається порядковий номер набору?

Поясніть алгебраїчний зв'язок між набором логічної функції та його порядковим номером.

Скільки різних наборів має логічна функція n змінних?

Чому дорівнює кількість усіх різних логічних функцій n змінних?

Поясніть поняття повністю та неповністю визначених логічних функцій. Яким чином неповністю визначена логічна функція приводиться до повністю визначеної?

Якими способами можна задати логічну функцію? Поясніть, в яких випадках переважно застосовується той чи інший спосіб.

Які логічні функції називають елементарними?

Скільки різних наборів мають логічні функції n змінних, якщо n=1, 2, 3, 4, 5?

Наведіть перелік логічних функцій однієї змінної та побудуйте для них таблиці істинності.

Наведіть перелік логічних функцій двох змінних та побудуйте для них таблицю істинності.

За якою ознакою логічні функції створюють клас базисних логічних функцій? Які з логічних функцій двох змінних складають систему базисних функцій?

Чи можна створювати складні логічні функції шляхом підстановки більш простих функцій як аргументів?

Яку функцію називають суперпозицією логічних функцій $f_1,\dots, f_n$?

Чи є обов'язковою заміна всіх аргументів логічної функції $F(x_1,\dots, x_n)$ функціями $f_1,\dots, f_n$ під час виконання операції підстановки з метою одержання суперпозиції?

Чи є допустимою заміна аргументів логічної функції $F(x_1,\dots, x_n)$ під час виконання операції підстановки з метою одержання суперпозиції такими функціями $f_1,\dots, f_n$, які не містять серед своїх аргументів деяких зі змінних $x_1,\dots, x_n$?

Дайте означення функціонально повної системи елементарних логічних функцій.

Наведіть приклади надлишкових функціонально повних систем.

Яка з надлишкових функціонально повних систем логічних функцій має мінімальну надлишківість?

Чи є система елементарних логічних функцій функціонально повною?

Чи достатньо операцій диз'юнкції, кон'юнкції та інверсії для подання формули будь-якої логічної функції?

Дайте означення базису логічних функцій. Наведіть перелік базисів.

У зв'язку з чим набір логічних функцій І–Або–Не умовно відносять до базисів?

Дайте означення інверсної логічної функції.

Наведіть перелік пар взаємно інверсних елементарних логічних функцій одного і двох аргументів.

Дайте означення рівносильності логічних функцій. Чи є такі функції серед елементарних логічних функцій?

Дайте означення дійсності та фіктивності аргументів логічної функції.

2.5 Способи технічної реалізації логічних функцій

У практичних застосуваннях логічні функції являють собою математичні моделі дискретних інформаційних процесів. В технічних, технологічних та організаційних системах за допомогою логічних функцій розв'яуються задачі ситуаційного аналізу та формування команд управління. Функціонування обчислювальних пристроїв (комп'ютерів, мікроконтролерів) здійснюється за алгоритмами, що містять логічні операції. Технічними засобами реалізації логічних функцій є пристрої дискретної дії (дискретні пристрої). Прикладами дискретних пристроїв, які ми розглянемо, є комбінаційні схеми (КС).

Комбінаційними схемами прийнято називати такі дискретні пристрої, вихідні сигнали яких визначаються лише комбінацією вхідних сигналів у поточний момент часу і не залежать від сигналів, що надійшли на входи пристрою в попередні моменти часу. Існують два види комбінаційних схем: контактні, що носять назву "Контактна схема", та безконтактні під назвою "Логічна схема".

Контактними схемами називають ділянки електричних кіл, які містять керовані ключі (контакти). Будь-яку ЛФ можна реалізувати контактною схемою, а кожній такій схемі можна поставити у відповідність формулу логічної функції – функції провідності ділянки електричного кола. У такий спосіб розв'язуються задачі аналізу контактних схем (виявлення можливостей спрощення) та їх синтезу (побудови схеми за допомогою опису її функцій провідності). Цю специфічну галузь застосування алгебри логіки називають *алгеброю контактів*.

Елементами алгебри контактів є прості висловлення: $x =$ "Замикальний контакт x замкнений" та $x =$ "Розмикальний контакт x розімкнений". Якщо висловлення x істинне (спрацювало реле або контактор, якому належить цей контакт), то значення істинності цього висловлення $x=1$ і $\bar{x}=0$. У протилежному разі (вихідний стан контакту) $x=0$ і $\bar{x}=1$.

Подальші міркування зводяться до наступного. Якщо ділянка електричного кола (контактна схема) проводить струм, то вважається, що вона існує, тобто її функція провідності $F(x_1, \ldots, x_n)=1$, в протилежному разі ця ділянка не існує, тобто $F(x_1, \ldots, x_n)=0$. Аргументами $x_1, \ldots, x_n$ функції провідності є стани контактів, що містяться у контактній схемі.

Для зручності складання функцій провідності позначення контакту x_i в контактній схемі та відповідної йому логічної змінної роблять однаковими , якщо контакт замикальний. У протилежному разі, якщо контакт розмикальний, його позначають $\overline{x_i}$, тобто відповідною змінною із запереченням (рис. 2.1, 2.2). Прийнявши цю домовленість, можемо будувати функції провідності контактних схем за законами алгебри логіки, а саме: стан ділянки з послідовним з’єднанням контактів визначається кон’юнкцією змінних стану цих контактів, а стан ділянки з паралельним з’єднанням нерозгалужених гілок (з одним або декількоми контактами) – диз’юнкцією змінних стану (функцій провідності) цих гілок. Покажемо це на наступному прикладі.

Приклад 2.4. Ділянка електричного кола (рис. 2.1) містить у своїх гілках контакти (керовані ключі) x_1, ..., x_5, а на її вхід подається напруга $U_{вх}$.

Напруга на виході цієї ділянки електричного кола

$$U_{вих}=U_{вх}\,F(x_1, x_2, x_3, x_4, x_5),$$

де

$F(x_1, x_2, x_3, x_4, x_5)=(x_1 \vee \overline{x_2} \wedge x_3) \wedge x_4 \wedge x_5 = x_1 \wedge x_4 \wedge x_5 \vee \overline{x_2} \wedge x_3 \wedge x_4 \wedge x_5$ – функція провідності, логічне значення якої (1 або 0) береться як натуральне число.

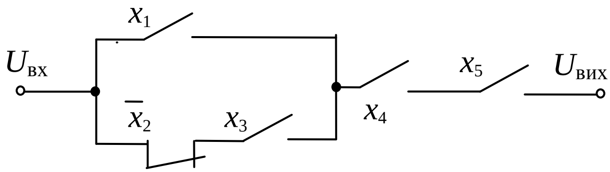


Рисунок 2.1 – Схема ділянки електричного кола

Значення $F(x_1, x_2, x_3, x_4, x_5) = 1$, якщо ділянка електричного кола проводить струм, має місце лише при таких наборах значень аргументів:

1) x_1=1, x_4=1, x_5=1 незалежно від стану контактів x_2 і x_3;

2) $x_2 = 0$ ($\overline{x_2}$=1, вихідний стан контакту), x_3=1, x_4=1, x_5=1 незалежно від стану контакта x_1;

3) x_1=1, x_2=0, x_3=1, x_4=1, x_5=1 (усі контакти замкнені).

Приклад 2.4 ілюструє процедуру аналізу умов роботи заданої контактної схеми. Якщо ми маємо задану логічну функцію, то розглядаючи її як функцію провідності, можемо виконати синтез відповідної до неї контактної схеми, як це показано у наступному прикладі.

Приклад 2.5. Побудуємо контактну схему, яка технічно реалізує логічну функцію

$F(x_1, x_2, x_3, x_4, x_5) = \overline{x}_1 \wedge (x_2 \vee x_3) \wedge \overline{x}_4 \wedge x_5$.

Аналіз структури заданої ЛФ показує, що вона містить три логічні співмножники: $\overline{x}_1$, $x_2 \vee x_3$ та $\overline{x}_4 \wedge x_5$. Тому контактна схема буде містити три послідовно з'єднані ланки з такими самими функціями провідності (рис. 2.2).

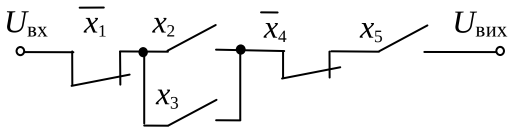


Рисунок 2.2 – Контактна схема до прикладу 2.5

У відповідності синтезованої контактної схеми до заданої ЛФ можна переконатися, склавши для неї функцію провідності та перевіривши збіжність цих функцій.

Таким чином, застосування алгебри логіки є зручним інструментом для моделювання та проектування контактних схем за заданими алгоритмами функціонування.

Будь-яка логічна функція n змінних, як це зазначено вище, може бути реалізована логічною схемою. *Логічна схема* являє собою безконтактний дискретний пристрій (комбінаційну схему) у вигляді певним чином з'єднаних між собою логічних елементів. *Логічний елемент* (ЛЕ) являє собою електронний пристрій, що фізично реалізує певну елементарну ЛФ – заперечення (інверсію), кон'юнкцію, диз'юнкцію, штрих Шеффера, стрілку Пірса та інші. Сигнали на виході та входах ЛЕ (потенціальні або імпульсні) є фізичними носіями значень ЛФ та її аргументів. Значення сигналу на виході ЛЕ однозначно визначається сигналами на його входах у поточний момент часу, тобто логічні елементи є дискретними пристроями без пам'яті.

Оскільки будь-яку ЛФ можна реалізувати за допомогою відповідної комбінації елементарних ЛФ, то і будь-яку логічну схему можна побудувати за допомогою відповідної комбінації логічних елементів. Сучасна елементна база електронної техніки досить розвинена, існують різноманітні види ЛЕ для виконання як елементарних, так і складних логічних функцій від 3, 4 і більше аргументів.

Для побудови логічних схем довільної складності необхідно мати сукупність ЛЕ, достатню для реалізації всіх функцій, що входять до однієї з функціонально повних систем логічних функцій. Сукупність ЛЕ, за допомогою яких здійснюється технічна реалізація вибраного функціонально повного набору логічних функцій, називають *функціонально повним набором (*або *базисом*) *логічних елементів*.

Кожна елементарна ЛФ реалізується логічним елементом певного типу, назва якого збігається з назвою ЛФ. Найбільшого поширення, з точки зору технічної реалізації, набули функціонально повні набори ЛЕ, що складаються: з елементів І та НЕ; з елементів Або та Не; з одного елемента Або-Не (стрілка Пірса); елемента І-Не (штрих Шеффера); з елементів І, Або та Не. Останній базис ЛЕ не є мінімальним, але часто найбільш зручним для подання складних логічних функцій.

Розглянемо реалізацію елементарних ЛФ типовими ЛЕ (повторювач, інвертор, І, Або, І-Не, Або-Не) з одним виходом. Під логічним значенням “0” сигналів на входах і виході ЛЕ будемо розуміти напругу (струм) низького рівня, фізичне значення якого близьке до нуля, а під значенням “1” – напругу (струм) високого рівня, додатного за знаком. Відзначимо, що всі вхідні сигнали повинні надходити на ЛЕ одночасно.

Умовне графічне позначення ЛЕ показує функцію, яку реалізує елемент, число входів і виходів. Умовне позначення ЛЕ має форму прямокутника, в якому розміщують символ функції. Входи ЛЕ розміщують з лівого боку прямокутника, виходи – з правого (рис. 2.3). Кількість входів елемента Або дорівнює кількості доданків у диз’юнктивному записі ЛФ, а логічний елемент І може мати стільки входів, скільки множників є в кон'юнкції.

Наведені на рис. 2.3 схеми ЛЕ називають додатною логікою, бо вони реалізують свої функції для прямих значень сигналів, якщо ними вважати сигнали високого рівня. Наступний приклад ілюструє побудову комбінаційної логічної схеми, яка реалізує складну ЛФ у базисі логічних елементів І–Або–Не.

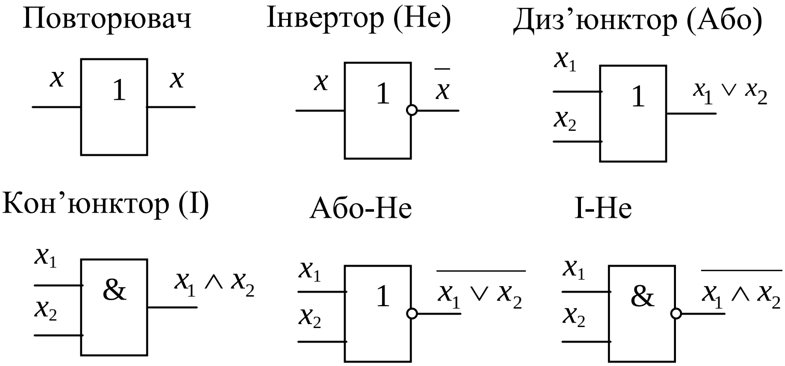


Рисунок 2.3 – Логічні елементи додатної логіки

Приклад 2.6. Побудуємо схемну реалізацію логічної функції $f = x_1 \wedge \overline{x_2} \wedge x_3 \vee x_1 \wedge x_2 \wedge \overline{x_3}$ у базисі І–Або–Не.

Задана ЛФ являє собою диз’юнкцію двох кон’юнкцій. Логічну схему її технічної реалізації подано на рис. 2.4.

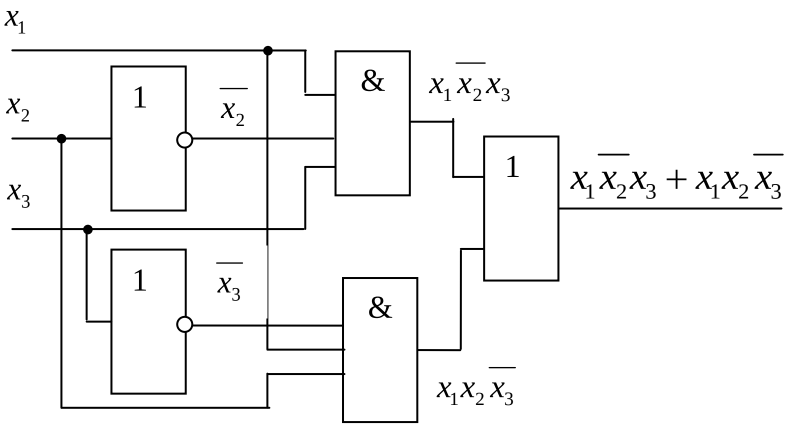


Рисунок 2.4 – Комбінаційна схема реалізації логічної функції $f(x_1, x_2, x_3) = x_1\overline{x_2}x_3 + x_1x_2\overline{x_3}$

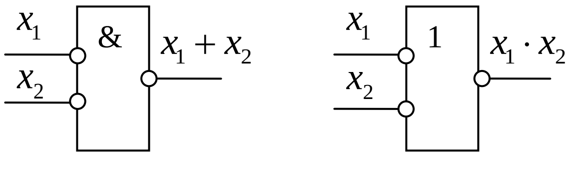
На підставі правил де Моргана

$$x_1 + x_2 = \overline{\overline{x}_1 \cdot \overline{x}_2}, \qquad (2.2)$$

та

$$x_1 \cdot x_2 = \overline{\overline{x}_1 + \overline{x}_2} \qquad (2.3)$$

бачимо, що операції І та Або можна виконати відповідно за допомогою логічних елементів Або-Не та І-Не (рис. 2.5).



$$x_1 + x_2 = \overline{\overline{x}_1 \cdot \overline{x}_2} \qquad\qquad x_1 \cdot x_2 = \overline{\overline{x}_1 + \overline{x}_2}$$

Рисунок 2.5 – Логічні елементи І-Не та Або-Не від’ємної логіки

Тотожності (2.2), (2.3) і рис. 2.5 свідчать про те, що логічні елементи І-Не та Або-Не для заперечень вхідних сигналів – це елементи Або та І відповідно. Ці елементи називають від’ємною логікою (через інвертування вхідних сигналів) за умови, що за логічну одиницю взято високий рівень сигналу. Кожен з них утворює функціонально повний набір (базис) логічних елементів, оскільки дає можливість реалізувати будь-яку складну логічну функцію.

Під час проектування інтегральних схем, які реалізують складні ЛФ, з точки зору технологічності їх виготовлення дуже важливо забезпечити однотипність використовуваних ЛЕ. Для цього прагнуть застосовувати ЛЕ одного базису. Тому потрібно навчитися зводити логічні вирази від одного базису ЛФ до іншого. Наприклад, щоб звести до базису І-Не схему рис. 2.4, потрібно логічну функцію, яка описує її роботу, з допомогою законів алгебри логіки (в цьому випадку застосовують закон подвійного заперечення і правила де Моргана) перетворити до потрібного вигляду, як це показано у прикладі 2.7.

Приклад 2.7. Побудуємо схемну реалізацію логічної функції $f = x_1 \wedge \overline{x_2} \wedge x_3 \vee x_1 \wedge x_2 \wedge \overline{x_3}$ з прикладу 2.6 у базисі І–Не. Для цього здійснемо перетворення цієї функції до вигляду заперечення кон’юнкції на підставі (2.2):

$$f = x_1 \overline{x_2} x_3 + x_1 x_2 \overline{x_3} = \overline{\overline{x_1 \overline{x_2} x_3} \cdot \overline{x_1 x_2 \overline{x_3}}}\,.$$

Цей вираз містить лише кон’юнкції та їх заперечення, схему його реалізації у базисі І–Не подано на рис. 2.6.

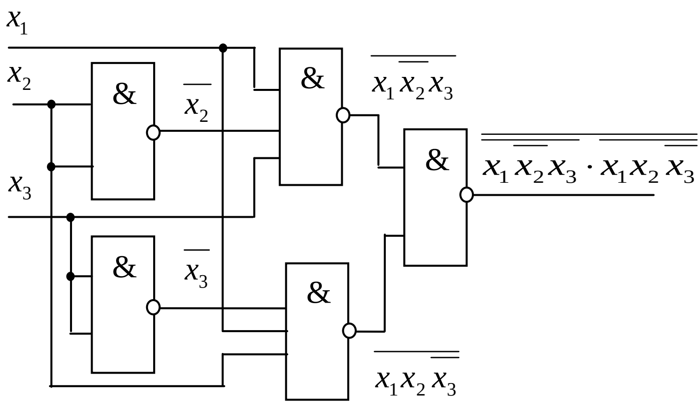


Рисунок 2.6 – Аналог схеми рис. 2.4 у базисі І-Не

Логічний елемент може мати два виходи, на першому з яких формується вихідний сигнал із прямим значенням, а на другому – із запереченням. Про такі ЛЕ говорять, що вони мають *парафазний* вихід. Якщо в логічній схемі використовується лише один із виходів ЛЕ, то інший зазвичай не показують, як на рис. 2.4 і 2.6.

На завершення зазначимо, що розглянуті тут способи технічної реалізації логічних функцій мають практичне значення в задачах конструювання дискретних пристроїв у апаратному виконанні. У програмно-технічних комплексах ЛФ реалізуються засобами програмування спеціалізованих інтегральних схем (програмованих логічних матриць) та мікропроцесорних обчислювальних пристроїв (мікроконтролерів).

Контрольні питання

Якими засобами здійснюють технічну реалізацію логічних функцій?

Дайте означення комбінаційної схеми. Які види комбінаційних схем Вам відомі?

Дайте означення контактної схеми та її функції провідності. Чи можна реалізувати контактною схемою будь-яку логічну функцію?

Які об’єкти є елементами алгебри контактів?

Яким чином пов’язані між собою стан ділянки електричного кола та значення її функції провідності?

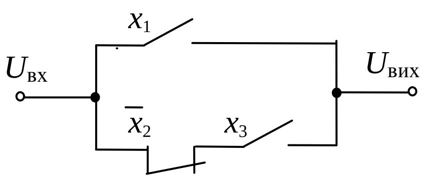
Які об’єкти є аргументами функції провідності ділянки електричного кола (контактної схеми)?

Яким чином пов’язані позначення контактів у контактній схемі та логічних змінних, значеннями яких відображається стан контактів?

Чи можна моделювати дискретні процеси в контактних схемах за допомогою логічних функцій?

Чи можна проектувати контактні схеми за допомогою логічних функцій?

Чи вірно, що показана тут контактна схема має функцію провідності



$$f = x_1 \cdot \overline{x_2} + x_3 ?$$

Якщо ні, то побудуйте контактну схему, яка відповідає цієї функції.

Дайте означення логічного елемента.

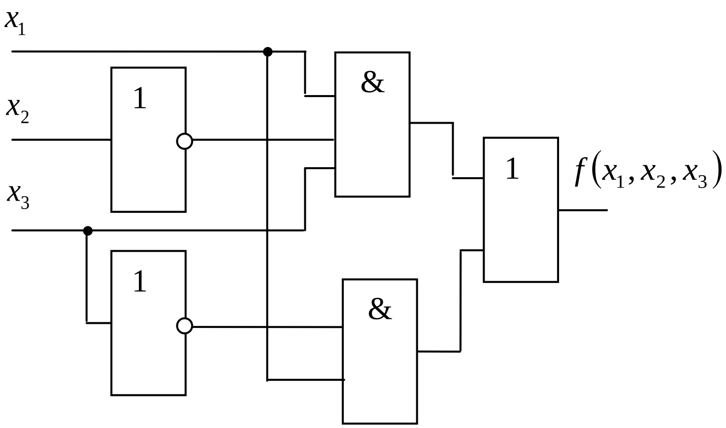
Дайте означення логічної схеми. Чи є логічні схеми різновидом комбінаційних схем?

Дайте визначення функціонально повного набору ЛЕ. Перелічіть ці набори. Які з них є мінімальними?

Зобразіть на рисунку схеми логічних елементів, які виконують функції І, Або, Не, І-Не, Або-Не. Назвіть операції алгебри логіки, що реалізують ці пристрої.

Що означає висловлення ”Елемент Або-Не має парафазний вихід”? Наведіть умовне позначення цього елемента.

Чи вірно, що ця комбінаційна схема реалізує логічну фун-



кцію $f = \overline{x_1 \cdot x_2} + \overline{x_1} \cdot x_2 + x_1 \cdot \overline{x_2 \cdot x_3}$? Якщо ні, то побудуйте комбінаційну схему, що відповідає функції *f* у тому ж самому базисі логічних елементів.

Чи вірно, що логічну функцію

$$f = x_1 \cdot x_2 + x_1 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3}$$

технічно реалізує наведена нижче комбінаційна схема ? Якщо ні, то побудуйте комбінаційну схему, що відповідає функції *f* у тому ж самому базисі логічних елементів.

2.6 Подання логічних функцій нормальними формами

Приклад 2.7 засвідчує, що одну й ту саму ЛФ описують різні логічні вирази (формули). У загальному випадку, зробивши низку еквівалентних перетворень деякої ЛФ згідно із законами алгебри логіки, можна одержати різні подання цієї ЛФ, наприклад,

$$f(x_1, x_2, x_3) = x_1\left(x_3 + \overline{x_2}\right) + x_2\left(x_3 + \overline{x_1}\right) =$$
$$= x_1x_3 + x_1\overline{x_2} + x_2x_3 + \overline{x_1}x_2 = x_3(x_1 + x_2) + \overline{x_1}x_2 + x_1\overline{x_2}.$$

Тому, розглядаючи низку ЛФ, необхідно переконатися в тому, що вони відрізняються одна від одної, щоб не виконувати зайвої роботи. У зв'язку з цим виникає потреба вибирати таку форму зображення логічних функцій, коли кожній функції відповідатиме одна і лише одна формула та навпаки, кожній формулі повинна відповідати єдина ЛФ. Такі форми зображення логічних функцій називають *канонічними* (*стандартними*).

У математичній логіці застосовуються форми логічних функцій, які називаються диз'юнктивною та кон'юнктивною *нормальними формами*. Ознайомимося докладніше із запровадженими термінами та способами одержання зображень логічних функцій такими формами. Із цією метою наведемо спочатку деякі означення.

Означення 2.12. Логічний добуток будь-якого скінченного числа змінних, взятих із запереченням чи без нього не більше ніж один раз, називається *елементарною кон'юнкцією* (елементарним добутком).

Наприклад, логічні добутки $x_1 \cdot x_3,\ x_2 \cdot \overline{x_3},\ \overline{x_1} \cdot x_2,\ x_2, \overline{x_3}$ є елементарними кон'юнкціями, а $x_1 \cdot \overline{x_1} \cdot \overline{x_2}$, $\overline{x_3 \cdot x_4}$ – ні.

Означення 2.13. Логічна функція, що подається диз'юнкцією елементарних кон'юнкцій, називається *диз'юнктивною нормальною формою* (ДНФ).

Наприклад, функція $f = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_4 + x_1 \cdot x_2 + x_3$ є диз'юнктивною нормальною формою.

Означення 2.14. Логічна сума будь-якого скінченного числа змінних, взятих із запереченням чи без нього не більше ніж один раз, називається *елементарною диз'юнкцією* (елементарною сумою).

Наприклад, логічні суми $x_1 + x_3$, $x_2 + \overline{x_3}$, $\overline{x_1} + \overline{x_2}$, x_2, $\overline{x_3}$ є елементарними диз'юнкціями, а $x_1 + \overline{x_1} + \overline{x_2}$, $\overline{x_3 + x_4}$ – ні.

Означення 2.15. Логічна функція, що подається кон'юнкцією елементарних диз'юнкцій, називається *кон'юнктивною нормальною формою* (КНФ).

Наприклад, функція $f = (x_1 + \overline{x_2} + \overline{x_3}) \cdot (\overline{x_1} + \overline{x_2} + x_4) \cdot x_3$ є кон'юнктивною нормальною формою.

Означення 2.16. *Розв'язуваною* називається формула, яка не є тотожно хибною або тотожно істинною.

Елементарні кон'юнкції (диз'юнкції) завжди розв'язувані. Це забезпечується тим, що в них кожна змінна зустрічається лише один раз із запереченням або без нього (немає випадків одночасної наявності змінної та її інверсії). Унаслідок цього жодна елементарна кон'юнкція (диз'юнкція) у ДНФ (КНФ) не може бути тотожно хибною (істинною) і тому не може бути виключеною із числа доданків (співмножників) без порушення умови рівносильності вихідної ЛФ та перетвореної у ДНФ (КНФ).

Як відмічалося раніше, можна будь-яку логічну формулу перетворити на інші, рівносильні формули. Аналогічно, будь-яку ЛФ можна подати різними, рівносильними ДНФ і КНФ. Це означає, що поданням ЛФ у вигляді ДНФ або КНФ однозначності нормальних форм ще не досягають. Способом вирішення цієї проблеми є зведення ДНФ (КНФ) до стандартної (канонічної) форми. Такі стандартні форми подання ЛФ називають *досконалими*.

Означення 2.17. Упорядкована множина (кортеж) аргументів логічної функції називається її *базою*.

Означення 2.18. *Рангом* елементарної кон'юнкції або диз'юнкції називають кількість змінних, які вона містить.

Очевидно, максимальне значення рангу елементарної кон'юнкції (диз'юнкції) дорівнює кількості аргументів ЛФ.

Означення 2.19. Якщо в елементарну кон'юнкцію (диз'юнкцію) входить кожна змінна із бази логічної функції, причому лише один раз, то вона називається *конституентою одиниці* (*нуля*).

Іншими словами, елементарна кон'юнкція (диз'юнкція) є *конституентою одиниці* (*нуля*), якщо вона має максимальний ранг, тобто містить всі змінні з бази ЛФ із запереченням чи без нього, причому не більше ніж один раз.

Конституента одиниці φ набуває значення 1 лише у тому разі, коли в ній всі змінні без заперечення та із запереченням мають значення відповідно 1 та 0. Серед 2^n наборів логічної функції n змінних немає однакових, тому можемо стверджувати, що лише на одному із цих наборів $\varphi=1$. На всіх інших наборах буде $\varphi=0$, оскільки серед співмножників кон'юнкції φ буде не менше одного зі значенням 0.

З іншого боку, кожний із наборів однозначно задає відповідну до нього конституенту одиниці φ, яка має значення 1 на даному наборі за умови, що значенню 1 (0) змінної x_i , $i\in\{1, \ldots, n\}$, у даному наборі відповідає змінна x_i ($\overline{x_i}$) у кон'юнкції φ. Побудована за таким правилом кон'юнкція φ є єдиною, що має значення 1 на даному наборі. Всі інші кон'юнкції, які можна скласти з n змінних, відрізняються від кон'юнкції φ наявністю змінної x_i замість $\overline{x_i}$ або $\overline{x_i}$ замість x_i принаймні для одного $i\in\{1, \ldots, n\}$. Унаслідок цього конституента одиниці φ', відмінна від конституенти φ, на даному наборі буде мати значення 0, оскільки вона містить принаймні один співмножник зі значенням 0.

Проведемо аналогічні міркування щодо властивостей конституент нуля. Конституента нуля ψ набуває значення 0 лише у тому випадку, коли в ній усі змінні без заперечення та з запереченням мають значення відповідно 0 та 1. Тому лише на одному з 2^n наборів логічної функції n змінних виконуються умови рівності $\psi=0$. На всіх інших наборах буде $\psi=1$, оскільки серед доданків диз'юнкції ψ буде не менше одного зі значенням 1. Кожний з 2^n наборів однозначно задає відповідну до нього конституенту нуля ψ, яка має значення 0 на даному наборі за умови, що значенню 0 (1) змінної x_i , $i\in\{1, \ldots, n\}$ у даному наборі відповідає змінна $\overline{x_i}$ (x_i) у диз'юнкції ψ. Побудована за таким правилом конституента нуля ψ є єдиною, що має значення 0 на даному наборі. Всі інші диз'юнкції, що можна скласти з n змінних, відрізняються від диз'юнкції ψ наявністю змінної x_i замість $\overline{x_i}$ або $\overline{x_i}$ замість x_i принаймі для одного $i\in\{1, \ldots, n\}$. Унаслідок цього конституента нуля ψ', відмінна від конституенти ψ, на даному наборі буде мати значення 1, оскільки вона містить принаймні один доданок зі значенням 1.

Приклад 2.8. Набору $(x_1 x_2 x_3)=(010)$ функції $f(x_1, x_2, x_3)$ відповідають єдині конституенти одиниці $\overline{x_1}\cdot x_2\cdot\overline{x_3}$ та нуля $x_1+\overline{x_2}+x_3$, що набувають відповідно значень 1 і 0 на цьому наборі. Інші конституенти одиниці та нуля на даному наборі мають значення відповідно 0 та 1.

Приклад 2.9. Конституенти $x_1\cdot x_2\cdot\overline{x_3}$ та $\overline{x_1}+\overline{x_2}+x_3$ набувають відповідно значення 1 та 0 лише на наборі $(x_1 x_2 x_3)=(110)$. На будь-якому іншому наборі трьох змінних вони будуть мати протилежні значення (відповідно 0 та 1).

Із наведених прикладів легко помітити, що конституенти одиниці та нуля, пов'язані відношенням взаємної однозначності з одним набором, *взаємно інверсні* (читачеві рекомендується довести цей факт за допомогою законів де Моргана).

Отже, кожному із 2^n наборів логічної функції n змінних відповідає єдина конституента одиниці (нуля), що має значення 1 (0) на даному наборі, і навпаки, будь-якій з конституент одиниці (нуля) відповідає єдиний із 2^n наборів, на якому ця конституента має значення 1 (0). Це означає, що існує взаємно однозначна відповідність між кожним набором логічної функції і конституентою одиниці (нуля), що має на даному наборі значення 1 (0). Відношення взаємної однозначності між конституентою одиниці (нуля) та єдиним набором $(x_1 \ldots x_n)$ логічної функції n змінних забезпечується тим, що в цій конституенті всі змінні без заперечення та із запереченням пов'язані з даним набором таким правилом: у конституенті одиниці (нуля) значенню 1 змінної x_i, $i \in \{1, \ldots, n\}$ даного набору відповідає змінна x_i ($\overline{x_i}$), а значенню 0 змінної x_i – змінна $\overline{x_i}$ (x_i).

Оскільки розв'язувана ЛФ не є тотожно хибною або істинною (як це відзначалося вище), то очевидно, що вона набуває кожне зі значень 1 і 0 не менше ніж на одному з наборів її аргументів. Це означає, враховуючи взаємно однозначну відповідність між наборами ЛФ і конституентами одиниці (нуля), які мають на цих наборах значення 1 (0), що кількість конституент одиниці (нуля) строго дорівнює кількості наборів, на яких ЛФ набуває значення 1 (0). Ця обставина наводить на думку про те, що згадану вище стандартизацію виду ДНФ (КНФ) буде досягнено, якщо цю нормальну форму подати у вигляді логічної суми (логічного добутку) всіх її конституент одиниці (нуля).

Означення 2.20. Логічна функція, що подається диз'юнкцією всіх її конституент одиниці, називається *досконалою диз'юнктивною нормальною формою* (ДДНФ).

Наприклад, функція $f = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$ трьох аргументів, яка набуває значення 1 лише на наборах (100), (001) та (111), є досконалою ДНФ.

Формула логічної функції у вигляді ДДНФ містить всі ті й лише ті доданки, які є конституентами одиниці цієї функції, надлишкових доданків немає. Цим гарантується можливість подання у ДДНФ будь-якої ЛФ, крім функції "Константа нуль" (див. п. 2.4), тому що остання має значення 0 на всіх наборах.

Якщо при поданні ЛФ у вигляді ДДНФ задати будь-який набір із числа 2^n наборів цієї функції, то її значення дорівнюватиме визначеному для цього набору. Правильне відтворення значень ЛФ для кожного з наборів забезпечується тим, що на наборі, якому відповідає задане значення 1 логічної функції, ДДНФ обов'язково містить один з доданків зі значенням 1, а на наборі, якому відповідає задане значення 0 логічної функції, всі доданки мають значення 0.

Означення 2.21. Логічна функція, що подається кон'юнкцією всіх її конституент нуля, називається *досконалою кон'юнктивною нормальною формою* (ДКНФ).

Наприклад, досконалою КНФ є функція трьох аргументів

$$f = \left(x_1 + \overline{x_2} + \overline{x_3}\right) \cdot \left(\overline{x_1} + \overline{x_2} + x_3\right) \cdot \left(x_1 + x_2 + x_3\right),$$

яка набуває значення 0 лише на наборах (011), (110) та (000).

Формула логічної функції у вигляді ДКНФ містить всі ті й лише ті співмножники, які є конституентами нуля цієї функції, надлишкових співмножників немає. Цим гарантується можливість подання у ДКНФ будь-якої ЛФ, крім функції "Константа одиниця" (див. п. 2.4), тому що остання має значення 1 на всіх наборах.

Правильне відтворення значень ЛФ для кожного з наборів забезпечується тим, що на наборі, якому відповідає задане значення 0 логічної функції, ДКНФ обов'язково містить один із співмножників зі значенням 0, а на наборі, якому відповідає задане значення 1 логічної функції, всі співмножники мають значення 1.

Саме ДДНФ і ДКНФ вважаються стандартними формами аналітичного подання логічних функцій. Підкреслимо основні властивості ДДНФ (ДКНФ):

– у ній немає двох однакових доданків (співмножників); жодний доданок (співмножник) не містить двох однакових співмножників (доданків);

– жодний доданок (співмножник) не містить деякої змінної разом з її запереченням;

– кожний доданок (співмножник) містить як співмножники (доданки) всі аргументи ЛФ, деякі або всі з яких можуть бути із запереченням.

Завдяки властивостям ДДНФ і ДКНФ розв'язується задача порівняння логічних функцій, а саме: дві ЛФ рівносильні в тому разі, коли збігаються їх досконалі нормальні форми.

Розглянемо методи одержання ДДНФ і ДКНФ логічних функцій, заданих різними способами (п. 2.4), основними з яких є задання таблицею істинності та формулою. При вербальному способі необхідно, базуючись на текстовому описі ЛФ, побудувати її таблицю істинності чи аналітичну формулу або визначити базу ЛФ та десяткові номери наборів, на яких вона набуває значення 1. Для успішного засвоєння методів одержання ДДНФ і ДКНФ достатньо проілюструвати їх на прикладах.

Приклад 2.10. Подамо у ДДНФ і ДКНФ логічну функцію $f(x_1, x_2, x_3)$ на підставі таблиці істинності (табл. 2.5).

Таблиця 2.5 – Таблиця істинності функції $f(x_1, x_2, x_3)$

№	База			Значення
	x_1	x_2	x_3	f
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0

4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

ДДНФ складають відповідно до такого правила: для кожного набору змінних, на якому ЛФ набуває значення 1, записують конституенту одиниці і всі ці конституенти одиниці об’єднують диз’юнктивно.

Задана функція (табл. 2.5) набуває значення 1 на 0, 1, 6 та 7-му наборах, а на решті наборів – значення 0. Конституенти одиниці для наборів 0, 1, 6 та 7 мають вигляд

$$\overline{x_1}\cdot\overline{x_2}\cdot\overline{x_3}\,,\quad \overline{x_1}\cdot\overline{x_2}\cdot x_3\,,\quad x_1\cdot x_2\cdot\overline{x_3}\,,\quad x_1\cdot x_2\cdot x_3\,.$$

Щоб наведені повні кон’юнкції мали значення 1 на відповідних наборах, змінні, які мають нульове значення на цих наборах, взяті із запереченням. ДДНФ заданої ЛФ має вигляд

$$f=\overline{x_1}\cdot\overline{x_2}\cdot\overline{x_3}+\overline{x_1}\cdot\overline{x_2}\cdot x_3+x_1\cdot x_2\cdot\overline{x_3}+x_1\cdot x_2\cdot x_3\,. \qquad (2.4)$$

Перевірка цієї формули за всіма наборами з табл. 2.5 доводить, що вона правильно відображає задану ЛФ.

ДКНФ заданої ЛФ можна одержати з таблиці істинності згідно з таким правилом: для кожного набору змінних, на яких функціях набуває значення 0, записують конституенту нуля і всі ці конституенти нуля об’єднують кон’юнктивно. Ті змінні, які в наборах мають значення 1, записують із запереченням.

Задана функція (табл. 2.5) набуває значення 0 на 2, 3, 4 і 5-му наборах, а на решті наборів – значення 1. Конституенти нуля для наборів 2, 3, 4 і 5 мають вигляд:

$$x_1+\overline{x_2}+x_3\,,\quad x_1+\overline{x_2}+\overline{x_3}\,,\quad \overline{x_1}+x_2+x_3\,,\quad \overline{x_1}+x_2+\overline{x_3}\,.$$

ДКНФ заданої ЛФ має вигляд

$$f=(x_1+\overline{x_2}+x_3)\cdot(x_1+\overline{x_2}+\overline{x_3})\cdot(\overline{x_1}+x_2+x_3)\cdot(\overline{x_1}+x_2+\overline{x_3}). \quad (2.5)$$

Перевірка цієї формули за всіма наборами з табл. 2.5 показує, що вона правильно відображає задану ЛФ.

Формулу ЛФ у вигляді ДКНФ легко одержати з її ДДНФ. Для цього спочатку побудуємо ДДНФ функції $\overline{f}$:

$$\overline{f}=\overline{\overline{x_1}\cdot\overline{x_2}\cdot\overline{x_3}+\overline{x_1}\cdot\overline{x_2}\cdot x_3+x_1\cdot x_2\cdot\overline{x_3}+x_1\cdot x_2\cdot x_3}\,.$$

Користуючись правилом де Моргана та правилами перетворення логічних формул, одержимо вираз функції $\overline{f}$, зведений до ДДНФ :

$$\overline{f}=\overline{x_1}\cdot x_2\cdot\overline{x_3}+\overline{x_1}\cdot x_2\cdot x_3+x_1\cdot\overline{x_2}\cdot\overline{x_3}+x_1\cdot\overline{x_2}\cdot x_3\,.$$

Тепер одержимо заперечення функції $\overline{f}$:

$$\overline{\overline{f}}=\overline{\overline{x_1}\cdot x_2\cdot\overline{x_3}+\overline{x_1}\cdot x_2\cdot x_3+x_1\cdot\overline{x_2}\cdot\overline{x_3}+x_1\cdot\overline{x_2}\cdot x_3}\,.$$

Використавши правило де Моргана, зведемо функцію $\overline{\overline{f}}$ до ДКНФ:

$$f=\overline{\overline{f}}=(x_1+\overline{x_2}+x_3)\cdot(x_1+\overline{x_2}+\overline{x_3})\cdot(\overline{x_1}+x_2+x_3)\cdot(\overline{x_1}+x_2+\overline{x_3}).$$

Як бачимо, одержана ДКНФ функції f збігається з (2.5).

Таким чином, для одержання ДКНФ логічної функції f потрібно виконати операцію подвійного заперечення виразу ДДНФ цієї функції.

Приклад 2.11. Подамо у ДДНФ логічну функцію п'яти аргументів $f=f(x_1, x_2, x_3, x_4, x_5)$, про яку відомо, що вона набуває значення 1 на наборах з десятковими номерами 14, 16 і 31, а на решті наборів – значення 0.

Складемо диз'юнкцію трьох повних кон'юнкцій та під кожною з них запишемо один із трьох наборів як двійковий еквівалент його десяткового номера:

$$x_1 \ x_2{\cdot}x_3{\cdot}\,x_4{\cdot}\,x_5 + x_1{\cdot}\,x_2{\cdot}\,x_3{\cdot}\,x_4{\cdot}\,x_5 + x_1{\cdot}\,x_2{\cdot}\,x_3{\cdot}\,x_4{\cdot}\,x_5.$$
$$0\ 1\ 1\ 1\ 0 \quad 1\ 0\ 0\ 0\ 0 \quad 1\ 1\ 1\ 1\ 1$$

Розставивши заперечення над змінними, які у цих наборах мають значення 0, одержимо результат:

$$f = \overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4 \cdot \overline{x_5} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \cdot \overline{x_5} + x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5.$$

Приклад 2.12. Подамо у ДКНФ логічну функцію п'яти аргументів $f = f(x_1, x_2, x_3, x_4, x_5)$, про яку відомо, що вона набуває значення 0 на наборах з десятковими номерами 14, 16 і 31, а на решті наборів – значення 1.

Складемо кон'юнкцію трьох повних диз'юнкцій та під кожною з них запишемо один із трьох наборів як двійковий еквівалент його десяткового номера:

$$(x_1 + x_2 + x_3 + x_4 + x_5) \cdot (x_1 + x_2 + x_3 + x_4 + x_5) \cdot (x_1 + x_2 + x_3 + x_4 + x_5).$$
$$0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$$

Розставивши заперечення над змінними, що у цих наборах мають значення 1, одержимо результат:

$$f = x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \cdot x_5 + \overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \ \overline{x_5}\ .$$

Вихідна ЛФ не завжди подається у ДНФ або КНФ. У таких випадках її перетворюють у ДНФ або КНФ, використовуючи закони алгебри логіки, що не викликає складнощів. Наступним кроком є виконання операції *розгортання* одержаної ДНФ (КНФ), результатом якої буде одержання ДДНФ (ДКНФ) вихідної ЛФ. Операція розгортання елементарної кон'юнкції (диз'юнкції) дозволяє перетворити її у диз'юнкцію (кон'юнкцію) повних кон'юнкції (диз'юнкції), які є конституентами одиниці (нуля), якщо перетворенню піддається ДНФ (КНФ). Розглянемо сутність операцій розгортання елементарних кон'юнкції та диз'юнкції на наступних прикладах.

Приклад 2.13. Нехай $x\cdot\overline{y}$ – елементарна кон'юнкція функції $f(x,y,z,u)$. Застосовуючи двічі операцію розгортання, одержимо:

$$x\cdot\overline{y}=x\cdot\overline{y}\,(z+\overline{z}\,)=x\cdot\overline{y}\cdot z+x\cdot\overline{y}\cdot\overline{z}=$$
$$=x\cdot\overline{y}\cdot z\cdot(u+\overline{u}\,)+x\cdot\overline{y}\cdot\overline{z}\cdot(u+\overline{u}\,)=$$
$$=x\cdot\overline{y}\cdot z\cdot u+x\cdot\overline{y}\cdot z\cdot\overline{u}+x\cdot\overline{y}\cdot\overline{z}\cdot u+x\cdot\overline{y}\cdot\overline{z}\cdot\overline{u}\,.$$

Приклад 2.14. Нехай $x+\overline{y}$ – елементарна диз'юнкція функції $f(x,y,z,u)$. Застосовуючи двічі операцію розгортання, одержимо:

$$x+\overline{y}=x+\overline{y}+z\cdot\overline{z}\;=(x+\overline{y}+z)\cdot(x+\overline{y}+\overline{z}\,)=$$
$$=(x+\overline{y}+z+u\cdot\overline{u}\,)\cdot(x+\overline{y}+\overline{z}+u\cdot\overline{u}\,)=$$
$$=(x+\overline{y}+z+u)\cdot(x+\overline{y}+z+\overline{u}\,)\cdot(x+\overline{y}+\overline{z}+u)\cdot(x+\overline{y}+\overline{z}+\overline{u}\,).$$

При розгортанні різні елементарні кон'юнкції (диз'юнкції) можуть утворювати одну й ту саму конституенту одиниці (нуля), одну з яких потрібно видалити як дублюючу, на підставі тотожностей $x+x=x$, $x\cdot x=x$ (законів ідемпотентності).

Приклад 2.15. Подамо у ДДНФ логічну функцію, задану формулою $f(x_1,x_2,x_3)=x_1\wedge(x_2\vee x_3)\vee\overline{x_3}$.

Можна побудувати таблицю істинности цієї функції і на її основі розв'язати поставлену задачу, як у прикладі 2.10. Однак ми скористаємось іншим методом.

Спочатку зведемо формулу $x_1\wedge(x_2\vee x_3)\vee\overline{x_3}$ до ДНФ:

$$f=x_1\wedge(x_2\vee x_3)\vee\overline{x_3}\;=x_1\wedge x_2\vee x_1\wedge x_3\vee\overline{x_3}\,.$$

Проведемо операцію розгортання цієї ДНФ:

$$f=x_1\wedge x_2\wedge(x_3\vee\overline{x_3}\,)\vee x_1\wedge(x_2\vee\overline{x_2}\,)\wedge x_3\vee(x_1\vee\overline{x_1}\,)\wedge(x_2\vee\overline{x_2}\,)\wedge\overline{x_3}=$$
$$=x_1\wedge x_2\wedge x_3\vee x_1\wedge x_2\wedge\overline{x_3}\vee x_1\wedge x_2\wedge x_3\vee x_1\wedge\overline{x_2}\wedge x_3\vee$$

$$\vee x_1 \wedge x_2 \wedge \overline{x_3} \vee x_1 \wedge \overline{x_2} \wedge \overline{x_3} \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \vee \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}\ .$$

Після видалення дублюючих доданків (із пар однакових) одержимо ДДНФ:

$$f = \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \vee x_1 \wedge \overline{x_2} \wedge \overline{x_3} \vee$$
$$\vee x_1 \wedge \overline{x_2} \wedge x_3 \vee x_1 \wedge x_2 \wedge \overline{x_3} \vee x_1 \wedge x_2 \wedge x_3. \qquad (2.6)$$

У цій формулі конституенти одиниці розміщені за десятковими номерами відповідних їм наборів, на яких функція f набуває значення 1, у послідовності 0, 2, 4, 5, 6, 7.

Читачеві може здатися зручнішимим такий метод. Запишемо вираз ДНФ $x_1 \wedge x_2 \vee x_1 \wedge x_3 \vee \overline{x_3}$ у вигляді

$$F = (1 \wedge 1 \wedge -) \vee (1 \wedge - \wedge 1) \vee (- \wedge - \wedge 0),$$

керуючись правилом: у кожній кон’юнкції в ДНФ змінна без заперечення замінюється числом 1, а змінна із запереченням – числом 0, на місцях відсутніх змінних ставляться дефіси. Одержаний вираз піддається розгортанню шляхом підстановки сум виду (1+0) замість дефісів та розкриття внутрішніх дужок, унаслідок чого одержимо:

$$F = (1 \wedge 1 \wedge (1 \vee 0)) \vee (1 \wedge (1 \vee 0) \wedge 1) \vee ((1 \vee 0) \wedge ((1 \vee 0) \wedge 0))=$$
$$= (1 \wedge 1 \wedge 1) \vee (1 \wedge 1 \wedge 0) \vee (1 \wedge 1 \wedge 1) \vee (1 \wedge 0 \wedge 1) \vee (1 \wedge 1 \wedge 0) \vee$$
$$\vee (1 \wedge 0 \wedge 0) \vee (0 \wedge 1 \wedge 0) \vee (0 \wedge 0 \wedge 0).$$

Видаливши дублюючі доданки та символи кон’юнкції, одержимо:

$$F = (000)_0 \vee (010)_2 \vee (100)_4 \vee (101)_5 \vee (110)_6 \vee (111)_7.$$

Кожний з доданків формули F являє собою набір, на якому задана функція f має значення 1, а підрядковими індексами позначені десяткові номери цих наборів.

Замінивши у формулі F нулі змінними із запереченням, а одиниці – змінними без заперечення, одержимо подання заданої логічної функції у вигляді ДДНФ (2.6).

Приклад 2.16. Подамо у ДКНФ логічну функцію, задану формулою $f(x_1, x_2, x_3) = x_1 \cdot (x_2 + x_3) + \overline{x_3}$.

Використавши закони дистрибутивності та поглинання (табл. 2.1), одержимо КНФ, що містить лише одну елементарну диз'юнкцію:

$$f = x_1 \cdot (x_2 + x_3) + \overline{x_3} = (\overline{x_3} + x_1) \cdot (\overline{x_3} + x_2 + x_3) = (x_1 + \overline{x_3}) \cdot (x_3 + \overline{x_3}) +$$
$$+ (x_1 + \overline{x_3}) \cdot x_2 = (x_1 + \overline{x_3}) + (x_1 + \overline{x_3}) \cdot x_2 = x_1 + \overline{x_3}\ .$$

Проведемо операцію розгортання та одержимо, як її результат, подання заданої функції у ДКНФ:

$$f = x_1 + \overline{x_3} = x_1 + \overline{x_3} + x_2 \overline{x_2} = \underset{(0\ \ 0\ \ 1)_1}{(x_1 + x_2 + \overline{x_3})} \cdot \underset{(0\ \ 1\ \ 1)_3}{(x_1 + \overline{x_2} + \overline{x_3})}.$$

Під кожною диз'юнкцією записано набір (із його десятковим номером), що відповідає даній конституенті нуля.

ДДНФ і ДКНФ є найбільш застосовними аналітичними формами подання ЛФ. Однак, вони містять найбільш можливе число символів у їх запису. Розглянувши ДДНФ і ДКНФ однієї ЛФ, потрібно з'ясувати, яка із цих форм простіша (має менше символів у формулі). Можна припустити, що коли таблиця істинності містить більше нульових значень, ніж одиничних, то ДДНФ простіша, і навпаки.

Практика реалізації логічних функцій ставить проблему скорочення числа символів у нормальних формах. Певною мірою ця проблема вирішується шляхом перетворення ДДНФ і ДКНФ в так звані *скорочені нормальні форми*.

Означення 2.22. Якщо функція φ має значення 0 (1) принаймні на тих наборах, на яких має значення 0 (1) інша функція f, то вона називається *імплікантою* (*імпліцентою*) функції f.

Це означення потребує пояснень. По-перше, функції φ і f мають одну й ту саму базу або база функції φ містить лише деякі з аргументів функції f. По-друге, функція φ, якщо вона є імплікантою (імпліцентою) функції f, може мати значення 0 (1) не тільки на тих наборах, на яких функція f має значення 0 (1), але й на наборах, на яких функція f має значення 1 (0).

Поставимо питання: чи є конституенти одиниці (нуля) імплікантами функції f, поданої у ДДНФ (ДКНФ)? Нагадаємо, що кожна з конституент одиниці (нуля) є логічною функцією з такою самою базою, що й сама ЛФ. Конституента одиниці (нуля) має значення 1 (0) тільки на одному з наборів, на яких ЛФ також має значення 1 (0), а на всіх інших наборах вона має значення 0 (1). Тому можемо стверджувати, що будь-яка конституента одиниці (нуля) логічної функції, поданої у ДДНФ (ДКНФ), є імплікантою (імпліцентою) цієї ЛФ. Більше того, всі доданки ДДНФ (конституенти одиниці) є імплікантами, а всі співмножники ДКНФ (конституенти нуля) є імпліцентами вихідної ЛФ.

Вихідну ЛФ можна подати як ДНФ (КНФ), доданки (співмножники) якої є логічними функціями, що носять назву елементарної кон’юнкції (диз’юнкції). Якщо база елементарної кон’юнкції (диз’юнкції) містить не всі змінні із бази вихідної ЛФ, то ця кон’юнкція (диз’юнкція) не є конституентою одиниці (нуля). Незалежно від того, чи є елементарна кон’юнкція (диз’юнкція) конституентою одиниці (нуля), завжди знайдеться такий набір (можливо, не єдиний) аргументів ЛФ, на якому ця елементарна кон’юнкція (диз’юнкція) одночасно з самою ЛФ набуває значення 0 (1). Отже, всі елементарні кон’юнкції (диз’юнкції) як доданки (співмножники) ДНФ (КНФ) є імплікантами (імпліцентами) вихідної ЛФ, поданої у ДНФ (КНФ). При цьому слід пам’ятати, що ДДНФ і ДКНФ є частковими випадками подання ЛФ у ДНФ і КНФ. Це положення ілюструють наступні приклади.

Приклад 2.17. Логічну функцію $f(x_1, x_2, x_3)$ подано у ДНФ: $f = \varphi_1+\varphi_2+\varphi_3$, де $\varphi_1 = x_1\cdot \overline{x_2} \cdot x_3$; $\varphi_2 = \overline{x_1} \cdot x_3$; $\varphi_3 = x_2 \cdot \overline{x_3}$. Складемо таблицю істинності функції f та кон'юнкцій φ_1, φ_2, φ_3.

Таблиця 2.6 – Таблиця істинності функцій f, φ_1, φ_2, φ_3

№	x_1	x_2	x_3	φ_1	φ_2	φ_3	f
0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1
2	0	1	0	0	0	1	1
3	0	1	1	0	1	0	1
4	1	0	0	0	0	0	0
5	1	0	1	1	0	0	1
6	1	1	0	0	0	1	1
7	1	1	1	0	0	0	0

Дані цієї таблиці показують, що елементарна кон'юнкція φ_1 є конституентою одиниці та імплікантою функції f, дорівнює одиниці лише на наборі №5, де f=1. Елементарні кон'юнкції φ_2 і φ_3 є імплікантами функції f, оскільки вони мають значення 0 не менше, а у даному прикладі більше ніж на одному наборі, де $f = 0$.

Приклад 2.18. Логічну функцію $f(x_1, x_2, x_3)$ подано у КНФ:

$$f = \varphi_1 \cdot \varphi_2 \cdot \varphi_3, \text{ де } \varphi_1 = x_1 + \overline{x_2} + x_3,\ \varphi_2 = \overline{x_1} + x_3,\ \varphi_3 = x_2 + \overline{x_3}.$$

Складемо таблицю істинності функції f та диз'юнкцій φ_1, φ_2, φ_3 (табл. 2.7).

Таблиця 2.7 – Таблиця істинності функцій f, φ_1, φ_2, φ_3

№	x_1	x_2	x_3	φ_1	φ_2	φ_3	f
0	0	0	0	1	1	1	1
1	0	0	1	1	1	0	0
2	0	1	0	0	1	1	0
3	0	1	1	1	1	1	1
4	1	0	0	1	0	1	0
5	1	0	1	1	1	0	0
6	1	1	0	1	0	1	0
7	1	1	1	1	1	1	1

Дані цієї таблиці показують, що елементарна диз’юнкція φ_1 є конституентою нуля та імпліцентою функції f, дорівнює нулю лише на наборі №2, де $f = 0$. Елементарні диз’юнкції φ_2 і φ_3 є імпліцентами функції f, оскільки вони мають значення 1 не менше, а у даному прикладі більше, ніж на одному наборі, де f =1.

Означення 2.23. Елементарна кон’юнкція (диз’юнкція), одержана шляхом вилучення однієї або кількох змінних із вихідної кон’юнкції (диз’юнкції), називається *власною частиною* останньої.

Наприклад, власними частинами елементарної кон’юнкції $x_1 \cdot \overline{x_2} \cdot x_3$ будуть кон’юнкції $x_1 \cdot \overline{x_2}$, $x_1 \cdot x_3$, $\overline{x_2} \cdot x_3$, x_1, $\overline{x_2}$, x_3, а власними частинами елементарної диз’юнкції $x_1 + \overline{x_2} + x_3$ будуть диз’юнкції $x_1 + \overline{x_2}$, $x_1 + x_3$, $\overline{x_2} + x_3$, x_1, $\overline{x_2}$, x_3.

Означення 2.24. Елементарні кон’юнкції (диз’юнкції), які є доданками (співмножниками) ДНФ (КНФ) за умови, що ніякі їх власні частини не є самостійними доданками (співмножниками) цієї ДНФ (КНФ), називаю*ться простими імплікантами* (*імпліцентами*).

Наприклад, елементарні кон’юнкції $x_1 \cdot \overline{x_2} \cdot x_3$ та $x_1 \cdot \overline{x_2}$ є доданками деякої ДНФ, а змінні x_1 та $\overline{x_2}$ самостійно до складу ДНФ не входять як елементарні кон’юнкції. Тоді кон’юнкція $x_1 \cdot \overline{x_2}$ буде простою імплікантою цієї ДНФ. Разом із цим елементарна кон’юнкція $x_1 \cdot \overline{x_2} \cdot x_3$ не буде простою імплікантою, оскільки вона містить кон’юнкцію $x_1 \cdot \overline{x_2}$ як власну частину. Аналогічно, якщо елементарні диз’юнкції $x_1 + \overline{x_2} + x_3$ та $x_1 + \overline{x_2}$ є співмножниками деякої КНФ, а змінні x_1 та $\overline{x_2}$ самостійно до складу КНФ не входять як елементарні диз’юнкції, то диз’юнкція $x_1 + \overline{x_2}$ буде простою імпліцентою цієї КНФ. Разом з цим елементарна диз’юнкція $x_1+\overline{x_2}+x_3$ не буде простою імпліцентою, оскільки вона містить диз’юнкцію $x_1+\overline{x_2}$ як власну частину.

Проста імпліканта (імпліцента) не склеюється (за законом неповного склеювання, табл. 2.1) із жодною іншою імплікантою (імпліцентою) у складі ДНФ (КНФ), оскільки вона не є власною частиною цих імплікант (імпліцент). Це означає, що скорочення кількості символів у ДНФ (КНФ) шляхом застосування операцій склеювання імплікант (імпліцент) неможливе, якщо вони всі є простими.

Означення 2.25. Якщо всі імпліканти (імпліценти) ДНФ (КНФ) є простими, то ця ДНФ (КНФ) називається *скороченою*.

Будь-яка логічна функція може бути подана у вигляді скороченої ДНФ (СДНФ) або скороченої КНФ (СКНФ). Цей факт утверджується наступною теоремою.

Теорема 2.1 (Теорема Квайна)**.** Якщо в ДДНФ (ДКНФ) логічної функції здійснити всі операції неповного склеювання, а потім всі операції поглинання та вилучення дублюючих членів, то буде одержана скорочена ДНФ (КНФ) цієї функції, тобто диз’юнкція (кон’юнкція) всіх її простих імплікант (імпліцент).

Доведення. Нехай функція $f(x_1,\ldots,x_n)$ подана у ДДНФ (це завжди можливо). Припустимо, що після здійснення всіх операцій неповного склеювання, а потім усіх операцій поглинання одержана ДНФ, яка містить елементарну кон’юнкцію φ, що не є простою імплікантою. Це означає, що (згідно з означенням 2.25 простої імпліканти) ДНФ містить, крім імпліканти φ, деяку її частину ν, яка є простою імплікантою функції f, тобто ДНФ містить диз’юнкцію $\varphi+\nu$, а імпліканта φ має вигляд $\varphi=\nu\cdot\mu$, де μ – деяка кон’юнкція. Тоді імпліканта φ згідно з рівністю $\varphi+\nu=\nu\cdot\mu+\nu=\nu$ поглинатиметься простою імплікантою ν. Таким чином, ДНФ складатиметься лише з простих імплікант, тобто вона буде скороченою ДНФ.

Тепер розглянемо випадок подання функції f у ДКНФ. Припустимо, що після здійснення всіх операцій неповного склеювання, а потім усіх операцій поглинання одержана КНФ, що містить елементарну диз'юнкцію φ, яка не є простою імпліцентою. Це означає, що КНФ містить, крім імпліценти φ, деяку її частину ν, яка є простою імпліцентою функції f, тобто КНФ містить кон'юнкцію $\varphi\cdot\nu$, а імпліцента φ має вигляд $\varphi=\nu+\mu$, де μ – деяка диз'юнкція. Тоді імпліцента φ, згідно з рівністю $\varphi\cdot\nu=(\nu+\mu)\cdot\nu=\nu$, поглинатиметься простою імпліцентою ν. Таким чином, КНФ складатиметься лише з простих імпліцент, тобто вона буде скороченою КНФ. Теорему доведено.

Для одержання СДНФ (СКНФ) логічної функції, поданої в довільній ДНФ (КНФ), спочатку слід перетворити її в ДДНФ (ДКНФ) шляхом розгортання вихідної ЛФ та видалення дублюючих членів, як у прикладах 2.15 і 2.16. Після цього необхідно, згідно з методом Квайна, послідовно виконати перетворення ЛФ наступними кроками.

1. У ДДНФ (ДКНФ) логічної функції $f(x_1,\ldots,x_n)$ виконують усі операції неповного склеювання конституент одиниці (нуля). При цьому слід мати на увазі, що кожна конституента одиниці (нуля) може склеюватися з кількома іншими (за різними змінними), тому після першого склеювання її не поглинають, а використовують для чергових операцій склеювання (у цьому є суть операції неповного склеювання). В результаті одержана таким чином ДНФ (КНФ) містить збережені конституенти одиниці (нуля) та імпліканти (імпліценти) рангу n–1, причому серед останніх можут бути і прості імпліканти (імпліценти).

2. Виконують поглинання імплікантами (імпліцентами) всіх конституент одиниці (нуля), що брали участь у неповно-

му склеюванні. Ці конституенти поглинаються обов'язково, оскільки кожна з них містить у своєму складі власну частину, що дорівнює одній з одержаних на першому кроці імплікант (імпліцент). Конституент одиниці (нуля), які не брали участь в операціях неповного склеювання, не можуть поглинатися, вони є простими імплікантами (імпліцентами) рангу n. Після цього проводять вилучення дублюючих членів.

3. Виконують операції неповного склеювання імплікант (імпліцент) рангу $n-1$ і наступного їх поглинання імплікантами (імпліцентами) рангу $n-2$ за аналогією з операціями за кроками 1 і 2. Ця процедура повторюється з одержанням імплікант (імпліцент) нижчих рангів доти, поки операції неповного склеювання залишаються можливими, а після проведення операцій поглинання у формулі ДНФ (КНФ) залишаться лише прості імпліканти (імпліценти), тобто буде одержано СДНФ (СКНФ).

Приклад 2.19. Перетворимо у СДНФ подану в ДДНФ логічну функцію

$$f(x_1, x_2, x_3) = x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} .$$

Спочатку виконаємо неповне склеювання кон'юнкцій 3-го рангу. Склеюванню підлягають конституенти 1 і 3 (за порядком розміщення у формулі ДДНФ) за змінною x_1, з одержанням імпліканти $x_2 \cdot x_3$; конституенти 1 і 2 за змінною x_3, з одержанням імпліканти $x_1 \cdot x_2$; конституенти 2 і 5 за змінною x_2, з одержанням імпліканти $x_1 \cdot \overline{x_3}$; конституенти 3 і 4 за змінною x_2, з одержанням імпліканти $\overline{x_1} \cdot x_3$. Кожна з конституент ДДНФ має власну частину, що дорівнює одній з одержаних імплікант 2-го рангу і тому поглинається цією імплікантою. Як результат одержимо ЛФ

$$f(x_1, x_2, x_3) = x_2 \cdot x_3 + x_1 \cdot x_2 + x_1 \cdot \overline{x_3} + \overline{x_1} \cdot x_3.$$

В одержаній формулі залишилися лише кон'юнкції 2-го рангу, які подальшому склеюванню не підлягають, тобто вони є простими імплікантами. Таким чином, одержано СДНФ вихідної ЛФ у вигляді

$$f(x_1, x_2, x_3) = x_1 \cdot x_2 + x_1 \cdot \overline{x_3} + \overline{x_1} \cdot x_3 + x_2 \cdot x_3.$$

Приклад 2.20. Перетворимо у СКНФ подану в ДКНФ логічну функцію

$$f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge$$
$$\wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee x_4) \wedge (x_1 \vee x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}).$$

Виконаємо неповне склеювання диз'юнкцій 4-го рангу. Конституента 1 (за порядком розміщення у формулі ДКНФ) не склеюється з жодною із інших 5 конституент, тому вона є простою імпліцентою. Склеюванню підлягають конституенти 2 і 5 за змінною x_1, з одержанням імпліценти $x_2 \vee x_3 \vee \overline{x_4}$; конституенти 2 і 3 за змінною x_2, з одержанням імпліценти $\overline{x_1} \vee x_3 \vee \overline{x_4}$; конституенти 3 і 6 за змінною x_3, з одержанням імпліценти $\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}$; конституенти 4 і 6 за змінною x_4, з одержанням імпліканти $\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$. Кожна з конституент 2–6 має власну частину, що дорівнює одній з одержаних імпліцент 3-го рангу і тому поглинається цією імпліцентою. В результаті одержимо логічну функцію у вигляді КНФ:

$$f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge$$
$$\wedge (\overline{x_1} \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}).$$

Перевірка можливості попарного склеювання імпліцент 2, 3 і 4 показує, що це неможливо. Це означає, що вони є простими імпліцентами, а одержана КНФ є скороченою.

Якщо вихідну ЛФ задано повною (за всіма наборами) таблицею істинності або формулою, що дозволяє побудувати пову таблицю істинності, то ця ЛФ є *повністю визначеною*. У таких випадках задану ЛФ можна шляхом відповідних перетворень подати у ДДНФ або ДКНФ і далі одержати СДНФ або СКНФ.

При вербальному описі вихідна ЛФ може бути задана як неповністю визначена (див. означення 2.5). Зведення її до повністю визначеної здійснюється присвоєнням їй довільних значень із множини {0, 1} на тих наборах, на яких вона не визначена. Зокрема, довизначення ЛФ може бути здійснене присвоєнням їй значень 0 або 1 на всіх цих наборах, якщо ми маємо намір подати повністю визначену ЛФ відповідно у ДДНФ або ДКНФ. Після цього для одержання СДНФ або СКНФ з меншою кількістю символів можна змінити деякі значення ЛФ з числа доданих для збільшення кількості склеюваних конституент, як це показано у наступному прикладі.

Приклад 2.21. Описом логічної функції $f(x_1, x_2, x_3, x_4)$ задано, що вона має значення 1 на наборах з десятковими номерами 1, 10 і 14, значення 0 на наборах з номерами 0, 9 і 13, а на інших наборах не визначена (на практиці така ситуація має місце, коли набори вхідних сигналів дискретного пристрою поділяються на дозволені та заборонені). На наборах з номерами 2 – 8, 11, 12, 15 і 16 присвоїмо ЛФ значення 0 і подамо цю функцію у ДДНФ:

$$f(x_1, x_2, x_3, x_4) = (\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4) + (x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4}) + (x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4}).$$

Неповному склеюванню підлягають лише конституенти 4-го рангу за номерами 2 і 3 (в порядку розміщення в ДДНФ) за змінною x_2, з одержанням імпліканти $x_1 \cdot x_3 \cdot \overline{x_4}$. Кожна з конституент 2 і 3 має власну частину, яка дорівнює імпліканті 3-го рангу $x_1 \cdot x_3 \cdot \overline{x_4}$ і тому поглинається цією імплікантою. Як результат одержимо СДНФ

$$f(x_1, x_2, x_3, x_4) = (\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4) + (x_1 \cdot x_3 \cdot \overline{x_4}) \qquad (2.7)$$

зі збереженням заданих значень 1 та 0 вихідної ЛФ на наборах 1, 10, 14 та 0, 9 і 13 відповідно.

Здійснимо заміну значень вихідної ЛФ з 0 на 1 на наборі за номером 3 і подамо цю функцію у ДДНФ:

$$f(x_1, x_2, x_3, x_4) = (\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4) + (\overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4) + \\ + (x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4}) + (x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4}).$$

Склеюванню підлягають конституенти 1 і 2 за змінною x_3, з одержанням імпліканти $\overline{x_1} \cdot \overline{x_2} \cdot x_4$;·конституенти 3 і 4 за змінною x_2, з одержанням імпліканти $x_1 \cdot x_3 \cdot \overline{x_4}$. Кожна з конституент має власну частину, що дорівнює одній з одержаних імплікант 3-го рангу і тому поглинається цією імплікантою. Як результат одержимо СДНФ

$$f(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_2} \cdot x_4 + x_1 \cdot x_3 \cdot \overline{x_4}. \qquad (2.8)$$

зі збереженням заданих значень 1 та 0 вихідної ЛФ на наборах 1, 10, 14 та 0, 9 і 13 відповідно.

Порівняння СДНФ (2.7) і (2.8) показує, що остання має менше символів, завдяки "вдалому" довизначенню неповністю визначеної вихідної ЛФ.

Скорочені нормальні форми не завжди є мінімальними за кількістю символів, як це бачимо з наступного прикладу.

Приклад 2.22. Нехай деяку ЛФ подано у ДДНФ:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 + \\ + \overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4} + x_1 \cdot x_2 \cdot x_3 \cdot x_4. \qquad (2.9)$$

За методом Квайна одержимо СДНФ

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot x_4 + x_1 \cdot x_2 \cdot x_3 + x_2 \cdot x_3 \cdot x_4. \qquad (2.10)$$

Деякі з простих імплікант у скороченій ДНФ (2.10) можуть з'явитися зайвими. Покажемо, що такою є імпліканта $x_2 \cdot x_3 \cdot x_4$. Для цього проведемо "випробування" функції (2.10) при виключенні з неї даної імпліканти, у результаті чого будемо мати функцію

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \cdot x_4 + x_1 \cdot x_2 \cdot x_3. \qquad (2.11)$$

Подана нижче таблиця 2.8 істинності функцій (2.9) – (2.11) показує, що формулами цих функцій визначається одна й та сама функція $f(x_1,x_2,x_3,x_4)$. Однак формула (2.11) містить значно менше символів, ніж формула (2.10), тобто імпліканта $x_2 \cdot x_3 \cdot x_4$ у СКНФ (2.10) є зайвою.

Таблиця 2.8 – Таблиця істинності функцій (2.9) – (2.11)

№	x_1	x_2	x_3	x_4	Формула (2.9)	Формула (2.10)	$\overline{x_1} \cdot x_4$	$x_1 \cdot x_2 \cdot x_3$	Формула (2.11)
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	**1**	**1**	1	0	**1**
2	0	0	1	0	0	0	0	0	0
3	0	0	1	1	**1**	**1**	1	0	**1**
4	0	1	0	0	0	0	0	0	0
5	0	1	0	1	**1**	**1**	1	0	**1**
6	0	1	1	0	0	0	0	0	0
7	0	1	1	1	**1**	**1**	1	0	**1**
8	1	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	0
10	1	0	1	0	0	0	0	0	0
11	1	0	1	1	0	0	0	0	0
12	1	1	0	0	0	0	0	0	0
13	1	1	0	1	0	0	0	0	0
14	1	1	1	0	**1**	**1**	0	1	**1**
15	1	1	1	1	**1**	**1**	0	1	**1**

Досконала та скорочена нормальні форми подання логічних функцій не є єдиними нормальними формами. Доповнимо перелік останніх ще двома їх видами.

Означення 2.26. Диз'юнкція (кон'юнкція) простих імплікант (імпліцент), жодна з яких не є зайвою, називається *тупиковою* ДНФ (КНФ) логічної функції.

Наприклад, ЛФ (2.11) з прикладу 2.22 є тупиковою ДНФ вихідної функції (2.9). У цьому читач може переконатися шляхом проведення випробувань функції (2.11) при вилученні з неї імплікант $\overline{x_1} \cdot x_4$ та $x_1 \cdot x_2 \cdot x_3$.

Кількість можливих тупикових ДНФ вихідної функції (2.9) дорівнює одиниці, у чому можна переконатись, виконавши випробування СДНФ (2.10) при вилученні з неї по черзі всіх простих імплікант $\overline{x_1} \cdot x_4$, $x_1 \cdot x_2 \cdot x_3$ та $x_2 \cdot x_3 \cdot x_4$. У загальному ж випадку логічна функція може мати декілька різних тупикових нормальних форм (ДНФ і КНФ), що відрізняються кількістю символів (у п. 2.7 ми зустрінемося з такими випадками). Це означає, що існує тупикова нормальна форма (ДНФ або КНФ) логічної функції з мінімальною кількістю символів у її формулі.

Означення 2.27. Тупикова нормальна форма логічної функції, що має мінімальну кількість символів, називається *мінімальною* (МДНФ, МКНФ).

Під час постановки задачі мінімізації кількості символів у формулі ЛФ з метою спрощення її технічної або програмної реалізації необхідно провести процедуру мінімізації ЛФ, базуючись на її поданні у вигляді СДНФ або СКНФ. Метод випробувань, застосований у прикладі 2.22, незручний через його трудомісткість, особливо при великій кількості простих імплікант (імпліцент) у складі СДНФ (СКНФ). Більш зручні методи мінімізації логічних функцій, які широко застосовуються на практиці, розглядаються у п. 2.7, 2.8.

Контрольні питання

Яку мету переслідує подання логічних функцій канонічними (стандартними) формами?

Дайте означення елементарної диз'юнкції та елементарної кон'юнкції.

Дайте означення диз’юнктивної та кон’юнктивної нормальних форм подання логічних функцій.

Які логічні формули є розв’язуваними? Якими структурними особливостями забезпечується розв’язуваність елементарних кон’юнкцій та диз’юнкцій?

Дайте означення бази логічної функції.

Дайте означення рангу елементарної кон’юнкції (диз’юнкції). Чому дорівнює максимальне значеня рангу елементарної кон’юнкції (диз’юнкції)?

Дайте означення конституенти одиниці (нуля).

Доведіть властивість конституенти одиниці (нуля), яка полягає в тому, що вона має значення 1 (0) на одному і тільки на одному наборі аргументів логічної функції.

Базуючись на законах де Моргана, доведіть, що конституенти одиниці та нуля, які пов’язані відношенням взаємної однозначності з одним набором, взаємно інверсні.

Доведіть, що кількість конституент одиниці (нуля) строго дорівнює кількості наборів, на яких логічна функція набуває значення 1 (0).

Дайте означення досконалої диз’юнктивної (кон’юнктивної) нормальної форми подання логічних функцій.

Перелічіть основні властивості досконалих нормальних форм подання логічних функцій.

За якої умови логічні функції, подані різними аналітичними формулами, є рівносильними?

Наведіть послідовність логічних операцій для одержання ДКНФ із заданої ДДНФ і навпаки, ДДНФ із ДКНФ.

Наведіть послідовність логічних операцій для одержання ДДНФ або ДКНФ логічної функції, яку задано довільною формулою, не відповідною за своєю структурою диз’юнктивній або кон’юнктивній нормальній формі.

Дайте означення імпліканти та імпліценти ЛФ.

Наведіть докази того, що конституента одиниці (нуля) логічної функції, поданої у ДДНФ (ДКНФ), є імплікантою (імпліцентою) цієї функції.

Наведіть докази того, що всі елементарні кон'юнкції (диз'юнкції) поданої у ДНФ (КНФ) логічної функції є імплікантами (імпліцентами) цієї функції.

Дайте означення власної частини кон'юнкції (диз'юнкції).

Дайте означення простої імпліканти (імпліценти) логічної функції, поданої у ДНФ (КНФ).

Чому скорочення кількості символів у ДНФ (КНФ) шляхом застосування операцій склеювання імплікант (імпліцент) неможливе, якщо вони всі є простими?

Дайте означення скороченої ДНФ (КНФ).

Наведіть докази того, що будь-яка ЛФ може бути подана у вигляді скорочених ДНФ та КНФ.

Наведіть опис процедури одержання СДНФ (СКНФ) логічної функції, заданої у вигляді довільної ДНФ (КНФ).

Яким чином здійснюється приведення неповністю визначеної ЛФ до повністю визначеної ДДНФ (ДКНФ) з подальшим перетворенням у СДНФ (СКНФ)?

Чи завжди скорочені нормальні форми логічних функцій є мінімальними за кількістю символів?

У чому полягає процедура випробувань СДНФ (СКНФ) логічної функції з метою мінімізації останньої за кількістю символів?

Дайте означення тупикової ДНФ (КНФ) логічної функції.

Чи можливі випадки існування декількох різних тупикових нормальних форм однієї ЛФ?

Дайте означення мінімальної ДНФ (КНФ).

Наведіть повний перелік нормальних форм логічних функцій. Які з них є канонічними (стандартними)?

2.7 Метод Квайна мінімізації логічних функцій

Мінімізація за *методом Квайна* передбачає подання вихідної логічної функції у ДДНФ (ДКНФ). Тому якщо функція задана довільною формулою, слід спочатку перетворити її в досконалу нормальну форму шляхом розгортання (приклади 2.15, 2.16). Після цього процедура мінімізації ЛФ виконується за наступним *алгоритмом*.

Крок 1. Перетворюють ДДНФ (ДКНФ) логічної функції у СДНФ (СКНФ), базуючись на теоремі Квайна (теорема 2.1, приклади 2.19, 2.20).

Крок 2. Будують *імплікантну* (*імпліцентну*) *таблицю* і визначають *ядро простих імплікант* (*імпліцент*). Поняття ядерної, тобто належної до ядра, простої імпліканти (імпліценти) пояснюється під час розгляду прикладів.

Крок 3. Визначають перелік конституент одиниці (нуля), що не покриваються ядерними імплікантами (імпліцентами). Поняття "*покриття*" пояснюється під час розгляду прикладів. Якщо таких конституент немає або налічується лише одна неядерна імпліканта (імпліцента), то тупикова ДНФ (КНФ) є єдиною і мінімальною. У першому випадку складовими мінімальної ДНФ (КНФ) є лише всі ядерні імпліканти (імпліценти), а у другому випадку – всі ядерні та лише одна неядерна імпліканти (імпліценти).

Якщо перелік конституент одиниці (нуля), що не покриваються ядерними імплікантами (імпліцентами), налічує не меше однієї конституенти за наявності двох або більше неядерних імплікант (імпліцент), то визначають усі можливі варіанти покриття цих конституент неядерними імплікантами (імпліцентами). Кількість тупикових ДНФ (КНФ) дорівнює кількості цих варіантів.

Крок 4. Складають формули всіх тупикових ДНФ (КНФ) і вибирають із них мінімальну за кількістю символів.

Розглянемо процедури мінімізації ЛФ на прикладах, окремо для вихідніх ДДНФ і ДКНФ.

Приклад 2.23. Логічну функцію подано у ДДНФ:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge x_4 \vee$$
$$\vee \overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4 \vee \overline{x_1} \wedge x_2 \wedge x_3 \wedge x_4 \vee x_1 \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4} \vee$$

$$\vee x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4} \vee x_1 \wedge x_2 \wedge x_3 \wedge x_4. \quad (2.12)$$

Крок 1. За методом Квайна одержимо СДНФ

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \wedge x_4 \vee x_1 \wedge x_2 \wedge x_3 \vee x_1 \wedge x_2 \wedge \overline{x_4} \vee x_2 \wedge x_3 \wedge x_4. \quad (2.13)$$

Крок 2. Для вилучення із (2.13) надлишкових (зайвих) простих імплікант побудуємо імплікантну таблицю (табл. 2.9). Чотири рядки таблиці відповідають простим імплікантам із (2.13), а сім стовпців – конституентам одиниці – доданкам функції (2.12).

Таблиця 2.9 – Імплікантна таблиця Квайна

Імплі-канти	Конституенти одиниці						
	$\overline{x_1} \wedge \overline{x_2} \wedge \wedge \overline{x_3} \wedge x_4$	$\overline{x_1} \wedge \overline{x_2} \wedge \wedge x_3 \wedge x_4$	$\overline{x_1} \wedge x_2 \wedge \wedge \overline{x_3} \wedge x_4$	$\overline{x_1} \wedge x_2 \wedge \wedge x_3 \wedge x_4$	$x_1 \wedge x_2 \wedge \wedge \overline{x_3} \wedge \overline{x_4}$	$x_1 \wedge x_2 \wedge \wedge x_3 \wedge \overline{x_4}$	$x_1 \wedge x_2 \wedge \wedge x_3 \wedge x_4$
$\overline{x_1} \wedge x_4$	×	×	×	×			
$x_1 \wedge x_2 \wedge x_3$						×	×
$x_1 \wedge x_2 \wedge \overline{x_4}$					×	×	
$x_2 \wedge x_3 \wedge x_4$				×			×

Якщо імпліканта є власною частиною конституенти, то клітинка на перетині видповідних рядка і стовпця відмічається хрестиком. По суті, хрестиком фіксується той факт, що під час перетворення ДДНФ у СДНФ імпліканта "*покриває*" відповідну до неї конституенту. Сутність покриття одною простою імплікантою єдиної або декількох конституент одиниці полягає в тому, що в результаті перетворення ДДНФ у СДНФ проста імпліканта поглинає ці конституенти (проста імпліканта набуває значення 1 на всіх тих і тільки тих наборах, на яких набувають значення 1 всі конституенти одиниці, власною частиною яких є ця імпліканта). Оскільки у процедурі перетворення ДДНФ у СДНФ здійснюються операції неповного склеювання, то одна конституента може покриватися декількома простими імплікантами.

Для кожної конституенти перевіряється, чи є вона такою, що покривається лише одною імплікантою. З табл. 2.9 бачимо, що такими є конституенти $\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4$, $\overline{x_1} \wedge \overline{x_2} \wedge x_3 \wedge x_4$, $\overline{x_1} \wedge x_2 \wedge \overline{x_3} \wedge x_4$ та $x_1 \wedge x_2 \wedge \overline{x_3} \wedge \overline{x_4}$. Імпліканти $\overline{x_1} \wedge x_4$ та $x_1 \wedge x_2 \wedge \overline{x_4}$, що покривають ці конституенти, називають ядерними, вони становлять "*ядро*" імплікант.

Крок 3. У нашому прикладі ядерні імпліканти покривають усі конституенти одиниці, крім $x_1 \wedge x_2 \wedge x_3 \wedge x_4$. Ця конституента має покриватись однією із двох неядерних імплікант: $x_1 \wedge x_2 \wedge x_3$ або $x_2 \wedge x_3 \wedge x_4$. Це означає, що кількість тупикових ДНФ дорівнює 2.

У загальному випадку, коли кількості неядерних імплікант і конституент, що не покриваються ядерними імплікантами, більше одиниці, для виявлення варіантів покриття цих конституент зручно скористатися *скороченою* (спрощеною) *імплікантною таблицею*.

Таблиця 2.10 – Скорочена імплікантна таблиця

Неядерні імпліканти	Конституенти одиниці
	$x_1 \wedge x_2 \wedge x_3 \wedge x_4$
$x_1 \cdot x_2 \cdot x_3$	×
$x_2 \cdot x_3 \cdot x_4$	×

Складання скороченої імплікантної таблиці (табл. 2.10) здійснюється шляхом вилучення із повної імплікантної таблиці (табл. 2.9) рядків, що відповідають імплікантам ядра, і стовпців, що відповідають конституентам одиниці, які покриваються імплікантами ядра.

У цьому прикладі побудова скороченої імплікантної таблиці не є необхідною. Ми навели таблицю 2.10 лише для ілюстрації принципів складання таких таблиць.

Крок 4. Очевидно, що логічна сума всіх ядерних імплікант має бути складовою частиною будь-якої тупикової ДНФ. Додаючи до неї варіативно кожну з імплікант $x_1{\cdot}x_2{\cdot}x_3$ і $x_2{\cdot}x_3{\cdot}x_4$, одержимо формули тупикових ДНФ:

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \wedge x_4 \vee x_1 \wedge x_2 \wedge x_3 \vee x_1 \wedge x_2 \wedge \overline{x_4}; \qquad (2.14)$$

$$f(x_1, x_2, x_3, x_4) = \overline{x_1} \wedge x_4 \vee x_1 \wedge x_2 \wedge \overline{x_4} \vee x_2 \wedge x_3 \vee x_4. \qquad (2.15)$$

Обидві тупикові функції (2.14) і (2.15) мають однакову кількість символів (букв, знаків диз’юнкції, кон’юнкції та заперечення), тому будь-яку з них можна бути взяти за мінімальну (ще одна особливість цього прикладу).

Формалізацію методу Квайна, орієнтовану на можливість програмної реалізації процедур мінімізації логічних функцій, запропонував Мак-Класкі. Формалізація полягає в тому, що у формулі елементарної кон’юнкції (диз’юнкції) змінна без заперечення (із запереченням) замінюється числом 1 (0), а змінна із запереченням (без заперечення) – числом 0 (1), на місцях відсутніх змінних ставляться дефіси (таку формалізацію ми вже застосовували у прикладі 2.14). Такі позначення елементарних кон’юнкцій (диз’юнкцій) називаються *узагальненими кодами*. У прикладі 2.23 табл. 2.9 із формалізованими за Мак-Класкі позначеннями конституент одиниці та простих імплікант матиме вигляд таблиці 2.11. По суті, позначення конституент одиниці заміняються двійковими номерами наборів, на яких ці конституенти мають значення 1, а в позначеннях простих імплікант дефісами позначаються вилучені розряди наборів, відповідних конституентам, що покриваються цими імплікантами. Помітимо, що напис функцій (2.12) і (2.13) також можна подати в узагальнених кодах (читач може зробити це самостійно, за аналогією з прикладом 2.14).

Таблиця 2.11 – Імплікантна таблиця Квайна–Мак-Класкі

Імплі-	Конституенти одиниці

канти	0001	0011	0101	0111	1100	1110	1111
0--1	×	×	×	×			
111-						×	×
11-0					×	×	
-111				×			×

На підставі таблиці 2.11 одержуємо тупикові ДНФ в узагальнених кодах:

$$f(x_1, x_2, x_3, x_4) = 0\text{--}1 \vee 111\text{-} \vee 11\text{-}0; \tag{2.16}$$

$$f(x_1, x_2, x_3, x_4) = 0\text{--}1 \vee 11\text{-}0 \vee \text{-}111. \tag{2.17}$$

Шляхом переходу від узагальнених кодів до позначень змінних буквами із (2.16) і (2.17) одержуються відповідно функції (2.14) і (2.15).

Приклад 2.24. Логічну функцію подано у ДКНФ:

$$f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee \overline{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3} \vee x_4) \wedge$$
$$\wedge (x_1 \vee x_2 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee x_3 \vee \overline{x_4}) \wedge$$
$$\wedge (\overline{x_1} \vee x_2 \vee \overline{x_3} \vee \overline{x_4}). \tag{2.18}$$

Крок 1. За методом Квайна одержимо СКНФ:

$$f(x_1, x_2, x_3, x_4) = (x_2 \vee x_3 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_4}) \wedge (x_1 \vee x_4). \tag{2.19}$$

Крок 2. Для вилучення із (2.19) зайвих простих імпліцент побудуємо імпліцентну таблицю (табл. 2.12). Чотири рядки таблиці відповідають простим імпліцентам із (2.19), а сім стовпців – конституентам нуля – співмножникам функції (2.18). Хрестиками відмічені покриття простими імпліцентами певних конституент нуля.

Таблиця 2.12 – Імпліцентна таблиця Квайна

Іпліценти	Конституенти нуля						
	$x_1 \vee x_2 \vee$ $\vee x_3 \vee x_4$	$x_1 \vee \overline{x_2} \vee$ $\vee x_3 \vee x_4$	$x_1 \vee \overline{x_2} \vee$ $\vee \overline{x_3} \vee x_4$	$x_1 \vee x_2 \vee$ $\vee \overline{x_3} \vee x_4$	$\overline{x_1} \vee x_2 \vee$ $\vee x_3 \vee x_4$	$\overline{x_1} \vee x_2 \vee$ $\vee x_3 \vee \overline{x_4}$	$\overline{x_1} \vee x_2 \vee$ $\vee \overline{x_3} \vee \overline{x_4}$
$x_2 \vee x_3 \vee x_4$	×				×	×	
$\overline{x_1} \vee x_2 \vee x_3$					×	×	
$\overline{x_1} \vee x_2 \vee \overline{x_4}$						×	×
$x_1 \vee x_4$	×	×	×	×			

Ядерними імпліцентами є $\overline{x_1} \vee x_2 \vee \overline{x_4}$ та $x_1 \vee x_4$.

Крок 3. Ядерні імпліценти покривають усі конституенти нуля, крім $\overline{x_1} \vee x_2 \vee x_3 \vee x_4$. Ця конституента має покриватися однією із двох неядерних імпліцент: $x_2 \vee x_3 \vee x_4$ або $\overline{x_1} \vee x_2 \vee x_3$. Це означає, що кількість тупикових КНФ дорівнює 2.

Крок 4. Додаючи до суми ядерних імпліцент варіативно кожну з імпліцент $x_2 \vee x_3 \vee x_4$ і $\overline{x_1} \vee x_2 \vee x_3$, одержимо формули двох тупикових КНФ:

$$f(x_1, x_2, x_3, x_4) = (\overline{x_1} \vee x_2 \vee \overline{x_4}) \wedge (x_1 \vee x_4) \wedge (x_2 \vee x_3 \vee x_4); \quad (2.20)$$

$$f(x_1, x_2, x_3, x_4) = (\overline{x_1} \vee x_2 \vee \overline{x_4}) \wedge (x_1 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee x_3). \quad (21)$$

Обидві тупикові функції (2.20) і (2.21) мають однакову “довжину”, однак функція (2.20) має на одиницю менше операцій заперечення, тому саме її слід взяти за мінімальну.

Рекомендуємо читачеві самостійно розв’язати задачу мінімізації функції (2.18) з використанням узагальнених кодів за Мак-Класкі.

Контрольні питання

У якій формі має бути подана ЛФ для її мінімізації за методом Квайна?

Викладіть алгоритм мінімізації ЛФ за методом Квайна.

Наведіть форму імплікантної (імпліцентної) таблиці Квайна, поясніть правила заповнення її клітинок.

Поясніть поняття покриття простими імплікантами (імпліцентами) конституент одиниці (нуля) вихідної ЛФ.

Чи може одна конституента покриватися декількома простими імплікантами?

Поясніть поняття ядра імплікант (імпліцент). Які імпліканти (імпліценти) є неядерними?

Поясніть зв'язок між кількістю неядерних імплікант (імпліцент) і кількістю тупикових ДНФ (КНФ).

У яких випадках тупикова ДНФ (КНФ) є єдиною?

Наведіть форму скороченої імплікантної (імпліцентної) таблиці Квайна, поясніть правила заповнення її клітинок. У чому полягає корисність побудови і аналізу цієї таблиці?

За якими правилами складаються формули тупикових ДНФ (КНФ) на підставі імплікантної (імпліцентної) таблиці Квайна?

У чому полягає формалізація методу Квайна мінімізації логічних функцій, запропонована Мак-Класкі?

Поясніть принципи подання елементарних кон'юнкцій (диз'юнкцій), зокрема конституент одиниці (нуля) і простих імплікант (імпліцент) узагальненими кодами.

За якими правилами складаються формули тупикових ДНФ (КНФ) на підставі імплікантної (імпліцентної) таблиці Квайна – Мак-Класкі?

За якими правилами здійснюється перехід від узагальнених кодів до натурних позначень змінних у тупиковій ДНФ (КНФ)?

2.8 Метод Вейча – Карно мінімізації логічних функцій

Розглянутий у п. 2.7 метод Квайна придатний для мінімізації логічних функцій будь-якої кількості аргументів, з можливістю програмної реалізації. У той же час при розв'язанні завдань розроблення дискретних пристроїв обробки даних, автоматики та управління ставиться задача мінімізації ЛФ невеликого числа змінних. Коли база ЛФ містить не більше 6 змінних, зручним інструментом мінімізації є метод, що базується на застосуванні *карт Карно* (діаграм або таблиць Вейча-Карно).

На карті Карно логічна функція n змінних задається спеціальною формою таблиці істинності – у вигляді розкресленого прямокутника, в якому кількість клітинок дорівнює 2^n можливих наборів у базі ЛФ. Якщо кількість змінних n парна, то кількості рядків та стовпців у таблиці дорівнюють $2^{n/2}$, якщо непарна, то кількість рядків дорівнює $2^{(n-1)/2}$, а кількість стовпців – $2^{(n+1)/2}$.

Правила складання карт Карно передбачають подання вихідної ЛФ звичайною таблицею істинності або у вигляді ДДНФ (ДКНФ), що рівносильно в сенсі можливості виявлення повного переліку конституент одиниці (нуля).

Задання ЛФ таблицею карти Карно здійснюється таким чином: якщо вихідна функцiя подається у ДДНФ (ДКНФ), то слід записати одиниці (нулі) в клітинках, що відповідають наборам, на яких вона набуває значення 1 (0), а решту клітинок залишити порожніми. Для того щоб вибір клітинок був однозначним, потрібно шляхом *розмітки таблиці* забезпечити взаємно-однозначну відповідність між 2^n клітинками таблиці та 2^n можливими n-елементними наборами в базі ЛФ.

Рядки та стовпці таблиці в карті Карно розмічаються за гранями таблиці (зліва направо та зверху вниз) символами змінних із запереченням та без нього, наборами змінних кількістю від 1 до n–1. Набори змінних на гранях таблиці розмічаються так, щоб два сусідні набори в стовпці або рядку відрізнялися значенням однієї змінної: в одному наборі вона із запереченням, а в сусідньому – без нього. При цьому вважається, що кожна клітинка таблиці, що примикає до грані таблиці, є сусідньою до клітки, яка прилягає до протилежній грані і розміщена на тій же самій горизонталі або вертикалі. Виконанням цих умов забезпечується входження кожної змінної в половину клітинок таблиці із запереченням, а в іншу половину – без заперечення. За такою розміткою кожна клітинка пов'язується відношенням взаємно-однозначної відповідності з єдиним набором із бази ЛФ, який є об'єднанням в кортеж довжиною n наборів змінних, якими помічені рядок і стовпець, на перетині яких знаходиться ця клітинка. Цією відповідністю забезпечується однозначність вибору клітинки, у який може бути записане значення ЛФ на будь-якому з наборів із бази ЛФ.

Ілюстрацією розмітки таблиці на карті Карно може служити рис. 2.6, де в клітинках таблиці записані формули кон'юнкцій для 8 наборів функції $f(x_1, x_2, x_3)$, поданої в ДДНФ, а верхніми індексами відмічені десяткові номери цих наборів.

	x_2		$\overline{x_2}$	
x_1	$x_1 \cdot x_2 \cdot \overline{x_3}$ (6)	$x_1 \cdot x_2 \cdot x_3$ (7)	$x_1 \cdot \overline{x_2} \cdot x_3$ (5)	$x_1 \cdot \overline{x_2} \cdot \overline{x_3}$ (4)
$\overline{x_1}$	$\overline{x_1} \cdot x_2 \cdot \overline{x_3}$ (2)	$\overline{x_1} \cdot x_2 \cdot x_3$ (3)	$\overline{x_1} \cdot \overline{x_2} \cdot x_3$ (1)	$\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$ (0)
	$\overline{x_3}$	x_3		$\overline{x_3}$

Рисунок 2.6 – Розмітка таблиці на карті Карно
для ДДНФ логічної функції 3 змінних

Під час задання конкретної ЛФ у ДДНФ певні кон'юнкції мають статус конституент одиниці й замість них у відповідних клітинках таблиці (рис. 2.6) записують одиниці, а решта клітинок залишаються порожніми (1-й та 4-й стовпці таблиці вважаються сусідніми). Якщо ж у клітинках, що не відповідають конституентам одиниці, проставити нулі, то ми одержимо таблицю істинності ЛФ у кон'юнктивній формі, в якій дотримується взаємно однозначна відповідність між наборами змінних ЛФ, значеннями ЛФ на цих наборах та кон’юнктивними доданками ДДНФ, включаючи конституенти одиниці та надлишкові кон’юнкції.

Під час задання ЛФ у ДКНФ у клітинках таблиці з такою самою розміткою рядків і стовпців, як на рис. 2.6, будуть записані диз’юнкції тих самих змінних (рис. 2.7).

	x_2		$\overline{x}_2$	
x_1	$x_1+x_2+\overline{x}_3$ (1)	$x_1+x_2+x_3$ (0)	$x_1+\overline{x}_2+x_3$ (2)	$x_1+\overline{x}_2+\overline{x}_3$ (3)
$\overline{x}_1$	$\overline{x}_1+x_2+\overline{x}_3$ (5)	$\overline{x}_1+x_2+x_3$ (4)	$\overline{x}_1+\overline{x}_2+x_3$ (6)	$\overline{x}_1+\overline{x}_2+\overline{x}_3$ (7)
	$\overline{x}_3$	x_3		$\overline{x}_3$

Рисунок 2.7 – Розмітка таблиці на карті Карно
для ДКНФ логічної функції 3 змінних

Для конкретної ЛФ певні диз’юнкції мають статус конституент нуля і замість них у відповідних клітинках таблиці записуються нулі, а решта клітинок залишаються порожніми. Якщо ж у клітинках, що не відповідають конституентам нуля, проставити одиниці, то ми одержимо таблицю істинності ЛФ у диз’юнктивній формі, в якій дотримується взаємнооднозначна відповідність між наборами змінних ЛФ, значеннями ЛФ на цих наборах та диз’юнктивними співмножниками ДКНФ, включаючи конституенти нуля та надлишкові диз’юнкції.

Із порівняння рис. 2.6 і 2.7 можна помітити неоднакове розміщення номерів наборів, а отже, і значень ЛФ по клітинках таблиць. Цей недолік карти Карно для ДКНФ (рис. 2.7) можна усунути, помінявши місцями диз’юнкції, що відповідають наборам за номерами 0 і 7, 1 і 6, 5 і 2, 3 і 4, в результаті чого розмітка клітинок у таблиці карти Карно стане такою, як на рис. 2.8. Тут кожна диз’юнкція є інверсією кон’юнкції з тієї самої клітинки таблиці рис. 2.6, наприклад, $\overline{x_1} + \overline{x_2} + x_3 \ ^{(6)} = \overline{x_1 \cdot x_2 \cdot \overline{x_3}}^{(6)}$ (закон де Моргана).

	x_2		$\overline{x_2}$	
x_1	$\overline{x_1} + \overline{x_2} + x_3 \ ^{(6)}$	$\overline{x_1} + \overline{x_2} + \overline{x_3} \ ^{(7)}$	$\overline{x_1} + x_2 + \overline{x_3} \ ^{(5)}$	$\overline{x_1} + x_2 + x_3 \ ^{(4)}$
$\overline{x_1}$	$x_1 + \overline{x_2} + x_3 \ ^{(2)}$	$x_1 + \overline{x_2} + \overline{x_3} \ ^{(3)}$	$x_1 + x_2 + \overline{x_3} \ ^{(1)}$	$x_1 + x_2 + x_3 \ ^{(0)}$
	$\overline{x_3}$	x_3		$\overline{x_3}$

Рисунок 2.8 – Розмітка таблиці на карті Карно для ДКНФ логічної функції 3 змінних

Правомірність заповнення клітинок таблиці за формою рис. 2.8 випливає з того, що за правилами складання конституент одиниці та нуля (див. приклад 2.10, п. 2.6) для одного й того ж самого набору вони є взаємно інверсними. Отже, за умови запису конституент одиниці та нуля за формулами, наведеними у табл. 2.6 і 2.8, можна скласти і використати в процедурі мінімізації ЛФ універсальну карту Карно, в якій клітинки таблиці, що відповідають конституентам одиниці у складі ДДНФ, заповнюються одиницями, а решта клітинок – нулями (вони відповідають конституентам нуля у складі ДКНФ).

Карта Карно, побудована за описаними правилами заповнення клітинок таблиці одиницями або (та) нулями, дає наочне уявлення про наявність серед конституент одиниці (нуля) таких, що мають однакові власні частини, тому дозволяє формально виконувати неповне склеювання і поглинання конституент одиниці (нуля) простими імплікантами (імпліцентами) без надлишкових, тобто одержувати тупикову ДНФ (КНФ).

Процедура мінімізації ЛФ, поданої у ДДНФ (ДКНФ) або таблицею істинності (що рівносильно), за допомогою побудованої карти Карно здійснюється таким чином.

Етап 1. Виконують побудову контурів, що охоплюють клітинки з одиницями (нулями), дотримуючись правил:

1) контур має бути замкненим (овальним, прямокутним або квадратним);

2) контур повинен охоплювати лише клітинки з одиницями (нулями);

3) кількість клітинок усередині контура повинна бути цілим степенем числа 2, тобто 1, 2, 4, 8, ...;

4) кожний контур повинен охопити максимально можливу кількість одиниць (нулів), оскільки чим більше одиниць (нулів) у контурі, тим коротша імпліканта (імпліцента), що покриває конституенти, відповідні клітинкам, охопленим цим контуром;

5) усі одиниці (нулі) повинні бути охоплені контурами (у зв'язку з неповним склеюванням конституент допускається перетинання контурів);

6) під час побудови контурів нижній і верхній рядки та лівий і правий стовпці таблиці вважаються сусідніми (змикаються при згортанні полотна таблиці у циліндр).

Етап 2. Одержують аналітичний вираз тупикової ДНФ (КНФ), зважаючи на таке:

а) спільною власною частиною всіх конституент одиниці (нуля), що відповідають клітинкам, охопленим контуром, визначається проста імпліканта (імпліцента), яка поглинає ці конституенти. Що більше одиниць (нулів) охоплює контур, то коротша імпліканта, яка покриває конституенти одиниці (нуля), відповідні клітинкам, охопленим контуром. Якщо контур охоплює лише одну одиницю (нуль), тобто одну клітинку, то простою імплікантою (імпліцентою) буде відповідна конституента одиниці (нуля);

б) загальна кількість імплікант (імпліцент) в тупиковій ДНФ (КНФ) буде дорівнювати числу побудованих контурів, причому зайвих імплікант (імпліцент) не буде. Якщо варіант побудови системи контурів є єдиним, то одержана тупикова ДНФ (КНФ) буде мінімальною. У протилежному разі з числа одержаних у різних варіантах побудови контурів тупикових ДНФ (КНФ) вибирають мінімальну за кількістю символів в її формулі.

Отже, вираз ЛФ у вигляді тупикової ДНФ (КНФ) записують за такими правилами:

1) кількість простих імплікант (імпліцент) дорівнює кількості контурів на карті Карно;

2) у кожній імпліканті (імпліценті) залишаються лише ті змінні охоплених відповідним контуром конституент одиниці (нуля), які є їх спільною власною частиною, а ті змінні, за якими ці конституенти склеюються, до імплікант (імпліцент) не входять.

Помітимо, що метод мінімізації за допомогою карт Карно може застосовуватися також для неповністю (частково) визначених ЛФ, поданих у ДНФ (КНФ), і мінімізувати їх шляхом "вдалого" довизначення. У цьому разі будуються також контури, що містять хоча б одну одиницю (нуль) і до яких входять, шляхом довизначення, додаткові одиниці (нулі).

Оскільки застосування карт Карно розглядається щодо задач мінімізації ЛФ, база яких містить невелику (не більше ніж 6) змінних, то найкращим способом засвоєння "технології" побудови карт Карно та розв'язання задач мінімізації ЛФ цим методом є розгляд зразкових прикладів. Для більшої наочності аргументи ЛФ будемо позначати прописними буквами *A*, *B*, *C*, *D*, *G*, *V*. Почнемо із простішого випадку, коли ставиться задача мінімізації логічної функції двох змінних.

Приклад 2.25. Поставимо завдання мінімізації ЛФ двох змінних, яку задано таблицею істинності (табл. 2.13).

Таблиця 2.13 – Таблиця істинності функції $F(A, B)$

№	A	B	$F(A,B)$
0	0	0	1

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

Із табл. 2.13 випливає подання функції у ДДНФ та ДКНФ:

$$F_{\text{ДДНФ}}(A,B)=\overline{A}\cdot\overline{B}+\overline{A}\cdot B+A\cdot\overline{B}, \qquad (2.22)$$

$$F_{\text{ДКНФ}}(A,B)=(\,\overline{A}+\overline{B}\,). \qquad (2.23)$$

Карту Карно для функції (2.22) подано на рис. 2.9. Як бачимо, на карті побудовані два контури, якими охоплені всі ті клітинки, в яких записані значення 1 функції (2.22) на наборах № 0, 1, 2.

Щоб знайти просту імпліканту, що покриває конституенти одиниці, охоплені контуром, необхідно з'ясувати, за якою змінною ці конституенти склеюються. Оскільки вертикальний контур охоплює рядки з A та $\overline{A}$, то змінна A не входить до простої імпліканту, що покриває конституенти $A\cdot\overline{B}$ та $\overline{A}\cdot\overline{B}$ даного контура (за нею склеюються ці конститенти).

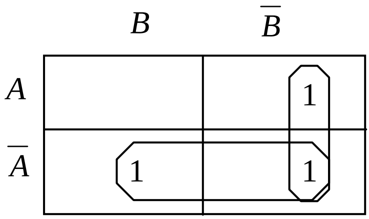


Рисунок 2.9 – Карта Карно функції $F_{\text{ДДНФ}}(A,B)$

Міркуючи аналогічно, з'ясовуємо, що до простої імпліканти, яка покриває конституенти $\overline{A}\cdot B$ та $\overline{A}\cdot\overline{B}$ горизонтального контура, змінна B входити не буде. Отже, мінімальна ДНФ логічної функції, що задана таблицею істинності (табл. 2.13) або ДДНФ (2.22), має вигляд

$$F_{\text{ДНФ}}^{min}(A,B)=\overline{A}+\overline{B}\,. \qquad (2.24)$$

Із практичної точки зору розв'язувати задачу мінімізації функції $F(A,B)$ в КНФ у цьому прикладі немає сенсу, оскільки вона має значення 0 лише на одному наборі (№3, табл. 2.13). Тому її аналітична формула (2.23) містить лише одну конституенту нуля $\overline{A}+\overline{B}$, яка з очевидністю є простою імпліцентою. Отже, мінімальна КНФ логічної функції, що задана таблицею істинності (табл. 2.13) або ДКНФ (2.23), має вигляд

$$F_{\text{КНФ}}^{min}(A,B)=\overline{A}+\overline{B}\,. \tag{2.25}$$

При формальному підході до розв'язання задачі мінімізації функції $F(A,B)$ у КНФ ми одержимо карту Карно, подану на рис. 2.10. На цій карті значення 0 функції (2.23) записане лише в одній клітинці на перетині рядка і стовпця, розмічених змінними $\overline{A}$ і $\overline{B}$. Єдиний контур охоплює саме цю клітинку, тобто єдину конституенту нуля $\overline{A}+\overline{B}$, яка із жодною іншою конституентою не склеюється (у даному випадку інших конституент нуля не існує).

Отже, діючи формально за методом Вейча – Карно, ми одержимо результат у вигляді (2.25).

У рівносильності функцій (2.22) – (2.25) читач може переконатись, побудувавши для них спільну таблицю істинності.

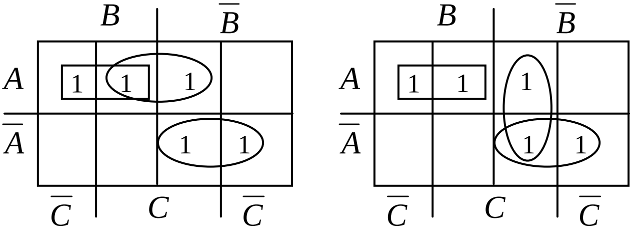
	B	$\overline{B}$
A		
$\overline{A}$		0

Рисунок 2.10 – Карта Карно функції $F_{\text{ДКНФ}}(A,B)$

Приклад 2.26. Знайдемо мінімальну ДНФ логічної функції трьох змінних, заданої в ДДНФ:

$$F(A,B,C)=A\cdot B\cdot\overline{C}+A\cdot B\cdot C+A\cdot\overline{B}\cdot C+\overline{A}\cdot\overline{B}\cdot\overline{C}+\overline{A}\cdot\overline{B}\cdot C \tag{2.26}$$

Карта Карно функції (2.26) подається у двох варіантах (рис. 2.11*а*, *б*).



а) *б)*

Рисунок 2.11 – Карти Карно функції (2.26)

Згідно зі схемами розміщення контурів на картах Карно рис. 2.11*а* і 2.11*б* одержуємо дві тупикові ДНФ:
за таблицею рис. 2.11*а*

$$F_{\text{ДНФ}}^{\text{туп}}(A, B, C) = A \cdot B + A \cdot C + \overline{A} \cdot \overline{B}\,; \tag{2.27}$$

за таблицею рис. 2.11*б*

$$F_{\text{ДНФ}}^{\text{туп}}(A, B, C) = A \cdot B + \overline{B} \cdot C + \overline{A} \cdot \overline{B}\,. \tag{2.28}$$

Очевидно, за мінімальну ДНФ слід взяти функцію (2.27) із меншою кількістю символів, ніж (2.28).

Приклад 2.27. Знайдемо мінімальну КНФ функції з прикладу 2.26. Для цього подамо її у ДКНФ:

$$F(A, B, C) = (A + \overline{B} + C) \cdot (A + \overline{B} + \overline{C}) \cdot (\overline{A} + B + C). \tag{2.29}$$

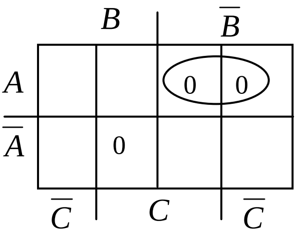


Рисунок 2.12 – Карта Карно функції (2.29)

Це можна зробити, взявши подвійне заперечення формули (2.26) за аналогією з прикладом 2.10 або з таблиці істинності ЛФ, побудованої за її заданою ДДНФ (2.26).

Карту Карно функції (2.29) подано на рис. 2.12. Конституенти нуля $A+\overline{B}+C$ і $A+\overline{B}+\overline{C}$, яким відповідають охоплені контуром дві клітинки з нулями, склеюються за змінною C і покриваються простою імпліцентою $A+\overline{B}$, а конституента нуля $\overline{A}+B+C$ сама є простою імпліцентою (відповідає клітинці з нулем у нижньому рядку таблиці).

Таким чином, мінімальна КНФ заданої логічної функції має вигляд

$$F^{min}_{\text{КНФ}}(A,B,C)=(A+\overline{B})\cdot(\overline{A}+B+C).$$

Приклад 2.28. Поставимо задачу мінімізації логічної функції трьох змінних, заданої в ДДНФ:

$$F(A,B,C)=\overline{A}\cdot\overline{B}\cdot\overline{C}+A\cdot\overline{B}\cdot\overline{C}+A\cdot\overline{B}\cdot C+\overline{A}\cdot\overline{B}\cdot C. \qquad (2.30)$$

Побудуємо карту Карно для цієї функції з розміткою всіх клітинок її значеннями на відповідних наборах (рис. 2.13).

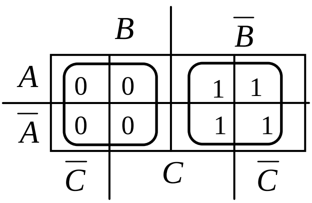


Рисунок 2.13 – Карта Карно функції (2.30)

Чотирьом конституентам одиниці із (2.30) відповідають 4 клітинки (помічені одиницями), що охоплюються одним контуром. Їх спільною власною частиною є змінна $\overline{B}$. Вона є імплікантою (рангу 1), що покриває ці конституенти.

Таким чином, мінімальна ДНФ має вигляд

$$F^{min}_{\text{ДНФ}}(A,B,C)=\overline{B}.$$

Клітинки таблиці, не помічені одиницями, відповідають конституентам нуля $\overline{A}+\overline{B}+C$, $\overline{A}+\overline{B}+\overline{C}$, $A+\overline{B}+C$ і $A+\overline{B}+\overline{C}$ (інверсіям кон'юнкцій $A\cdot B\cdot\overline{C}$, $A\cdot B\cdot C$, $\overline{A}\cdot B\cdot\overline{C}$ і $\overline{A}\cdot B\cdot C$). Вони помічені нулями й охоплені одним контуром. Спільною власною частиною цих конституент нуля є змінна $\overline{B}$, що покриває ці конституенти. Тому мінімальна КНФ має вигляд

$$F^{min}_{\text{КНФ}}(A, B, C) = \overline{B}\ .$$

Збіг формул мінімальних ДНФ і КНФ у загальному випадку не є обов'язковим. У цьому прикладі збіг обумовлений тим, що ці формули одноелементні при їх рівносильності.

Приклад 2.29. Знайдемо мінімальні ДНФ і КНФ логічної функції трьох змінних, заданої в ДДНФ:

$$F(A, B, C) = A{\cdot}B{\cdot}\overline{C} + \overline{A}{\cdot}B{\cdot}\overline{C} + \overline{A}{\cdot}B{\cdot}C + A{\cdot}\overline{B}{\cdot}C +$$
$$+ A{\cdot}\overline{B}{\cdot}\overline{C} + \overline{A}{\cdot}\overline{B}{\cdot}C. \qquad (2.31)$$

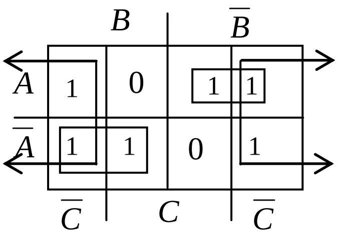


Рисунок 2.14 – Карта Карно функції (2.31)

Карту Карно для функції (2.31) подано на рис. 2.14. Чотири конституенти одиниці, яким відповідають 4 клітинки 1-го і 4-го стовпців, охоплені одним контуром, покриваються змінною (простою імплікантою) $\overline{C}$. Ще 2 клітинки у 2-му та 3-му стовпцях, що помічені одиницями, охоплюються контурами з поміченими одиницями клітинками нижнього та верхнього рядків. Конституенти одиниці, що відповідають клітинкам 1-го і 2-го стовпців нижнього рядка, покриваються простою імплікантою $\overline{A}{\cdot}B$. Конституенти одиниці, що відповідають кліткам 3-го і 4-го стовпців верхнього рядка, покриваються простою імплікантою $A{\cdot}\overline{B}$. У результаті одержуємо мінімальну ДНФ

$$F^{min}_{\text{ДНФ}}(A, B, C) = \overline{C} + \overline{A}{\cdot}B + A{\cdot}\overline{B}\ .$$

Помічені нулями конституенти нуля $\overline{A} + \overline{B} + \overline{C}$ та $A + B + \overline{C}$ є простими імпліцентами мінімальної КНФ:

$$F^{min}_{\text{КНФ}}(A,B,C) = (A + B + \overline{C})\cdot(\overline{A} + \overline{B} + \overline{C}).$$

Як бачимо, мінімальна ДНФ має менше символів у формулі, ніж мінімальна КНФ.

Приклад 2.30. Знайдемо мінімальну ДНФ частково визначеної логічної функції трьох змінних, заданої таблицею істинності на карті Карно (рис. 2.15). Дефісами помічені клітинки, що відповідають наборам, на яких значення ЛФ не визначені.

Якщо зробити довизначення ЛФ нульовими значеннями на всіх цих наборах, то ДДНФ цієї функції матиме вигляд

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

а з таким довизначенням, як показано на рис. 2.16, ДДНФ матиме вигляд

$$F(A,B,C)=A\cdot\overline{B}\cdot\overline{C}+\overline{A}\cdot\overline{B}\cdot C+A\cdot\overline{B}\cdot C. \qquad (2.32)$$

	B, $\overline{C}$	B, C	$\overline{B}$, C	$\overline{B}$, $\overline{C}$
A	–	0	–	1
$\overline{A}$	0	–	1	0

Рисунок 2.15 – Карта Карно частково визначеної функції

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

Рисунок 2.16 – Карта Карно довизначеної функції (2.32)

Охопивши пари сусідніх одиниць контурами, як показано на рис. 2.16, одержимо мінімальну ДНФ:

$$F^{min}_{\text{ДНФ}}(A,B,C)=A\cdot\overline{B}+\overline{B}\cdot C.$$

Надамо можливість читачеві самостійно одержати мінімальну КНФ функції (2.32), довизначеної її нульовими значеннями згідно з рис. 2.16.

Приклад 2.31. Знайдемо мінімальну ДНФ логічної функції чотирьох змінних

$$F(A,B,C,D)=A\cdot B\cdot\overline{C}\cdot\overline{D}+$$
$$+A\cdot B\cdot C\cdot\overline{D}+\overline{A}\cdot B\cdot C\cdot\overline{D}+$$
$$+\overline{A}\cdot\overline{B}\cdot C\cdot D+A\cdot\overline{B}\cdot C\cdot\overline{D}+$$
$$+\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D}. \qquad (2.33)$$

Карту Карно функції (2.33) подано на рис. 2.17. Як випливає з розмітки контурів, мінімальна ДНФ має вигляд

$$F^{min}_{ДНФ}(A,B,C,D)=C\cdot\overline{D}+$$
$$+A\cdot B\cdot\overline{D}+\overline{A}\cdot\overline{B}\cdot C. \qquad (2.34)$$

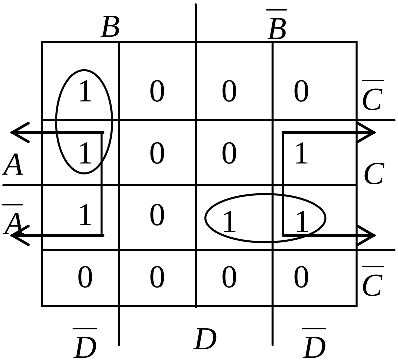


Рисунок 2.17 – Карта Карно функції (2.33)

Спосіб розмітки рядків і стовпців таблиці на картах Карно, що ми використовували у прикладах 2.25 – 2.31, не є єдиним із можливих. Зручнішим, особливо в задачах мінімізації функцій 4 – 6 змінних, є спосіб, що передбачає розмітку рядків і (або) стовпців таблиці наборами з двох (можливо й більше) змінних. Наприклад, карта Карно функції (2.33) із такою розміткою матиме вигляд рис.2. 18*a*.

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	0	1	1
$\overline{A}B$	0	0	0	1
AB	1	0	0	1
$A\overline{B}$	0	0	0	1

а)

AB \ CD	00	01	11	10
00	0	0	1	1
01	0	0	0	1
11	1	0	0	1
10	0	0	0	1

б)

Рисунок 2.18 – Карти Карно функції (2.33)

Із карти Карно рис.2.18а нескладно помітити, що охоплені контурами клітинки визначають ті самі кон'юнктивні доданки, що складають мінімальну ДНФ (2.34). Модифікація такої розмітки, що показана на рис. 2.18*б*, відрізняється тим, що набори змінних позначені їх двійковими кодами.

Результатом розв’язання задачі мінімізації ЛФ у разі застосування карти Карно з кодовою розміткою таблиці буде її подання в узагальнених кодах. У прикладі 2.31, застосовуючи кодову розмітку таблиці в карті Карно, буде одержано вираз мінімальної ДНФ:

$$F_{\text{ДНФ}}^{min}(A,B,C,D) = (\text{--}10) + (11\text{-}0) + (001\text{-}), \qquad (2.35)$$

що при інтерпретації в буквених позначеннях змінних збігається з (2.34).

Застосування розмітки таблиці в карті Карно наборами змінних особливо зручне під час розв’язування задач мінімізації функцій 5 і 6 змінних, як це показано у наступних прикладах.

Приклад 2.32. Знайдемо мінімальну ДНФ логічної функції 5 змінних

$$F(A,B,C,D,G) = A\cdot B\cdot \overline{C}\cdot \overline{D}\cdot \overline{G} + A\cdot B\cdot C\cdot D\cdot G + A\cdot B\cdot \overline{C}\cdot D\cdot G +$$
$$+ A\cdot B\cdot \overline{C}\cdot \overline{D}\cdot G + \overline{A}\cdot B\cdot \overline{C}\cdot \overline{D}\cdot \overline{G} + \overline{A}\cdot B\cdot C\cdot D\cdot G + \overline{A}\cdot B\cdot \overline{C}\cdot D\cdot G +$$
$$+ \overline{A}\cdot B\cdot \overline{C}\cdot \overline{D}\cdot G + A\cdot B\cdot C\cdot D\cdot \overline{G} + \overline{A}\cdot B\cdot C\cdot D\cdot \overline{G} + A\cdot \overline{B}\cdot C\cdot D\cdot \overline{G}\,. \qquad (2.36)$$

Карту Карно функції (2.36) подано на рис. 2.19.

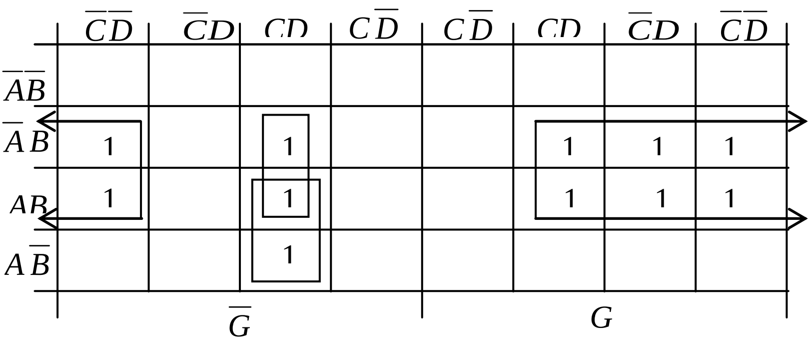


Рисунок 2.19 – Карта Карно функції (2.36)

Із розмітки контурів випливає, що перші 8 конституент одиниці з (2.36) склеюються за змінними *A, C, D, G* і покриваються простою імплікантою *B*. Конституенти $A{\cdot}B{\cdot}C{\cdot}D{\cdot}\overline{G}$ і $\overline{A}{\cdot}B{\cdot}C{\cdot}D{\cdot}\overline{G}$ склеюються за змінною *A* (неповне склеювання) і покриваються простою імплікантою $B{\cdot}C{\cdot}D{\cdot}\overline{G}$. Конституенти $\overline{A}{\cdot}B{\cdot}C{\cdot}D{\cdot}\overline{G}$ і $A{\cdot}\overline{B}{\cdot}C{\cdot}D{\cdot}\overline{G}$ склеюються за змінною *B* і покриваються простою імплікантою $A{\cdot}C{\cdot}D{\cdot}\overline{G}$. В результаті одержуємо мінімальну ДНФ:

$$F^{min}_{\text{ДНФ}}(A,B,C,D) = B + B{\cdot}C{\cdot}D{\cdot}\overline{G} + A{\cdot}C{\cdot}D{\cdot}\overline{G}.$$

Приклад 2.33. Знайдемо мінімальну ДНФ логічної функції 6 змінних

$$\begin{aligned} F(A,B,C,D,G,V) = {} & \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}\cdot\overline{G}\cdot\overline{V} + \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}\cdot\overline{G}\cdot V + \\ & + \overline{A}{\cdot}B{\cdot}\overline{C}{\cdot}D{\cdot}G{\cdot}\overline{V} + \overline{A}{\cdot}B{\cdot}\overline{C}{\cdot}D{\cdot}\overline{G}{\cdot}\overline{V} + A{\cdot}B{\cdot}\overline{C}{\cdot}D{\cdot}G{\cdot}\overline{V} + \\ & + A{\cdot}B{\cdot}\overline{C}{\cdot}D{\cdot}G{\cdot}V + A{\cdot}\overline{B}{\cdot}C{\cdot}\overline{D}{\cdot}G{\cdot}\overline{V} + A{\cdot}\overline{B}{\cdot}C{\cdot}\overline{D}{\cdot}\overline{G}{\cdot}\overline{V} + \\ & + A{\cdot}B{\cdot}C{\cdot}\overline{D}{\cdot}G{\cdot}\overline{V} + A{\cdot}B{\cdot}C{\cdot}\overline{D}{\cdot}\overline{G}{\cdot}\overline{V}. \end{aligned} \qquad (2.37)$$

Карту Карно функції (2.37) побудуємо з розміткою рядків і стовпців таблиці (рис. 2.20) наборами змінних у двійкових кодах (змінні без заперечень із (2.37) заміняються одиницями, а змінні із запереченнями – нулями).

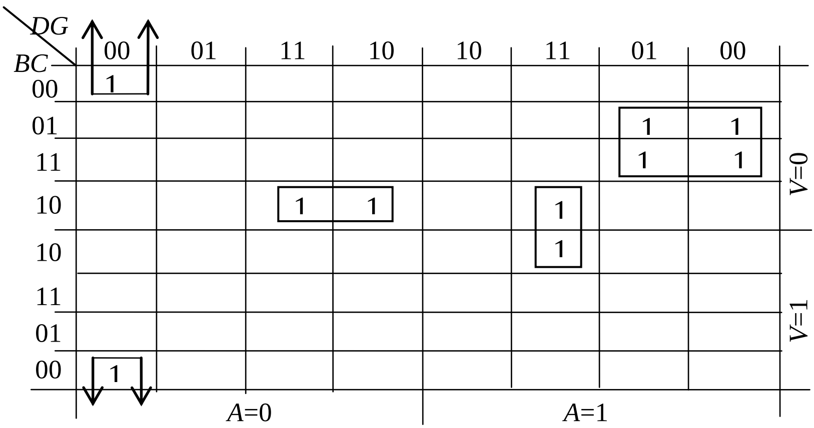


Рисунок 2.20 – Карта Карно функції (2.37)

Із розмітки контурів випливає, що мінімальна ДНФ в узагальнених кодах має вигляд

$$F_{\text{ДНФ}}^{min}(A, B, C, D, G, V) = (00000\text{-}) + (0101\text{-}0) + (11011\text{-}) + (1\text{-}10\text{-}0),$$

а при застосуванні позначень змінних буквами

$$F_{\text{ДНФ}}^{min}(A, B, C, D, G, V) = (\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{G}) + (\overline{A} \cdot B \cdot \overline{C} \cdot D \cdot \overline{V}) +$$

$$+ (A \cdot B \cdot \overline{C} \cdot D \cdot G) + (A \cdot C \cdot D \cdot \overline{V}).$$

Отже, карту Карно можна оформлювати з будь-якою розміткою рядків і стовпців таблиці, проте лише так, щоб кожна клітинка відповідала певному (і єдиному) набору з бази ЛФ і щоб сусідні набори, как по горизонталі так і по вертикалі, відрізнялися між собою тільки значенням однієї змінної: в одній клітинці вона із запереченням, а в сусідній – без нього. При цьому слід мати на увазі, що клітинки, розміщені на протилежних гранях таблиці у кожному рядку або стовпці, також є сусідніми.

Контрольні питання

Скільки рядків і стовпців містить таблиця істинності логічної функції *n* змінних на карті Карно, якщо число *n* парне (непарне)?

Назвіть форми подання ЛФ, що підлягає мінімізації за допомогою карти Карно.

Яким чином здійснюється задання логічної функції таблицею на карті Карно?

Яким чином забезпечується однозначність вибору клітинок в таблиці карти Карно, що помічаються одиницями (нулями)?

Викладіть правила розмітки рядків і стовпців таблиці на карті Карно.

Побудуйте шаблон карти Карно для ЛФ двох (трьох) змінних, поданої в ДДНФ, з розміткою рядків і стовпців таблиці.

У клітинках таблиці запишіть логічні вирази всіх можливих конституент одиниці. Яким чином на базі цього шаблона задати таблицю істинності конкретної ЛФ?

Яким чином помічаються клітинки таблиці карти Карно, для ЛФ, поданої у ДДНФ або звичайною таблицею істинності та мінімізованої в ДНФ?

Побудуйте шаблон карти Карно для ЛФ двох (трьох) змінних, поданої в ДКНФ, з розміткою рядків і стовпців таблиці. У клітинках таблиці запишіть логічні вирази всіх можливих конституент нуля. Чи відрізняється розмітка рядків і стовпців від такої як для ЛФ, поданої в ДДНФ?

Якими символами та за якими правилами помічаються клітинки таблиці карти Карно для ЛФ, поданої звичайною таблицею істинности та мінімізованої в КНФ?

Викладіть правила побудови універсальної карти Карно, придатної для розв’язання задач мінімізації ЛФ як у ДНФ, так і КНФ.

Викладіть процедуру мінімізації ЛФ, поданої у ДДНФ (ДКНФ) або звичайною таблицею істинності за допомогою карти Карно.

Викладіть правила побудови контурів, що охоплюють клітинки з одиницями (нулями) в таблиці карти Карно. З якою метою це робиться? Чи можлива багатоваріантність множини контурів?

Викладіть правила одержання формули ЛФ у вигляді мінімальної ДНФ (КНФ) на підставі карти Карно з вибраною множиною контурів, якими охоплюються клітинки таблиці, що помічені одиницями (нулями).

Чи може метод мінімізації за допомогою карт Карно застосовуватися для неповністю визначених ЛФ?

Базуючись на аналізі прикладів 2.25 – 2.31, відтворіть шаблони карт Карно для функцій двох, трьох і чотирьох змінних з розміткою рядків і стовпців таблиць змінними з бази ЛФ.

Базуючись на аналізі прикладів 2.31 – 2.33, відтворіть шаблони карт Карно для функцій 4, 5 і 6 змінних із розміткою рядків і стовпців наборами змінних із бази ЛФ з позначеннями змінних буквами та двійковими кодами.

2.9 Вправи

1. Визначте, які з наведених речень є висловленнями:

а) $2 > 3$; б) $10 > 5 > 3$; в) $5 < 6$; г) $13 < 13$?, д) $x = 10$.

Чи є ці висловлення простими?

2. Визначте істинність таких висловлень:

а) 28 ділиться на 3 та 4; б) 100 ділиться на 3 або 100 ділиться на 4 (“або” не виключне). Чи є ци висловлення простими (складними)?

3. Знайдіть серед зазначених нижче речень висловлення, вкажіть значення їх їстинності.

а) Яка година?

б) Ціле число 1 є щонайменше додатним цілим числом.

в) Якщо $x = 3$, то $x^2 = 6$.

г) Бережися автомобіля!

4. Знайдіть серед зазначених нижче речень висловлення, вкажіть значення їх їстинності.

а) Усі парні числа діляться на 2.

б) Завантажте пакети в машину.

в) Це твердження не може бути істинним.

г) Юпітер — найближча до Сонця планета.

д) Не варто зберігати компакт-диски в мікрохвильовій печі.

5. Які з висловлень є істинними?

а) Якщо $2^2 = 4$, то $3^2 = 9$.

б) Якщо $2^2 = 5$, то $3^2 = 9$.

в) Якщо $2^2 = 5$, то $3^2 = 10$.

г) Якщо $2^2 = 4$, то $3^2 = 10$.

6. Нехай x = “100 ділиться на 4”. Сформулюйте мовні висловлення y, що є результатами логічних операцій $y = f(x)$ над

висловленням x, та визначте значення істинності висловлень y, якщо: а) $f(x)=0$; б) $f(x)=1$; в) $f(x)=\overline{x}$; г) $f(x)=x$. Які значення істинності матимуть ці висловленя y як результати тих самих операцій, якщо $x=$ “100 ділиться на 3”?

7. Нехай A і B – деяки істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=A\wedge B$; б) $Y=\overline{A}\wedge\overline{B}$; в) $Y=\overline{A}\wedge B$; г) $Y=A\wedge\overline{B}$.

8. Нехай A і B – деяки істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=A\vee B$; б) $Y=\overline{A}\vee\overline{B}$; в) $Y=\overline{A}\vee B$; г) $Y=A\vee\overline{B}$.

9. Нехай A і B – істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=A\to B$; б) $Y=\overline{A}\to\overline{B}$; в) $Y=\overline{A}\to B$; г) $Y=A\to\overline{B}$.

10. Нехай A і B – істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=\overline{A\to B}$; б) $Y=\overline{\overline{\mathrm{A}}\to\overline{B}}$; в) $Y=\overline{\overline{\mathrm{A}}\to B}$; г) $Y=\overline{\mathrm{A}\to\overline{B}}$.

11. Нехай A і B – істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=B\Delta A$; б) $Y=\overline{B}\Delta\overline{A}$; в) $Y=\overline{B}\Delta A$; г) $Y=B\Delta\overline{A}$.

12. Нехай A і B – істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=A\sim B$; б) $Y=\overline{A}\sim\overline{B}$; в) $Y=\overline{A}\sim B$; г) $Y=A\sim\overline{B}$.

13. Нехай A і B – істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=A\oplus B$; б) $Y=\overline{A}\oplus\overline{B}$; в) $Y=\overline{A}\oplus B$; г) $Y=A\oplus\overline{B}$.

14. Нехай A і B – істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=A\downarrow B$; б) $Y=\overline{A}\downarrow\overline{B}$; в) $Y=\overline{A}\downarrow B$; г) $Y=A\downarrow\overline{B}$.

15. Нехай A і B – істинні висловлення (логічні змінні). Визначте значення істинності складних висловлень Y, якщо: а) $Y=A|B$; б) $Y=\overline{A}\,|\,\overline{B}$; в) $Y=\overline{A}\,|B$; г) $Y=A|\,\overline{B}$.

16. Виконати операції $x \vee y$, $x \wedge y$, $x \oplus y$ і $\overline{x}$ над двійковими числами x=1101001 та y=100110.

17. За допомогою таблиць істинності перевірити такі рівності (тотожності):

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

$\overline{x} \oplus \overline{y} = x \oplus y$; $x \oplus 0 = x$; $x \oplus 1 = \overline{x}$.

18. Знайти x та y із рівнянь: а) $(1 \rightarrow x) \rightarrow y = 0$; б) $x \vee y = \overline{x}$ (можна скористатися таблицями істинності).

19. Логічні функції f_1 і f_2 задані векторами їх значень:

f_1= (0,0,1,1,0,0,1,1,); f_2 = (0,0,1,1,1,1,0,0).

Пояснення: вектор значень ЛФ –це кортеж значень, упорядкований за зростанням номера (двійкового або десяткового) набору значень аргументів функції.

Скільки змінних містять бази цих функцій?

Побудуйте таблиці істинності цих функцій.

Чи є серед аргументів цих функцій фіктивні?

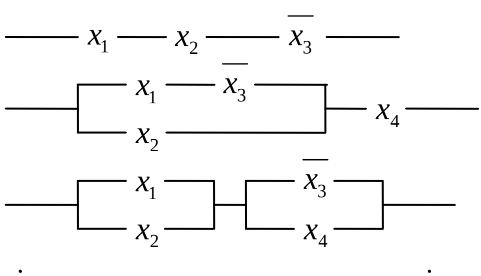
20. Чи є змінна x дійсною змінною функцій, поданих такими формулами?

$f = ((x \rightarrow y) \vee x) \wedge (y \rightarrow x) \wedge z \wedge \overline{x}$;

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

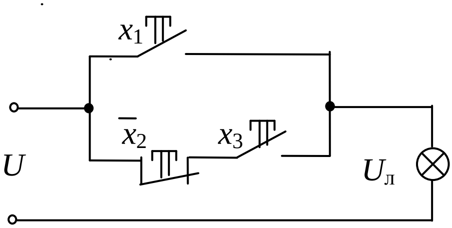
21. Логічну функцію $f = x_1 \wedge x_2 \vee \overline{x_2} \wedge (x_3 \vee x_4)$ подати в базисах $\{\wedge, \neg\}$, $\{\vee, \neg\}$, $\{\rightarrow\}$, $\{\downarrow\}$, $\{|\}$.

22. Скласти функцію провідності контактної схеми та визначити умови її роботи.



Розриви в гілках схеми означають наявність контакту; замикальні контакти позначаються змінними без заперечення, а розмикальні – змінними із запереченням; під умовами роботи схеми слід розуміти перелік наборів, на яких функція провідності приймає значення 1 та 0.

23. Скласти функцію провідності контактної схеми та визначити умови її роботи (ввімкнення лампи за допомогою кнопок).



24. Скласти контактні схеми, які технічно реалізують задані логічні функції:

$f = (\overline{x} \vee y) \wedge (z \wedge y \vee x) \vee u;$

$f = (x \vee y) \wedge \overline{z} \vee \overline{x} \wedge z \vee y;$

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

25. Скласти контактні схеми pf заданиvb умоваvb роботи (умови роботи зазначені наборами значень змінних, на яких логічні функції мають значення 1):

$f(1,0,1) = f(0,1,0) = f(1,1,1) = 1;$

$f(1,0,1) = f(1,1,0) = 1;$

$f(0,0,1) = f(0,0,0) = f(0,1,1) = 1.$

26. Кожний із 3 членів і голова комітету голосують “за”, натискуючи свою кнопку. Рішення вважається ухваленим, якщо загоряється сигнальна лампа (після натискання та утримання не менше ніж трьох кнопок). Голова комісії має ще одну кнопку, за допомогою якої він відкриває та закриває голосування. Скласти функцію голосування та електричну (контактну) схему, що реалізує цю функцію .

27. В умовах вправи 26 замінити контактну схему комбі-

наційною схемою на логічних елементах у базисі І–Або – Не.

28. Скласти комбінаційні схеми на логічних елементах, які реалізують функцію $f=x\cdot(y\cdot z+\overline{y}\cdot\overline{z})+\overline{x}\cdot(\overline{y}\cdot z+y\cdot\overline{z})$ в базисах І – Не та Або – Не.

29. Для функції $f(1,1,0,0,1,1,0,0)$, що задана вектором значень (кортежем значень функції *3* змінних на всіх наборах у послідовності їх десяткових номерів від 0 до 7), побудувати таблицю істинності. На підставі таблиці істинності скласти ДДНФ і ДКНФ цієї функції. Використовуючи закони алгебри логіки, перетворити ДДНФ у ДКНФ і навпаки, ДКНФ у ДДНФ.

30. Використовуючи закони алгебри логіки, подати функцію $f=x\cdot(y\cdot z+\overline{y}\cdot\overline{z})+\overline{x}\cdot(\overline{y}\cdot z+y\cdot\overline{z})$ у ДДНФ і ДКНФ.

31. Використовуючи закони алгебри логіки, подати функцію $f=(\overline{x}+y+z)\cdot(x+y+\overline{z})\cdot(\overline{x}+\overline{y}+\overline{z})$ у ДДНФ.

32. Знайти скорочену та мінімільну ДНФ функції $f=x\cdot y+x\cdot z+\overline{x}\cdot\overline{y}+\overline{x}\cdot\overline{z}+\overline{x}\cdot\overline{y}\cdot\overline{z}$ за методом Квайна.

33. За методом Квайна знайти скорочену та мінімальну КНФ функції $f=(x+y)\cdot(x+z)\cdot(\overline{x}+\overline{y})\cdot(\overline{x}+\overline{z})\cdot(\overline{x}+\overline{y}+\overline{z})$.

34. Логічна функція $f=(x_1,x_2,x_3,x_4)$ визначена частково. На наборах з десятковими номерами 0, 2, 5 вона має значення 1, на наборах №1, 6, 8 – значення 0, а на решті наборів значення функції не задані. За методом Квайна знайти скорочену ДНФ цієї функції. Спробуйте шляхом довизначення знайти тупикову ДНФ із меншою кількістю символів, ніж у скороченій ДНФ.

35. Накресліть шаблони карт Карно для логічних функцій 2, …, 6 змінних із розміткою рядків і стовпців таблиці наборами змінних із позначеннями останніх буквами та двійковими кодами.

36. Методом Вейча – Карно знайти мінімальні ДНФ та КНФ функції $f_{\text{дднф}}(x_1,x_2)=x_1\cdot x_2+\overline{x_1}\cdot x_2+x_1\cdot\overline{x_2}$.

37. Методом Вейча – Карно знайти мінімальні ДНФ та КНФ функції

$$f_{\text{ддноф}}(x_1, x_2, x_3) = x_1 \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot x_3 .$$

38. Методом Вейча – Карно знайти мінімальні ДНФ та КНФ функції f(1,0,0,1,1,0,0,0,1,1,1,0,0,0,1,0), заданої вектором її значень на всіх (16) наборах.

39. Методом Вейча – Карно знайти мінімальні ДНФ та КНФ функції 5 змінних, що має значення 1 на наборах №0, 1, 4, 5, 6, 9, 10, 22, 23, 24, 25, 26, 30, а на решті наборів вона має значення 0.

40. Методом Вейча – Карно знайти мінімальні ДНФ та КНФ функції 6 змінних, що має значення 1 на наборах №0, 1, 4, 5, 9 – 12, 24 – 26, 40 – 44, 56 – 60, а на решті наборів вона має значення 0.

41. Методом Вейча – Карно спробуйте мінімізувати в ДНФ та КНФ частково визначену функцію 4 змінних, що має значення 1 на наборах №0, 1, 5, 6, 12, 13, значення 0 на наборах №15, 16, а на решті наборів значення функції не визначені.

Розділ 3
ЕЛЕМЕНТИ ТЕОРІЇ ГРАФІВ

Теорія графів є одним з розділів математики. Елементи теорії графів були відомі та використовувалися в математиці ще у XVIII столітті, зокрема у працях Л. Ейлера. Проте становлення теорії графів як окремого розділу математики відбулося лише в середині XX століття. Із цього часу граф стає однією із найпопулярніших математичних моделей у багатьох сферах науки і техніки. Методи теорії графів широко застосовуються під час вирішення практичних завдань проектування й дослідження електронних засобів обробки інформації та промислової автоматики, інформаційно-управляючих систем тощо. Великою мірою це пов'язано з бурхливим розвитком та поширенням сфери застосування засобів комп'ютерної техніки в системах автоматики та управління і, як наслідок, значним зростанням ролі задач дискретного характеру. Метою цього розділу є ознайомлення читача з базовими поняттями теорії графів, необхідними для розв'язання практичних задач системної інженерії, зокрема аналізу і синтезу пристроїв дискретної дії та дослідження операцій.

3.1 Основні поняття та означення

Широка галузь завдань науки і техніки, а саме – задачі системної інженерії, до яких належать, зокрема, аналіз і синтез дискретних пристроїв обробки та перетворення інформації у складі інформаційних та керуючих систем, задачі дослідження операцій, такі як оптимізація транспортних потоків (рідини, газу, вантажів) у мережах трубопроводів і доріг, розробка ефективних операційних схем технологічних процесів у промисловому виробництві та багато інших задач потребують моделювання систем (технічних, біологічних, економічних, соціальних) і процесів їх функціонування.

Зручним методом моделювання дискретних процесів і систем є застосування графів. Наприклад, операційна схема алгоритму обчислювального процесу – це граф, яким задається процедура послідовного виконання обчислювальних операцій. Як інші приклади моделювання графами можна назвати електричні кола, схеми доріг, підприємства із двобічними зв’язками між ними, групи людей ыз зазначенням психологічної сумісності між особами, структури управління з переліком об’єктів та їх взаємопідпорядкованості та ін. В інформаційно-керуючих системах застосовуються складні перетворювачі інформації, структурно складені з простих перетворювачів, між якими здійснюється обмін сигналами різних видів за певними правилами. В теорії керуючих систем розглядаються теоретико-графові моделі таких перетворювачів.

Сукупність об’єктів і відношень між ними може бути зображена на площині множиною точок (образом множини об’єктів), і множини ліній, що з’єднують пари точок (образом відношень). Графічний образ сукупності об’єктів і відношень являє собою *діаграму графа*, яку зазвичай називають графом (якщо не відступати від строгості означень, то не графічний образ, а сама сукупність об’єктів і відношень є графом). Множину точок називають вершинами графа, а множину ліній – ребрами (або дугами) графа. Таке неформальне подання графа ми вже використовували для задання відношення між множинами (розд. 1, п. 1.3). Базуючись на неформальному уявленні про графи, введемо необхідні формальні означення.

Нехай V – деяка непорожня множина, а $V^2 = V \times V$ – множина всіх двоелементних підмножин множини V (декартів добуток множини V).

Означення 3.1. *Графом* $G = (V, E)$ називається пара множин (V, E), де $E \subseteq V^2$ – підмножина множини V^2.

Якщо кількість елементів множини V є скінченною, то граф називають *скінченним*. Надалі будемо розглядати лише скінченні графи.

Означення 3.2. Елементи множини V називаються *вершинами* графа $G=(V,E)$.

Вершини графа $G=(V,E)$ будемо позначати символами v_1, $v_2,\ldots,v_n$. У такому разі елементами множини E будуть пари (v_i,v_k) елементів множини V.

Означення 3.3. Якщо елементами множини E графа $G=(V,E)$ є невпорядковані пари (v_i,v_k) елементів множини V, то вони називаються *ребрами* графа G, а сам граф G називається *неорієнтованим*.

Означення 3.4. Якщо елементами множини E графа $G=(V,E)$ є впорядковані пари (v_i,v_k) елементів множини V, то вони називаються *дугами* графа G, а сам граф G називається *орієнтованим* (*орграфом*).

Означення 3.5. Якщо серед елементів множини E графа $G=(V,E)$ є впорядковані та невпорядковані пари (v_i,v_k) елементів множини V, то граф G називається *змішаним*.

Ребра (дуги) графа $G=(V,E)$ будемо позначати символами e_i, $i=1, \ldots, m$, а в розгорнутому вигляді $e_i=(v_i,v_k)$, чим відображається математична сутність ребра (дуги) як елемента множини V^2 (пари елементів множини V).

Вершини та ребра (дуги) графа називаються його *елементами.* Кількість n вершин графа G називається його *порядком* і позначається через $|V|$. Якщо $|V| = n$, а кількість ребер (дуг) $|E| = m$, то G називають *n,m-графом*. Кількості n і m вершин і ребер (дуг) графа та його порядок належать до його *числових характеристик*.

Означення 3.6. Вершини v_i і v_k називаються *кінцевими вершинами* ребра (дуги) $e_i=(v_i,v_k)$ графа (орграфа) $G=(V,E)$.

Вершини v_i і v_k орграфа, що з’єднані дугою $e_i=(v_i,v_k)$, називають також *початковою* і *кінцевою* відповідно, а саму дугу називають такою, що виходить із вершини v_i і заходить у вершину v_k.

Вершинами графа можуть бути вузли транспортної або електричної мережі, оператори програми, команди операційної системи, події в технологічній або керуючій системі, а ребрами (дугами) – лінії зв'язку в транспортній або електричній мережі, переходи в технологічній схемі виробництва або в процедурі обчислення на комп'ютері. Програма або електронна схема, виробничий або інформаційний процес можуть бути поданими орграфом, схема електричної мережі – неорієнтованим графом, схема транспортної мережі – неорієнтованим або змішаним графом.

Означення 3.7. Граф $G=(V,\varnothing)$, що не містить жодного ребра (дуги), називається *порожнім* (порожній граф з n вершинами позначається O_n).

Означення 3.8. Граф, що складається з однієї вершини, називається *тривіальним*.

Означення 3.9. Граф, що не містить жодної вершини, називається *нуль-графом*: $G=(\varnothing,\varnothing)$.

Означення 3.10. Якщо вершини v_i і v_k з'єднані ребром (дугою), то вони називаються *суміжними*, а ребро (дуга) $e_i=(v_i,v_k)$ називається *інцидентним* (інцидентною) до своїх кінцевих вершин v_i і v_k.

Означення 3.11. Вершина v_i графа (орграфа) $G=(V,E)$ називається *ізольованою*, якщо вона не має жодного інцидентного до неї ребра (дуги).

Означення 3.12. Вершина v_i графа (орграфа) $G=(V,E)$ називається *висячою*, якщо вона має лише одну суміжну до неї вершину.

Означення 3.13. Два ребра (дві дуги) називаються *суміжними*, якщо вони мають спільну вершину.

Граф (орграф) може містити пари вершин, що з'єднані кількома ребрами (дугами). Такі ребра в неорієнтованому графі називають *паралельними*. В орграфі такі дуги називають паралельними, якщо вони одного напряму, а в протилежному разі – *контрпаралельними* (зустрічно-паралельними).

Елементами множини E можуть бути пари виду (v_i, v_i), що означає збіг кінцевих вершин відповідного ребра (дуги). Ребра (дуги) виду $e_i = (v_i, v_i)$ називають *петлями*. Очевидно, спрямованість петлі не має практичного сенсу, тому в орграфі можна вважати її неорієнтованою.

Деякі особливі властивості графів відображаються їхними спеціальними назвами.

Означення 3.14. *Степенем* $deg(v_i)$ вершини v_i називається кількість ребер (дуг), інцидентних цій вершині, причому петля враховується двічі.

Для орграфа кількість дуг, що виходять із вершины v_i, називається *напівстепенем виходу* і позначається $deg^-(v_i)$, а кількість дуг, що заходять у вершину v_i, – *напівстепенем заходу* і позначаєтся $deg^+(v_i)$. Очевидно, що

$$deg(v_i) = deg^-(v_i) + deg^+(v_i). \tag{3.1}$$

Означення 3.15. Вершина орграфа з *напівстепенем заходу* $deg^+(v_i) = 0$ називається *витоком*, а вершина з *напівстепенем виходу* $deg^-(v_i) = 0$ – *стоком*.

Лема 3.1 (про рукостискання). Нехай $G_{n,m} = (V, E)$ – довільний граф. Тоді

$$\sum_{i=1}^{n} deg(v_i) = 2m. \tag{3.2}$$

Доведення. При підрахуванні суми степенів кожне ребро (дуга) $e_i = (v_i, v_k)$ вносить свій вклад, що дорівнює 1, одночасно в $deg(v_i)$ і $deg(v_k)$, причому петля буде враховуватися двічі. Лему доведено.

Наслідок: довільний граф містить парне число вершин непарного степеня. Дійсно, якщо V_0 і V_1 – множини вершин відповідно парного та непарного степенів, то

$$\sum_{v_i \in V_0} deg(v_i) + \sum_{v_i \in V_1} deg(v_i) = 2m.$$

Очевидно, що перший доданок парний. Тому і другий доданок парний, оскільки сума (3.2) парна. Оскільки у другій сумі всі доданки непарні, їх кількість парна. Отже, множина V_1 містить парне число вершин.

Означення 3.16. Граф (орграф), у якого будь-які дві вершини суміжні, називається *повним*.

Повний граф із n вершинами позначається K_n. Очевидно, степінь кожної вершини в графі K_n дорівнює $n-1$. Тому з леми 3.1 про рукостискання випливає, що кількість ребер (дуг) у K_n дорівнює $n(n-1)/2$.

Означення 3.17. Якщо у графі без петель хоча б одна пара вершин сполучена більше ніж одним ребром (дугою), то такий граф називається *мультиграфом* (неорієнтованим, орієнтованим).

Під сполученням пари вершин більш ніж одним ребром (дугою) в означенні мультиграфа розуміється наявність у неорієнтованому графі паралельних ребер, а в орграфі – паралельних або контрпаралельних дуг.

Означення 3.18. Граф, що не містить пар вершин, сполучених більш ніж одним ребром (дугою), та вершин з більш ніж одною петлею, називається *графом Бержа*.

Означення 3.19. Граф (орграф), що містить петлі, називається *псевдографом* (графом із петлями).

Петлі у псевдографі можуть бути кратними. Псевдограф може містити паралельні та контрпаралельні ребра (дуги).

Означення 3.20. Граф (орграф), що не містить петлі й кратні ребра (дуги), називається *простим*.

Очевидно, що в простому графі степінь вершини v_i дорівнює кількості суміжних з нею вершин.

Означення 3.21. Граф $G=(V,E)$ називається *зваженим* (навантаженим), якщо його ребра (дуги) або вершини мають характеристичні ознаки, які називаються *вагами*.

Вагою ребра (дуги) або вершини можуть бути тривалість у часі, протяжність у просторі, надійність функціонування та ін. Наприклад, час виконання команди операційної системи або оператора програми, ймовірність події або надійність експлуатації вузла і т. п. зазначаються як вага вершини графа, а протяжність ліній транспортних або електричних мереж, їх пропускна здатність за якімось параметром – як вага ребра або дуги.

Паралельні ребра (дуги) та петлі незваженого графа (орграфа) називають також *кратними*. Очевидно, що у зваженому графі (орграфі) паралельні ребра (дуги) та петлі є кратними лише тоді, коли вони мають однакову вагу. Якщо контрпаралельні дуги у зваженому орграфі мають однакову вагу, то їх можна еквівалентно замінити неорієнтованим ребром із такою самою вагою.

Аналізуючи назви видів графів за означеннями 3.3 –3.5, 3.11 –3.12, 3.16 – 3.21, неважко помітити, що один і той самий граф може бути віднесений до графів різних видів. Тому вид графа визначають за найбільш істотною ознакою, залежно від мети його застосування у конкретній задачі.

Помітимо, що в літературних джерелах переважно використовується назва “граф” для графа будь-якого виду. Конкретизація виду графа його специфікованою назвою робиться лише в тих випадках, коли характеристичні ознаки виду графа не випливають із контексту з очевидністю. Надалі ми будемо дотримуватися цієї традиції.

Наведений тут короткий огляд понять теорії графів не є вичерпним. Надалі, під час вивчення способів задання графів, їх властивостей та операцій над ними ми зустрінемось із багатьма іншими видами графів, “архітектура” яких має специфічні риси.

Контрольні питання

Складіть перелік систем і процесів, моделями яких можуть бути графи.

Наведіть приклади застосування діаграм графів у теорії множин.

Визначте перелік елементів множини V^2, якщо V – п'ятиелементна скінченна множина.

Який математичний об'єкт називається графом? За яких умов граф буде скінченним?

Які математичні об'єкти називаються: неорієнтованим графом, орграфом, змішаним графом, ребром, дугою, петлею, паралельними та кратними ребрами (дугами)?

Яким чином визначається порядок графа?

Який математичний об'єкт називається n, m-графом?

Які вершини називають кінцевими вершинами ребра графа $G = (V, E)$?

Поясніть поняття степеня вершини графа.

Поясніть поняття напівстепенів виходу і заходу вершини орграфа.

Які вершини орграфа називають витоком і стоком?

Наведіть приклади програмних, технічних і технологічних об'єктів, що можуть бути подані вершинами, ребрами (дугами) графів, неорієнтованим графом, орграфом, змішаним графом.

Відтворіть означення порожнього і тривіального графів та нуль-графа.

Поясніть поняття суміжності вершин графа.

Поясніть поняття інцидентності ребра (дуги) графа його (її) кінцевим вершинам.

Які вершини графа називають ізольованою та висячою? Чому дорівнюють їх степені?

Поясніть поняття суміжності ребер (дуг) графа.

Поясніть поняття повного графа. Чи є він простим? Чому дорівнює степінь кожної з його вершин? Скільки ребер (дуг) містить такий граф?

Відтворіть означення мультиграфа, псевдографа, простого графа, графа Бержа.

За яких умов один і той самий граф є одночасно графом Бержа і псевдографом?

Чому дорівнює сума степенів вершин графа? Чи є вона парною?

Чи є парною кількість вершин парного (непарного) степеня в довільному графі?

За яких умов граф $G=(V,E)$ називається зваженим?

Наведіть приклади технічних і технологічних параметрів, кількісні оцінки яких можуть бути вагами ребер (дуг) і вершин зважених графів.

Поясніть поняття кратності для ребер і дуг відповідно в неорієнтованих графах і орграфах, незважених і зважених.

Чи може один й той самий граф бути графом різних видів (за означеннями 3.3 – 3.5, 3.11 – 3.12, 3.16 – 3.21)?

3.2 Способи задання графів

Існують різні способи задання графів, серед яких найбільш зручними та широко застосовними є такі: аналітичний, геометричний, списковий та матричний. Вибір та зручність того чи іншого з цих способів залежать від особливостей задачі, що розв’язується з використанням графа.

Перед тим як розглядати ці способи, введемо наступне формальне означення (яке ми неформально вже застосовували вище).

Означення 3.22. Граф називається *поміченим,* якщо його вершини відрізняються одна від іншої деякими позначками, наприклад $v_1, v_2, \ldots, v_n$ або 1, 2, …, n.

Усі перелічені способи застосовуються для задання помічених графів, а геометричний спосіб застосовується також і для задання непомічених графів. Перейдемо до розгляду цих способів.

1. *Аналітичний спосіб задання графів*. Для n,m-графа

$G_{n,m}=(V, E)$ з вершинами $v_1,v_2,\ldots,v_n \in V$ і ребрами $e_1,e_2,\ldots,e_m \in E$ задається функція відображення φ: $V^2 \to E$, наприклад переліком пар $e_i=(v_i, v_k)$, упорядкованих у разі, якщо $G_{n,m}$ – орграф. Цього достатньо для визначення всіх відношень суміжності та інцидентності у заданому графі.

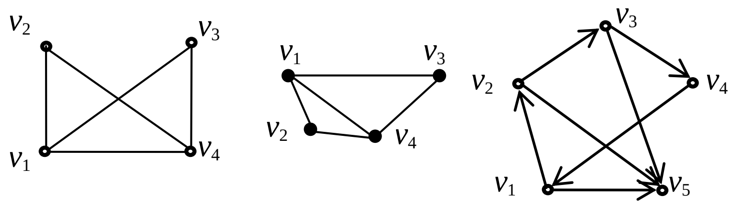
Приклад 3.1. $G_1=(V_1, E_1)$, $V_1=\{v_1, v_2, v_3, v_4\}$, $E_1=\{(v_1, v_2),(v_1, v_3),(v_1, v_4),(v_2, v_4),(v_3, v_4)\}$ – це граф із чотирма вершинами і п'ятьма ребрами, а граф $G_2=(V_2, E_2)$, $V_2=\{v_1, v_2, v_3, v_4, v_5\}$, $E_2=\{(v_1, v_2),(v_2, v_3),(v_3, v_4),(v_4, v_1),(v_1, v_5),(v_2, v_5),(v_3, v_5)\}$ – орграф із п'ятьма вершинами і сімома дугами.

2. *Геометричний спосіб задання графів*. Графи наочно зображаються за допомогою рисунків на площині, які називають *діаграмами* графів. Вершинам графа ставляться у відповідність точки площини. Точки, що відповідають суміжним вершинам, з'єднуються лінією (відрізком або кривою, в орграфі – зі стрілкою), яка графічно відображає ребро (дугу). Діаграми графів часто скорочено називають графами, якщо це не викликає непорозумінь.

На рис. 3.1 зображені діаграми графів G_1 і G_2 з прикладу 3.1. Діаграмами рис. 3.1а і 3.1б задається граф G_1, який є поміченим, неорієнтованим, простим, незваженим. Як бачимо з порівняння діаграм на рис. 3.1а і 3.1б, діаграма графа змінює свій вигляд залежно від розміщення точок (вершин) на площині.

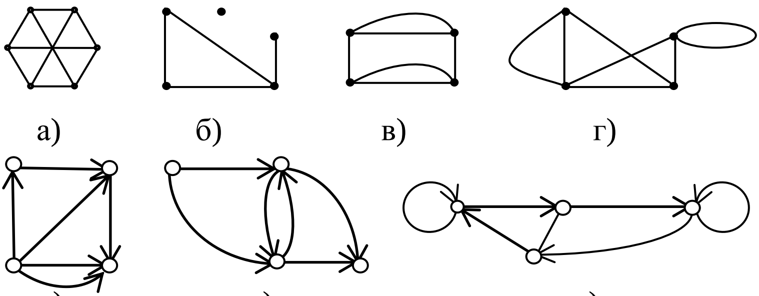
Діаграмою рис. 3.1в задається орграф G_2 із прикладу 3.1, який є поміченим і простим. На рис. 3.2 подано інші приклади діаграм графів.

Для більшої наочності діаграми графів на рис. 3.2 показані непоміченими. Безумовно, цими зразками графів не охоплюється їх різноманітність. Надаємо читачеві можливість самостійно визначити види графів (за означеннями 3.3 – 3.5, 3.11 – 3.12, 3.16 – 3.20), заданих діаграмами на цьому рисунку.



а) G_1 б) G_1 в) G_2

Рисунок 3.1 – Діаграми графів з прикладу 3.1



а) б) в) г)

д) е) ж)

Рисунок 3.2 – Геометричне задання графів

3. *Списковий спосіб задання графів*. Це спосіб задання графів списками суміжностей, інциденцій та ваг. Зручною формою подання цих списків є таблиці відношень. Розглянемо цей спосіб на наступному прикладі.

Приклад 3.2. Вихідними умовами зазначено, що граф містить вершини v_1, v_2 та v_3 з ваговими параметрами відповідно h_1, h_2 та h_3 і дуги $e_1^{(1)} = (v_1, v_2)$, $e_1^{(2)} = (v_1, v_2)$, $e_2 = (v_2, v_3)$, $e_3^{(1)} = (v_3, v_3)$, $e_3^{(2)} = (v_3, v_1)$ з ваговими параметрами відповідно $p_1^{(1)}$, $p_1^{(2)}$, p_2, $p_3^{(1)}$ та $p_3^{(2)}$. (Умовними позначеннями вершин та їх ваг відображаються, наприклад, стани системи, що подається графовою моделлю, або деякі технічні характеристики цих станів, а позначеннями дуг та їх ваг відображаються переходи з одного стану до іншого і тривалість цих переходів). Такого опису орграфа $G_{3,5} = (V, E)$ достатньо для побудови його діаграми, показаної на рис. 3.3.

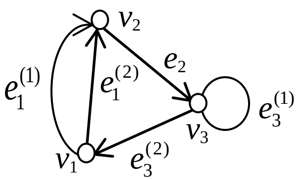


Рисунок 3.3 – Діаграма орграфа до прикладу 3.2

Потрібно: скласти задання зваженого орграфа списками суміжностей, інциденцій та ваг на підставі наведеного вище його вихідного опису.

Список суміжностей та ваг вершин орграфа $G_{3,5} = (V, E)$ подано таблицею 3.1.

Таблиця 3.1 – Список суміжностей та ваг вершин орграфа

v_i:	v_1	v_2	v_3
$S(v_i) = \{\ldots\}$:	v_2, v_3	v_1, v_3	v_1,v_2
h_i:	h_1	h_2	h_3

Список інциденцій та ваг дуг орграфа $G_{3,5} = (V, E)$ подано таблицею 3.2.

Таблиця 3.2 – Список інциденцій та ваг дуг орграфа

e_i:	$e_1^{(1)}$	$e_1^{(2)}$	e_2	$e_3^{(1)}$	$e_3^{(2)}$
(v_i,v_k):	(v_1,v_2)	(v_1,v_2)	(v_2,v_3)	(v_3,v_3)	(v_3,v_1)
p_i:	$p_1^{(1)}$	$p_1^{(2)}$	p_2	$p_3^{(1)}$	$p_3^{(2)}$

На підставі таблиць 3.1 і 3.2 можна успішно накреслити діаграму орграфа $G_{3,5} = (V, E)$. Умовні позначення ваг вершин і дуг можна не розміщати на діаграмі графа, якщо це надто її ускладнює.

Ми не будемо окремо розглядати технологію спискового задання неорієнтованих та незважених графів, оскільки вона очевидна з прикладу 3.2. Лише помітимо, що для незважених графів в таблицях 3.1 і 3.2 відсутні рядки ваг, а якщо граф неорієнтований, то слід мати на увазі, що пари (v_i, v_k) є невпорядкованими.

Задання графа таблицями відношень (табличними списками) зручно для зберігання та обробки інформації про ваги ребер (дуг) і вершин, пошуку вершин, інцидентних ребрам (дугам), пошуку суміжних вершин.

4. *Матричний спосіб задання графів*. Матричний опис графа зручний для обчислення числових характеристик графа і виконання різних алгебраїчних операцій, оскільки спирається на глибоко розроблену теорію матриць.

Для задання графа за допомогою матриць потрібно всі його вершини занумерувати натуральними числами від 1 до n, а всі ребра (дуги) – числами від 1 до m. Саме так ми робили вище, тобто елементами графа $G_{n,m}=(V,E)$ є вершини v_i, $i=1, \ldots, n$ і ребра (дуги) e_i, $i=1, \ldots, m$.

Основними видами матриць, якими описуються графи, є матриці суміжностей, інциденцій та ваг.

Матриця суміжностей A графа (орграфа) $G_{n,m}=(V,E)$ без петель являє собою квадратну $n\times n$-матрицю, в якій елемент a_{ij} i-го рядка і j-го стовця дорівнює 1, якщо вершини v_i та v_j, $i\neq j$ суміжні, і дорівнює 0 при $i=j$. В матриці суміжностей мультиграфа елемент a_{ij}, $i\neq j$ дорівнює кількості ребер (дуг), що з'єднують вершини v_i та v_j.

Приклад 3.3. Складемо матриці суміжностей A_1 і A_2 відповідно для неорієнтованого графа G_1 та орграфа G_2 з прикладу 3.1. Вершини і ребра (дуги) занумеруємо в тому порядку, в якому вони виписані у цьому прикладі (діаграми графів подані на рис. 3.2, читачеві рекомендується самостійно зробити позначення ребер і дуг). Матриці суміжностей A_1 і A_2 графів G_1 і G_2 мають вигляд:

$$A_1 = \begin{Vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{Vmatrix}; \qquad A_2 = \begin{Vmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{Vmatrix}.$$

Помітимо, що в матриці суміжностей графа без петель головна діагональ ($i=j$) заповнена нулями. За наявності петель умовно вважають, що вершина a_i з петлею (можливо, не єдиною) суміжна до себе. Отже, в матриці суміжностей графа з петлею (псевдографа) елемент a_{ii} головної діагоналі має значення 1.

Матриця інциденцій A простого неорієнтованого графа $G_{n,m}=(V, E)$ являє собою $n{\times}m$-матрицю, в якій елемент a_{ij} i-го рядка і j-го стовпця дорівнює 1, якщо вершині v_i з номером i інцидентне ребро e_j з номером j, і дорівнює 0 у протилежному разі.

Приклад 3.4. Складемо матрицю інциденцій B_1 графа G_1 з прикладу 3.1:

$$B_1 = \begin{Vmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{Vmatrix}.$$

Кожний стовпець матриці інциденцій простого неорієнтованого графа містить дві одиниці. Це обумовлено тим, що кожне ребро (дуга) має дві кінцеві вершини. Кількість одиниць у кожному рядку дорівнює степеню $deg(v_i)$ відповідної вершини графа.

У загальному випадку в рядку за номером i матриці інциденцій будь-якого неорієнтованого графа B сума значень елементів b_{ij} має бути рівною степеню $deg(v_i)$ вершини v_i, з урахуванням наявності петель і кратних ребер. Інцидентність кратних ребер і петель вершині v_i графа враховується повним списком його ребер і петель з відповідними до них стовпцями матриці інциденцій. Інцидентність кожної петлі вершині графа враховується двічі (див. означення 3.14), оскільки вона пов'язана з вершиною двома своїми кінцями. Наявність петлі відображається в матриці інциденцій таким чином: якщо вершині v_i інцидентна петля e_j, то $b_{ij} = 2$.

Матриця інциденцій простого орграфа будується по-іншому, а саме: елемент b_{ij} матриці B дорівнює: –1, якщо дуга e_j виходить із вершини v_i; 1, якщо дуга e_j заходить у вершину v_i; 0, якщо дуга e_j не інцидентна вершині v_i. Тому в матриці інциденцій простого орграфа кожний стовпець містить два числа: –1 і 1 (початок і кінець відповідної дуги). У кожному рядку кількість чисел "1" дорівнює напівстепеню заходу $deg^+(v_i)$ вершини v_i, а кількість чисел "–1" – напівстепеню виходу $deg^-(v_i)$.

Приклад 3.5. Складемо матрицю інциденцій B_2 для графа G_2 з прикладу 3.1:

$$B_2 = \begin{Vmatrix} -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{Vmatrix}.$$

Наявність в орграфі паралельних і контрпаралельних дуг і петель враховується відповідними стовпцями матриці інциденцій. У випадку, коли вершина v_i має інцидентну до неї петлю e_j, слід враховувати, що петля одночасно заходить у вершину і виходить із неї, що еквівалентно наявності двох контрпаралельних дуг. Тому в матриці інциденцій мають бути два стовпці для петлі e_j за номерами j та $j+1$, а елементи матриці b_{ij} та b_{ij+1} матимуть значення відповідно –1 та 1. Тоді у кожному рядку кількість чисел "1" буде дорівнювати напівстепеню заходу $deg^+(v_i)$ вершини v_i, а кількість чисел "–1" – напівстепеню виходу $deg^-(v_i)$.

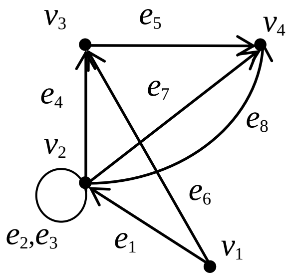


Рисунок 3.4 – Діаграма орграфа до прикладу 3.6

Приклад 3.6. На рис. 3.4 подано діаграму зваженого орграфа. Ваги дуг дорівнюють: $p_1 = 2$, $p_2 = p_3 = 1$, $p_4 = 5$, $p_5 = 3$, $p_6 = 8$, $p_7 = 6$, $p_8 = 7$ (індексація ваг співпадає з індексацією дуг і петлі).

Складемо матриці суміжностей (А) та інциденцій (В) цього орграфа:

$$A = \begin{Vmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{Vmatrix}; \; B = \begin{Vmatrix} -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{Vmatrix}.$$

У цьому орграфі вершина v_1 є витоком, а вершина v_1 – стоком. Цей факт також можна бачити з матриці B: $deg^{+}(v_1) = 0$, $deg^{-}(v_4) = 0$.

Щоб скласти матрицю інциденцій змішаного графа, потрібно замінити його еквівалентним орграфом. Для цього неорієнтоване ребро змішаного графа заміняється двома контрпаралельними дугами.

Матриця ваг. Матрицями ваг зазначають ваги ребер (дуг) графів (орграфів). Для простого графа (орграфа) матрицю ваг ребер (дуг) можна скласти такою самою за структурою, як матриця суміжностей. Елементи c_{ij} матриці ваг C визначаються таким чином: $c_{ij} = p(e_l)$, тобто елемент c_{ij} на перетині i-го рядка та j-го стовпця матриці дорівнює вазі $p(e_l)$ ребра (дуги), що з'єднує суміжні вершини v_i та v_j, $i \neq j$; $c_{ii} = 0$; $c_{ij} = 0$ або $c_{ij} = \infty$, якщо вершини v_i та v_j, $i \neq j$ не суміжні. Вибір значення 0 або ∞ елемента c_{ij} в останньому випадку залежить від практичного сенсу ваг ребер (дуг). Наприклад, якщо під вагою розуміється відстань між пунктами (вершинами) v_i та v_j на карті доріг, то у разі їх несуміжності потрібно взяти $c_{ij} = \infty$, а якщо під вагою розуміється пропускна здатність трубопроводу, то для несуміжних v_i та v_j слід прийняти $c_{ij} = 0$.

Незалежно від того, простий граф (орграф) чи ні, матрицю ваг ребер (дуг) можна скласти за структурою матриці інциденцій, в якій одиниці заміняються значеннями ваг відповідних ребер (дуг).

Приклад 3.7. Матриця ваг орграфа з прикладу 3.6, складена за описаними правилами, матиме такий вигляд:

$$C = \begin{Vmatrix} -2 & 0 & 0 & 0 & 0 & -8 & 0 & 0 \\ 2 & -1 & 1 & -5 & 0 & 0 & -6 & -7 \\ 0 & 0 & 0 & 5 & -3 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 6 & 7 \end{Vmatrix}.$$

Для вершинно зваженого графа можна задати ваги вершин матрицею ваг, яка з очевидністю буде являти собою n-елементну матрицю-стовпець. Наприклад, транспонована матриця ваг тривершинного графа (орграфа) матиме вигляд

$$H^{\text{т}} = \begin{Vmatrix} h_1 & h_2 & h_3 \end{Vmatrix},$$

де h_1, h_2, h_3 – ваги вершин з порядковою нумерацією 1, 2, 3.

Контрольні питання

Перелічіть відомі вам способи задання графів.

Які графи називають поміченими?

У чому полягає процедура задання графів аналітичним способом? Чи існують якісь особливості задання орграфів цим способом?

Який геометричний об’єкт називають діаграмою графа? Які елементи містить діаграма графа? Зображеннями яких елементів відрізняється діаграма орграфа від діаграми неорієнтованого графа?

Чи накладаються якісь обмеження на розміщення вершин в діаграмі графа? Чи можна намагатись усунути перетинання ребер (дуг) в діаграмі графа шляхом зміни розміщення вершин графа на площині?

Чи можна зображати петлю в діаграмі орграфа у вигляді замкненої лінії без стрілки? Якщо так, то обґрунтуйте.

Яким чином (в якій формі) реалізується списковий спосіб задання графів?

За якою формою будується таблиця відношень, якою задається список суміжностей і ваг вершин графа?

За якою формою будується таблиця відношень, якою задається список інциденцій та ваг ребер (дуг) графа?

Поясніть структуру матриці суміжностей графа (орграфа) без петель. Яким чином визначаються значення її елементів? Які особливості в структурі матриці суміжностей виникають за наявності петель в деяких вершинах графа?

Поясніть структуру матриці інциденцій простого неорієнтованого графа. Яким чином визначаються значення її елементів? Який зв'язок існує між кількістю одиниць у рядку матриці та степенем відповідної вершини графа? Чому кожний стовпець матриці інциденцій простого неорієнтованого графа містить дві одиниці?

Який зв'язок існує між сумою значень елементів у рядку матриці інциденцій неорієнтованого графа та степенєм відповідної цьому рядку вершини графа? Яким чином наявність кратних ребер та петель ураховується в структурі матриці інциденцій та значеннях її елементів?

Викладіть правила побудови матриці інциденцій простого орграфа.

Яким чином можна дізнатися про значення напівстепенів заходу та виходу будь-якої з вершин простого орграфа на підставі його матриці інциденцій?

Чому кожний стовпець матриці інциденцій простого орграфа містить два числа: –1 і 1, а інші елементи стовпця подаються числом 0?

Викладіть правила побудови матриці інциденцій орграфа з урахуванням наявності паралельних і контрпаралельних дуг і петель.

Яким чином здійснюється заміна змішаного графа еквівалентним орграфом?

Ваги яких елементів графа (орграфа) можна задати за допомогою матриці ваг?

Викладіть правила побудови матриці ваг ребер (дуг) простого графа (орграфа) за структурою матриці суміжностей.

За структурою якої матриці – суміжностей або інциденцій – можна побудувати матрицю ваг ребер (дуг) будь-якого графа (незалежно від того, простий цей граф чи ні)?

Викладіть правила побудови матриці ваг ребер (дуг) графа (орграфа) за структурою матриці інциденцій. Наведіть приклади побудови матриці ваг ребер (дуг) графа (орграфа), що містить паралельні та контрпаралельні ребра (дуги) та петлі.

Який вигляд має матриця ваг вершинно зваженого графа?

3.3 Ізоморфізм графів

У теорії графів та її застосуваннях часто суттєвим є лише існування об'єктів (вершин графа) і зв'язків між ними (ребер). Тоді один граф можна утворити з відомого іншого шляхом зміни позначень вершин та конфігурації діаграми графа. Ці міркування приводять до поняття ізоморфізму графів.

Означення 3.23. Нехай $G_1=(V_1,E_1)$, і $G_2=(V_2,E_2)$ – два графи. Бієктивне відображення ψ: $V_1 \to V_2$ називається *ізоморфізмом* G_1 на G_2, якщо число ребер, що з'єднують вершини u і v в G_1, дорівнює числу ребер, що з'єднують вершини $\psi(u)$ і $\psi(v)$ в G_2 (зрозуміло, що число петель у вершині u дорівнює числу петель у вершині $\psi(u)$).

Означення 3.24. Графи $G_1=(V_1,E_1)$ і $G_2=(V_2,E_2)$ називаються *ізоморфними*, якщо існує ізоморфізм G_1 на G_2 (позначення $G_1 \cong G_2$).

Сформульоване означення ізоморфізму придатне і для орграфів, необхідно лише ребра вважати орієнтованими (тобто дугами), а пари вершин, з’єднані дугами, вважати упорядкованими.

На рис. 3.5 наведені діаграми двох ізоморфних графів.

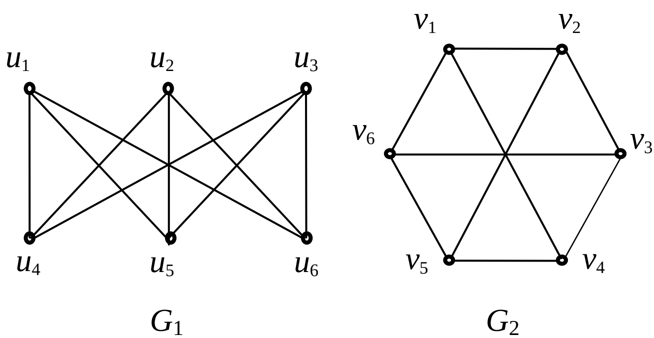


Рисунок 3.5 – Діаграми ізоморфних графів

Ізоморфізм G_1 на G_2 є взаємно однозначним відображенням множини вершин V_1 на множину вершин V_2, що зберігає відношення суміжності вершин. Іншими словами, ізоморфізм ψ характеризується властивістю: довільні вершини u і v суміжні в графі G_1 тоді й лише тоді, коли вершини $\psi(u)$ і $\psi(v)$ суміжні в графі G_2.

Ізоморфізм – це бінарне взаємно однозначне відношення на множині графів, тобто він є відношенням еквівалентності. Ізоморфні графи відрізняються лише ідентифікаторами вершин. Ця відмінність не є суттєвою, тому звичайно ізоморфні графи ототожнюють, зображаючи їх діаграми без позначення вершин, або з нумерацією вершин натуральними числами. Отже, діаграма задає граф з точністю до ізоморфізму.

Для доведення ізоморфності двох графів потрібно показати відповідну бієкцію. У загальному випадку доведення ізоморфізму графів може виявитися складною задачею. Якщо формально виходити з означення ізоморфізму, то потрібно здійснювати перебір бієкцій множини вершин одного графа на множину вершин іншого і для кожної з них перевіряти, чи є вона ізоморфізмом. Практично такий перебір бієкцій здійснюється на основі наступної теореми.

Теорема 3.1. Графи G_1 та G_2 ізоморфні тоді й лише тоді, коли матрицю суміжностей одного з цих графів можна одержати з матриці суміжностей іншого графа за допомогою відповідних перестановок рядків та стовпців.

Доведення. Оскільки ізоморфні графи G_1 і G_2 порядку n відрізняються між собою лише нумерацією (позначеннями) вершин, то існує бієктивне відображення φ множини номерів (позначень) вершин першого графа на множину номерів (позначень) вершин другого. Отже, кожен елемент a_{ij} матриці суміжностей A графа G_1 збігається з елементом $b_{\varphi(i)\,\varphi(j)}$, що знаходиться в рядку з номером $\varphi(i)$ і стовпці з номером $\varphi(j)$ матриці суміжностей B графа G_2. Якщо відображення φ відоме, то шляхом послідовної перестановки рядків з номерами i та $\varphi(i)$, а потім і стовпців з тими самими номерами для всіх $i=1, ..., n$ матрицю суміжностей A можна перетворити у матрицю суміжностей B і навпаки.

Якщо потрібно перевірити за допомогою матриць суміжностей ізоморфність двох графів порядку n, необхідно здійснити різні (за сполученнями номерів i та j) перестановки рядків і стовпців однієї з цих матриць. Якщо після чергової перестановки одержимо матрицю, що збігається з іншою, то дійдемо до висновку, що ці графи ізоморфні.

Кількість усіх можливих перестановок рядків і стовпців квадратної матриці порядку n дорівнює $n!$. Якщо після виконання всіх $n!$ перестановок збігу матриць суміжностей обох графів не одержимо, то дійдемо висновку, що ці графи не ізоморфні. *Теорему доведено*.

Приклад 3.8. Матриці A і B суміжностей вершин графів і G_1 і G_2, наведених на рис. 3.5, мають вигляд:

$$A=\begin{Vmatrix}0&0&0&1&1&1\\0&0&0&1&1&1\\0&0&0&1&1&1\\1&1&1&0&0&0\\1&1&1&0&0&0\\1&1&1&0&0&0\end{Vmatrix};\quad B=\begin{Vmatrix}0&1&0&1&0&1\\1&0&1&0&1&0\\0&1&0&1&0&1\\1&0&1&0&1&0\\0&1&0&1&0&1\\1&0&1&0&1&0\end{Vmatrix}.$$

Матриця A одержується з матриці B одночасною перестановкою другого та третього, а потім третього та п'ятого рядків і стовпців. Отже, ці графи ізоморфні. Ізоморфізм A на B можна подати в табличній формі:

$u_i \in U$	u_1	u_2	u_3	u_4	u_5	u_6
$v_i \in V$	v_1	v_3	v_5	v_4	v_2	v_6

Для n вершин число можливих бієкцій дорівнює $n!$. Щоб з'ясувати розглянутим способом, чи є графи ізоморфними, може стати потрібним виконати всі $n!$ перестановок рядків і стовпців матриці суміжностей одного з них. Уже для порівняно невеликих значень n здійснити такий перебір бієкцій практично неможливо (наприклад, при $n=20$ маємо $n!>2\cdot 10^{18}$).

У прикладній теорії алгоритмів розробляються різноманітні алгоритми перевірки ізоморфізму графів, які для більшості графів (або окремих типів графів) дозволяють суттєво скоротити обсяг необхідних перевірок.

Одержаний (шляхом відповідних перестановок рядків і стовпців) збіг матриць суміжностей є достатньою умовою ізоморфності неорієнтованих графів без паралельних ребер, тому що при цьому збігаються також їх матриці інциденцій. У загальному ж випадку для будь-яких графів (орграфів) достатньою умовою ізоморфності є збіг матриць інциденцій, що гарантує збіг матриць суміжностей цих графів. За аналогією з теоремою 3.1 наведемо наступну теорему, на підставі якої можна здійснювати перевірку ізоморфності будь-яких графів.

Теорема 3.2. Графи (орграфи) G_1 і G_2 ізоморфні тоді й лише тоді, коли матрицю інциденцій одного із цих графів можна одержати з матриці інциденцій іншого за допомогою перестановок рядків та стовпців.

Для матриць інциденцій графів G_1 і G_2 з n вершинами і m ребрами (дугами) справедливі аналогічні міркування щодо перестановок рядків і стовпців, як і для матриць суміжностей. Відмінність у тому, що коли G_1 і G_2 ізоморфні, тоді для їх множин вершин існує бієкція φ, а для множин ребер – інша бієкція ψ. Тому перестановки рядків і стовпців здійснюються в будь-якої послідовності як взаємно незалежні кроки. Загальна кількість таких кроків для перевірки ізоморфізму графів G_1 і G_2 не перевищує $n!\cdot m!$.

Приклад 3.9. Перевіримо ізоморфність графів G_1 і G_2, діаграми яких наведені на рис. 3.6.

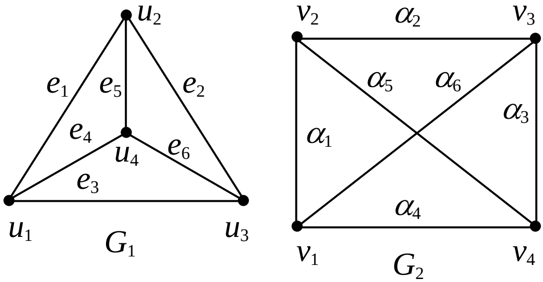


Рисунок 3.6 – Діаграми графів до прикладу 3.9

Матриці A і B інциденцій графів G_1 і G_2 мають вигляд:

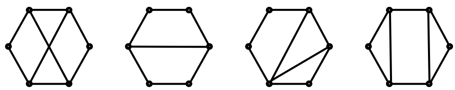
$$A = \begin{Vmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{Vmatrix}; \quad B = \begin{Vmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{Vmatrix}.$$

Матриця A одержується з матриці B послідовними перестановками 3-го і 5-го, а потім 5-го і 6-го стовпців. Отже, ці графи ізоморфні. Ізоморфізм G_1 і G_2 подається бієкціями φ: $u_i \leftrightarrow v_i,\ i=\overline{1,4}$ та ψ: $e_1 \leftrightarrow \alpha_1$, $e_2 \leftrightarrow \alpha_2$ $e_3 \leftrightarrow \alpha_5$ $e_4 \leftrightarrow \alpha_4$, $e_5 \leftrightarrow \alpha_6$, $e_6 \leftrightarrow \alpha_3$.

Приклади 3.8 і 3.9 ілюструють випадки, коли неважко здогадатися, якими перестановками можна довести ізоморфність графів. Частіше ж доведення ізоморфності виявляється дуже трудомістким, потребує великих витрат часу. Однак у багатьох випадках можна досить легко дізнатися, що два графи не ізоморфні.

Достатньою умовою ізоморфності графів є збіг матриць інциденцій та суміжностей, а необхідними умовами є: однакові кількості вершин, ребер (дуг) і петель; однакові набори степенів (напівстепенів виходу та заходу) вершин; інші числові та структурні характеристики графів. Ці характеристики називають *інваріантами*, оскільки вони зберігаються при ізоморфізмі. Якщо вдається виявити, що для двох графів деякі інваріанти не збігаються, то ці графи не ізоморфні. У теорії графів повну систему інваріантів ще не визначено.

Приклад 3.10. Перевіримо на ізоморфність графи G_1, G_2, G_3, G_4, діаграми яких наведені на рис. 3.7.



G_1 G_2 G_3 G_4

Рисунок 3.7 – Діаграми графів до прикладу 3.10

Кожен із цих графів має 6 вершин. Граф G_2 відрізняється від усіх інших кількістю ребер. Граф G_3 відрізняється від графів G_1 та G_4 тим, що на відміну від них містить вершину, степінь якої дорівнює 4. Граф G_4 відрізняється від графа G_1 тим, що містить два цикли (замкнені контури) довжиною 3 (із трьох ребер). Отже, серед розглянутих графів немає жодної пари ізоморфних.

Контрольні питання

Яка властивість графів дозволяє утворити новий граф із відомого шляхом зміни позначень вершин та конфігурації діаграми графа?

Наведіть поняття ізоморфізму графа G_1 на граф G_2.

За яких умов графи G_1 та G_2 ізоморфні?

Вкажіть на особливості поняття ізоморфізму графа G_1 на граф G_2, якщо вони є орграфами.

Яким чином відношення ізоморфізму графа G_1 на граф G_2 пов'язане із суміжністю вершин у цих графах?

Чи існує ізоморфізм графа G_2 на граф G_1 при ізоморфізмі G_1 на граф G_2? Якщо так, то завдяки чому?

Чи є ізоморфізм графів відношенням їх еквівалентності? Якщо так, то завдяки чому?

Якими способами можна виявити ізоморфність простих графів (орграфів)?

Якими способами можна виявити неізоморфність простих графів (орграфів)?

Якими способами можна виявити неізоморфність мультиграфів?

Викладіть процедуру перевірки ізоморфності двох графів шляхом перестановок рядків і стовпців матриці суміжностей одного з них.

Викладіть процедуру перевірки ізоморфності двох графів шляхом перестановок рядків і стовпців матриці інцидентностей одного з них.

Які характеристики графів називають інваріантами?

Які умови є достатніми умовами ізоморфності графів, а які – необхідними?

3.4 Операції над графами

Для одержання нових графів можна здійснювати різні операції над графами. Розглянемо два види операцій – *алгебраїчні*, коли новий граф будується за визначеними правилами з інших графів, та *унарні* (локальні), за яких вилучаються або додаються окремі елементи графа.

Для графів, як і для множин, можна означити операції об'єднання, перетину, різниці, доповнення і симетричної різниці (розд. 1, означення 1.1 – 1.5).

Означення 3.25. *Об'єднанням* графів $G_1 = (V_1, E_1)$ і $G_2 = (V_2, E_2)$ називається граф $G = G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$.

Об'єднання $G = G_1 \cup G_2$ називається *диз'юнктивною (*або прямою*) сумою* графів G_1 і G_2, якщо $V_1 \cap V_2 = \varnothing$, тобто графи G_1 і G_2 не мають спільних вершин.

Означення 3.26. *Перетином графів* $G_1 = (V, E_1)$ і $G_2 = (V, E_2)$ називається граф $G = G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$.

Якщо $V_1 \cap V_2 = \varnothing$, то $G = (\varnothing, \varnothing)$.

Означення 3.27. Р*ізницею графів* $G_1 = (V, E_1)$ і $G_2 = (V, E_2)$ з однаковими множинами V вершин називається граф $G = G_1 \backslash G_2 = (V, E_1 \backslash E_2)$.

Означення 3.28. *Доповненням графа* $G = (V, E)$ називається граф $\overline{G} = (V, V^2 \backslash E)$.

Граф $\overline{G}$ має ту саму множину вершин V, що й граф G, а вершини графа $\overline{G}$ суміжні тоді й лише тоді, коли вони несуміжні в G. Для графа G із n вершинами виконується рівність $\overline{G} = K_n \backslash G$ (див. означення 3.16).

Означення 3.29. *Симетричною різницею* графів $G_1 = (V, E_1)$ і $G_2 = (V, E_2)$ з однаковими множинами V вершин називається граф $G = G_1 - G_2 = (V, (E_1 \backslash E_2) \cup (E_2 \backslash E_1))$.

Розглянуті операції над графами становлять групу *алгебраїчних операцій*, оскільки вони виконуються за законами алгебри графів $< \Gamma, \cup, \cap, \bar{} >$, носієм якої є множина Γ усіх графів, а сигнатурою – операції об'єднання $\cup$, перетину $\cap$ та доповнення $\bar{}$.

Приклад 3.11. На рис. 3.8 наведені діаграми псевдографів G_1 і G_2 та $G = G_1 \cup G_2$.

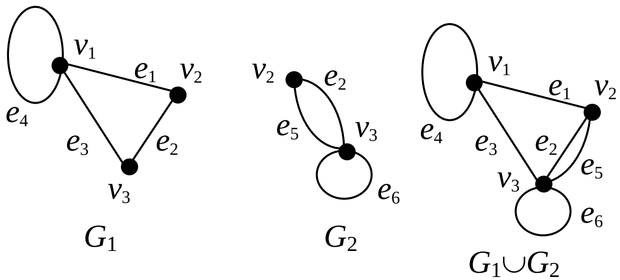


Рисунок 3.8 – Діаграми графів до прикладу 3.11

Приклад 3.12. На рис. 3.9 наведені діаграми графів $G_1 = (V, E_1)$ і $G_2 = (V, E_2)$ та їх об'єднання $H_1 = G_1 \cup G_2$, перетин $H_2 = G_1 \cap G_2$, різниця $H_3 = G_1 \setminus G_2$, доповнення $H_4 = K_n \setminus G_1$, симетрична різниця $H_5 = G_1 - G_2$.

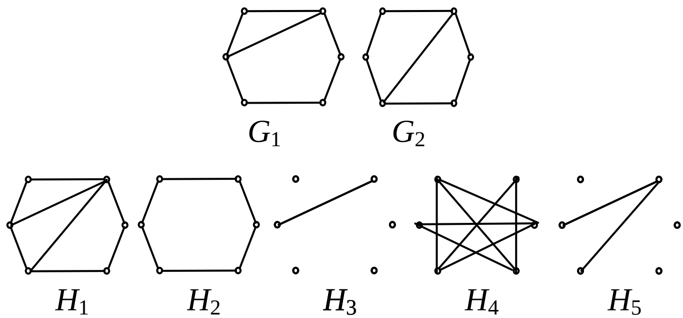


Рисунок 3.9 – Діаграми графів до прикладу 3.12

Іншу групу операцій над графами становлять унарні (локальні) операції, що здійснюються над єдиним графом. Простішими з них є операції вилучення вершини або ребра (дуги) і додавання таких самих елементів.

При *вилученні ребра* (дуги) у графі зберігаються всі його елементи, крім вилученого ребра (дуги). Операція *додавання ребра* (дуги) є зворотною до операції вилучення, при цьому в графі можуть з'являться паралельні ребра (дуги) і навіть петлі.

При *вилученні вершиши* з графа вилучаються також і всі ребра (дуги), що інцидентні цій вершині. Операція *додавання вершини* полягає в тому, що до елементів графа додається ізольована вершина.

До числа простіших можна віднести також операцію *стягування вершин* графа, коли задану множину вершин (зазвичай дві) з'єднують в єдину вершину, а отримані з ребер (дуг) петлі вилучаються.

Будь-які інші унарні операції над графом, що мають за мету зміну його елементного складу, зводяться до послідовного виконання низки операцій вилучення та додавання вершин і ребер (дуг) та стягування вершин.

Контрольні питання

Назвіть два види операцій над графами.

Наведіть перелік алгебраїчних операцій над графами.

Наведіть означення об'єднання графів. За яких умов об'єднання графів називається їх диз'юнктивною (прямою) сумою? Наведіть графічний приклад.

Наведіть означення перетину графів з однаковими множинами вершин. Наведіть графічний приклад.

Наведіть означення різниці графів з однаковими множинами вершин. Наведіть графічний приклад.

Наведіть означення доповнення графа. Наведіть графічний приклад.

Наведіть означення симетричної різниці графів. Наведіть графічний приклад.

Назвіть перелік унарних операцій над графами.

Поясніть сутність операції одержання нового графа шляхом вилучення ребра (дуги) з відомого графа. Наведіть графічний приклад.

Поясніть сутність операції одержання нового графа шляхом додавання ребра (дуги) у відомому графі. Наведіть графічний приклад.

Поясніть сутність операції одержання нового графа шляхом вилучення вершини з відомого графа. Наведіть графічний приклад.

Поясніть сутність операції одержання нового графа шляхом додавання вершини у відомому графі. Наведіть графічний приклад.

Поясніть сутність операції одержання нового графа шляхом стягування вершин відомого графа. Наведіть графічний приклад.

3.5 Графи та підграфи

Під час використання графових моделей систем і процесів часто виникає необхідність виділяти у графі більш складні елементи, ніж ребра та вершини, зокрема складові частини графа, що самі є графами. Такими складовими частинами є *підграфи*.

Означення 3.30. Граф $G'=(V',E')$ називається *підграфом* графа (орграфа) $G=(V,E)$, якщо $V'\subseteq V$ і $E'\subseteq E$, причому $e_i=(v_i,v_k)\in E'$ лише за умови, що $v_i, v_k\in V'$.

Якщо $E'\subset E$ чи $V'\subset V$, то граф G' називають *власним* підграфом графа G.

Означення 3.31. Якщо $V'=V$, тобто всі вершини графа (орграфа) $G=(V,E)$ наявні в графі $G'=(V',E')$, $E'\subset E$, причому G' є деревом (див. п. 3.7.4), то граф G' називається *кістяковим* (каркасним) підграфом графа (орграфа) G.

Приклад 3.13. На рис. 3.10 показані власний G_1 та кістяковий G_2 підграфи орграфа G .

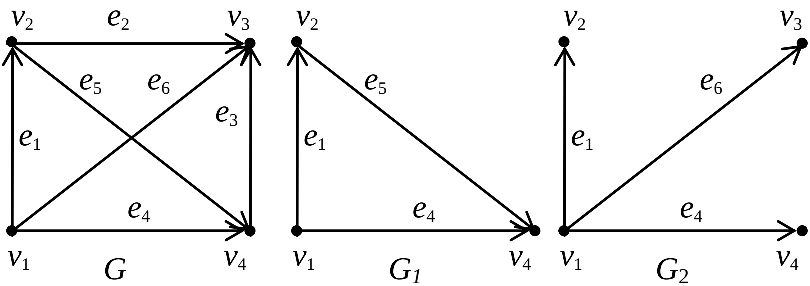


Рисунок 3.10 – Орграф та його підграфи

Очевидно, один і той самий граф (орграф) може мати декілька власних і кістякових підграфів, які одержують шляхом вилучення вершин та ребер (дуг).

Контрольні питання

Який математичний об’єкт носить назву підграфа? Наведіть означення цього об’єкта.

За яких умов підграф $G^{'}$ є власним підграфом графа G?

За яких умов підграф $G^{'}$ є кістяковим підграфом графа G? Чи є кістяковий підграф $G^{'}$ одночасно власним підграфом графа G?

За допомогою яких операцій над графом можна виділяти з нього різні підграфи?

3.6 Структурні характеристики графів

Із деякими структурними характеристиками графів ми вже зустрічалися при введенні основних понять (п. 3.1). Ці характеристики випливають з означень 3.3 –3.5, 3.11 –3.12, 3.16 – 3.21 видів графів. Однак застосування графів під час проектування й дослідження об'єктів і систем потребує також уявлень і про більш складні структурні характеристики графів, зокрема про різноманітні види маршрутів. Уявлення про види маршрутів необхідні для розв’язання проблеми досяжності, тобто можливості переходу по ребрах (дугах) графа (орграфа) з однієї вершини в іншу. Задачі аналізу досяжності на графах є складовими комплексних завдань проектування і дослідження таких складних технологічних систем, як транспортні мережі, мережі електро-, газо-, водопостачання та ін.

Означення 3.32. *Маршрутом* у графі називається скінченна послідовність вершин і ребер v_i, e_i, v_k, e_k, …, v_l, e_l, v_m, що починається і закінчується вершинами, причому кожне ребро в цій послідовності інцидентне вершинам, що примикають до нього зліва і справа.

Мірою маршруту називається його *довжина*, яка дорівнює числу ребер, що входять у цей маршрут.

Поняття маршруту згідно з цим означенням відноситься і до орграфів, при цьому орієнтація дуг не має значення, тобто дуга розглядається як неорієнтоване ребро.

Означення 3.33. Маршрут називається *замкненим*, якщо його початкова та кінцева вершини збігаються.

Означення 3.34. Маршрут називається *відкритим*, якщо його початкова та кінцева вершини різні.

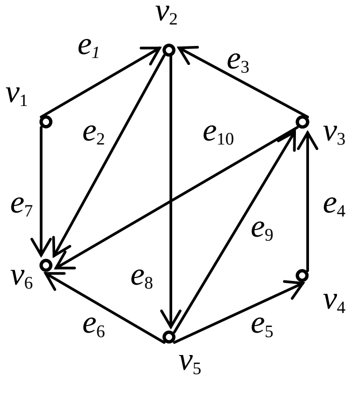


Рисунок 3.11 – Орграф до прикладу 3.14

Приклад 3.14. В орграфі на рис. 3.11 послідовності

v_2, e_2, v_6, e_6, v_5, e_9, v_3, e_{10}, v_6, e_7, v_1; v_1, e_1, v_2, e_2, e_6, e_{10}, v_3, e_3, v_2, e_2, e_6; v_3, e_3, v_2, e_8, v_5, e_9, v_3

є маршрутами, причому два перших є відкритими, а останній з них є замкненим.

Частіше використовуються окремі види маршрутів, яким властиві певні особливості.

Означення 3.35. *Шляхом* (орієнтованим маршрутом) в орграфі називається послідовність дуг, в якій кінцева вершина будь-якої дуги, крім останньої в порядку слідування, є початковою вершиною наступної дуги.

Наприклад, у графі рис. 3.11 маршрут v_5, e_9, v_3, e_3 є шляхом. Очевидно, в неорієнтованому графі будь-який маршрут є шляхом. Оскільки шлях є маршрутом, то означення *замкненого і відкритого шляхів* будуть такими самими, як замкненого і відкритого маршрутів. Наприклад, у графі рис. 3.11 шлях v_5, e_5, v_4, e_4, v_3, e_3, v_2, e_8, v_5 є замкненим, а шлях v_5, e_5, v_4, e_4, v_3, e_3, v_2 – відкритим.

Означення 3.36. *Довжиною* (або потужністю) шляху називається число ребер (дуг), що в нього входять (позначається $d(v_i, v_j)$).

Означення 3.37. Довжина найкоротшого шляху між парою вершин v_i і v_j називається *відстанню* між цими вершинами.

Шлях v_5, e_5, v_4, e_4, v_3, e_3, v_2 у графі рис. 3.11 має довжину $d(v_5,v_2)=3$, а найкоротшим шляхом між вершинами v_5 і v_2 є v_5, e_9, v_3, e_3, v_2 довжиною $d(v_5, v_2)=2$, тобто відстань між вершинами дорівнює 2.

Поняття довжини щляху дозволяє ввести наступні дві *числові характеристики* графів.

Означення 3.38. Максимальний за довжиною шлях із вершини v_i графа (орграфа) G до всіх інших вершин називається *ексцентриситетом* цієї вершини (позначення $\varepsilon(v_i)$).

Означення 3.39. Максимальний ексцентриситет вершин графа (орграфа) називається його *діаметром*.

Означення 3.40. Мінімальний ексцентриситет вершин графа (орграфа) називається його *радіусом*.

Продовжимо огляд різновидів маршрутів.

Означення 3.41. Шлях у графі (орграфі) називається *ланцюгом* (*орієнтованим ланцюгом*), якщо всі його ребра (дуги) різні.

Означення 3.42. Шлях, в якому кожна вершина (крім, можливо, першої та останньої в порядку слідування) зустрічається не більше одного разу, називається *простим шляхом (простим ланцюгом*).

Оскільки ланцюги є шляхами, то означення *замкненого і відкритого ланцюгів* будуть такими самими, як замкненого і відкритого шляху. Наприклад, у графі рис. 3.11 ланцюг v_5, e_5, v_4, e_4, v_3, e_3, v_2, e_8, v_5 є простим замкненим, а шлях v_5, e_5, v_4, e_4, v_3, e_3, v_2 – простим відкритим.

Означення 3.43. Простий змкнений ланцюг називається *циклом*.

Наприклад, у графі рис. 3.11 ланцюг v_5, e_5, v_4, e_4, v_3, e_3, v_2, e_8, v_5 є циклом.

Шляхи (ланцюги, цикли) у графі можна зображати також послідовністю вершин або ребер (дуг). Таке їх подання корисне, коли здійснюється пошук простих шляхів або ланцюгів.

Якщо граф (орграф) є зваженим, то будь-який шлях w, що подається послідовністю ребер (дуг), має вагу

$$P(w) = \sum_{(e_i, e_j) \in w} p(e_i, e_j),$$

де $p(e_i, e_j)$ та $P(w)$ – відповідно ваги ребра (дуги) та шляху.

Базуючись на поняттях маршруту і шляху, розглянемо ще одну структурну характеристику графів, що має назву "*зв'язність*".

Означення 3.44. Вершини v_i і v_j у графі (орграфі) G називаються *зв'язаними*, якщо існує маршрут із вершини v_i у вершину v_j.

Означення 3.45. Граф (орграф) G називається *зв'язним*, якщо у ньому будь-які дві вершини є зв'язаними.

З очевидністю справедлива наступна теорема.

Теорема 3.3. Для зв'язності графа (орграфа) необхідно і достатньо, щоб у ньому існував маршрут із якої-небудь фіксованої вершини v_i у кожну іншу вершину v_j.

Означення 3.46. Якщо для деякої пари вершин графа (орграфа) не існує маршруту, що їх з'єднує, то такий граф (орграф) називається *незв'язним*.

Означення 3.47. Орграф називається *сильно зв'язним*, якщо для будь-якої пари різних вершин v_i та v_j існує принаймні один шлях із v_i в v_j.

Це означення стосується і неорієнтованих графів, оскільки маршрути в цих графах одночасно є шляхами.

Очевидно, сильно зв'язні орграфи (графи) одночасно є зв'язними, а зв'язний орграф не завжди є сильно зв'язним.

Означення 3.48. Вершина v_j графа (орграфа) G називається *досяжною* з вершини v_i, якщо існує принаймні один шлях, що з'єднує v_i з v_j.

Означення 3.47 означає, що будь-які дві вершини сильно зв'язного графа (орграфа) *взаємно досяжні*, тобто існують шляхи з вершини v_i у вершину v_j та у зворотному напрямку – з вершини v_j у вершину v_i.

Приклад 3.15. Орграф, поданий на рис. 3.12, зв'язний, але не сильно зв'язний, тому що немає шляхів із вершини v_3 у вершини v_1 та v_2. Якщо у цьому графі замінити дуги ребрами, то одержимо сильно зв'язний (і зв'язний) неорієнтований граф.

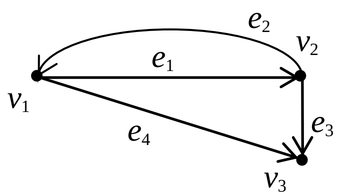


Рисунок 3.12 – Орграф до прикладу 3.15

Поняття маршрутів, шляхів, ланцюгів, циклів, зв'язності, розглянуті щодо графів (орграфів), відносяться без змін і до підграфів. Зв'язні підграфи називають *компонентами зв'язності* графів (орграфів).

Практичні завдання, що розв'язуються за допомогою графів, часто потребують проведення аналізу досяжності одних вершин графа (орграфа) з інших. Це зручно виконувати за допомогою матриць досяжностей.

Матриця досяжностей графа (орграфа) G_n являє собою $n \times n$ матрицю $R = \|r_{ij}\|$, елементи якої визначаються таким чином: $r_{ij} = 1$, якщо існує принаймні один шлях із вершини v_i у вершину v_j, і $r_{ij} = 0$, якщо шляху з вершини v_i до вершини v_j не існує. Всі діагональні елементи r_{ii} в матриці R дорівнюють 1, оскільки кожна вершина завжди досяжна із себе.

Приклад 3.16. Матриця досяжностей графа рис. 3.12 має такий вигляд:

$$R = \begin{Vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{Vmatrix}.$$

Множина вершин графа G, досяжних із вершини v_i, складається з вершин v_j, для яких $r_{ij} = 1$ в i-му рядку матриці досяжностей $R = \|r_{ij}\|$, побудованої за описаними правилами.

Матриця досяжностей $R = \|r_{ij}\|$ може мати деякі модифікації. Зокрема, елементи r_{ij} можна визначити як такі, що дорівнюють кількості шляхів заданої довжини $d(v_i, v_j)$ із v_i в v_j.

Побудовою матриць досяжностей для заданих значень $d(v_i, v_j) = 1, 2, \ldots$ можна знайти найкоротший шлях із вершини v_i у вершину v_j (відстань між вершинами v_i і v_j), що може бути необхідним для розв’язання деякої практичної задачі з використанням графової моделі.

У матриці досяжностей зваженого графа (орграфа) елементи r_{ij} можна визначити як такі, що дорівнюють сумарній вазі шляху із v_i у v_j в деякому діапазоні. Таке подання матриць досяжностей зваженого графа корисне під час розв’язування задач оптимізації переходу з однієї вершини в іншу за критерієм мінімізації (або максимізації) ваги переходу (вартості, витрати часу, пропускної здатності та ін.).

Контрольні питання

Який структурний елемент графа (орграфа) називається маршрутом?

Яким чином визначається довжина маршрута в графі (орграфі)?

Який маршрут у графі (орграфі) називається замкненим?

Який маршрут у графі (орграфі) називається відкритим?

Який маршрут у графі (орграфі) називається шляхом?

Яким чином визначається довжина шляху в графі (орграфі)?

Поясніть поняття відстані між вершинами графа (орграфа). Яким чином вона визначається?

Поясніть поняття ексцентриситету вершини графа (орграфа).

Поясніть поняття діаметра і радіуса графа (орграфа).

Який шлях у графі (орграфі) називається ланцюгом (орієнтованим ланцюгом)?

Який ланцюг у графі (орграфі) називається простим?

Який ланцюг у графі (орграфі) називається циклом?

Яким чином визначається вага шляху в графі (орграфі)?

Які вершини в графі (орграфі) називаються зв'язними?

За яких умов граф (орграф) називається зв'язним?

Яка умова є необхідною і достатньою для зв'язності графа (орграфа)?

Який граф (орграф) називається незв'язним?

Який граф (орграф) називається сильно зв'язним? Чи є сильно зв'язний граф (орграф) одночасно зв'язним?

Чи завжди зв'язний граф (орграф) є одночасно сильно зв'язним?

За яких умов вершина v_j графа (орграфа) G є досяжною з вершини v_i?

Які структурні елементи графа (орграфа) називаються компонентами зв'язності?

За якими правилами будується матриця досяжностей графа (орграфа)? В яких модифікаціях можуть бути подані матриці досяжностей незважених та зважених графів (орграфів) для розв’язання практичних задач дослідження операцій, таких як пошук найкоротшого шляху та оптимізація переходів за критерієм мінімізації (максимізації) ваги переходу?

3.7 Спеціальні види графів

Графи спеціальних видів характеризуються тим, що вони мають певні структурні особливості. Таких графів багато, ми обмежимося розглядом лише тих, що часто зустрічаються в задачах проектування і дослідження об’єктів системотехніки.

3.7.1 *Двочасткові графи*. Відмітною ознакою графів цього виду є те, що вершини графа поділені на дві підмножини таким чином, що кожне ребро (дуга) з’єднує вершини різних підмножин. Із такими графами ми зустрічалися під час вивчення відношень на множинах (п. 1.3).

Означення 3.49. Граф (орграф) $G=(V,E)$ називається *двочастковим,* якщо $V=V_1 \cup V_2$, $V_1 \cap V_2 = \varnothing$, і кожне ребро (дуга) із E з’єднує одну вершину з підмножині V_1 з однією або декількома вершинами в підмножині V_2.

Пара підмножин (V_1, V_2) є двочастковим розбиттям множини V.

Означення 3.50. Двочастковий граф (орграф) $G=(V,E)$ з розбиттям (V_1, V_2) множини V називається *повним двочастковим графом* (орграфом), якщо кожна вершина $v_i \in V_1$ з’єднана з кожною вершиною $v_j \in V_2$ одним ребром (дугою).

Повний двочастковий граф (орграф) з r і s вершинами позначається K_{rs}. При $r=s$ повний двочастковий граф (орграф) називається *збалансованим.*

На рис. 3.13 наведені приклади діаграм двочасткових графів: а) неповного, б) повного $K_{2,5}$, в) збалансованого $K_{2,2}$.

3.7.2 *Роздільні графи*. Структурна особливість таких графів (орграфів) полягає в тому, що в результаті виконання операції вилучення ребер або вершини граф стає (або залишається) незв'язним, набираючи вигляду двох відокремлених підграфів, між якими немає жодного зв'язку.

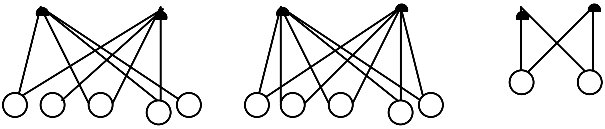


Рисунок 3.13 – Приклади двочасткових графів

Означення 3.51. Множина ребер (дуг), вилучення яких розділяє граф на два незв'язаних між собою підграфи, називається *розрізною множиною*, або *розрізом* (перерізом).

Наслідком вилучення із графа G ребер розрізаючої множини є створення нового графа $H = G_1 \cup G_2$, який складається з графів $G_1 = (V_1, E_1)$, $V_1 \subset V$, $E_1 \subset E$ і $G_2 = (V_2, E_2)$, $V_2 \subset V$, $E_2 \subset E$, які є підграфами графа G з різними множинами вершин і ребер: $V_1 \cap V_2 = \varnothing$, $E_1 \cap E_2 = \varnothing$.

Очевидно, в багатореберному графі $G = (V, E)$ буває можливим виділити декілька розрізних множин.

Означення 3.52. Ребро (дуга), що являє собою одноелементний розріз графа, не належний жодному з його циклів, називається *мостом*, а вершини, інцидентні цьому ребру, називаються *точками зчленування*.

Означення 3.53. Граф, одне з ребер (дуг) якого є мостом, називається *роздільним*.

Помітимо, що граф може мати єдину точку зчленування, вилучення якої (з інцидентними до неї ребрами) робить граф незв'язним, що складається з двох підграфів, які не перетинаються. Графи з єдиною точкою зчленування також відносять до роздільних. Беручи це до уваги, можемо дати наступне означення.

Означення 3.54. Граф (орграф), що не містить жодної точки зчленування, називається *нероздільним*.

На рис. 3.14 показані приклади: а) графа з виділеним розрізом, б) графа з мостом, що з'єднує вершини v і w, в) графа рис.3.14,б без вилученого моста.

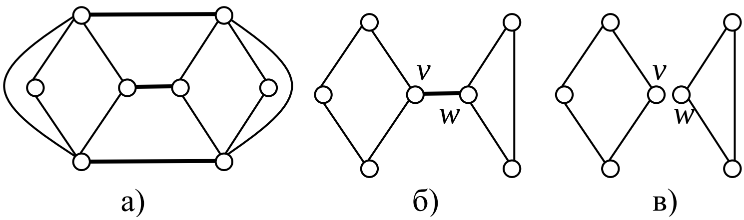


Рисунок 3.14 – Діаграми графів з розрізом (а), мостом (б) та розділеного вилученням моста

Роздільні графи широко застосовуються в задачах проектування складних систем і процесів, що піддаються декомпозиції на підсистеми (підпроцеси). Взаємно незалежне проектування і дослідження останніх з подальшим агрегуванням спрощує процедуру створення системи (процесу) в цілому. Як приклад можна назвати завдання дослідження надійності складного технічного пристрою, що розв'язується на графовій моделі, підмоделями якої є підграфи, якими моделюється надійність окремих складових пристрою.

3.7.3 *Мережі*. Важливе значення в практичному застосуванні теорії графів має поняття мережі.

Означення 3.55. Реберно зважений орграф $G_{n,m}=(V,E)$ називається *мережею*, якщо він:

а) містить дві й лише дві вершини v_1 і v_n, перша з яких (початкова) є витоком ($deg^{+}(v_1)=0$), а друга (кінцева) – стоком ($deg^{-}(v_n)=0$);

б) вага кожної з дуг зазначається невід'ємним числом, яке називається *пропускною здатністю* дуги (позначення $c(e_i)$, $e_i \in E$, $i=\overline{1,m}$).

На рис. 3.15 наведений приклад діаграми мережі.

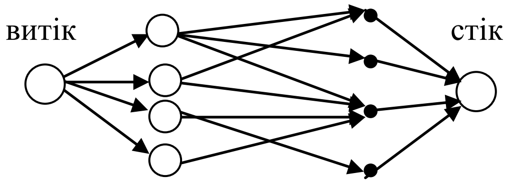


Рисунок 3.15 – Діаграма мережі

Основним поняттям теорії мереж є *потік*. Множину дуг, що входять у проміжну вершину v_k, $k=2, \ldots, n-1$, позначимо E_k^+, а множину дуг, що виходять з цієї вершини, позначимо E_k^-. На множині дуг E, що становлять мережу, визначимо функцію $f(e_i)$, що відображає множину E у множину цілих невід'ємних чисел, що відповідають таким умовам:

а) $f(e_i) \leq c(e_i)$, тобто числа $f(e_i)$ не перевищують пропускної здатності $c(e_i)$ дуг мережі;

б) для кожної вершини, за винятком витоку та стоку, виконується рівність:

$$\sum_{e_i \in E_k^+} f(e_i) - \sum_{e_i \in E_k^-} f(e_i) = 0.$$

Умова б) означає, що для проміжних вершин мережі сума потоків дуг, що виходять з вершини v_k, дорівнює сумі потоків дуг, що входять у цю вершину. Умова б) означає, що в проміжних вершинах мережі потоки не створюються і не зникають.

Означення 3.56. Функція $f(e_i)$, що визначається умовами а) і б), називається *потоком в дузі*.

Означення 3.57. Функція

$$F_k = \sum_{e_i \in E_k^+} f(e_i) = \sum_{e_i \in E_k^-} f(e_i),\ k=2,\ \ldots,\ n-1$$

називається *потоком у вершині*.

Потоки у вершинах v_1 і v_n, що є витоком та стоком, відповідно дорівнюють:

$$F_1 = \sum_{e_i \in E_1^-} f(e_i); \quad F_n = \sum_{e_i \in E_n^+} f(e_i).$$

Мережа завжди має принаймні один розріз. Множину дуг розрізу позначимо як E_t.

Означення 3.58. Функція

$$F_t = \sum_{e_i \in E_t} f(e_i)$$

називається *потоком у мережі*.

Очевидно, має місце рівність

$$F_t = F_1 = F_n. \tag{3.3}$$

Прикладами реальних мереж є транспортні, дільниці яких (дуги орграфа) мають різну пропускну здатність щодо можливості перевезення певної кількості вантажу, пропускання води або газу, передачі електроенергії за одиницю часу. Важливим завданням для реальних мереж є максимізація величини потоку.

Будь-який розріз мережі перетинає всі шляхи, що проходять від витоку до стоку. З усіх розрізів, які можна виявити в мережі, є такий, що його пропускна здатність (сума пропускних здатностей усіх дуг, що він включає) найменша. Тому *пропускна здатність мережі* дорівнює мінімальній пропускній здатності розрізів, а *величина потоку в мережі* не може перевищувати пропускну здатність мережі.

Мережа рис. 3.15 має лише один витік та один стік. Це простіший випадок. Реальні мережі можуть містити декілька витоків та стоків. У таких випадках потоки F_1 та F_n в рівнянні

(3.3) являють собою сумарні потоки відповідно всіх витоків та стоків мережі.

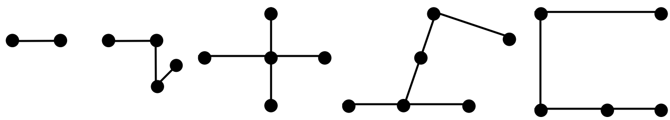
3.7.4. *Дерева*. Дерева є найбільш простим і поширеним класом графів. Структурною особливістю дерев є відсутність циклів. Вихідним поняттям цього класу графів є ліс.

Означення 3.59. Буд-який зв'язний граф, що не містить циклів, називається *ациклічним*, або *лісом*.

Компонентами зв'язності (підграфами) лісу є дерева.

Означення 3.60. *Деревом* називається скінченний зв'язний граф без циклів, що містить не менше двох вершин.

Приклади дерев показані на рис. 3.16.



а) б) в) г) д)

Рисунок 3.16 – Приклади дерев

Як реальний приклад дерева можна назвати структуру дисків, папок і файлів персонального комп'ютера.

Для будь-якого дерева має місце рівність $m=n-1$. Дійсно, ця рівність справедлива для однореберного дерева (рис. 3.16,а). Додавання ребра до вершини збільшує m і n на одиницю зі збереженням рівності $m=n-1$. Нарощуючи число ребер приєднанням їх до будь-яких вершин з появою нових висячих вершин, ми будемо одержувати нові дерева зі збереженням рівності $m=n-1$.

Крім властивостей зв'язності, ациклічності та співвідношення $m=n-1$, деревам властиві ще такі очевидні властивості: кожне ребро є ребром розрізу (мостом); будь-які дві вершини з'єднані одним і лише одним простим відкритим ланцюгом; при додаванні ребра між двома несуміжними вершинами граф перестає бути деревом у зв'язку із появою в ньому одного циклу; після вилучення будь-якого ребра граф перестає бути деревом у зв'язку із втратою зв'язності; дерево є *планарним* графом, оскільки його діаграму можна зобразити на площині без перетину ребер.

Орієнтоване дерево (ордерево) визначається аналогічно неорієнтованому. Воно являє собою орграф без циклів, в якому напівстепінь заходу єдиної вершини, яка називається *коренем*, дорівнює нулю, а напівстепінь заходу кожної висячої вершини, крім кореня, дорівнює одиниці.

Означення 3.61. Вершина ордерева називається *коренем*, якщо існує орієнтований шлях із цієї вершини у кожну з інших вершин.

Означення 3.62. Вершини ордерева, напівстепені виходу яких дорівнюють нулю (висячі вершини), називаються його *листям*.

Ордерева відрізняються від неорієнтованих дерев тим, що не всі впорядковані пари вершин пов'язані оршляхами.

Оскільки дерева (ордерева) є графами (орграфами), то до них рівною мірою належать усі теоретичні положення за пп. 3.1 – 3.6. Зокрема, компонентами зв'язності (підграфами) дерева є піддерева.

На рис. 3.17 показано діаграму орграфа, який є ордеревом. Коренем цього дерева є вершина v_0, а листями є вершини v_2, v_3, v_6, v_7, v_8.

Для більшої наочності структуру діаграми ордерева зображають як *ранжовану* за рівнями вершин (рис.3.17).

Означення 3.63. Якщо в ордереві існує шлях з вершини v_i у вершину v_k, то вершина v_i називається *предком* вершини v_k, а вершина v_k – *нащадком* вершини v_i.

В ордереві рис. 3.17 v_0 – предок вершин v_1, v_2, …, v_8, а отанні є нащадками вершини v_0.

Означення 3.64. Якщо в ордереві вершини v_i та v_k суміжні, причому дуга виходить з вершини v_i, то вершина v_i називається *прямим предком* вершини v_k, а вершина v_k – *прямим нащадком* вершини v_i.

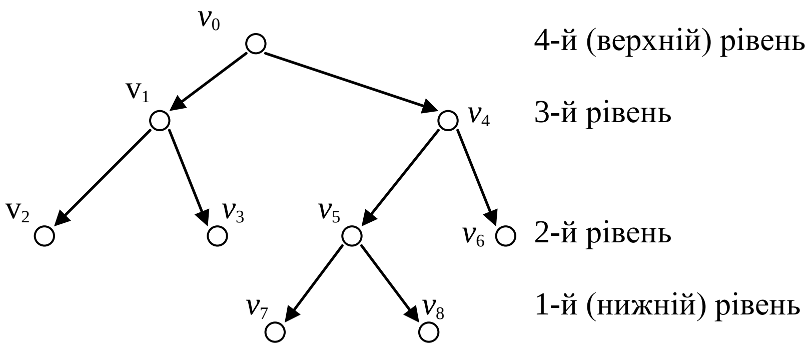


Рисунок 3.17 – Діаграма орієнтованого дерева

В ордереві рис. 3.17 v_0 – прямий предок вершин v_1 і v_4, а вершини v_1 і v_4 – прямі нащадки вершини v_0. Наприклад, якщо дерево рис. 3.17 є генеалогічним деревом роду, то вершина v_0 – це засновник роду, вершини v_1 і v_4 – його діти, v_2, v_3, v_5, v_6 – онуки, v_7, v_8 – правнуки.

Вершина ордерева, яка не має прямих нащадків, є листком. В ордереві рис. 3.17 вершини v_2,v_3,v_6,v_7,v_8 є листям.

Графи виду "дерево" широко використовуються в задачах розроблення та дослідження систем оброблення даних, складання розкладів, управління проектами, планування робіт і багатьох інших застосуваннях.

3.7.5. *Турніри*. Така назва орграфів грунтується на тому, що їх можна можна застосовувати для запису результатів будь-яких змагань (турнірів), в яких не дозволені нічиї.

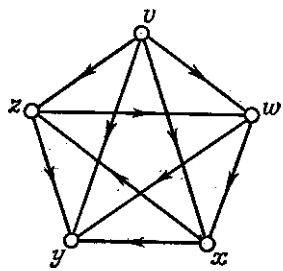


Рисунок 3.18 – Турнір

Означення 3.65. *Турніром* називається орграф, у якому будь-які дві вершини з'єднані рівно однією дугою.

Приклад турніру показаний на рис. 3.18. Його можна інтерпретувати, припустимо, так: команда z завдала поразки команді w, але програла команді v, і т. д.

Контрольні питання

Який граф називається двочастковим? Охарактеризуйте відмітну ознаку графів цього виду. Проілюструйте двочастковий граф прикладом його діаграми.

Який двочастковий граф називається: повним, збалансованим?

Охарактеризуйте відмітну ознаку роздільних графів.

Яку множину ребер (дуг) називають розрізом графа? Чи може граф мати декілька розрізів?

Який граф називається роздільним? Проілюструйте роздільний граф прикладом його діаграми.

Яке ребро графа називається мостом?

Які вершини графа називаються точками зчленування?

Які графи є роздільними (нероздільними)?

Який граф називається мережею? Охарактеризуйте відмітні ознаки графів цього виду. Проілюструйте мережу прикладом її діаграми.

Поясніть поняття потоку в дузі мережі.

Поясніть сутність закону збереження об'єму для вершини мережі.

Поясніть поняття потоку у вершині мережі.

Поясніть поняття потоку в мережі. В якому співвідношенні перебувають потоки в мережі, її витоці та стоці?

Поясніть поняття пропускної здатності мережі.

Який граф називається ациклічним?

Який граф називається лісом?

З яких компонентів складається ліс?

Наведіть означення дерева. Наведіть графічні приклади.

Яке співвідношення між кількостями вершин та ребер (дуг) існує у дереві?

Наведіть перелік специфічних властивостей дерев.

Який граф називається планарним?

Поясніть поняття орієнтованого дерева. Які з його вершин носять назви кореня та листя?

Чи можуть бути всі впорядковані пари вершин ордерева пов’язаними оршляхами?

Назвіть компоненти зв’язності дерева.

Наведіть графічний приклад ордерева з ранжуванням за рівнями.

Які вершини ордерева є предками та нащадками (прямими предками та нащадками) одна по відношенню до іншої?

Чи є у листів ордерева прямі нащадки?

Наведіть означення турніру та його графічний приклад.

3.8 Числові характеристики графів

Для кожного графа можна визначити ряд кількісних характеристик, пов’язаних з його структурою, властивостями та необхідних для різних розрахунків, зокрема визначення ізоморфізму графів.

Найпростішими кількісними показниками графа G є кількості $n(G)$ вершин та $m(G)$ ребер, набори степенів (напівстепенів) вершин. Зв’язність графа характеризується кількістю $q(G)$ його зв’язних компонент (зв’язних підграфів). Три зі згаданих показників визначають *циклохроматичне число* графа

$$\lambda(G) = m(G) - n(G) + q(G).$$

Циклохроматичне число тісно пов’язане з кількістю циклів, які можна побудувати на графі. В теорії графів розроблен методи визначення параметра $q(G)$. Для зв’язного графа без циклів $q(G) = 1$, $\lambda(G) = 0$.

Розглянемо характеристику графа, що називається хроматичним числом. Уявімо, що граф вершинно розфарбований різними кольорами. *Хроматичне число* $\gamma(G)$ дорівнює мінімальній кількості кольорів, якими можна розфарбувати вершини графа таким чином, що ніякі дві суміжні вершини не будуть зафарбовані однаковим кольором (рис. 3.19).

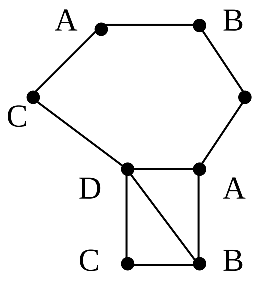


Рисунок 3.19 – Приклад розфарбованого графа. Кольори: A– червоний, B– жовтий, C– зелений, D– синій.

У теорії графів доведено, що будь-який планарний граф можна розфарбувати п’ятьма кольорами так, щоб ніякі дві суміжні вершини не були одного кольору (хроматичне число не перевищує 5). Таке розфарбування графа називають *правильним*. Існує гіпотеза (до цього часу не доведена), що будь-який *планарний* граф, зображений діаграмою на площині, можна розфарбувати правильно чотирма кольорами (проблема чотирьох кольорів).

Для реберно розфарбованих графів визначають хроматичний клас (індекс) графа. *Хроматичний клас* $\chi(G)$ дорівнює мінімальній кількості кольорів, якими можна правильно розфарбувати ребра графа.

Якщо граф містить багато вершин та ребер, то підбір мінімальної кількості кольорів важко здійснити. У такому разі для визначення хроматичного числа або класу застосовуються спеціальні алгоритми, які є в “арсеналі” цілочислового лінійного програмування.

Різноманітні задачі, що виникають при плануванні виробництва, складанні графіків ремонту обладнання або перевезень товарів і т. д., часто можуть бути подані як задачі розфарбування графів.

Тут ми розглянули лише головні числові характеристики графів. У теорії графів розроблено ще багато інших числових характеристик, причому повного їх переліку поки не визначено. Важливими задачами теорії графів є вираження складних для обчислення числових характеристик (хроматичного числа, хроматичного класу та ін.) через такі, що легко визначаються (кількості вершин та ребер, циклохроматичне число, степені та напівстепені вершин тощо).

Контрольні питання

Перелічіть усі відомі вам числові характеристики графів.

Наведіть означення циклохроматичного числа графа.

Наведіть означення хроматичного числа графа.

Наведіть означення хроматичного класу графа.

Яку мінімальну кількість фарб різного кольору необхідно використати, щоб розфарбувати правильно (вершинно, реберно) планарний граф (орграф)?

Наведіть графічні приклади правильно розфарбованих (вершинно та реберно) графів (орграфів, дерев).

3.9 Вправи

1. Граф (неорієнтований) $G(V,E)$ заданий множинами: $V=\{1,2,3,4,5,6\}$; $E=\{e_1,e_2,e_3,e_4,e_5\}$, $e_1=(1,2)$, $e_2=(1,5)$, $e_3=(2,5)$, $e_4=(3,4)$, $e_5=\{5,6\}$, $e_6=\{6,6\}$. Ребра графа мають вагові параметри: $p(e_1)$, $p(e_2)$, $p(e_3)$, $p(e_4)$, $p(e_5)$, $p(e_6)$.

Завдання: а) побудувати діаграму графа; б) скласти табличні списки суміжностей вершин та інциденцій і ваг ребер; в) скласти матриці суміжностей, інциденцій та ваг ребер; г) визначити степені всіх вершин; д) вважаючи граф $G(V,E)$ орієнтованим із зазначеною упорядкованістю пар вершин, інцидентних дугам $e_1 - e_5$, виконати завдання за пп. а), б), в) та визначити напівстепені заходу і виходу всіх вершин.

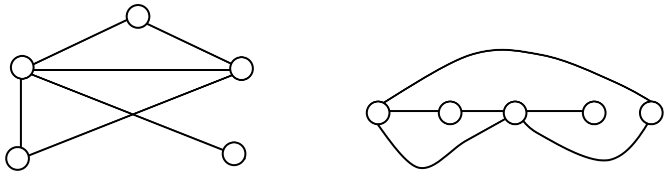
2. За заданою матрицею A суміжностей графа $G(V,E)$ побудувати його діаграму.

$$A=\begin{Vmatrix}0&1&1&0\\1&0&1&1\\1&1&0&1\\0&1&1&0\end{Vmatrix}$$

3. За заданою матрицею B інциденцій орграфа $G(V, E)$ побудувати його діаграму.

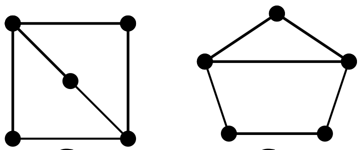
$$B = \begin{Vmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 \end{Vmatrix}$$

4. Довести ізоморфність графів G_1 і G_2.



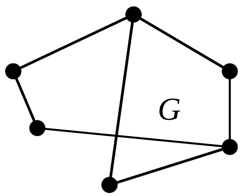
G_1 $\qquad\qquad$ G_2

5. Довести, що графи G_1 і G_2 неізоморфні.

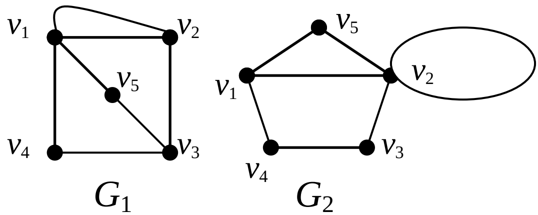


G_1 $\qquad\qquad$ G_2

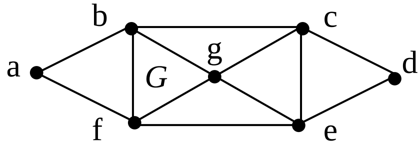
6. Побудувати діаграму доповнення $\overline{G} = (V, V^2 \setminus E)$ графа $G = (V, E)$.



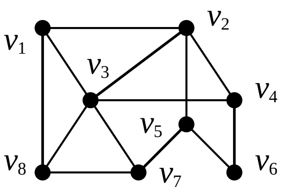
7. Для заданих графічно графів $G_1 = (V, E_1)$ і $G_2 = (V, E_2)$ побудувати діаграми графів: $G = G_1 \cup G_2$, $G = G_1 \cap G_2$, $G = G_1 \setminus G_2$, $G = G_1 - G_2$.



8. Побудувати власний та каркасний підграфи графа G (декілька варіантів кожного).

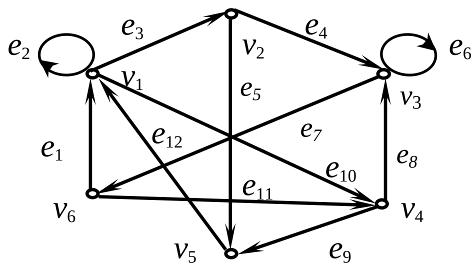


9. У даному графі визначити: всі маршрути між вершинами v_2 і v_6; всі прості шляхи $v_1 \rightarrow v_6$ і довжину кожного з них; прості ланцюги з визначених шляхів; найкоротший шлях $v_1 \rightarrow v_6$ (методом простого перебору); всі цикли.



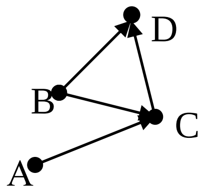
Дати відповідь на запитання: чи є цей граф зв'язним (сильно зв'язним)? Скласти матрицю досяжності.

10. У даному орграфі визначити: всі маршрути між вершинами v_2 і v_4; всі оршляхи із v_2 у v_4 ($v_2 \rightarrow v_4$) і довжину кожного з них; орланцюги з числа визначених ор-шляхів; найкоротший шлях



$v_2 \to v_4$ (методом простого перебору); всі орцикли (контури). Дати відповідь на питання: чи є цей граф зв'язним (сильно зв'язним)? Скласти матрицю досяжності.

11. Для даного графа визначити числові характеристики: кількості вершин і дуг; кількості вершин, степінь, напівстепені заходу і виходу яких мають значення відповідно 0, 1, 2, 3; кількість орциклів; кількості вершин – витоків та стоків; циклохроматичне та хроматичне числа; хроматичний клас.



Розділ 4

СКІНЧЕННІ АВТОМАТИ

(теоретичні основи)

Автомат (від грецького слова $\alpha\upsilon\tau o\mu\alpha\tau o\zeta$ – самодіючий) – це пристрій, що виконує свої функції без участі людини. Теорія автоматів як наукова дисципліна виникла в межах теорії керуючих систем у середині XX століття, в період бурхливого розвитку засобів електронної обчислювальної техніки та відповідних галузей математичного знання. На сьогодні ця теорія є одним із розділів дискретного аналізу.

Розвиток інформаційних технологій вивів сферу застосування теорії автоматів далеко за межі моделювання апаратних засобів цифрової електроніки, розширивши її до фундаментальних основ сучасної теоретичної інформатики. На цей час абстракції і моделі, розроблені в теорії автоматів, затребувані такими науковими дисциплінами, як теорія алгоритмів, теорія формальних граматик, математична лінгвістика, теорія логічних моделей, теорія кодування, теорія обчислювальної складності та іншими.

Теорію автоматів як теоретико-прикладну науку умовно поділяють на *теорію абстрактних автоматів* і *теорію структурних автоматів*, перша з яких є теоретичною базою другої. Математичні моделі в теорії абстрактних автоматів розглядають пристрої дискретної дії з точки зору алгоритмів їх функціонування, тобто послідовностей дій з перетворення дискретної інформації. Під час абстрактного аналізу та синтезу автомата головну роль відіграє не спосіб побудови автомата, а ті переходи зі стану в стан, які робить автомат під впливом вхідних сигналів, і ті вихідні сигнали, які автомат при цьому формує. Предметом дослідження теорії структурних автоматів є структурно-функціональна організація автоматів як технічних засобів реалізації їхніх математичних моделей. Під структурним аналізом і синтезом автомата розуміється розгляд структури самого автомата, вхідних і вихідних сигналів, а також дослідження способів його побудови із набору логічних елементів та елементарних автоматів, способів кодування внутрішніх станів і т. ін.

Знання з теорії автоматів є необхідними для успішного розроблення принципів побудови дискретних пристроїв обробки інформації, удосконалення відомих алгоритмів обробки інформації, грамотного застосування обчислювальної техніки в системах управління і автоматики та розроблення програмного забезпечення таких систем. У даному розділі розглядаються вихідні теоретичні положення теорії абстрактних автоматів. До числа основних належать задачі аналізу, синтезу, еквівалентних перетворень і мінімізації автоматів.

Задача аналізу полягає в тому, щоб за заданим автоматом описати його поведінку або за неповними даними про алгоритм функціонування автомата встановити його властивості.

Задача синтезу полягає в побудові автомата з наперед заданим алгоритмом функціонування. Задачу синтезу прийнято розглядати двояко: абстрактний синтез як побудова математичної моделі автомата і структурний синтез як розроблення його функціональної логічної схеми.

Задача еквівалентних перетворень формулюється так: визначити повну систему правил, що дозволяє перетворювати довільний автомат в еквівалентний йому автомат. Окремим випадком цієї задачі є перехід від однієї моделі автомата до іншої.

Задача мінімізації полягає в побудуві *мінімального* автомата, еквівалентного заданому. Мінімальний автомат має найменше число компонентів моделі (зокрема мінімальну потужність множини станів) і при цьому функціонально еквівалентний заданому автомату.

4.1 Поняття автомата. Види автоматів

У технічних системах автоматами називають пристрої, за допомогою яких здійснюються операції прийому, перетворення і передачі матеріальних об'єктів або інформації у відповідності із певною програмою, без безпосередньої участі людини. Задачі синтезу та аналізу технологічних автоматів, тобто таких, що здійснюють операції прийому, перетворення і передачі матеріальних об'єктів, належать до галузі машинобудування.

За функціональним призначенням автомати, що застосовуються в системах керування, поділяються на три види: інформаційні, керуючі та обчислювальні. До інформаційних автоматів належать різноманітні довідкові таблиці, наприклад, на вокзалах, інформаційні табло на стадіонах, різні пристрої аварійної сигналізації. До керуючих автоматів прийнято відносити пристрої для керування технічними об'єктами шляхом формування команд керування в дискретній формі. До обчислювальних автоматів відносять ЕОМ, мікроконтролери. Однак усі ці автомати, по суті, є одночасно обчислювальними, керуючими та інформаційними за сукупністю процесів, що в них відбуваються. Такі автомати є основою сучасної обчислювальної техніки та систем автоматичного контролю і керування.

Під терміном "*автомат*" у дискретному аналізі розуміють абстрактну модель пристрою (*абстрактний автомат*), що функціонує у дискретному часі, переробляючи потоки вхідних сигналів в потоки вихідних сигналів. На мові теорії множин це означає, що абстрактний автомат відображає множину вхідних сигналів у множину вихідних сигналів. В абстрактному автоматі вхідні та вихідні сигнали мають вид послідовностей символів, тобто слів у відповідних (вхідному, вихідному) *алфавітах*. В процесі роботи автомата здійснюється послідовна зміна його станів у дискретні моменти часу під впливом вхідних сигналів.

Дискретизація реального часу базується на понятті *автоматного часу*. Шкала дискретного часу зображена на рис. 4.1. На осі часу цілими числами 0, 1, 2, ..., i позначені моменти часу t_i, якими виділяються дискретні інтервали часу Δt_i,упродовж яких вхідні і вихідні сигнали автомата можуть змінювати своє значення. Інтервали часу Δt_i називаються *тактами*, пронумерованими в порядку слідування числами 0, 1, 2, ..., i .

Рисунок 4.1 – Шкала дискретного часу: i – номер такта

За фіксований інтервал часу $\Delta t_i = t_{i+1} - t_i$ (рис. 4.1) здійснюється перетворення одного символа вхідного алфавіту $Z = \{z_1, \ldots, z_f, \ldots, z_F\}$ в один символ вихідного алфавіту $W = \{w_1, \ldots, w_g, \ldots, w_G\}$ та перехід абстрактного автомата зі стану $q(t_i)$ в стан $q(t_{i+1})$. Поняття автоматного часу випливає з описаного принципу дискретизації реального часу. Автоматний час є безрозмірною величиною та має цілочислове значення, *мірою* якого є кількість тактів. Довільному інтервалу Δt (будь-якої тривалості) реального часу ставиться у відповідність момент t автоматного часу, який має бізрозмірне цілочислове значення.

Будь-який дискретний пристрій обробки інформації в загальному виді являє собою перетворювач, що здійснює перетворення кортежу вхідних сигналів $X = (x_1, x_2, \ldots, x_n)$ у кортеж вихідних сигналів $Y = (y_1, y_2, \ldots, y_m)$. На рис. 4.2 такий пристрій представлено у вигляді “чорного” ящика.

Алфавіти, в яких відображаються можливі значення вхідних та вихідних сигналів дискретного перетворювача, є скінченними, вхідні та вихідні сигнали є дискретними, і перетворення сигналів здійснюється в дискретні моменти часу. Тому такі перетворювачі інформації називаються *скінченними*.

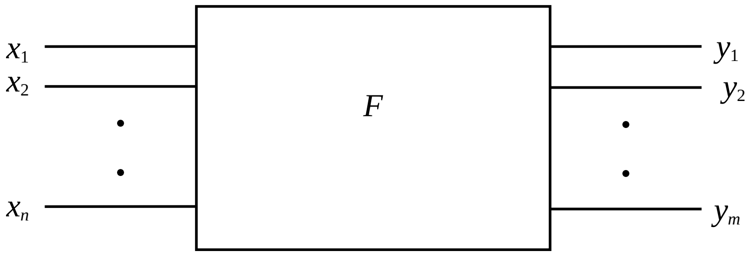


Рисунок 4.2 – Пристрій обробки дискретної інформації

Абстрактний автомат за сутністю процесу його функціонування є скінченним перетворювачем дискретної інформації. Символи z_f і w_g алфавітів $Z=\{z_1,\ldots,z_f,\ldots,z_F\}$ і $W=\{w_1,\ldots,w_g,\ldots,w_G\}$ входу і виходу в загальному випадку являють собою скінченні кортежі $(x_1,x_2,\ldots,x_n)$ і $(y_1,y_2,\ldots,y_m)$ вхідних і вихідних сигналів автомата (рис. 4.2), необов’язково однорозрядні ($n\geq1$, $m\geq1$).

У всіх пристроях збірання, обробки та передачі інформації, що застосовуються в системах управління та автоматики, вхідні і вихідні сигнали подаються в цифровій формі, тобто у процедурах перетворення інформації об’єктом обробки є цифрові сигнали, над якими здійснюються обчислювальні операції або логічні перетворення. Логічні сигнали віднесені до цифрових на тій підставі, що вони є квантованими за двома рівнями і набувають числових значень 0 та 1 (алфавіт логічних змінних). База (упорядкований список n аргументів) логічної функції також є цифровим сигналом (n-розрядним). У подальшому як алфавіти вхідних та вихідних сигналів скінченних автоматів ми будемо використовувати множину символів, що в цифрових автоматах набувають значення із множини $\{0,1\}$.

Відзначимо, що в літературних джерелах з дискретного аналізу зустрічаються назви “дискретний автомат”, “дискретний пристрій”, “пристрій дискретної дії”, що рівнозначні за сенсом узагальненій назві “скінченний автомат” і використовуються в окремих випадках з метою віддзеркалення певних характеристичних ознак реальних автоматів. Якщо мова йде про дискретні пристрої обчислювальної техніки та автоматики, вхідний та вихідний алфавіти яких являють собою множини цифр, то такі пристрої називають *цифровими автоматами*. Нижче ми дамо формалізовані означення скінченного та цифрового автоматів.

Оскільки предметом вивчення у цьому розділі є елементи теорії абстрактних автоматів, надалі будемо користуватися термінами “скінченний автомат”, або просто “автомат”. Продовжимо огляд видів автоматів.

Залежно від того, одночасно чи послідовно автомат здійснює прийом вхідного сигналу та зміну свого стану, автомати поділяються на синхронні та асинхронні.

У *синхронних* автоматах тривалості вхідних сигналів і час переходу з одного стану в інший погоджені. Моменти часу, в які фіксуються зміни стану автомата, задаються спеціальним пристроєм – генератором синхросигналів. Він видає імпульси, якими задаються такти автоматного часу. Такі автомати застосовуються в обчислювальних комплексах, автоматизованих системах керування технічними об’єктами та ін.

В *асинхронних* автоматах тривалість вхідних сигналів і час переходів не погоджені. Ці параметри залежать від зовнішніх джерел – різних подій, а інтервал дискретності Δt є змінним (наприклад, у кодових замках). В асинхронних автоматах чергова зміна значень вхідних сигналів може відбутися лише за умови, що закінчився перехідний процес, викликаний попередньою зміною цих сигналів.

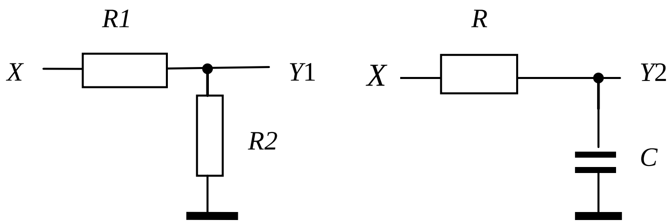
Якщо принципом класифікації є механізм випадкового вибору, то розрізняють *детерміновані* та *ймовірнісні* (стохастичні) автомати. Поведінка детермінованого автомата в кожний момент дискретного часу визначається поточною вхідною інформацією та станом автомата у попередній момент часу. У детермінованих автоматах виконується умова однозначності переходів: якщо автомат знаходиться в деякому стані $q_i \in Q$, де Q – множина станів, то під впливом довільного вхідного сигналу $z_k \in Z$ автомат може перейти в один і тільки один стан $q_j \in Q$, причому ситуація $q_i = q_j$ зовсім не виключається. У ймовірнісних автоматах ця залежність пов'язана ще і з деяким випадковим вибором. Ймовірнісний автомат — це дискретний перетворювач інформації, функціонування якого в кожний момент часу залежить лише від стану пам'яті та описується статистичними законами. У ймовірнісних автоматах під впливом одного й того самого вхідного сигналу можливі переходи зі стану q_i в різні стани з множини Q із заданою ймовірністю.

Якщо принципом класифікації автоматів є об'єм пам'яті, то різниця полягає в тому, що автомат має скінченне або нескінченне число внутрішніх станів. За цим принципом розрізняють наступні типи скінченних автоматів.

Автоматами 1-го типу, або *комбінаційними схемами* (КС), прийнято називати такі автомати, сукупність вихідних сигналів яких визначається лише комбінацією вхідних сигналів у поточний момент часу t_i і не залежить від сигналів, що надходили на входи автомата у попередні моменти часу. Такі автомати називають також *автоматами без пам'яті*, або *скінченними функціональними перетворювачами*. З автоматами цього типу ми зустрічалися, розглядаючи технічні засоби реалізації логічних функцій (розд. 2, п. 2.4).

В автоматах 2-го типу вихідний сигнал, що виробляється в поточний момент часу, залежить не лише від вхідних сигналів, що надійшли в цей самий момент, але й від сигналів, що надійшли у попередні моменти часу. Попередні вхідні сигнали фіксуються в автоматі шляхом зміни його внутрішнього стану. Такі автомати називають *автоматами з пам'яттю*.

Приклад 4.1. Поставимо таке завдання: проаналізувати роботу електричних схем, показаних на рис. 4.3а,б, та зробити висновок щодо типів автомата, яким відповідають принципи роботи даних схем. Вважати при цьому, що вхідні напруги *X* надходять на входи схем у дискретні моменти часу t_1 і t_4, а вихідні напруги *Y1* та *Y2* в момент t_0 дорівнювали 0.



а) б)

Рисунок 4.3 – Електричні схеми до прикладу 4.1

Розв'язування

Для аналізу поведінки схем визначимо напруги на їхніх виходах. На виході схеми рис. 4.3,а вихідна напруга в момент часу t_1

$$Y1 = Y1(t_0) + Y1(t_1) = 0 + X \cdot R1/(R1+R2) = X \cdot R1/(R1+R2).$$

Оскільки в схемі немає реактивних елементів, енергія не накопичується, і після зняття вхідної напруги у момент t_2 вихідна напруга дорівнюватиме 0. Отже, в момент t_4 подачі напруги *X* на вхід схеми її вихідна напруга знову стане дорівнювати

$$Y1 = X \cdot R1/(R1+R2).$$

Отже, робимо висновок, що вихідний сигнал схеми рис. 4.3а у поточний момент часу залежить лише від вхідного сигналу у цей самий момент часу, тому її поведінка подібна до поведінки автомата 1-го типу, тобто комбінаційної схеми.

Напруга на виході схеми рис. 4.3б після подачі вхідної напруги X у момент часу t_1 і зняття її у момент часу t_2 стане такою, що дорівнює

$$Y2(t_2)=X\cdot(1-e^{-\tau/RC}),$$

де $\tau=t_2-t_1$ – тривалість одного такту часу.

Після зняття вхідної напруги до моменту t_4 вихідна напруга зменшиться до величини

$$Y2(t_4)=X\cdot(1-e^{-\tau/RC})\cdot e^{-2\tau/RC}.$$

Отже, після подачі напруги на вхід цієї схеми в момент t_4 і зняття її у момент t_5 вихідна напруга стане такою, що дорівнює

$$Y2(t_5)=Y2(t_4)+Y2(t_4)\,(1-e^{-\tau/RC})=X\cdot(1-e^{-\tau/RC})(1+e^{-2\tau/RC}).$$

Отже, через наявність реактивного елемента в схемі рис. 4.3б, який запасає енергію попередніх станів, ми бачимо, що її виходний сигнал залежить не лише від вхідного сигналу в поточний момент часу, але й від вхідних сигналів у попередні моменти часу. Це робить її поведінку схожою на поведінку автоматів 2-го типу, тобто автоматів з пам'яттю.

Наведений приклад не описує поведінку реальних автоматів, а лише показує на прикладі аналогових схем, що вихідні сигнали можуть залежати, а можуть і не залежати від значень вхідних сигналів у попередні моменти часу залежно від того, які елементи входять до даних схем, чи є серед них накопичувачі енергії.

Сукупність вихідних сигналів автомата з пам'яттю залежить не лише від вхідного впливу, але й від стану автомата в поточний та попередній моменти часу. Цими самими факторами визначається і той стан, до якого автомат переходить.

Якщо скінченний автомат забезпечити (теоретично) необмеженою пам'яттю, то отримаємо автомат 3-го типу – *нескін-*

ченний автомат, який називається *машиною Тьюринга*. За допомогою машини Тьюринга можна реалізувати будь-який алгоритм перетворення інформації. Нескінченний автомат являє собою певну математичну ідеалізацію скінченного автомата, який має нескінченне число станів. Абстрактний автомат – машина Тьюринга є нескінченним, але ЕОМ або її окремі частини є скінченними автоматами.

Контрольні питання

Дайте найбільш загальне визначення автомата.

З яких двох частин скаладається теорія автоматів?

Що є предметом теорії абстрактних автоматів?

Які задачі розв’язуються у теорії структурних автоматів?

Опишіть суть задачі аналізу автомата.

Опишіть суть задачі синтезу автомата.

У чому полягає задача еквівалентних перетворень автомата?

Як формулюється задача мінімізації автомата?

Назвіть види автоматів за функціональним призначенням.

Який об’єкт розуміється під терміном “*автомат*” у дискретному аналізі?

Поясніть поняття автоматного часу. Що є його мірою?

Яку функцію виконує дискретний пристрій обробки інформації у загальному виді? Чи є цей пристрій скінченним автоматом?

У якому випадку скінченний автомат є цифровим?

У чому полягає різниця функціональних властивостей синхронних та асинхронних автоматів?

За якою ознакою автомат є детермінованим?

Поясніть поняття автомата без пам’яті.

Поясніть поняття автомата із пам’яттю.

За якими ознаками автомат класифікується як такий, що належить до автоматів 1-го типу?

За якими ознаками автомат класифікується як такий, що належить до автоматів 2-го типу?

Назвіть характеристичну ознаку, за якою автомат класифікується як нескінченний.

Чи здатна машина Тьюринга реалізувати будь-який алгоритм обробки інформації?

4.2 Абстрактні автомати

Математичну модель дискретного пристрою називають *абстрактним автоматом* згідно з наступним означенням.

Означення 4.1. *Абстрактний автомат* – це шестикомпонентний кортеж $A=(Z, W, Q, \delta, \lambda, q_0)$, де:

$Z=\{z_1, z_2,\dots,z_f,\dots, z_F\}$ – множина символів вхідного алфавіту (вхідних сигналів);

$W=\{w_1, w_2,\dots, w_g,\dots, w_G\}$ – множина символів вихідного алфавіту (вихідних сигналів);

$Q=\{q_1,\dots,q_m,\dots,q_M\}$ – множина станів (алфавіт станів);

δ: $Q\times Z\rightarrow Q$ – функція переходів, що деяким парам (q_m, z_f) "стан – вхідний сигнал" ставить у відповідність певні стани автомата $q_s=\delta(q_m, z_f)$, $q_s\in Q$;

λ: $Q\times Z\rightarrow W$ – функція виходів, що парам (q_m, z_f) ставить у відповідність вихідні сигнали автомата $w_s=\lambda(q_m, z_f)$, $w_s\in W$;

$q_0\in Q$ – початковий стан автомата.

Абстрактний автомат має один вхідний і один вихідний канали (рис. 4.4). Абстрактний автомат, в якому виділено один із станів q_0, називається *ініціальним*. Це стан, із якого автомат починає роботу і в який він переходить по її закінченні. У деяких задачах такий початковий стан не є необхідним, тоді абстрактний автомат описується п'ятикомпонентним кортежем $A=(Z, W, Q, \delta, \lambda)$.

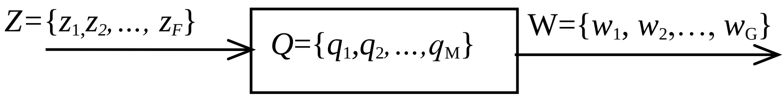


Рисунок 4.4 – Абстрактний автомат

Означення 4.2. Автомат $A=(Z,W,Q,\delta,\lambda,q_0)$ називається *скінченним*, якщо алфавіти Z і W входу і виходу та множина Q його станів скінченні.

Надалі ми будемо мати справу лише зі скінченними автоматами.

Означення 4.3. Якщо символи вхідного і вихідного алфавітів Z і W автомата кодуються цифрами, то такий автомат називається *цифровим*.

Функціонування автомата розглядається в дискретні моменти автоматного часу $t_1, t_2, \ldots, t_i$, які називаються тактовими моментами. Припускається, що поведінка автомата не залежить від інтервалу часу між t_i і t_{i+1}. Таким чином, фактично змінною є не сам час, а порядкові номери тактових моментів $t=0, 1, 2, \ldots, i, \ldots$. Реальна ж тривалість тактів може бути довільною. Береться також припущення, що перехід автомата із одного стану в наступний та формування нового (а можливо, і того ж самого) символу на виході здійснюється стрибкоподібно, тобто миттєво в момент t автоматного часу. В реальних же автоматах завжди має місце кінцева тривалість перехідних процесів, тобто функції δ і λ переходів і виходів технічно реалізовуються упродовж інтервалу реального часу, що відповідає даному моменту t автоматного часу.

Процес функціонування абстрактного автомата можна уявити наступним чином. На кожному такті t автомат перебуває в деякому стані $q(t)=q_m \in Q$, а на початковому такті ($t=0$) – у початковому стані $q(0)=q_0$. Сприймаючи в момент t автоматного часу деякий символ вхідного алфавіту $z(t)=z_f \in Z$ на вхідному каналі, автомат видає на вихідному каналі в цей самий момент символ вихідного алфавіту $w(t) = w_g \in W$ і переключається на новий стан $q(t+1) = q_s \in Q$, який зберігатиметься протягом наступного такту часу. Вихідний сигнал і наступний стан визначають функції виходів λ і переходів δ:

$$w(t) = \lambda(q(t), z(t)), \tag{4.1}$$

$$q(t+1) = \delta(q(t), z(t)). \tag{4.2}$$

Таким чином, функціонування автомата може розглядатись на рівні абстрактної теорії як процес перетворювання вхідних слів (послідовностей вхідних символів) у вихідні слова.

Залежно від способу одержання значень вихідних сигналів є два типи автоматів – автомати Мілі та Мура. Виразам (4.1) та (4.2) відповідає автомат, вихідний сигнал якого залежить від його поточного стану і від сигналу на його вході. Це *автомат Мілі.*

Автомат, вихідний сигнал $w(t)$ якого у такті t формується незалежно від вхідного сигналу $z(t)$, а лише залежно від його поточного внутрішнього стану, називають *автоматом Мура*. Функціонування такого автомата описують вирази:

$$w(t) = \lambda(q(t)), \tag{4.3}$$

$$q(t+1) = \delta(q(t), z(t)). \tag{4.4}$$

Будь-який реальний автомат можна спроектувати за моделлю Мілі або Мура, що доведено у п. 4.5.

Контрольні питання

Наведіть означення абстрактного автомата.

За яких умов абстрактний автомат вважається ініціальним?

Наведіть означення скінченного автомата. Чим воно відрізняється від означення абстрактного автомата?

Наведіть аналітичний вираз функції переходів автомата. Який процес вона відображає?

Наведіть аналітичні вирази функції виходів для автоматів Мілі і Мура. В чому полягає різниця процесів формування вихідних сигналів у цих автоматах?

Дайте опис процесу функціонування автомата.

Наведіть математичні моделі процесів функціонування автоматів Мілі та Мура.

Наведіть означення цифрового автомата. Чи є цифровий автомат скінченним?

4.3 Способи задання автоматів

Від задачі побудови реального автомата до формальної моделі переходять, описуючи насамперед алгоритм його роботи. Потім одним із способів задання автомата формують його математичну модель. Щоб задати абстрактний автомат A, з умови задачі треба описати всі елементи кортежу $A=(Z, W, Q, \delta, \lambda, q_0)$, тобто алфавіти вхідний, вихідний та станів, а також функції переходів і виходів. Задають автомат найчастіше табличним, графічним та аналітичним (з допомогою логічних виразів, див. п. 2.4) способами.

4.3.1 Табличний спосіб задання автомата Мілі

У разі табличного способу функції виходів λ (4.1) та переходів δ (4.2), які описують роботу автомата Мілі, задають у вигляді таблиць виходів (табл. 4.1) і переходів (табл. 4.2). Рядки цих таблиць позначають символами вхідних сигналів з множини Z, а стовпці – символами станів з множини Q. У клітинці на перетині стовпця $q(t)=q_m$ і рядка z_f у таблиці переходів (табл. 4.2) поставлено стан $q(t+1)=q_s$, в який переходить автомат із стану q_m під дією сигналу z_f. У відповідній клітинці таблиці виходів (табл. 4.1) поставлено вихідний сигнал $w(t) = w_g$, сформований під час переходу автомата із стану q_m у стан q_s.

У повністю визначеного автомата функції λ і δ визначено на всій множині пар $(q_m, z_f) \in Q \times Z$, тобто усі клітинки таблиць заповнено. Табл. 4.1 та 4.2 повністю задають закон функціонування автомата Мілі, оскільки за ними можна довідатися про його поведінку в будь-який момент дискретного часу. Наприклад, у момент t на автомат, заданий табл. 4.1 та 4.2, який перебуває у стані q_1, надходить вхідний сигнал z_2. Треба з'ясувати, як поводитиметься цей автомат. Для цього за табл. 4.2 знаходимо, що автомат перейде в новий стан q_2 і на його виході з'явиться сигнал $w_3 \in W$ (див. табл. 4.1). Клітинки в табл. 4.1 та 4.2, які відповідають цьому, обведені рамками.

Таблиця 4.1

$z(t)$ \ $q(t)$	q_0	q_1	q_2	q_3
z_1	w_3	w_2	w_1	y_1
z_2	w_4	$\boxed{w_3}$	w_4	w_2
z_3	w_1	w_2	w_4	w_3

Таблиця 4.2

$z(t)$ \ $q(t)$	q_0	q_1	q_2	q_3
z_1	q_1	q_2	q_0	q_3
z_2	q_3	$\boxed{q_2}$	q_0	q_0
z_3	q_2	q_1	q_2	q_3

Таблиця 4.3

$z(t) \backslash q(t)$	q_0	q_1	q_2	q_3
z_1	q_1 / w_3	q_2 / w_2	q_0 / w_1	q_3 / w_1
z_2	q_3 / w_4	q_2 / w_3	q_0 / w_4	q_0 / w_2
z_3	q_2 / w_1	q_1 / w_2	q_2 / w_4	q_3 / w_3

Легко бачити, що табл. 4.1 та 4.2 можна з'єднати в одну суміщену таблицю (табл. 4.3), у клітинках якої буде записано значення функцій δ і λ.

Означення 4.4. Автомат називають *частково визначеним* якщо в табл. 4.1 і 4.2 заповнено не всі клітинки, тобто функції λ=(q_m, z_f) та δ=(q_m, z_f) задані не для всіх пар $(q_m, z_f) \in Q \times Z$. Приклад такого автомата наведено в таблицях:

	q_1	q_2	q_3
z_1		w_1	w_1
z_2	w_1	w_2	
z_3			w_3

	q_1	q_2	q_3
z_1		q_1	
z_2	q_2	q_3	
z_3			q_3

4.3.2 Графічний спосіб задання автомата Мілі

Графічний спосіб задання автомата Мілі передбачає подання його у вигляді орієнтованого графа, вершини якого відповідають станам автомата, а дуги – переходам між ними (рис. 4.5).

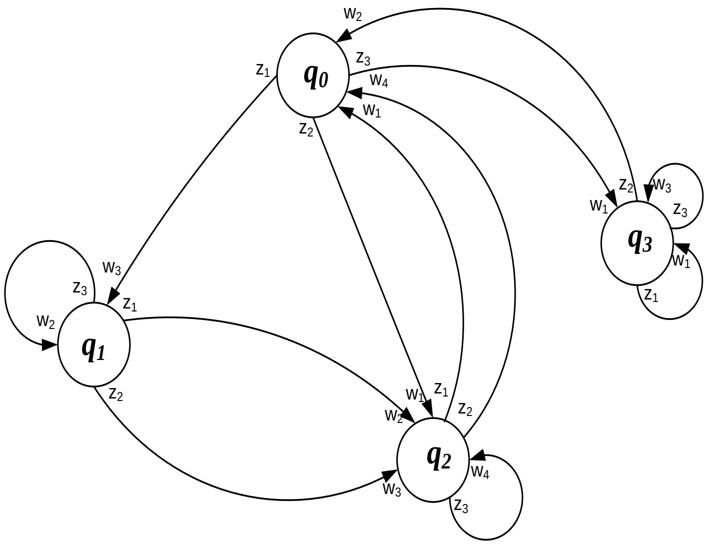


Рисунок 4.5 – Граф автомата Мілі, заданого табл. 4.1 і 4.2

Дві вершини графа автомата q_m та q_s (стани початковий та переходу) з'єднують дугою, спрямованою від q_m до q_s, коли в автоматі є перехід з q_m у q_s, тобто коли $q_s = \delta(q_m, z_f)$ для деякого вхідного сигналу $z_f \in Z$.

Початок дуги (q_m, q_s) графа автомата помічають вхідним z_f, а кінець – вихідним сигналом $w_g = \lambda(q_m, z_f)$ (або рискою, коли w_g не визначено у частково визначеному автоматі).

4.3.3 Табличний спосіб задання автомата Мура

Як і для автомата Мілі, для автомата Мура використовують табличний та графічний способи задання. Автомат Мура задають так званою позначеною таблицею переходів (табл. 4.4), в якій кожен стовпець позначено станом $q(t) = q_m$ і вихідним сигналом $w(t) = w_g = \lambda(q_m)$, що відповідає цьому стану.

Таблиця 4.4

$w(t)$	w_1	w_2	w_3	w_3, w_4	w_3	w_3, w_4
$q(t)$ / $z(t)$	q_0	q_1	q_2	q_3	q_4	q_5
z_1	q_1	q_0	q_3	q_1	q_5	q_0
z_2	q_2	q_4	q_3	q_1	q_5	q_0

Таблиця 4.4 описує одночасно і функцію переходів δ, і функцію виходів λ.

4.3.4 Графічний спосіб задання автомата Мура

Так само, як і для автомата Мілі, графічний спосіб задання автомата Мура передбачає подання його у вигляді орієнтованого графа, вершини якого відповідають станам автомата, а дуги – переходам між ними. В графі автомата Мура його вершини також ототожнюють із станами, але значення вихідних сигналів записують біля вершин графа, а дуги позначають лише вхідними сигналами (рис. 4.6).

Позначення дуг графа символом “1” на рис. 4.6 відповідає тому, що перехід між певними станами наявний завжди, незалежно від значень вхідних сигналів.

4.3.5 Задання автоматів граф-схемами алгоритмів

Для задання автомата за граф-схемою алгоритму (рис. 4.7) необхідно виконати розмітку граф-схеми відповідно до обраного типу автомата. Для автомата Мілі розмітка повинна здійснюватися наступним чином.

1. Усі операторні вершини повинні бути позначені символами вихідного алфавіту w_1, w_2, ..., w_G. Однакові операторні вершини позначають однаковими символами w_g.

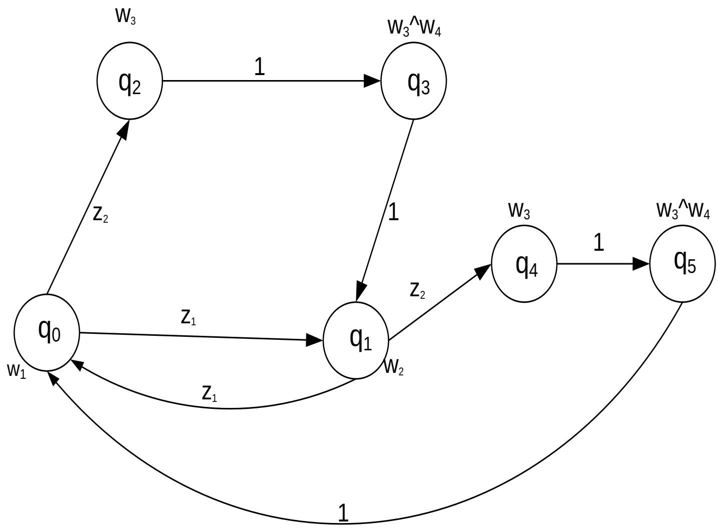


Рисунок 4.6 – Граф автомата Мура, заданого табл. 4.4

2. Усі умовні вершини позначають символами z_1, z_2, ..., z_F. Однакові умовні вершини позначають однаковими символами z_f.

3. Виходи операторних вершин позначаються символами внутрішніх станів автомата q_0, q_1, ..., q_M.

Для автомата Мура розмітка виконується аналогічно, за винятком станів, які проставляються біля операторних вершин. На рис. 4.7 стани автомата Мура позначені символами b_0, b_1, ..., b_M (щоб відрізнити їх від станів q_0, q_1, ..., q_M , які вже були використані для розмітки автомата Мілі).

Переходи автомата зі стану в стан здійснюються під впливом вхідних сигналів, що відповідають виконанню або не- виконанню умови z_f. Невиконанню умови відповідає інверсне значення змінної z_f, виконанню умови – пряме значення змінної.

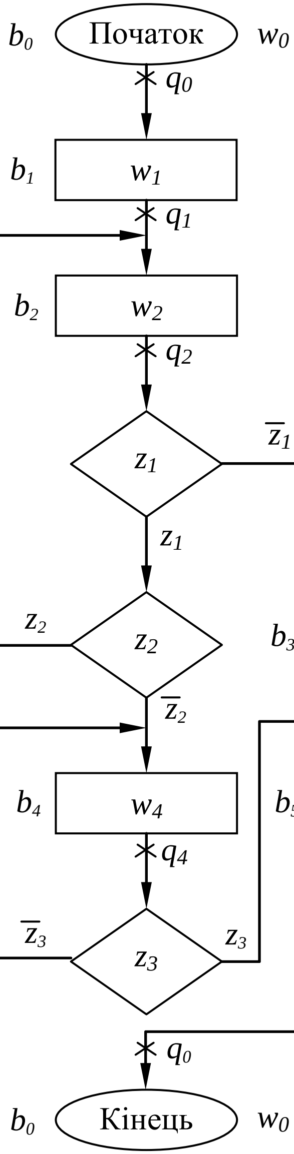


Рисунок 4.7 – Граф-схема алгоритму роботи автомата

При виконанні або невиконанні кількох умов одночасно перехід здійснюється під впливом сигналу, що подаються кон'юнкцією відповідних прямих або інверсних значень змінних.

Після закінчення виконання алгоритму автомати обох типів повинні повертатися до початкового стану.

Граф-схеми алгоритмів роботи автоматів дуже часто використовуються для задання керуючих автоматів процесорів, в яких робота операційного блока з виконання деякої операції описується насамперед алгоритмом.

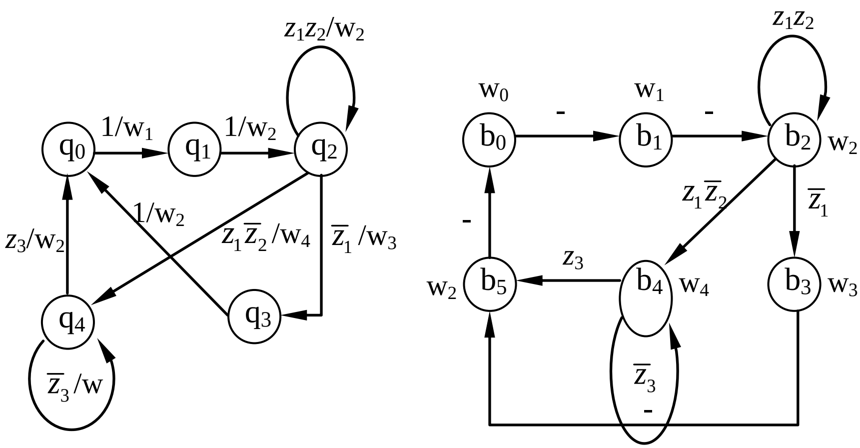
Граф-схема алгоритму, що показана на рис. 4.7, розмічена для задання як автомата Мілі, так і автомата Мура з урахуванням зазначеної вище різниці позначень операторних вершин для цих автоматів. Відповідні цьому алгоритму графи автоматів Мілі і Мура подані на рис. 4.8 а і 4.8 б відповідно.

Порівнюючи між собою ці графи, помічаємо, що в загальному випадку кількість станів в автоматі Мура більша, ніж в автоматі Мілі, що призводить до вищих затрат апартних ресурсів на пам’ять автомата.

4.3.6 Перехід від вербального до формального опису автомата

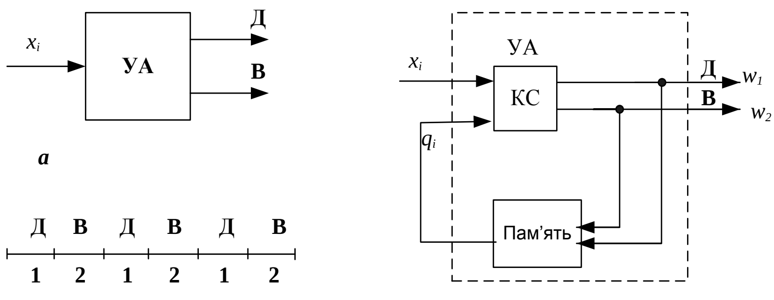
Одним із найпростіших є автомат регулювання дорожнього руху на перехресті (автомат керує світлофором, рис. 4.9).

Вхідний сигнал x_i автомата *УА* (рис. 4.9 а) – це черговий i-й імпульс тактової послідовності, який поділяє час на інтервали з однаковою тривалістю, наприклад 1 хв. Цей сигнал дозволяє рух вулицею *Д*, якщо в (i–1)-й інтервал часу було відкрито рух вулицею *В*, і навпаки. Інакше кажучи, залежно від того, в якому стані перебував автомат у попередній момент часу, він видає вихідний сигнал *Д* чи сигнал *В* після надходження вхідного сигналу x_i.



а) б)

Рисунок 4.8 – Графи автомата Мілі (а) і автомата Мура (б), що відповідають граф-схемі алгоритму рис. 4.7



в *б*

Рисунок 4.9 – Структурна схема дорожнього автомата (а), його склад (б) і часова діаграма роботи (в)

Отже, автомат повинен мати пам'ять, яка б запам'ятовувала його попередній стан q_i, і комбінаційну схему КС, що формувала б вихідний сигнал (w_1, w_2) за сигналом (x_i, q_i) (рис. 4.9 б).

Щоб краще зрозуміти процес формування математичної моделі автомата, розглянемо вже згадуваний дорожній автомат, трохи ускладнивши його. Припустимо, що автомат керує не лише світлофором, що регулює рух транспорту на перехресті, а й рухом пішоходів. З цією метою кнопкою виклику можна надіслати автоматові сигнал запиту z_1, щоб рух транспорту припинився. Ця кнопка має пам'ять і повертається до початкового стану за сигналом *С* "Скид". Як уже було зазначено, автомат завдяки генераторові тактових імпульсів працює в дискретному часі. Він виробляє сигнали *Д, В, П, С* – дозволи рухатися транспорту вулицями *Д, В*, переходити пішоходам *П* (вулицями *Д* і *В*) та скид запиту *С* (заборона переходів).

Автомат працює так (рис. 4.10 а). Коли немає запитів $z_1 = 1$ на припинення руху транспорту, він щохвилини перемикає сигнали *Д* і *В*. Із надходженням запиту $z_1 = 1$ після закінчення поточного відрізка часу 1 хв автомат перериває послідовність сигналів *Д, В* на 2 хв, упродовж цього часу сигналом *П* засвітлює транспарант, що дозволяє пішоходам переходити, формує сигнал скиду *С*, після чого поновлює перервану послідовність видачі сигналів *Д, В*.

Тепер зробимо формальний опис автомата, задавши алфавіти вхідних, вихідних сигналів та його станів, а також описавши реакцію автомата на вхідні сигнали.

На підставі часової діаграми та граф-схеми алгоритму роботи автомата дорожнього руху (рис. 4.10) можна довідатися, що він має шість різних станів. Перші чотири з них відрізняються один від одного вихідними сигналами: 1 і 2 – дозволи проїжджати вулицями відповідно *Д* і *В*; *П* – дозвіл переходити пішоходам; 4 – дозвіл переходити й вилучення запиту пішоходів (*П*, *С*).

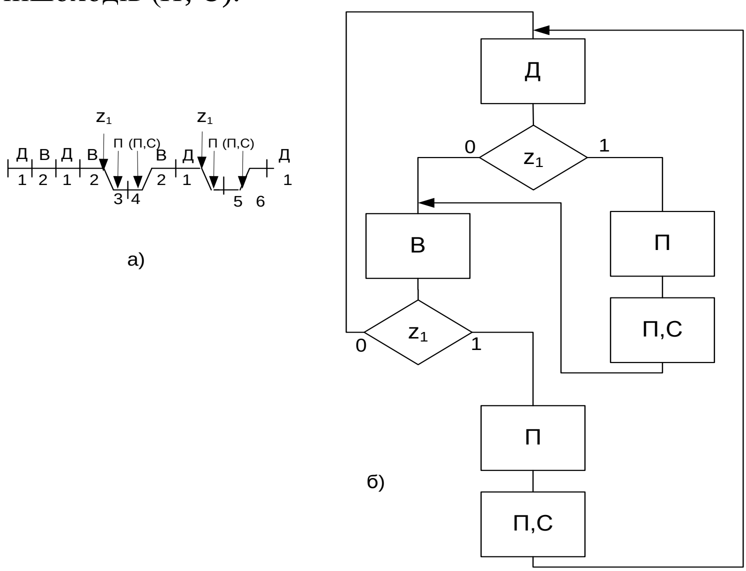


Рисунок 4.10 – Автомат дорожнього руху: а) часова діаграма роботи; б) граф-схема алгоритму

Крім того, з рис. 4.10 б бачимо, що до стану 3 можна перейти із стану 1 або 2, а із стану 4 – до стану 1 або 2. Тому є ще два стани – 5 і 6. Таким чином, алфавіт станів складається з шести символів:

$$Q_{ДА} = \{q_0, q_1, q_2, q_3, q_4, q_5\}.$$

Якщо за початковий взяти стан *Д* (хоча це для даного автомата не принципово), то символи алфавіту станів можна закріпити за конкретними станами (рис. 4.10 б). Алфавіт вихідних сигналів $W_{ДА} = \{w_1, w_2, w_3, w_4\}$ можна означити так: $w_1 = Д; w_2 = В; w_3 = П; w_4 = С$.

Сигнали зовнішнього середовища стосовно дорожнього автомата – це запити пішоходів, тому алфавіт вхідних сигналів має вигляд $Z_{ДА} = \{z_1, \overline{z}_1\}$.

Таким чином, виконано першу частину процедури формування математичної моделі дорожнього автомата – задано алфавіти вхідних, вихідних сигналів і станів Z, W та Q, також визначено початковий стан q_0.

Контрольні питання

Назвіть відомі вам способи задання автоматів.

Наведіть приклад табличного задання автомата Мілі.

Які елементи кортежу A абстрактного автомата описують таблиця переходів і таблиця виходів автомата Мілі?

Наведіть приклад графічного задання автомата Мілі.

Наведіть приклад табличного задання автомата Мура.

Як відрізняється позначена таблиця переходів автомата Мура від таблиці переходів автомата Мілі і які функції вона описує?

Наведіть приклад графічного задання автомата Мура.

Як здійснюється розмітка ГСА для автоматів Мілі і Мура?

4.4 Аналіз абстрактних автоматів

Як було визначено на початку цього розділу, *задача аналізу* полягає в тому, щоб за заданим автоматом описати його поведінку або за неповними даними про алгоритм функціонування автомата встановити його властивості. Розглянемо приклад розв'язання даної задачі.

Приклад 4.2 Розглянемо поведінку автомата Мура A_1 (розв'яжемо задачу аналізу для цього автомата), заданого суміщеною таблицею переходів і виходів (табл. 4.5). Автомат встановлений у початковий стан q_0 і на його вхід подається слово $p = z_1 z_2 z_2 z_1 z_2 z_2$.

Таблиця 4.5

	w_1	w_2	w_3	w_2
	q_0	q_1	q_2	q_3
z_1	q_1	q_2	q_3	q_3
z_2	q_3	q_0	q_0	q_0

Розв'язування

Оскільки для автомата Мура вихідний сигнал $w_g = \lambda(q_m)$, то послідовність вироблюваних ним вихідних сигналів (реакція автомата $w = \lambda(q_0, p)$) визначається лише послідовністю станів, які автомат A_1 проходить, сприймаючи вхідне слово p. Початково автомат перебуває в стані q_0, тому на виході виробляється символ w_1 (табл. 4.5); подача на вхід символа z_1 переводить його в стан q_1 (він стоїть у клітинці табл. 4.5 на перетині q_0 і z_1), в якому він виробляє символ w_2 (він стоїть у верхньому рядку над станом q_1); подача символа z_2 в стані q_1 переводить його в стан q_0 (він стоїть у клітинці табл. 4.5 на перетині q_1 і z_2), в цьому стані автомат виробляє вихідний символ w_1; подача символа z_2 в стані q_0 переводить його в стан q_3 (він стоїть в клітинці табл. 4.5 на перехресті q_0 і z_2), в цьому стані автомат виробляє вихідний символ w_2; подача символа z_1 в стані q_3 переводить його в стан q_3 (він стоїть в клітинці табл. 4.5 на перетині q_3 і z_1), в цьому стані автомат виробляє вихідний символ w_2; подача символа z_2 в стані q_3 переводить його в стан q_0 (він стоїть в клітинці табл. 4.5 на перетині q_3 і z_2), в цьому стані автомат виробляє вихідний символ w_1; подача символа z_2 в стані q_0 переводить його в стан q_3 (він стоїть у клітинці табл. 4.6 на перетині q_0 і z_2), в цьому стані автомат виробляє вихідний символ w_2. Таким чином, проведений аналіз поведінки автомата Мура, заданого таблицею 4.5, показав, що у відповідь на вхідне слово $p = z_1z_2z_2z_1z_2z_2$ довжини n він пройде послідовність станів $q_0q_1q_0q_3q_3q_0q_3$ довжини $n+1$ і видасть вихідне слово $w = w_1w_2w_1w_2w_2w_1w_2$ тієї самої довжини $n+1$. Така поведінка автомата описується таблицею 4.6.

Таблиця 4. 6

Вхідне слово p	z_1	z_2	z_2	z_1	z_2	z_2	
Послідовність станів	q_0	q_1	q_0	q_3	q_3	q_0	q_3
Вихідне слово w	w_1	w_2	w_1	w_2	w_2	w_1	w_2

Контрольні питання

У чому полягає задача аналізу абстрактного автомата?

Яким чином здійснюється аналіз поведінки автомата Мура, заданого табличним способом, при заданому вхідному слові?

Як потрібно аналізувати поведінку автомата Мілі, заданого графом, при заданому вхідному слові?

4.5 Еквівалентні перетворення моделей автоматів

У теорії автоматів доведено, що кожному автомату Мілі можна поставити у відповідність еквівалентний автомат Мура і навпаки.

Означення 4.5. Два автомати, що мають однакові вхідні і вихідні алфавіти, називаються *еквівалентними*, якщо вони, починаючи з того самого початкового стану, на будь-які однакові вхідні послідовності відповідають однаковими вихідними послідовностями. При цьому число їх внутрішніх станів може бути різним.

Іншими словами, два автомати є еквівалентними, якщо після установки в початковий стан їх реакції на будь-яке вхідне слово збігаються.

Подання реального автомата абстрактною моделлю Мура чи Мілі є суто умовним і залежить від його функціонального призначення. Наприклад, дорожній автомат зручніше зображувати моделлю Мура, оскільки сигнал автомата, який керує світлофором, повинний зберігатись весь час, поки останній "горить" певним кольором (задає напрям руху). У цьому випадку керувальний (вихідний) сигнал автомата зручно ототожнювати з його станом, що й описує модель Мура. В інших автоматах, наприклад, коли період їх стійкості відповідає часу дії вхідних сигналів, а вихідний сигнал з'являється лише короткочасно і під час переходу між цими станами (наприклад, як у схемі десяткового лічильника), зручно користуватися моделлю у вигляді автомата Мілі. Отже, під "станом" розуміють не деяку істотну властивість автомата, а лише спосіб для опису минулого (передісторії) автомата, що дає можливість усунути час як явну змінну.

У зв'язку з викладеним постає потреба переходити від автомата одного типу до еквівалентного автомата іншого типу.

Коли задано граф автомата Мура, то за ним можна перейти до графа еквівалентного автомата Мілі таким чином: вихідний сигнал w_g, записаний поруч з вершиною q_s, переносять на всі дуги, що входять до цієї вершини (рис. 4.11). Під час такого перетворення функції переходів в автоматі A Мура і автоматі A' Мілі залишаються однаковими:

$$\delta_A(q_m, z_f) = q_s \text{ і } \delta_{A'}(q_m, z_f) = q_s .$$

Функції виходів: для автомата Мура $\lambda_A(q_s) = w_g$, а для автомата Мілі $\lambda_{A'}(q_m, z_f) = w_g$.

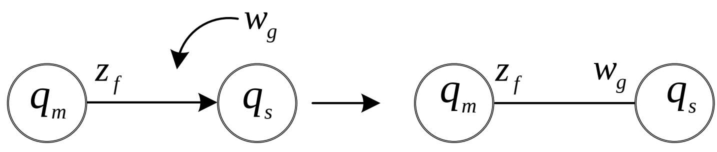


Рисунок 4.11 – Ілюстрація переходу від моделі Мура до моделі Мілі

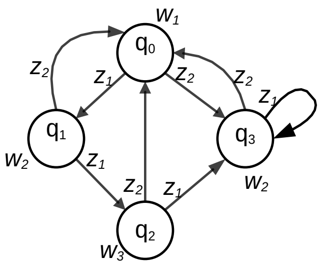
Із самої побудови графа автомата Мілі можна побачити, що він еквівалентний автомату Мура. Справді, коли деякий вхідний сигнал $x_f \in X$ надійде на вхід автомата A, що перебуває у стані q_m, то він перейде в стан $\delta_A(q_m, z_f) = q_s$ і видасть вихідний сигнал $\lambda_A(q_s) = w_g$. Але відповідний автомат Мілі A' із стану q_m також перейде в стан q_s, оскільки $\delta_{A'}(q_m, z_f) = \delta_A(q_m, z_f) = q_s$, і видасть той самий вихідний сигнал згідно зі способом побудови функції λ_A.

Таким чином, для вхідної послідовності певної довжини поведінка автоматів A та A' повністю збігається. За індукцією неважко поширити цей висновок на будь-яке вхідне слово p скінченної довжини і завдяки цьому довести, що автомати A та A' еквівалентні.

Приклад 4.3. Перетворимо автомат Мура A_1, заданий табл. 4.5 із прикладу 4.2, до еквівалентного йому автомата Мілі A_2.

Розв'язування

Перетворення здійснюється тривіально при використанні графічного способу задання автомата. Скориставшись відомостями з п. 4.3.3, зобразимо автомат A_1 Мура у вигляді орієнтованого графа (рис. 4.12).



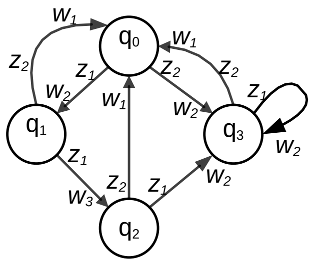


Рисунок 4.12 – Автомат Мура   Рисунок 4.13 – Автомат Мілі

Для отримання графа автомата Мілі (рис. 4.13) необхідно вихідний сигнал w_g, записаний поруч з вершиною q_s автомата Мура (рис. 4.12), перенести на всі дуги, що входять до цієї вершини.

Реакція на вхідне слово p автомата Мілі, отриманого в результаті перетворення графа, буде збігатися з реакцією автомата Мура. При цьому необхідно зазначити, що в еквівалентному автоматі Мілі кількість станів залишиться такою самою, як в автоматі Мура.

Приклад 4.4. Перетворимо автомат Мілі A_1, заданий таблицею 4.7 переходів і таблицею 4.8 виходів, до еквівалентного йому автомата Мура A_2.

Таблиця 4.7

	q_0	q_1	q_2
z_1	q_1	q_2	q_2
z_2	q_2	q_0	q_0

Таблиця 4.8

	q_0	q_1	q_2
z_1	w_2	w_3	w_2
z_2	w_2	w_1	w_1

Розв'язування

Перейдемо до графічного способу задання автомата, зобразивши орграф автомата Мілі A_1, наведений на рис. 4.14.

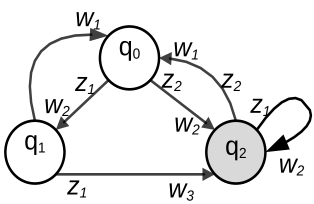


Рисунок 4.14 - Автомат Мілі

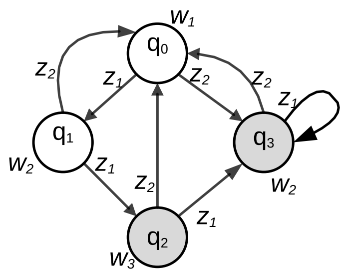


Рисунок 4.15 – Автомат Мура

Кожен стан q_i вихідного автомата Мілі (рис. 4.14) породжує стільки станів автомата Мура (рис. 4.15), скільки різних вихідних сигналів виробляється у вихідному автоматі при потраплянні у стан q_i.

Як випливає з рис. 4.14, при траплянні автомата у стан q_0 виробляються вихідні сигнали w_1, при траплянні автомата в q_2 – сигнал w_2, в q_3 – сигнали w_2 і w_3. Кожній парі "стан q_i – вихідний сигнал w_j", який виробляється при траплянні автомата у цей стан, поставимо у відповідність стан q_s еквівалентного автомата Мура з тим самим вихідним сигналом w_j: $q_0=(q_0, w_1)$, $q_1=(q_1, w_2)$, $q_2=(q_2, w_3)$, $q_3=(q_2, w_2)$. Таким чином, стан q_2 автомата Мілі породжує два стани (q_2 і q_3) еквівалентного автомата Мура. Оскільки в автоматі Мілі є перехід з q_2 у q_0 під дією z_2 і перехід з q_2 у q_2 під дією z_1, то з станів q_2 і q_3, породжуваних станом q_2 автомата Мілі, в автоматі Мура повинні бути переходи в стан q_0 під дією сигналу z_2, а також переходи $q_2 \rightarrow q_3$ і $q_3 \rightarrow q_3$ під дією сигналу z_1.

Контрольні питання

Які автомати називаються еквівалентними?

Що розуміють під терміном “стан” автомата?

Від яких умов залежить вибір моделі автомата (Мілі чи Мура) для його адекватного представлення?

Із чим пов’язана доцільність переходу від автомата одного типу до автомата іншого типу?

Чи можна кожному автомату Мілі поставити у відповідність еквівалентний йому автомат Мура і навпаки?

Яким чином здійснюється перехід від автомата Мілі до автомата Мура?

Чи містить одержаний у результаті перетворення еквівалентний автомат Мура однакову кількість станів з еквівалентним йому автоматом Мілі? Якщо ні, то чому?

Опишіть за допомогою графів, як здійснюється перехід від автомата Мура до автомата Мілі.

4.6 Мінімізація повністю визначених абстрактних автоматів

Еквівалентні автомати можуть мати різне число станів. У зв'язку з цим виникає задача знаходження мінімального автомата (з мінімальним числом станів) у класі еквівалентних між собою автоматів.

Означення 4.6. Два стани q_m і q_s автомата називаються *еквівалентними* (q_m=q_s), якщо $w(q_m, p) = w(q_s, p)$ для всіх можливих вхідних слів, тобто реакції автомата в цих станах на всі можливі вхідні слова p збігаються. Якщо q_m і q_s не еквівалентні, то вони *розрізнювані*.

Більш слабкою еквівалентністю є k-еквівалентність.

Означення 4.7. Стани q_m і q_s *k-еквівалентні* (q_m=q_s), якщо функції виходу λ від станів q_m і q_s однакові для всіх можливих вхідних слів p довжини k. Якщо ж стани не k-еквівалентні, то вони k – *розрізнювані*.

Поняття еквівалентності і k-еквівалентності використовується для розбиття множини станів автомата на попарно розрізнювані класи еквівалентності.

Розглянемо алгоритм мінімізації абстрактного автомата, запропонований Ауфенкампом і Хоном. Основна ідея алгоритму полягає в розбитті всіх станів вихідного абстрактного автомата на класи еквівалентних станів, що попарно не перетинаються. Після розбиття відбувається заміна кожного класу еквівалентності одним станом. Одержаний у результаті мінімальний абстрактний автомат має стільки станів, на скільки класів еквівалентності розбивається стан вихідного абстрактного автомата.

Приймемо позначення:

π – еквівалентність;

π_k – k-еквівалентність.

Розбиття станів автомата на класи еквівалентності дозволяє визначити надлишкові елементи в множині станів Q.

Нехай стани q_m і q_s еквівалентні. Це означає, що з точки зору реакції автоматів на всі можливі вхідні слова не важливо, в якому стані знаходиться автомат: q_m чи q_s. Тому один з цих станів можна вилучити з алфавіту станів, наприклад q_s.

Якщо клас еквівалентності містить більше одного елемента, то всі елементи, крім одного, можуть бути виключені з цього класу еквівалентності. Якщо кожен клас еквівалентності містить лише один стан, то множина Q нескоротлива.

У результаті розбиття множини станів автомата на попарно розрізнювані класи еквівалентності отримуємо автомат з мінімальним числом станів.

Алгоритм мінімізації числа станів автомата $A = (Z, W, Q, \delta, \lambda, q_0)$ складається з наступних кроків.

1. Знаходяться послідовні розбиття π_1 π_2, ... , π_K, π_{K+1} множини станів Q на класи одно-, дво-, ..., $k+1$-еквівалентних між собою станів. Розбиття на класи проводиться до тих пір, поки на якомусь до $k+1$ кроці не виявиться, що $\pi_{k+1}=\pi_k$. Можна показати, що тоді розбиття $\pi_k = \pi$, тобто k-еквівалентні стани є еквівалентними і число кроків k, при якому $\pi_k = \pi$, не перевищує M–1, де M – число станів у множині $Q = \{q_1, ..., q_m, ..., q_M\}$.

2. У кожному класі еквівалентності розбиття π вибирається за одним елементом, що утворює множину Q' станів мінімального автомата $A' = (Z, W, Q', \delta', \lambda', q_0')$, еквівалентного вихідному автомату A.

3. Функції переходів і виходів автомата S' визначаються на безлічі $Q' \times Z$, тобто δ': $Q' \times Z \to Q'$, λ': $Q' \times Z \to W$. Для цього в таблицях переходів і виходів викреслюють стовпці, що відповідають тим станам, які не увійшли до множини Q', а в тих стовпцях таблиці переходів, що залишилися, всі стани замінюються на еквівалентні з множини Q'.

4. Як q_0' вибирається один зі станів, еквівалентних q_0.

Приклад 4.5. Заданий таблицею 4.9 автомат Мура має три вхідних сигнали $Z=\{z_1, z_2, z_3\}$, два вихідних сигнали $W=\{w_1, w_2\}$ та сім станів автомата $Q=\{q_1, q_2,...,q_6, q_7\}$. Мінімізуємо автомат, використовуючи описаний вище алгоритм.

Таблиця 4.9

λ	w_1	w_1	w_2	w_2	w_1	w_2	w_1
δ	q_1	q_2	q_3	q_4	q_5	q_6	q_7
z_1	q_2	q_1	q_3	q_4	q_7	q_1	q_5
z_2	q_4	q_3	q_4	q_1	q_3	q_2	q_4
z_3	q_6	q_5	q_2	q_7	q_2	q_6	q_6

Розв'язування

Визначаємо класи k-еквівалентних станів. На першому кроці це 1-еквівалентні стани. Знаходимо класи 1-еквівалентних станів (за виходами w_1 таблиці 4.9):

$B_1=\{q_1,q_2, q_5, q_7\}$;

$B_2=\{q_3, q_4,q_6\}$;

$\pi_1=\{B_1, B_2\}$.

Будуємо таблицю розбиття π_1 (на k-му кроці π_k), замінюючи стани в таблиці переходів на відповідні класи еквівалентності.

	B_1				B_2		
	q_1	q_2	q_5	q_7	q_3	q_4	q_6
z_1	B_1	B_1	B_1	B_1	B_2	B_2	B_2
z_2	B_2	B_2	B_2	B_2	B_2	B_1	B_1
z_3	B_2	B_1	B_1	B_2	B_1	B_1	B_2

Проводимо перевірку. Якщо $\pi_к = \pi_{к+1}$, то переходимо до пункту 4. Якщо $\pi_к$ і $\pi_{к+1}$ не збіглися, то повертаємося до пункту 2 і знаходимо відповідні класи еквівалентності, поки перевірка не буде виконана.

$C_1 = \{q_1, q_7\};$

$C_2 = \{q_2, q_5\};$

$C_3 = \{q_3\};$

$C_4 = \{a_4\};$

$C_5 = \{q_6\};$

$\pi_2 = \{C_1, C_2, C_3, C_4, C_5\}.$

Перевірка після другого кроку показує, що $\pi_1 \neq \pi_2$. Виконуємо знову пункти 2 і 3 алгоритму.

	C_1		C_2		C_3	C_4	C_5
	q_1	q_7	q_2	q_5	q_3	q_4	q_6
z_1	C_2	C_2	C_1	C_1	C_3	C_4	C_1
z_2	C_4	C_4	C_3	C_3	C_4	C_4	C_2
z_3	C_5	C_5	C_2	C_2	C_2	C_1	C_5

$D_1 = \{q_1, q_7\};$

$D_2 = \{q_2, q_5\};$

$D_3 = \{q_3\};$

$D_4 = \{q_4\};$

$D_5 = \{q_6\};$

$\pi_3 = \{D_1, D_2, D_3, D_4, D_5\};\ \pi_2 = \pi_3;$

$q_1 = q_7;$

$q_2 = q_5.$

Таким чином, таблиця переходів мінімізованого автомата Мура матиме вигляд:

λ	w_1	w_1	w_2	w_2	w_2
δ	q_1	q_2	q_3	q_4	q_6
z_1	q_2	q_1	q_3	q_4	q_1
z_2	q_4	q_3	q_4	q_1	q_2
z_3	q_6	q_2	q_2	q_1	q_6

Приклад 4.6. Мінімізувати автомат Мілі (знайти еквівалентний автомат із мінімальною кількістю станів), заданий таблицями переходів 4.10 і виходів 4.11.

Таблиця 4.10 – $\delta : Q \times Z \to Q$

	q_0	q_1	q_2	q_3	q_4	q_5
z_1	q_2	q_3	q_2	q_3	q_4	q_5
z_2	q_4	q_5	q_4	q_5	q_0	q_1

Таблиця 4.11 – $\lambda : Q \times Z \to W$

	q_0	q_1	q_2	q_3	q_4	q_5
z_1	w_1	w_1	w_1	w_1	w_1	w_1
z_2	w_1	w_1	w_2	w_2	w_1	w_1

Розв'язування

З таблиці виходів отримуємо розбиття на класи 1-еквівалентності π_1, об'єднуючи в еквівалентні класи B_i стани з однаковими стовпцями:

$\pi_1=\{B_1, B_2\}$; $B_1=\{q_0, q_1, q_4, q_5\}$; $B_2=\{q_2, q_3\}$.

Для одержання 2-еквівалентних станів будуємо таблицю розбиття π_1 (табл. 4.12), замінюючи в таблиці переходів стани q_i відповідними класами еквівалентності B_1 або B_2.

Таблиця 4.12

	B_1				B_2	
	q_0	q_1	q_4	q_5	q_2	q_3
z_1	B_2	B_2	B_1	B_1	B_2	B_2
z_2	B_1	B_1	B_1	B_1	B_1	B_1

З отриманої таблиці одержимо класи 2-еквівалентності C_i і розбиття $\pi_2=\{C_1, C_2, C_3\}$, де $C_1=\{q_0, q_1 \quad\}$, $C_2=\{q_4, q_5\}$, $C_3=\{q_2, q_3\}$. Порівнюючи π_2 і π_1, зазначаємо, що ці розбиття відрізняються одне від одного, $\pi_2 \neq \pi_1$. Тому аналогічно будуємо таблицю розбиття π_2 (табл. 4.13), знову замінюючи в таблиці переходів стани q_i відповідними класами еквівалентності C_i.

Таблиця 4.13

	C_1		C_1		C_1	
	q_0	q_2	q_4	q_5	q_2	q_3
z_1	C_3	C_3	C_2	C_2	C_3	C_3
z_2	C_2	C_2	C_1	C_1	C_2	C_2

Порівнюючи π_2 і π_3, помічаємо, що $D_1=C_1$, $D_2=C_2$, $D_3=C_3$, отже $\pi_2=\pi_3$. Таким чином, ми отримали розбиття Q на *3*-еквівалентні класи. Оскільки таких класів усього три, то мінімальний автомат буде мати лише три стани. Вибираємо з кожного класу D_i по одному стану і отримуємо множину станів Q' мінімального автомата. Нехай, наприклад, $Q'=\{q_0, q_3, q_4\}$. Для отримання мінімального автомата із вихідних таблиць переходів і виходів (табл. 4.10 і 4.11) викреслюємо стовпці, що відповідають “зайвим станам” q_1, q_3, q_6. У результаті одержимо мінімальний автомат Мілі, еквівалентний вихідному автомату (табл. 4.14, 4.15).

Таблиця 4.14

	q_0	q_3	q_4
z_1	q_3	q_3	q_4
z_2	q_4	q_4	q_0

Таблиця 4.15

	q_0	q_3	q_4
z_1	w_1	w_1	w_1
z_2	w_1	w_2	w_1

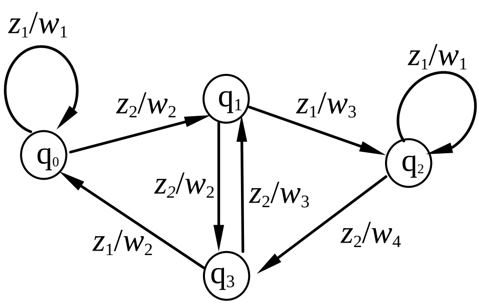
Мінімізацією числа внутрішніх станів автомата закінчується етап абстрактного синтезу.

Контрольні питання

Які два стани автомата називаються еквівалентними?
Які стани автомата називаються розрізнюваними?
Дайте визначення k-еквівалентності станів автомата.
Опишіть алгоритм мінімізації.
Що є умовою припинення алгоритму мінімізації?
Які стани залишають в мінімальному автоматі і скільки їх?

4.7 Вправи

1. На рисунку зображено граф автомата. Визначте тип автомата (Мура чи Мілі) і задайте його табличним способом.



Проаналізуйте поведінку автомата при надходженні на його вхід слова $p = z_1z_1z_2z_1z_2z_2$.

2. Автомат заданий таблицею. Як називається ця таблиця, автомат якого типу вона описує? Побудуйте граф цього автомата.

w	w_1	w_4	w_2	w_3	w_5
$z \setminus q$	q_1	q_2	q_3	q_4	q_5
z_1	q_1	q_3	q_4	q_1	q_1
z_2	q_2	q_4	q_3	q_4	q_2
z_3	q_4	q_1	q_2	q_3	q_1

3. Автомат Мілі заданий суміщеною таблицею переходів/виходів. Запишіть математичну модель автомата, а також подайте його графічним методом.

	q_0	q_1	q_2	q_3
z_1	q_1/w_1	q_1/w_1	q_0/w_2	a_0/w_4
z_2	q_3/w_5	q_2/w_3	q_3/w_4	q_2/w_5

4. Відомо, що скінченний автомат має вхідний алфавіт $Z=\{z_1, z_2\}$, вихідний алфавіт $W=\{w_1, w_2\}$ і множину станів $Q=\{q_0, q_1, q_2\}$. Задайте автомат табличним і графічним способами.

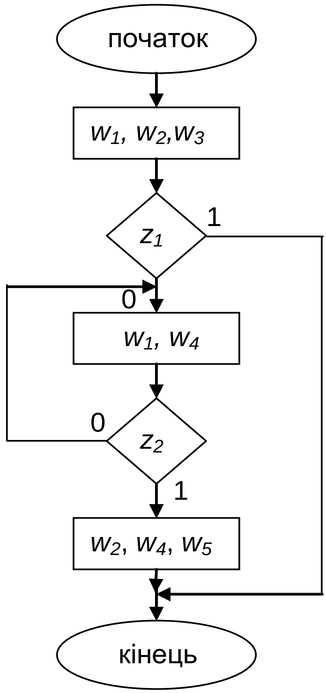
5. Вантажний ліфт, що обслуговує триповерховий магазин, має кнопку виклику на кожному поверсі і працює за такими правилами: якщо натиснута одна кнопка, то ліфт рухається на поверх, на якому розміщена ця кнопка; якщо натиснуті одночасно дві або три кнопки, то ліфт рухається на нижчий з усіх поверхів, на яких натиснуті кнопки. Жодна кнопка не може бути натиснута під час руху ліфта (наприклад, передбачено механічне блокування).

На підставі наведеного вербального опису автомата перейдіть до його математичної моделі у вигляді абстрактного автомата.

6. Задайте автомат, описаний у вправі 5, табличним і графічним способами.

7. Проаналізуйте поведінку автомата, заданого таблицею в задачі 2, при надходженні на його вхід слова $p = z_3z_1z_2z_1z_2z_2z_3$.

8. Зробіть розмітку граф-схеми алгритму, зображеної на рисунку, для автомата Мура. Подайте автомат у вигляді орграфа і табличним способом.

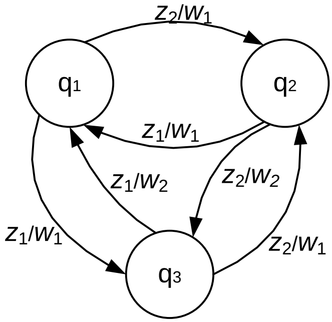


Зробіть розмітку цієї граф-схеми також для автомата Мілі і подайте його у вигляді орграфа і табличним способом.

9. Автомат Мура задано таблично. Перейти до автомата Мілі.

	w_1	w_1	w_3	w_2	w_3
$z\backslash q$	q_1	q_2	q_3	q_4	q_5
z_1	q_2	q_5	q_5	q_3	q_3
z_2	q_4	q_2	q_2	q_1	q_1

10. Автомат задано орграфом на рисунку. Визначити, який тип цього автомата, і якщо це автомат Мура, то перейти до автомата Мілі, а якщо Мілі – то перейти до автомата Мура.



11. Для автомата $U=\{Z, W, Q, \delta, \lambda\}$, де $Z=\{z_1, z_2, z_3\}$, $W=\{w_1, w_2\}$, задана автоматна таблиця. Знайти мінімальний за кількостю станів автомат.

$Q \backslash Z$	z_1	z_2	z_3
q_1	q_2, w_1	q_4, w_2	q_4, w_2
q_2	q_1, w_2	q_1, w_1	q_5, w_1
q_3	q_1, w_2	q_6, w_1	q_5, w_1
q_4	q_8, w_1	q_1, w_2	q_1, w_2
q_5	q_6, w_2	q_4, w_2	q_3, w_1
q_6	q_8, w_1	q_9, w_2	q_6, w_2
q_7	q_6, w_2	q_1, w_2	q_3, w_1
q_8	q_4, w_2	q_4, w_1	q_7, w_1
q_9	q_7, w_1	q_9, w_2	q_7, w_2

Розділ 5
СТРУКТУРНИЙ СИНТЕЗ АВТОМАТІВ

У розділі 4 було показано, що інформацію, подану дискретними фізичними сигналами, обробляють керуючі автомати, які поділяються на два класи – комбінаційні схеми (автомати без пам'яті) та скінченні автомати з пам'яттю. У системах управління та автоматики за допомогою автоматів здійснюються процеси обробки інформації, що подається в цифровій формі, тобто сигнал, що несе інформацію, дискретизований як за часом, так і за рівнем. Оскільки символи вхідного й вихідного алфавітів X і Y автомата в цьому випадку кодуються цифрами 0 та 1, що відповідають низькому чи високому рівню сигналу в деякому такті автоматного часу, то такий автомат називається цифровим. У цьому розділі розглядаються практичні питання, що стосуються до розв'язання задач структурного аналізу й синтезу комбінаційних схем і цифрових автоматів.

5.1 Методи синтезу і аналізу комбінаційних схем

Нагадаємо, що логічну схему називають комбінаційною, коли сукупність вихідних сигналів у будь-який момент часу однозначно визначають сигнали, що надходять на її входи.

Комбінаційну схему (КС) можна задати абстрактною чи структурною моделлю. Абстрактною моделлю КС є або таблиця істинності, або логічні вирази (формули логічних функцій), які пов'язують певною залежністю вихідні сигнали з вхідними. Структурну модель КС використовують для побудови її з конкретних логічних елементів (ЛЕ).

У ході розроблення комбінаційних схем доводиться розв'язувати задачі аналізу й синтезу. Задача аналізу полягає у визначенні статичних і динамічних властивостей комбінаційної схеми. У статиці визначається логічна функція (ЛФ), реалізована комбінаційної схемою за відомою її структурою. У динаміці розглядається здатність надійного функціонування КС у перехідних процесах при зміні значень змінних на її входах, тобто визначається наявність на виходах КС можливих небажаних імпульсних сигналів, що не випливають безпосередньо з формули ЛФ, реалізованої цією схемою.

Задача синтезу полягає в побудові комбінаційної схеми, що реалізує задану ЛФ. Зазвичай на етапі проектування закон функціонування КС спочатку задають у вигляді таблиці істинності ЛФ. За нею потім складають логічний вираз (формулу ЛФ), згідно з яким будують структуру комбінаційної схеми в заданому базисі логічних елементів.

5.1.1 Канонічний метод синтезу комбінаційних схем

Комбінаційна схема може мати кілька виходів. При канонічному методі передбачається, що кожна функція виходу реалізується своєю схемою, сукупність яких і дає необхідну КС. Тому синтез складної КС з n виходами замінюється синтезом n схем з одним виходом.

Згідно з канонічним методом процедура синтезу КС містить наступні етапи.

1. Вербальний опис функціонування КС перетворюється в її абстрактну модель, подану спочатку таблицею істинності, а потім логічною функцією (або системою ЛФ). Логічні функції, що підлягають схемній реалізації, подаються у вигляді досконалої диз'юнктивної нормальної форми (ДДНФ).

2. Із використанням методів мінімізації визначаються мінімальні диз'юнктивна і кон'юнктивна нормальні форми (МДНФ і МКНФ). З отриманих двох мінімальних форм вибирається більш проста за критерієм складності.

3. Логічну функцію в мінімальній формі подають в заданому (або обраному розробником) базисі.

4. За поданням функції у заданому базисі будують КС.

Розглянемо методику синтезу комбінаційних схем на прикладі мажоритарної схеми.

Приклад 5.1. Поставимо завдання: спроектувати КС на три входи і один вихід, що виробляє вихідний сигнал лише в тому випадку, коли сигнал “1” з’являється більш ніж на одному вході, тобто схема повинна видавати сигнал на виході при появі сигналів на будь-яких двох входах або на всіх трьох. У цьому і проявляється “мажоритарність”. Наприклад, для 4-входової і 5-входової мажоритарних схем вихідний сигнал повинен з’являтися при кількості сигналів “1” більш ніж на двох входах, для 6-входової – більш ніж на трьох і т. д.

Розв’язування

Етап 1. Перехід від вербального опису спочатку до таблиці істинності, а потім до формули ЛФ. Таблиця істинності для мажоритарної схеми на три входи має вигляд таблиці 5.1.

Таблиця 5.1 – Таблиця істинності для мажоритарної схеми

№ набору	Входи			Вихід F
	x_1	x_2	x_3	
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

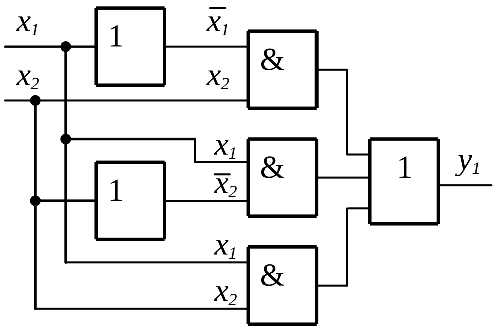
На підставі таблиці істинності подамо функцію F досконалими нормальними формами (див. розд. 2, п. 2.6):

$$F_{\text{ДДНФ}}=\overline{x_1}\cdot x_2\cdot x_3+x_1\cdot\overline{x_2}\cdot x_3+x_1\cdot x_2\cdot\overline{x_3}+x_1\cdot x_2\cdot x_3, \qquad (5.1)$$

$$F_{\text{ДКНФ}}=(x_1+x_2+x_3)\cdot(x_1+x_2+\overline{x_3})\cdot(x_1+\overline{x_2}+x_3)\cdot(\overline{x_1}+x_2+x_3). \qquad (5.2)$$

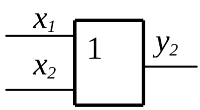
Здебільшого логічну функцію під час синтезу комбінаційної схеми подають у вигляді ДДНФ (якщо ДКНФ не є суттєво простішою). Далі ми будемо розв'язувати задачу синтезу КС для функції (5.1).

Етап 2. Мінімізація логічної функції (5.1). Будь-яку ЛФ, зображену формулою, можна реалізувати у вигляді логічної схеми (див. розд. 2, п. 2.5). Наприклад, для реалізації виразу $y_1 = \overline{x}_1 \cdot x_2 + x_1 \cdot \overline{x}_2 + x_1 \cdot x_2$ треба мати три елементи І (кон'юнктори), два елементи Не (інвертори) і один елемент Або (диз'юнктор) з трьома входами (рис. 5.1а).



$y_1 = \overline{x}_1 \cdot x_2 + x_1 \cdot \overline{x}_2 + x_1 \cdot x_2$

а)



$y_2 = x_1 + x_2$

б)

Рисунок 5.1 – Структурні схеми логічних схем реалізації функції: а) $y_1 = \overline{x}_1 \cdot x_2 + x_1 \cdot \overline{x}_2 + x_1 \cdot x_2$; б) $y_2 = x_1 + x_2$

У той самий час після проведення процедури мінімізації логічної функції y_1 (див. розд. 2, п. 2.8) отримуємо рівносильну до неї функцію $y_2 = x_1 + x_2$. Схема, що реалізує функцію $y_2 = x_1 + x_2$, складається з одного елемента Або (рис. 5.1б).

Таким чином, один і той самий процес перетворення цифрових сигналів можуть виконувати логічні схеми, що мають різну складність. Складність логічної схеми залежить від сумарної кількості входів логічних елементів.

Визначимо складність схем, поданих на рис. 5.1а, б. З урахуванням того, що схема на рис.5.1а містить 2 одновходові логічні елементи Не, 3 двовходові логічні елементи І та один трився входовий логічний елемент Або, вираховуємо:

$$C_{\text{у}_1} = 2{\cdot}1 + 3{\cdot}2 + 1{\cdot}3 = 2 + 6 + 3 = 11.$$

Аналогічно для схеми на рис.5.1б маємо: $C_{\text{у}_2} = 1{\cdot}2 = 2$.

Отже, складність КС можна зменшити шляхом мінімізації ЛФ, під якою розуміють процедуру знаходження найпростішого виразу для неї, еквівалентного початковому.

Отримуємо МДНФ логічної функції (5.1):

$$F_{\text{МДНФ}} = x_1{\cdot}x_2 + x_1{\cdot}x_3 + x_2{\cdot}x_3. \tag{5.3}$$

Після того як в процесі синтезу отримано МДНФ (чи МКНФ) логічної функції, виконують перехід до обраного функціонального базису.

Етап 3. Під час проектування логічних схем, що реалізують складні ЛФ, з погляду технологічності їх виготовлення бажано використовувати ЛЕ одного функціонального базису. Тому треба навчитися зводити логічні вирази від одного базису до іншого. Наприклад, щоб звести до базису І-Не отриману логічну функцію (5.3), потрібно за допомогою законів алгебри логіки перетворити її до потрібного вигляду:

$$F_{\text{МДНФ}} = x_1 x_2 + x_1 x_3 + x_2 x_3 = \overline{\overline{x_1x_2 + x_1x_3 + x_2x_3}} = \overline{\overline{x_1x_2} \cdot \overline{x_1x_3} \cdot \overline{x_2x_3}}\,. \tag{5.4}$$

Етап 4. На основі отриманого виразу ЛФ складають її функціональну схему з логічних елементів заданого базису. В нашому випадку такими є елементи І-Не. Згідно з отриманим виразом схема складається з трьох двовходових і одного трився входового елементів І-Не (рис.5.2).

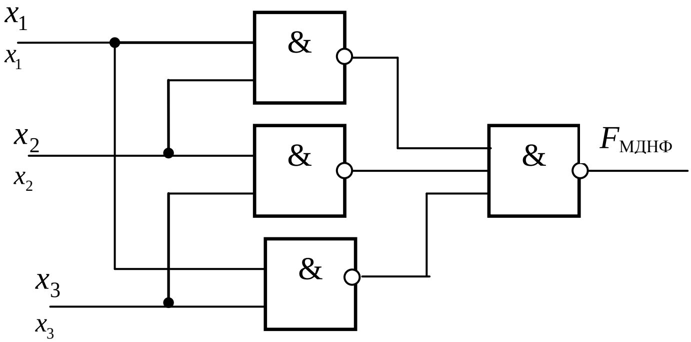


Рисунок 5.2 – Функціональна схема синтезованої КС
для прикладу 5.1 (згідно формули (5.4))

5.1.2 Аналіз комбінаційних схем

Задачі аналізу комбінаційних схем є зворотними щодо задач синтезу. Для аналізу КС у статиці необхідно описати схему аналітичними формами функцій, які реалізує КС, у заданому функціональному базисі.

На першому етапі аналізу необхідно вилучити зі схеми елементи, які не несуть функціонального навантаження. Прикладом таких елементів є повторювачі, що використовуються як підсилювачі сигналів.

На другому етапі складають аналітичну форму функції, яку реалізує задана КС. Для цього виходи логічних елементів позначають різними символами і записують формулу у вигляді поетапної суперпозиції згідно з топологією схеми.

Приклад 5.2. Наведемо приклад аналізу КС без пам'яті (рис. 5.3). Вона містить 2 елементи Не, 2 елементи І, елемент Або, елемент Або-Не.

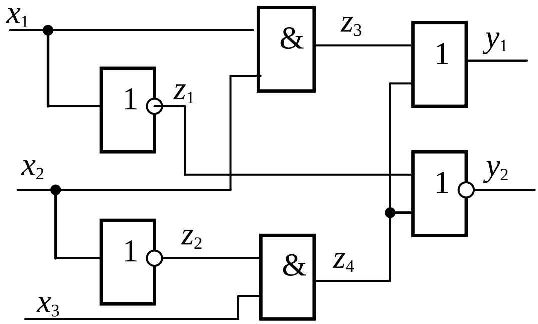


Рисунок 5.3 – Функціональна схема КС, що підлягає аналізу

Розв'язування

1. На функціональній схемі проміжні виходи логічних елементів позначаємо символами проміжних змінних z_1, z_2, z_3, z_4.

2. Визначаємо функції безпосередніх зв'язків, що встановлюють залежності виходу кожного ЛЕ від його входів на основі елементарних логічних функцій:

$$z_1 = \bar{x}_1, \quad z_2 = \bar{x}_2, \quad z_3 = x_1 x_2, \quad z_4 = \bar{x}_2 x_3;$$

$$y_1 = z_3 + z_4; \quad y_2 = \overline{z_1 + z_4}.$$

3. Шляхом підстановок виключаємо всі внутрішні змінні:

$$y_1 = x_1 x_2 + \bar{x}_2 x_3; \quad y_2 = \overline{z_1 + z_4} = \overline{\bar{x}_1 + \bar{x}_2 x_3}.$$

Отримуємо залежності виходів y_1 і y_2 комбінаційної схеми від її входів x_1, x_2 і x_3 у вигляді ДДНФ:

$$y_1 = x_1 x_2 + \bar{x}_2 x_3 = x_1 x_2 \cdot 1 + 1 \cdot \bar{x}_2 x_3 = x_1 x_2 (x_3 + \bar{x}_3) +$$

$$+ (x_1 + \bar{x}_1)\bar{x}_2 x_3 = \bar{x}_1 \bar{x}_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3;$$

$$y_2 = \overline{\bar{\bar{x}}_1 + \bar{\bar{x}}_2 x_3} = x_1 \cdot \overline{\bar{x}_2 x_3} = x_1 (x_2 + \bar{x}_3) = x_1 x_2 (x_3 + \bar{x}_3) +$$

$$+ x_1 \bar{x}_3 (x_2 + \bar{x}_2) = x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3.$$

4. Складаємо таблицю істинності функцій y_1 та y_2:

Номер набору	Входи			Виходи	
	x_1	x_2	x_3	y_1	y_2
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	0	0
3	0	1	1	0	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	1
7	1	1	1	1	1

5. З розгляду таблиці істинності встановлюється відповідність між наборами вхідних змінних x_1, x_2, x_3 і сигналами виходів y_1, y_2.

Під час аналізу КС у динаміці необхідно визначити, чи не формує вона в перехідних процесах вихідні сигнали, не передбачені таблицею істинності. Це обумовлено наявністю в схемі шляхів з різною тривалістю проходження сигналів від входів схеми до виходів. Наприклад, у схемі на рис. 5.3 від входів до виходів сигнал x_2 може проходити 2 або 3 елементи. Ця проблема відома як проблема “*гонок*”, що обумовлює так званий “ризик збою” в комбінаційних схемах.

Для усунення збою використовуються декілька методів. *Перший метод* – це синхронізація передачі сигналів від однієї логічної схеми в іншу. При цьому інформаційний сигнал тактується синхросигналом, який забезпечує прийом інформаційного сигналу в подальших пристроях після закінчення перехідних процесів у схемі, що формує цей сигнал. Недолік такого методу – зменшення швидкодії КС.

Другий метод пов’язаний із встановленням фільтрів для вихідних сигналів на тих виходах комбінаційної схеми, де можуть виникати помилкові сигнали (рис. 5.4).

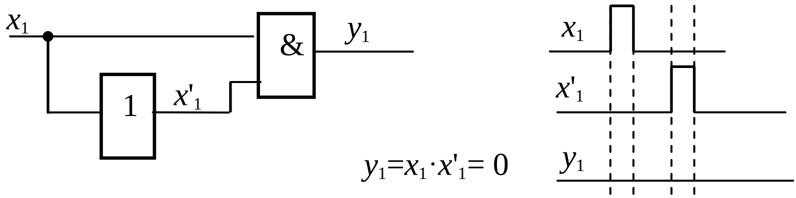


Рисунок 5.4 – Метод усунення “гонок” із використанням “фільтрів” – введенням додаткових елементів затримки

Короткочасні сигнали затримуються на елементі затримки перед поданням на один з входів вихідного елемента фільтра, наприклад, на повторювачі. За рахунок такого рознесення в часі короткочасних сигналів помилкові сигнали на виході фільтра відсутні. Цей спосіб може застосовуватися лише для окремих, зазвичай експериментальних схем.

Третій метод (метод Хаффмена) – введення структурної надлишковості. Основна ідея методу: для отримання схеми, вільної від змагань, необхідно і достатньо для кожної пари суміжних (тобто таких, що відрізняються на один розряд) станів входів, для яких вихідна функція схеми має однойменні значення (нульові для ДКНФ або одиничні для ДДНФ), знайти принаймні один терм функції, який покриває обидва стани. Щодо діаграм Вейча - Карно це означає, що кожна пара сусідніх однаково зазначених клітин діаграми повинна бути включена в загальний контур (рис.5.5).

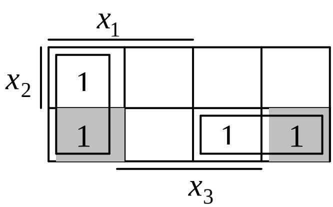


Рисунок 5.5 –Включення в покриття надлишкового контуру

Мінімальне покриття діаграми Вейча - Карно (рис. 5.5) визначає функцію $y_1 = x_1\,\overline{x_3} + \overline{x_1}\,\overline{x_2}$, що відповідає найбільш простій КС. Введення надлишкового контуру (на рисунку цей контур залитий сірим) призводить до функції

$$y_2 = x_1\overline{x_3} + \overline{x_1}\,\overline{x_2} + \overline{x_2}\,\overline{x_3}\,,$$

що описує більш складну, але і більш надійну схему.

5.2 Синтез типових комбінаційних пристроїв систем автоматики

До числа типових комбінаційних пристроїв можна віднести суматори, перетворювачі кодів різного роду, у тому числі шифратори й дешифратори, пристрої порівняння та ін. Ці пристрої досить широко застосовуються в схемах автоматизації, тому заслуговують на окремий розгляд.

5.2.1 Синтез суматорів

Двійковим *суматором* називають схему, що виконує операцію додавання цифрових кодів двох чисел. Суматори є ядром арифметико-логічних пристроїв, що виконують арифметичні й логічні операції з числами.

Комбінаційними називають суматори, реалізовані у вигляді комбінаційної схеми. Окрім комбінаційних, існують ще *накопичувальні* суматори, що містять ще й регістрову пам’ять. Регістр акумулює результати підсумовування так, що наступний доданок додається до збережуваного в ньому результату попереднього додавання. Такі суматори часто називають *акумуляторами*.

За кількістю входів розрізняють півсуматори, повні однорозрядні та багаторозрядні суматори.

У *багаторозрядних* суматорах паралельної дії числа-доданки надходять на входи суматора в паралельному коді, результат видається також паралельним кодом на його виходах. Таким чином, багаторозрядний суматор додає n-розрядні двійкові числа і має $2n$ входів і n виходів. Багаторозрядний суматор будується з потрібної кількості однорозрядних суматорів, кожен з яких формує значення одного з розрядів суми.

Однорозрядні суматори можна класифікувати за кількістю входів на такі типи:

– двовходовий, який називається півсуматором;

– тривходовий, який називається повним однорозрядним суматором.

Півсуматором називають пристрій, що виконує операцію додавання однорозрядних двійкових чисел без урахування перенесення з попереднього (молодшого) розряду. Таблицю істинності для нього (табл. 5.2) можна скласти на базі таблиці додавання зазначених чисел.

Таблиця 5.2 – Таблиця істинності півсуматора

Двійкові числа-доданки		Сума	Перенесення
a_0	b_0	S_0	P_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

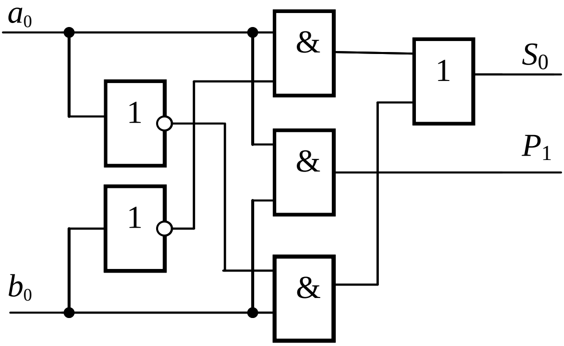
З таблиці 5.2 видно, що півсуматор повинен мати два входи і два виходи, один з яких – вихід суми S_0, а другий – вихід перенесення P_1. Відповідно до таблиці ДДНФ для S_0 та P_1 мають такий вигляд:

$$S_0 = \bar{a}_0 b_0 + a_0 \bar{b}_0, \quad (5.5)$$

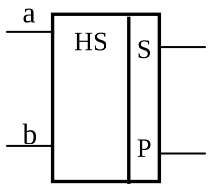
$$P_1 = a_0 b_0. \quad (5.6)$$

Із цих виразів видно, що вони не містять кон’юнкцій, що склеюються між собою, тому етап мінімізації не виконуємо.

Побудувавши ці функції в заданому базисі, отримаємо функціональну схему півсуматора (рис. 5.6).



а)



б)

Рисунок 5.6– Логічна схема півсуматора (а)
та його позначення на функціональних схемах (б)

Кількість логічних елементів у схемі півсуматора можна скоротити. Для цього виразимо функцію S_0 в КНФ і проведемо перетворення:

$$S_0 = \overline{a}_0(a_0 + b_0) + \overline{b}_0(a_0 + b_0) = (a_0 + b_0)(\overline{a}_0 + \overline{b}_0) =$$

$$= (a_0 + b_0) \cdot \overline{a_0 b_0} = (a_0 + b_0) \cdot \overline{P_1}.$$

Функціональна схема такого півсуматора представлена на рис. 5.7 (в базисі І, Або, Не).

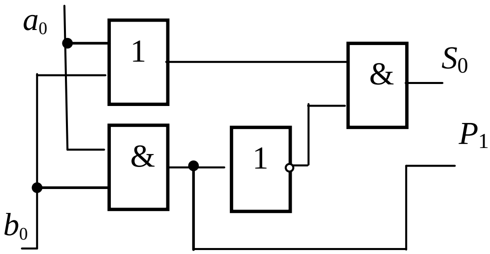


Рисунок 5.7 – Модифікована логічна схема півсуматора

У випадку додавання багаторозрядних чисел півсуматор можна використовувати лише у найправішому, молодшому розряді багаторозрядного суматора, оскільки в ньому не передбачено можливе перенесення з попереднього розряду.

Пристрій, який за значеннями a_i та b_i доданків A і B та за значенням перенесення p_i з попереднього молодшого розряду формує значення розрядної суми S_i та перенесення P_{i+1} у старший розряд, називають *повним однорозрядним суматором*.

Умови роботи однорозрядного суматора визначаються його таблицєю істинності (табл. 5.3).

Таблиця 5.3 – Таблиця істинності однорозрядного суматора

Входи			Виходи		Значення суми
p_i	a_i	b_i	S_i	P_{i+i}	
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	0

На підставі табл. 5.3 можемо записати логічні вирази для суми S_i та перенесення P_{i+1} у вигляді ДДНФ:

$$S_i = a_i\bar{b}_i\bar{p}_i + a_i b_i p_i + \bar{a}_i\bar{b}_i p_i + \bar{a}_i b_i \bar{p}_i \tag{5.7}$$

$$P_{i+1} = a_i b_i \bar{p}_i + \bar{a}_i b_i p_i + a_i \bar{b}_i p_i + a_i b_i p_i, \tag{5.8}$$

Вираз (5.8) можна звести попарним склеюванням перших трьох кон’юнкцій з останньою:

$$P_{i+1} = a_i b_i \vee b_i p_i \vee a_i p_i\,. \tag{5.9}$$

Позначення однорозрядного комбінаційного суматора зображено на рис. 5.8, а його логічну схему, що реалізує співвідношення (5.7) та (5.9), показано на рис. 5.9.

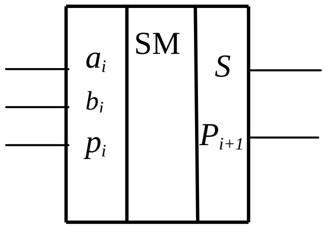


Рисунок 5.8 – Функціональне позначення однорозрядного суматора

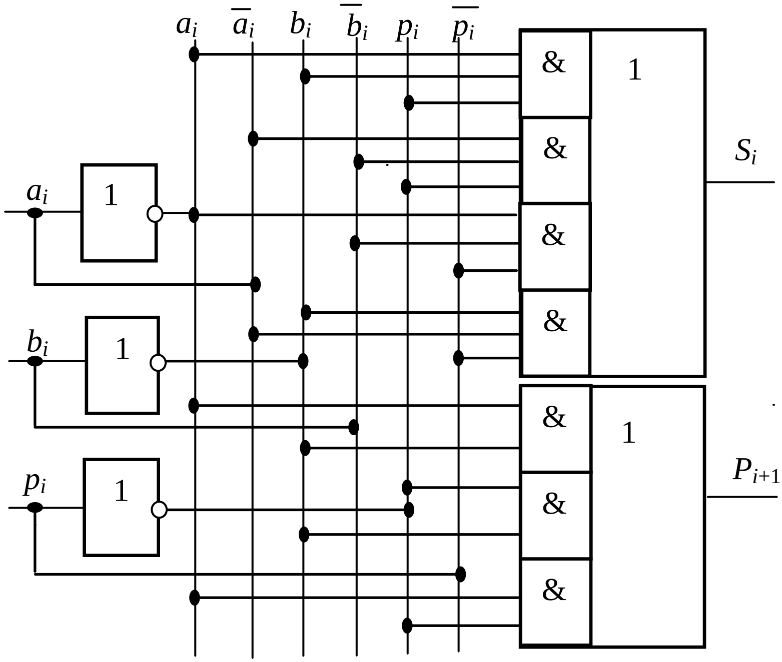


Рисунок 5.9 – Логічна схема однорозрядного суматора

Для реалізації схеми повного суматора на півсуматорах перетворимо логічні функції S і P так:

$$S = a_i\overline{b}_i\overline{p}_i + a_ib_ip_i + \overline{a}_i\overline{b}_ip_i + \overline{a}_ib_i\overline{p}_i = a_i(\overline{b}_i\overline{p}_i + b_ip_i) + \overline{a}_i(\overline{b}_ip_i + b_i\overline{p}_i) =$$

$$= a_i\overline{(\overline{b}_ip_i + b_i\overline{p}_i)} + \overline{a}_i(\overline{b}_ip_i + b_i\overline{p}_i) = a_i\overline{S_{HS1}} + \overline{a}_iS_{HS1} = S_{HS2}.$$

$$P = a_ib_i + b_ip_i + a_ip_i = b_ip_i + a_i(\overline{b}_ip_i + b_i\overline{p}_i) =$$

$$= b_ip_i + a_iS_{HS1} = P_{HS1} + P_{HS2}.$$

Перший півсуматор (HS_1) має входи b_i і P_i, другий (HS_2) – входи a_i і S_{HS1}. Відповідна цим функціям структурна схема повного суматора зображена на рис. 5.10.

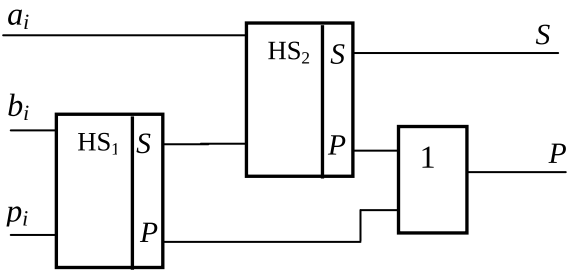


Рисунок 5.10 – Логічна схема однорозрядного суматора на півсуматорах

За допомогою однозарядних суматорів можна побудувати підсумовуючі пристрої послідовної або паралельної дії для складання багаторозрядних двійкових чисел.

Паралельний багаторозрядний суматор n-розрядних чисел являє собою в найпростішому випадку об’єднання n однорозрядних суматорів, з’єднаних колами перенесення послідовно від молодших до старших розрядів (рис. 5.11).

Схема паралельного суматора із послідовним перенесенням має порівняно невелику швидкодію, тому що сигнали суми S та перенесення P у кожному i-му розряді формуються лише після того, як надійде сигнал перенесення з (i -1)-го розряду.

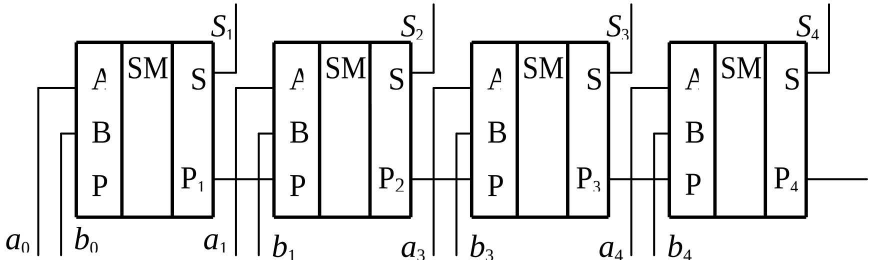


Рисунок 5.11 - Структурна схема паралельного 4-розрядного суматора з послідовним перенесенням

Із схеми рис. 5.11 бачимо, що максимальна затримка формування суми на виході старшого розряду в найгіршому випадку складатиметься із суми затримки всіх попередніх розрядів і власної затримки формування суми:

$$\tau_{max} = (n\text{-}1)\ \tau_{pn} + \tau_s,$$

де τ_{pn} і τ_s – затримки формування відповідно перенесення й суми в однорозрядному суматорі.

Швидкість перенесення підвищують різнимим способами: мінімізацією довжини з’єднань і шкідливих ємностей у монтажі, використанням швидкодійних елементів у колі перенесення і спеціальних структурних методів прискорення останнього, зокрема застосуванням кіл паралельного перенесення.

Крім комбінаційних підсумовуючих пристроїв, застосовуються накопичувальні суматори (суматори з пам’яттю, або акумулятори), які не тільки підсумовують доданки, а й запам'ятовують одержану суму.

Накопичувальним називають суматор, який реалізує операцію підсумовування згідно з формулою $S = S+A$. Цей запис означає, що до вмісту суматора, який зберігається в його пам’яті, додається черговий доданок, і результат лишається в пам’яті, замінюючи собою старий вміст. Організація накопичувальних суматорів на базі комбінаційних схем потребує використання запам’ятовувальних регістрів.

У схемі на рис. 5.12 регістри *A* і *B* – це регістри відповідно доданка й суми. Перед початком підсумовування регістр *B* ставлять у нульовий стан. Після виконання першого циклу додавання результат підсумовування з виходу комбінаційного суматора через схеми Або переписується в регістр *B*. Результат підсумовування таким чином можна знову подавати на додавання з новим значенням доданка *A*. Зрозуміло, що регістр *B* треба виготовляти на двотактних тригерах.

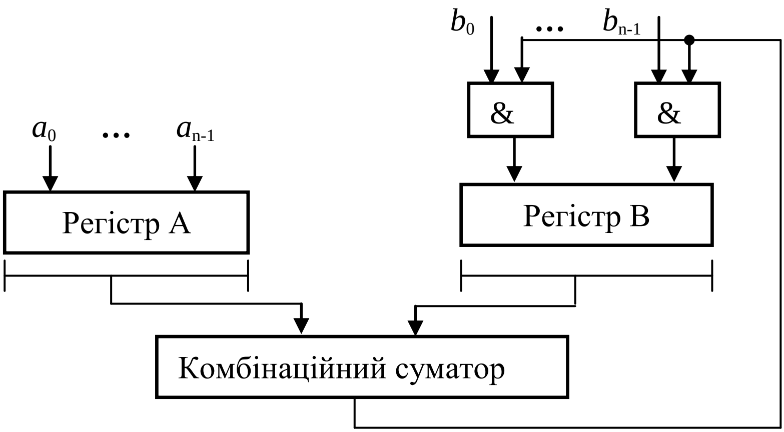


Рисунок 5.12 - Структурна схема накопичувального суматора

5.2.2 Синтез дешифраторів і шифраторів

Дешифратором називають комбінаційну схему з кількома входами й виходами, що перетворює код, поданий на входи, на сигнал одного з виходів.

Двійкові дешифратори перетворюють двійковий код у код "1 із *N*". Якщо на *n* входів такого дешифратора подано двійкові змінні, то на одному з 2^n виходів виробляється сигнал 1, а на решті зберігається сигнал 0. Дешифратор, який має $N = 2^n$

виходів, називається повним. Коли частини вхідних кодів не використано, то $N < 2^n$ і дешифратор неповний.

Синтезуємо схему повного двійкового дешифратора на два входи. Така схема повинна мати чотири виходи, а її функціонування можна описати таблицею істинності (табл. 5.4).

Згідно з табл. 5.4 логічні вирази функції виходів можна зобразити у вигляді ДДНФ:

$$D_0 = \bar{a}_0\bar{a}_1, \qquad D_1 = a_0\bar{a}_1,$$

$$D_2 = \bar{a}_0 a_1, \qquad D_3 = a_0 a_1.$$

Таблиця 5.4 – Таблиця істинності дешифратора

Входи		Функції виходів			
a_1	a_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

У загальному випадку вихідні функції $D_i(a_0, a_1, \ldots)$ дешифратора на n входів описує система логічних виразів, зображених за ДДНФ конституентами одиниці

$$D_0 = \bar{a}_0\bar{a}_1\bar{a}_2\ldots\bar{a}_{n-1},$$
$$D_1 = a_0\bar{a}_1\bar{a}_2\ldots\bar{a}_{n-1},$$
$$D_2 = \bar{a}_0 a_1\bar{a}_2\ldots\bar{a}_{n-1},$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$D_{2^n-1} = a_0 a_1 a_2\ldots a_{n-1}.$$

Із системи рівнянь випливає, що для побудови повного дешифратора треба мати N логічних елементів “І” з n входами у кожному (рис. 5.13).

Дешифратори цього типу називають *лінійними*. Вони мають найбільшу швидкодію. Проте коли вхідний код має велику розрядність, реалізовувати їх важко, оскільки треба застосувати кон’юнктори з великою кількістю входів, тому буде велике навантаження на джерела вхідних сигналів. Сумарну складність лінійного дешифратора визначає вираз $C_1 = n \cdot 2^n$.

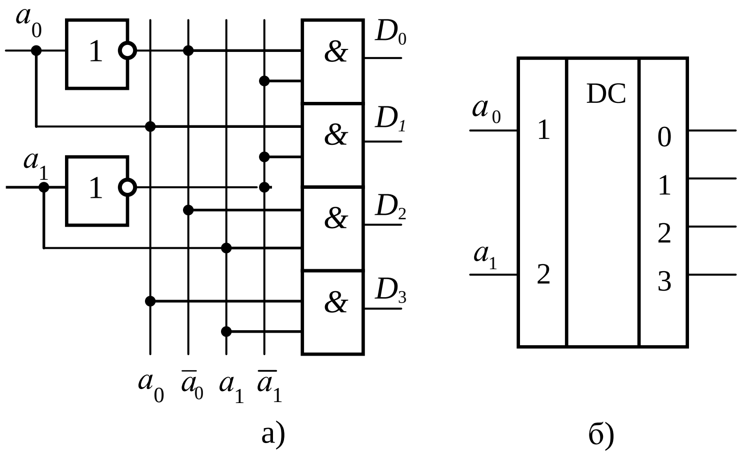


Рисунок 5.13 – Логічна схема дешифратора на 2 входи (а) та його позначення на функціональних схемах (б)

Двійковим шифратором називають комбінаційну схему, що перетворює код “1 з N” у двійковий. Шифратор виконує перетворення, зворотне дешифратору.

Коли збуджено одне з вхідних кіл шифратора, на його виходах формується двійковий код номера збудженого кола. Повний двійковий шифратор має 2^n входів і n виходів. Одне з основних призначень шифратора – перетворення десяткових чисел, введених з клавіатури, у двійковий код (тетради двійково-десяткового коду 8-4-2-1). У цьому випадку потрібний неповний шифратор “1 з 10” в двійковий код 8-4-2-1. Розглянемо на його прикладі принципи синтезу схеми шифратора.

Закон функціонування шифратора зобразимо його таблицею істинності (табл. 5.5). На підставі таблиці логічні вирази вихідних сигналів у формі ДДНФ запишемо так:

$$a_0 = D_1 + D_3 + D_5 + D_7 + D_9;\ a_1 = D_2 + D_3 + D_6 + D_7;$$
$$a_2 = D_4 + D_5 + D_6 + D_7; \qquad a_3 = D_8 + D_9.$$

Таблиця 5.5 – Таблиця істинності шифратора

Збуджуючий вхід	Виходи			
	a_3	a_2	a_1	a_0
D_0	0	0	0	0
D_1	0	0	0	1
D_2	0	0	1	0
D_3	0	0	1	1
D_4	0	1	0	0
D_5	0	1	0	1
D_6	0	1	1	0
D_7	0	1	1	1
D_8	1	0	0	0
D_9	1	0	0	1

Схемно реалізовувати шифратор зручно на елементах І-НЕ, тому зробимо еквівалентне перетворення одержаних співвідношень:

$$a_0 = \overline{\overline{D}_1 \overline{D}_3 \overline{D}_5 \overline{D}_7 \overline{D}_9};\ a_1 = \overline{\overline{D}_2 \overline{D}_3 \overline{D}_6 \overline{D}_7};$$
$$a_2 = \overline{\overline{D}_4 \overline{D}_5 \overline{D}_6 \overline{D}_7}; \qquad a_3 = \overline{\overline{D}_8 \overline{D}_9}.$$

Схему шифратора зображено на рис. 5.14.

5.2.3 Синтез мультиплексорів і демультиплексорів

Мультиплексором називають операційний елемент комбінаційного типу, що передає сигнали з однієї із N вхідних ліній у вихідну.

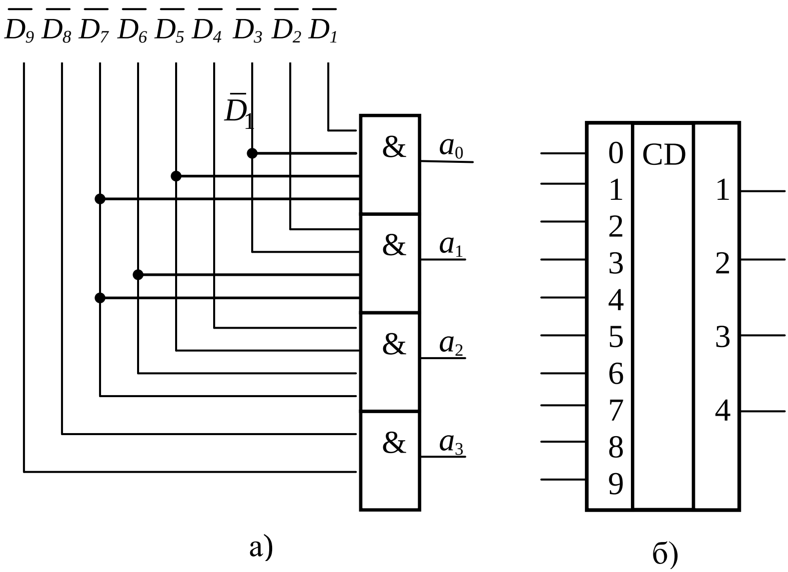


Рисунок 5.14 – Схема шифратора (а) та його умовне позначення (б)

Він являє собою програмований комутатор каналів, що має один вихід і кілька входів, а також керуючі входи (входи адресування), що задають номер входу, який підключається до виходу мультиплексора. Вхідну лінію, з якої передаються дані, вибирає код, що надходить на керуючі входи мультиплексора. Мультиплексор з n керуючими входами $x_{n-1},\dots,x_1$, x_0 може керувати 2^n інформаційними лініями $D_0, D_1, \dots, D_{2^n-1}$ (рис. 5.15).

Із схеми рис. 5.15 бачимо, що залежно від вхідного коду X на одному з виходів дешифратора з'являється логічна одиниця, що дозволяє передавати відповідну змінну D_i на вихід y.

Отже, функцію мультиплексора на чотири входи можна записати так:

$$y = D_0\bar{x}_1\bar{x}_0 + D_1\bar{x}_1 x_0 + D_2 x_1\bar{x}_0 + D_3 x_1 x_0 \,. \qquad (5.10)$$

Узагальнюючи цей вираз на мультиплексор із 2^n входами і n-розрядним керуючим словом, його функцію можна записати у вигляді

$$y = D_0\bar{x}_{n-1}\bar{x}_{n-2}...\bar{x}_0 + D_1\bar{x}_{n-1}\bar{x}_{n-2}...x_0 + \cdots + D_{2^n-1}x_{n-1}x_{n-2}...x_0\,, \quad (5.11)$$

де $\bar{x}_{n-1}\bar{x}_{n-2}...\bar{x}_0$ ... і т. д. – конституенти одиниці.

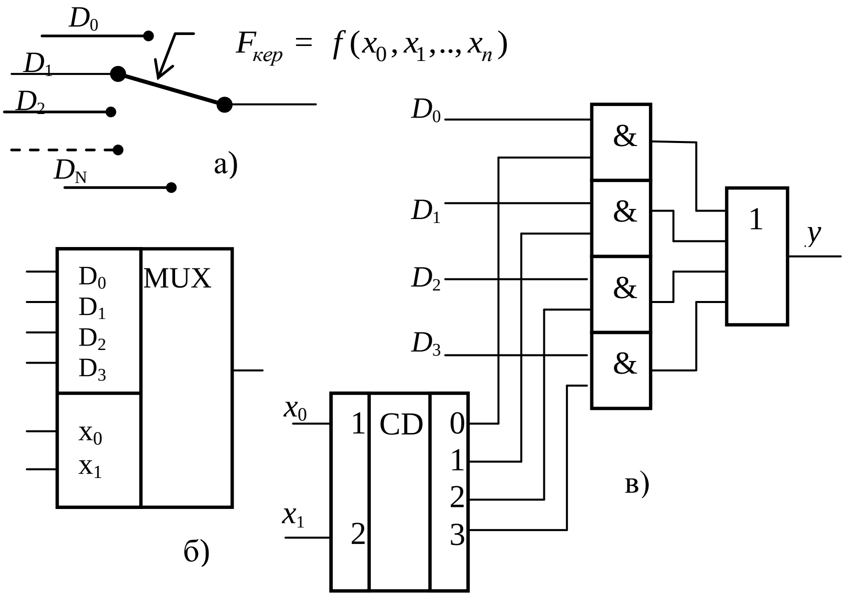


Рисунок 5.15 – Принцип роботи мультиплексора (а), його умовне позначення (б) та схемна реалізація (в)

Вираз (5.11) для виходу мультиплексора відповідає виразу теореми розкладання будь-якої ЛФ за її одиничними значеннями. Це дає можливість реалізувати з допомогою мультиплексора будь-яку ЛФ, що спрощує конструювання цифрових схем. Розглянемо це на прикладі логічної функції (5.1) для мажоритарної схеми з прикладу 5.1 (рис. 5.16):

$$y = \overline{x_1}\cdot x_2\cdot x_3 + x_1\cdot \overline{x_2}\cdot x_3 + x_1\cdot x_2\cdot \overline{x_3} + x_1\cdot x_2\cdot x_3.$$

Логічні змінні			Функція
x_3	x_2	x_1	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Рисунок 5.16 – Реалізація логічної функції (5.1) за допомогою мультиплексора

Із рис. 5.16 випливає, що коли на інформаційні входи мультиплексора подано логічні 0 і 1 згідно із стовпцем значень ЛФ у таблиці істинності, то в разі подавання на його керуючі входи різних комбінацій логічних змінних x_2, x_1, x_0 на виході з’являтимуться значення функції y цих змінних. Наприклад, у випадку подавання коду 111 до виходу мультиплексора підключається інформаційний вхід 7 і передає на вихід логічну 1, що відповідає значенню функції (5.1) на цьому наборі змінних. Таким чином, замість 4 інтегральних схем на друкованій платі буде використано лише одну, що значно спрощує конструкцію цифрового пристрою.

Демультиплексор – це операційний елемент комбінаційного типу, який передає дані із вхідної лінії на одну із N вихідних (рис. 5.17).

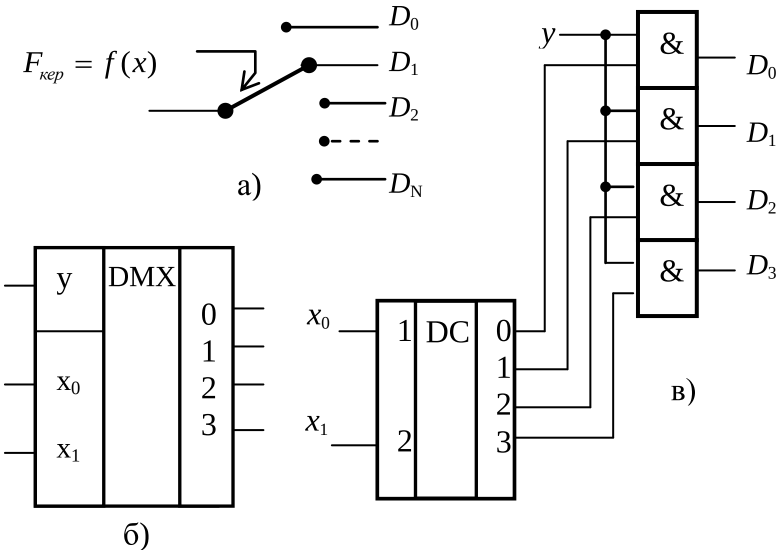


Рисунок 5.17 – Принцип дії демультиплексора (а),
його умовне позначення (б) і зразок
схемної реалізації на базі дешифратора (в)

Вхідна змінна y на інформаційному вході (рис.5.17 в) передається на один з чотирьох виходів, що його визначає вхідний код $X = (x_1 x_0)$:

$$D_0 = y\bar{x}_1\bar{x}_0;\ \ D_1 = y\bar{x}_1 x_0;\ \ D_2 = yx_1\bar{x}_0;\ \ D_3 = yx_1x_0.$$

Щоб одержати демультиплексор n-розрядного слова $y = (y_{n-1},...,y_0)$ на N виходів, паралельно вмикають n демультиплексорів, що мають один вхід і N виходів.

5.2.4 Синтез комбінаційних схем на основі програмованих логічних матриць

Логічним елементом, досить зручним для використання у схемах автоматів, є програмована логічна матриця (ПЛМ), яка в літературі називається програмованою логічною інтегральною схемаю (ПЛІС). Структурна схема ПЛМ показана на рис. 5.18.

ПЛМ складається з двох матриць. Матриця М1 формує k кон'юнкцій вхідних змінних, а матриця М2 – n диз'юнкцій від кон'юнкцій М1. Число входів може досягати десятків, а число ланцюгів $z_1, z_2, \dots z_k$ – більше ста.

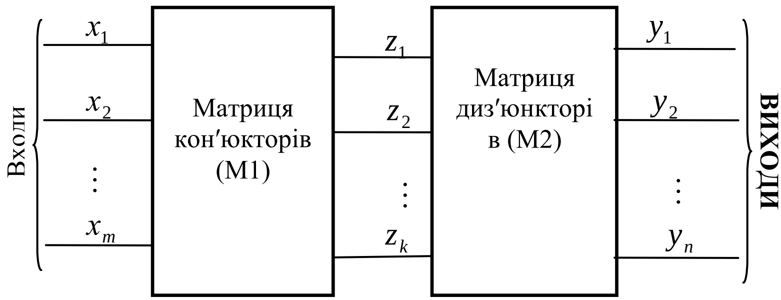


Рисунок 5.18 – Структурна схема ПЛМ

Функціональна схема ПЛМ (рис. 5.19) допускає безліч варіантів обробки вхідних сигналів. Вхідні елементи ПЛМ (рис. 5.19) дозволяють мати усі вхідні змінні як у прямій, так і в інверсній формі. На входи будь-якого елемента "І" подані усі вхідні змінні та їхні інверсії. До входів кожного елемента АБО підключені виходи всіх елементів "І". Нарешті, вихідні елементи дозволяють одержати кожну з вихідних функцій у прямому або інверсному виді.

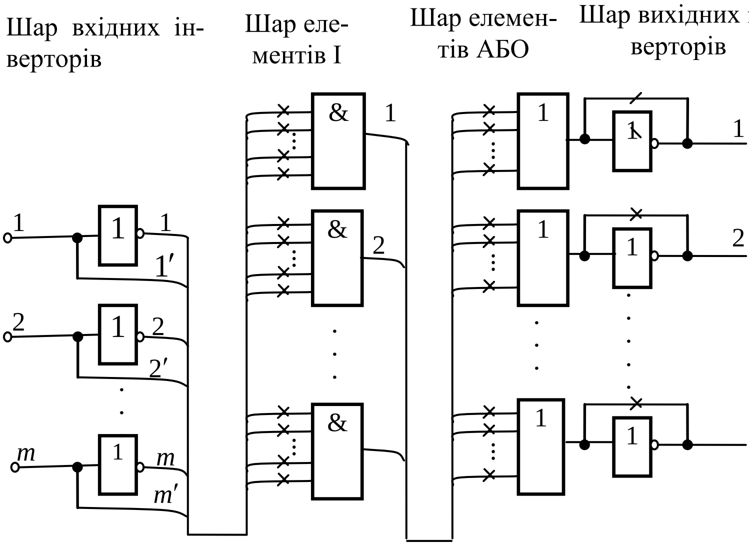


Рисунок 5.19 – Функціональна схема ПЛМ

Електрична принципова схема ПЛМ зображена на рис. 5.20. При побудові матриць М1 і М2 на перетині горизонтальних і вертикальних ліній включаються напівпровідникові біполярні або МОП-транзистори. Вхідні сигнали x_1, x_2, …, x_n або їх інверсії з’єднуються через транзистори з інформаційними шинами, утворюючи кон’юнкції z_k тих вхідних змінних, з’єднання із шинами яких були зроблені:

$$z_k = x_1 x_2 ... x_n = \prod_{k=1}^{n} x_k \ , \tag{5.12}$$

де x_k може набирати значення x_k, $\overline{x_k}$ або 1 (одиниця буде у випадку відсутності зв'язку зі змінною x_k).

Кількість функцій z_k буде залежати від числа логічних елементів, що формують вертикальні шини. Сформовані аналогічно кон’юнкціям z_k вихідні сигнали y_n відповідають рівнянню $y_n = z_1 \vee z_2 \vee ... \vee z_k$.

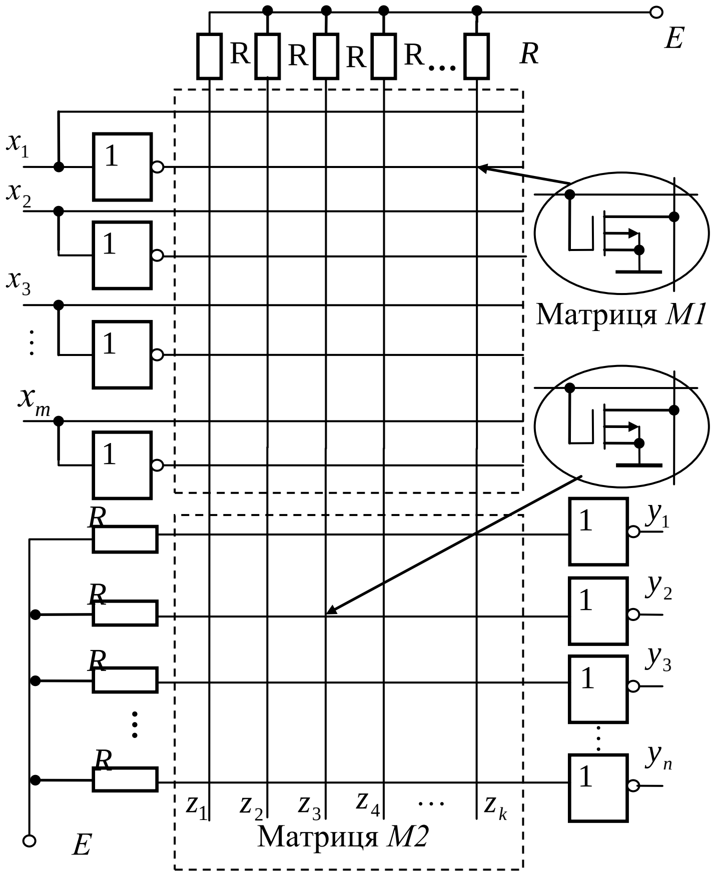


Рисунок 5.20 – Принципова схема ПЛМ

Принцип реалізації операції диз’юнкції в схемі рис. 5.20 показаний на рис. 5.21, а кон’юнкції – на рис. 5.22.

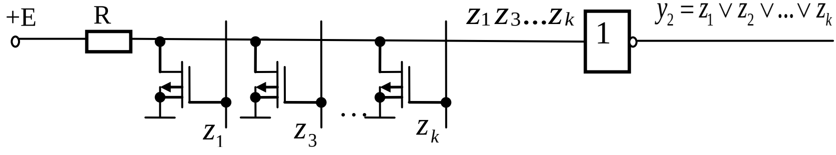


Рисунок 5.21 – Принцип реалізації операції диз’юнкції

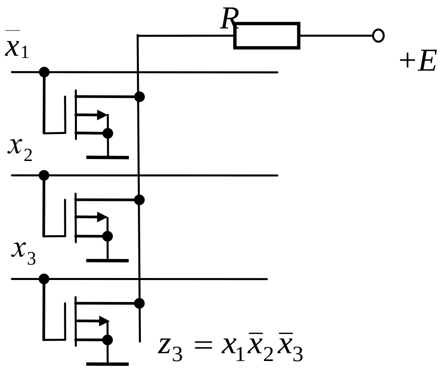


Рисунок 5.22 – Принцип реалізації операції кон’юнкції

Програмування ПЛМ залежно від її типу може проводитися або виробником ПЛМ, або у споживача. Є також такі ПЛМ, що допускають повторне програмування. Нехай потрібно запрограмувати ПЛМ для синтезу функції F і Q, заданих у ДДНФ (або таблицею істиннності):

$$F = \bar{x}_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1\bar{x}_2x_3x_4 + x_1x_2x_3\bar{x}_4;$$

$$Q = x_1\bar{x}_2\bar{x}_3x_4 + x_1x_2x_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1x_2\bar{x}_3\bar{x}_4.$$

Розглянемо перший член ДДНФ функції F: $\bar{x}_1\bar{x}_2\bar{x}_3x_4$. У ньому значення вхідних змінних x_1, x_2, x_3, x_4 дорівнюють 0001. Це означає, що в ПЛМ, в матриці “І” необхідно першу вертикальну шину z_1 з’єднати з прямою шиною x_4 та інверсними шинами x_1, x_2 $і$ x_3. Аналогічно формуються і решта елементарних кон'юнкцій (члени ДДНФ) для функцій F і Q.

На матриці Або потрібні для виходу кожної функції вертикальні шини об’єднуються перемичками-транзисторами і утворюють диз'юнкції

$$F = z_1 + z_2 + z_3 + z_4 + z_5\,,$$

$$Q = z_2 + z_5 + z_6 + z_7\,.$$

Для зручності використовують спрощене накреслення схем ПЛМ у вигляді таблиць, де перемички-транзистори для прямих шин замінені на “1”, а для інверсних шин – на “0”.

Новий імпульс розвитку ПЛІС надав розвиток комунікаційних технологій. Саме тут, на великих потоках і граничних швидкостях обробки інформації ПЛІС дозволяє реалізувати спеціалізовану структуру, тому при однаковій тактовій частоті реалізований пристрій має істотні переваги у швидкодії порівняно з мікропроцесорами. З цією особливістю пов’язана й інша область застосування ПЛІС, що швидко розвивається – реалізація функцій цифрової обробки сигналів. На сьогодні великі інтегральні схеми програмувальної логіки мають ступінь інтеграції до декількох мільйонів еквівалентних вентилів, швидкодія (введення/виведення) більше 400 МГц. Як засоби опису проектів застосовуються мови високого рівня типу HDL (Hardware Description Language), наприклад, Altera HDL, VNDL, Veriolog HDL.

Контрольні питання

Якими моделями можна описати комбінаційну схему?

З яких етапів складається синтез КС?

Як визначається складність комбінаційної схеми?

В чому полягає аналіз комбінаційної схеми?

Назвіть методи усунення “гонок” в комбінаційних схемах.

Які типові вузли у вигляді комбінаційних схем використовуються в цифрових автоматах?

Назвіть відомі вам типи суматорів.

Який пристрій називається суматором?

Наведіть схему півсуматора.

Наведіть схему повного однорозрядного суматора.

Накресліть схему 4-розрядного суматора з паралельним перенесенням.

Що таке дешифратор і як він використовується в цифровому автоматі?

Які переваги й недоліки можна помітити у лінійних, каскадних і прямокутних дешифраторах?

Дайте визначення шифратора.

Де використовуються шифратори?

Чим відрізняються мультиплексор і демультиплексор і де вони використовуються?

Накресліть функціональну схему ПЛМ і поясніть принцип її дії.

Які переваги надає використання ПЛМ в схемах автоматів?

5. 3 Структурний синтез цифрових автоматів

Кінцева мета проектування будь-якого цифрового пристрою – побудова його структурної схеми, що визначає склад логічних елементів та електричні зв’язки між ними. Процес одержання такої схеми називають структурним синтезом (рис.5.25).

На етапі початкового опису цифрового пристрою задають закон його функціонування у вигляді таблиці істинності, коли це КС, або блок-схеми алгоритму, коли це цифровий автомат (ЦА). На другому етапі розпочинають формалізований опис цифрового пристрою. Для КС його задають у вигляді логічних виразів, а для ЦА – таблицями переходів–виходів або граф-схемою алгоритму роботи абстрактного автомата.

Загальноприйнятий канонічнй метод структурного синтезу ЦА дозволяє звести задачу побудови автомата з пам’яттю до задачі синтезу комбінаційних схем. Права гілка алгоритму рис. 5.25 відбиває етапи, виконувати які необхідно під час структурного синтезу ЦА. Щоб краще зрозуміти їх суть, уточнимо поняття структурного автомата, який є кінцевою метою синтезу (рис. 5.26).

На відміну від абстрактного автомата, що має один вхідний та один вихідний канали, на які надходять сигнали в алфавітах $X = \{x_1, ..., x_f, ..., x_F\}$ та $W = \{w_1, ..., w_g, ..., w_G\}$ відповідно, *структурний автомат* має L вхідних каналів $x_1,\ldots,x_l,\ldots,x_L$ і N вихідних $y_1,\ldots,y_n,\ldots,y_N$, на яких з’являються сигнали у структурному алфавіті автомата.

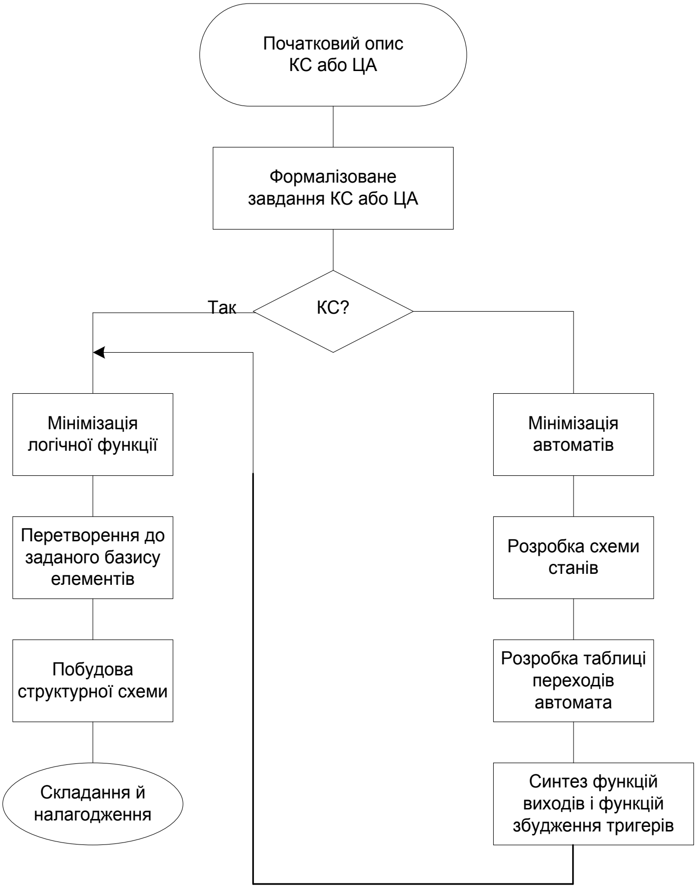


Рисунок 5.25- Етапи структурного синтезу ЦА

Кожен вхідний сигнал z_f абстрактного автомата можна закодувати вектором вхідних сигналів структурного автомата завдовжки L: $z_f = (x_{f_1},\ldots,x_{f_l},\ldots,x_{f_L}),\quad f=\overline{1,F},\quad x_{f_l}\in\{0,1\}$.

Кожен вихідний сигнал y_g абстрактного автомата можна закодувати вектором вихідних сигналів структурного автомата завдовжки N:

$$w_g = (y_{g_1},\ldots,y_{g_n},\ldots,y_{g_N}),\quad g=\overline{1,G},\quad y_{g_n}\in\{0{,}1\}.$$

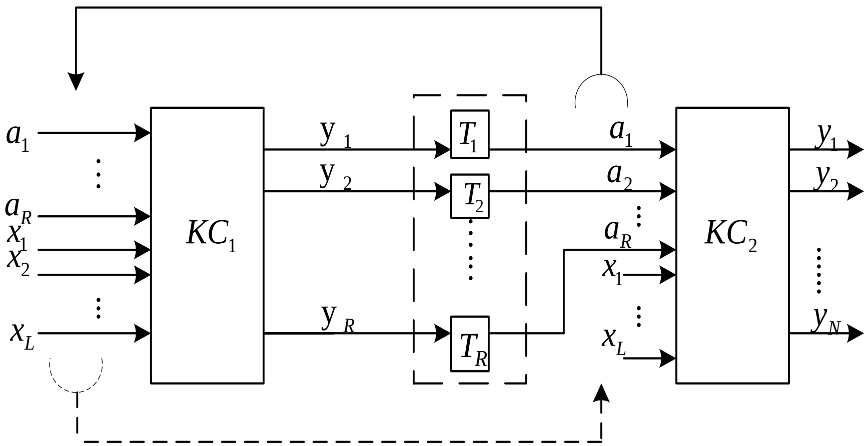


Рисунок 5.26 - Функціональна схема структурного автомата

Стани у структурному автоматі фіксує пам'ять, що містить R елементарних запам'ятовувальних елементів $T_1, T_2, \ldots, T_R$. Кожен стан q_m абстрактного автомата закодовано у структурному автоматі вектором довжини R вихідних сигналів елементів пам'яті:

$$q_m = (a_{m_1},\ldots,a_{m_r},\ldots,a_{m_R}), m=\overline{1,M},\quad a_m\in\{0{,}1\}.$$

Подання автомата схемою рис. 5.26 дозволяє звести задачу побудови автомата з пам'яттю до задачі синтезу комбінаційних схем $КС_1$ і $КС_2$ та пристрою пам'яті на елементах пам'яті T_1, T_2, …, T_R. Як елементи пам'яті використовуються тригери різних типів, сукупності двійкових значень виходів яких (на рис. 5.26 це $a_{m_1},\ldots,a_{m_r},\ldots,a_{m_R}$) утворюють двійкові вектори, що кодують стани q_m автомата. Тому проектування і побудова цифрових автоматів вимагає знання принципів побудови і роботи тригерів, які розглядаються в наступному підрозділі.

5.3.1 Тригери

Базовою структурною ланкою для побудови комбінаційних схем є логічний елемент. У скінченних автоматах (послідовнісних схемах) базова ланка – тригер. З'єднуючи тригери один з одним, можна побудувати такі найпростіші елементи цифрових автоматів, як лічильники, регістри, запам'ятовувальні пристрої.

Тригер являє собою пристрій з двома стійкими станами, одному з яких приписується значення 1, а іншому 0. В ці стани тригер переходить під дією вхідних сигналів, які надходять на один або декілька його входів. Узагальнена структурна схема тригерного пристрою наведена на рис. 5.27.

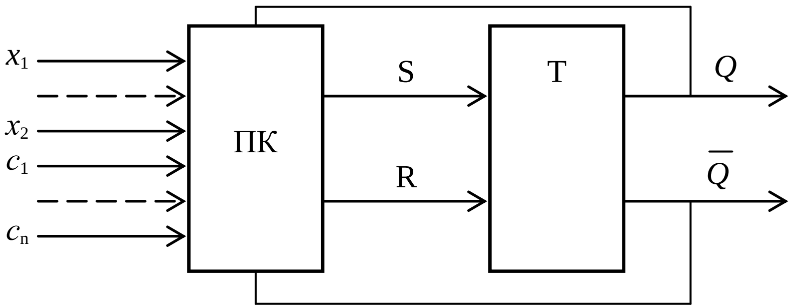


Рисунок 5.27 - Узагальнена структурна схема тригера

На схемі рис. 5.27 застосовані такі позначення: T – власне тригер, ПК – пристрій керування, x_1, ..., x_n – інформаційні входи тригерного пристрою, c_1, ..., c_n – тактові входи, S – вхід встановлення тригера в стан 1, R – вхід встановлення тригера в стан 0, Q – прямий вихід тригера, $\overline{Q}$ – інверсний вихід тригера.

Існує велика різноманітність тригерів, які відрізняються один від одного виконуваною функцією, схемотехнічною реалізацією, способом запису інформації та іншими ознаками.

За способом запису інформації тригери класифікують на асинхронні та синхронізовані. В асинхронних тригерах запис інформації здійснюється безпосередньо з надходженням сигналу на вхід тригера. У синхронізованих тригерах запис здійснюється лише за поданням дозволяючого тактувального імпульсу.

За способом синхронізації розрізняють однотактні та двотактні тригери. За способом організації логічних зв'язків розрізняють тригери з роздільним встановленням станів 0 та 1 (RS - тригери), тригери з лічильним входом (T - тригери), універсальні тригери з окремим установленням станів 1 та 0 (JK - тригери), тригери із прийняттям інформації одним входом (D - тригери) тощо.

На детальний розгляд заслуговує тригер RS – типу, оскільки на його основі створюються тригери всіх інших типів. Тригером RS-типу називають елементарний автомат з двома стійкими станами, який має два інформаційних входи R та S, такі, що при $S=1$ та $R=0$ тригер набуває одиничного стану $Q=1$, а при $R=1$ та $S=0$ тригер набуває нульового стану $Q=0$. Вхід S називається входом встановлення, вхід R – входом скидання.

Таблиця 5.6 – Таблиця істинності RS-тригера

t_n		t_{n+1}
R_n	S_n	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	x

Із таблиці 5.6 видно, що при одночасному надходженні на входи R і S логічних сигналів, які мають значення 1, тригер набуває невизначеного стану (позначено як “х”). Тому логічні пристрої на основі RS-тригерів повинні будуватися з урахуванням виключення комбінації сигналів $R \cdot S = 1$. При цьому логічне рівняння RS-тригера записується у вигляді

$$Q_{n+1} = S_n + \overline{R_n} \cdot Q_n,\; R_n \cdot S_n = 0,$$

а логічний вираз для інверсного виходу буде мати вигляд

$$\overline{Q}_{n+1} = \overline{S_n + \overline{R_n} \cdot Q_n}\,. \tag{5.15}$$

Шляхом еквівалентного перетворення приведемо (5.15) до вигляду, який зручний для реалізації на інтегральних логічних елементах Або-Не:

$$\begin{aligned} \overline{Q}_{n+1} &= \overline{S}_n \cdot (\overline{Q_n \cdot \overline{R_n}}) = \overline{S}_n \cdot (\overline{Q_n} + R_n) = \\ &= \overline{\overline{\overline{S}_n \cdot (\overline{Q_n} + R_n)}} = \overline{S_n + \overline{(\overline{Q}_n + R_n)}}. \end{aligned} \tag{5.16}$$

Схема RS-тригера на логічних елементах Або-Не, яка реалізує (5.16), показана на рис. 5.28 а, а її функціональне позначення наведено на рис. 5.28 б.

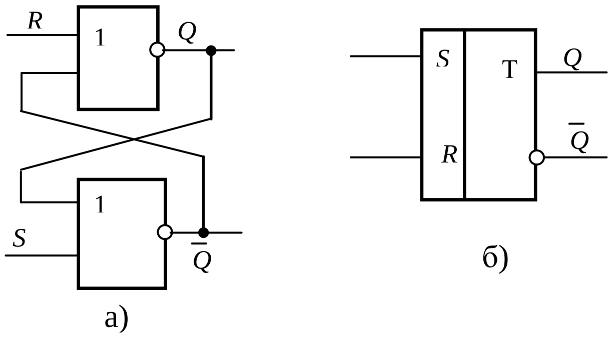


Рисунок 5.28 - Схема асинхронного RS-тригера

У синхронізованому RS-тригері на вході кожного плеча є додаткові схеми збігу, перші входи яких з'єднані і є входами синхроімпульсів. Інформація, що надходить на входи R і S, може передаватися лише за наявності на вході C синхронізувального імпульсу. Вхідна інформація повинна бути подана в парафазному коді, тобто $S=0$ та $R=1$ або $S=1$ та $R=0$. Схема синхронного RS-тригера на елементах Або – Не та його умовне позначення показані на рис. 5.29а та рис. 5.29б відповідно, а часова діаграма його роботи – на рис.5.29в.

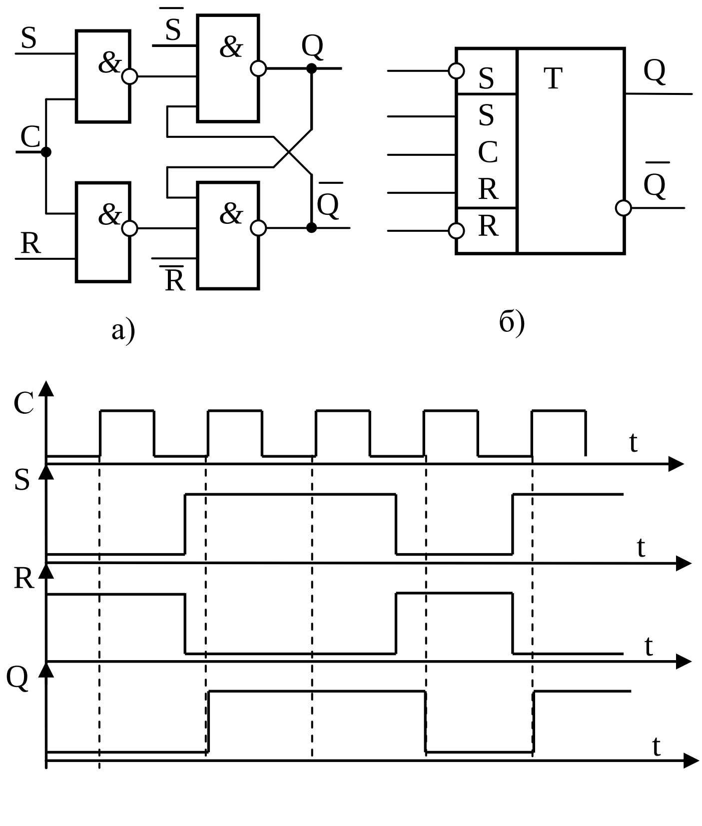


Рисунок 5.29 – Схема синхронного RS-тригера (а) на елементах І-Не, його умовне позначення (б) та часові діаграми (в)

У регістрових пристроях та лічильниках інформацію в тригер можна заносити тільки після того, як завершено передавання інформації про попередній його стан в наступний тригер. Цього досягають, використовуючи двоступеневі тригери, які можна побудувати з допомогою з'єднання двох одноступеневих тригерів і використання протифазних тактувальних сигналів, подаваних на синхровходи цих ступенів (рис. 5.30).

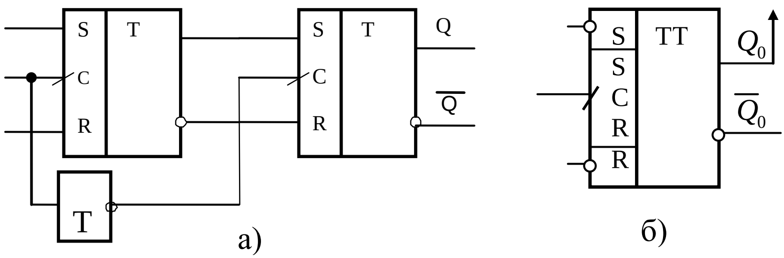


Рисунок 5.30 – Схема двоступеневого RS - тригера (а) та його умовне позначення (б)

Фронт синхроімпульсу *C* опитує входи *R* та *S*, і інформація запам'ятовується в першому тригері. Далі вона передається в другий тригер, входи якого і є виходами двоступеневого тригера. Часова діаграма цього тригера зображена на рис. 5.31.

Тригер D-типу відомий під назвою "тригер затримки". Тригером D-типу називається логічний елемент з двома стійкими станами та з одним інформаційним входом, закон функціонування якого можна описати логічним рівнянням

$$Q_{n+1} = D_n \qquad (5.17)$$

Рівняння (5.17) показує, що в $(n+1)$ - й тактовий момент часу вихідний стан збігається із сигналом, що діє на вході *D* у попередній (n - й) момент часу. Структурна схема синхронізованого D-тригера показана на рис. 5.32 а, а на рис. 5.32 б показане його умовне позначення.

Тригером T-типу (рис. 5.33), або лічильним тригером, називають елементарний автомат із двома стійкими та одним інформаційним входом *T*, надходження сигналу на який змінює стан Т-тригера на протилежний (табл. 5.7).

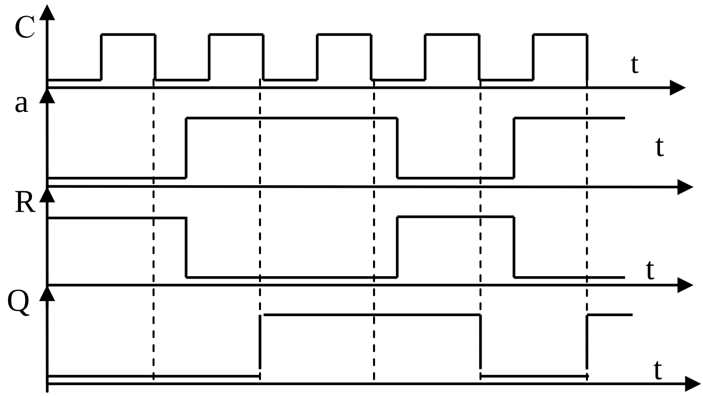


Рисунок 5.31 – Часові діаграми сигналів
для синхронізованого двоступеневого *RS*-тригера

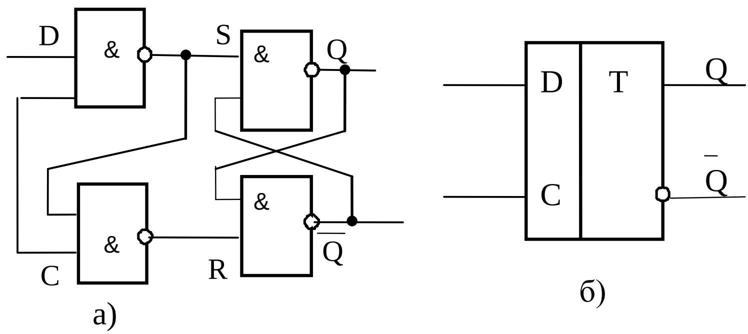


Рисунок 5.32 - Структурна схема D-тригера (а)
та його умовне позначення (б)

Логічне рівняння Т-тригера подано виразом

$$Q_{n+1} = Q_n \cdot \overline{T_n} + \overline{Q}_n \cdot T_n \quad . \qquad (5.18)$$

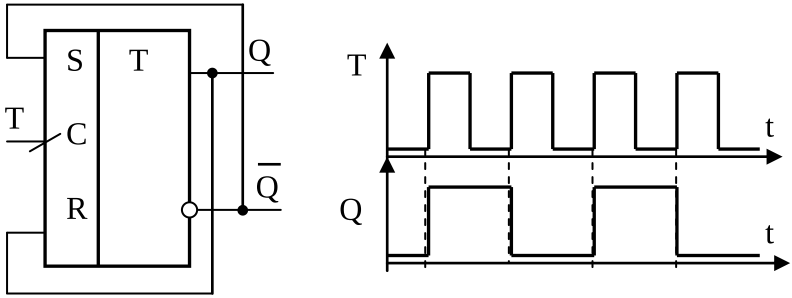


Рисунок 5.33 - Несинхронізований Т-тригер і часова діаграма його роботи

Таблиця 5.7 - Таблиця переходів Т-тригера

Режим роботи	Вхід	Вихід	
	Tn	Qn+1	$\overline{Q}_{n+1}$
Перемикання	1	$\overline{Q}_n$	Q_n
Зберігання	0	Q_n	Q_n

JK-тригер – найпоширеніший універсальний тригер, який має характеристики всіх інших типів тригерів. Його закон функціонування можна задати таблицею 5.8.

Тригером JK-типу називають елементарний автомат з двома стійкими станами та двома інформаційними входами *J* і *K*, який за умови *J*·*K*=1 робить інверсію попереднього стану $Q_{n+1}=\overline{Q}_n$, а в інших випадках функціонує згідно з таблицею істинності *RS*-тригера, коли вхід *J* еквівалентний входу *S*, а вхід *K* – входу *R* (табл. 5.6). Логічне рівняння JK-тригера

$$Q_{n+1} = J_n\overline{Q}_n + \overline{K}_n Q_n \qquad (5.19)$$

Таблиця 5.8 – Таблиця переходів JK-тригера.

Режими роботи	Входи		Виходи	
	J	*R*	*Q*	$\overline{Q}_{n+1}$
Зберігання	0	0	*Q*	$\overline{Q}_n$
Установлення	1	0	0	1
Установлення	1	0	1	0
Перемикання	1	1	$\overline{Q}_n$	Q_n

Схема синхронного JK-тригера, побудованого на основі синхронізованого RS-тригера, показана на рис. 5.34 а, а його часова діаграма роботи – на рис. 5.34 б.

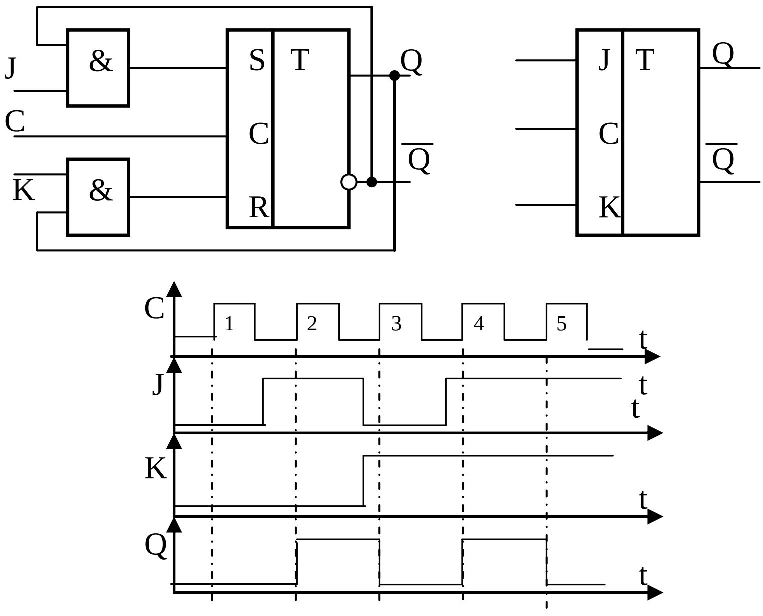


Рисунок 5.34 – Структурна й функціональна схеми JK-тригера (а) і часові діаграми його роботи (б)

Із рисунка 5.34 б бачимо, що тактовий імпульс 1 не змі-нює стану тригера за нульових рівнів сигналів на входах *J* та *K* (режим зберігання), тактовий імпульс 2 ставить тригер в одиничний стан за значень сигналів J=1 і K=0 (режим установлення в 1), а тактовий імпульс 3 – в нульовий за значень J=0 і K=1 (режим установлення в 0), тактові імпульси 4 та 5 щоразу перемикають тригер у протилежний до попереднього стан за значень вхідних сигналів J=1 і K=1 (режим перемикання або лічильний).

Під час синтезу схем цифрових автоматів зручними для використання є *характеристичні таблиці тригерів* (табл. 5.9 – 5.12). Вони визначають значення вхідних сигналів в залежності від заданих перемикань тригерів

Таблиця 5.9 – RS-тригер

Q_t	Q_{t+1}	S	R
0	0	0	*
0	1	1	0
1	0	0	1
1	1	*	0

Таблиця 5.10 – JK-тригер

Q_t	Q_{t+1}	J	K
0	0	0	*
0	1	1	*
1	0	*	1
1	1	*	0

Таблиця 5.11 – Т-тригер

Q_t	Q_{t+1}	T
0	0	0*
0	1	10
1	0	1
1	1	0

Таблиця 5.12 – D-тригер

Q_t	Q_{t+1}	D
0	0	0*
0	1	1*
1	0	0
1	1	1

5.3.2 Синтез схеми автомата Мура за заданим графом

Розглянемо приклад (розрахований на узагальнення) синтезу схеми автомата Мура на тригерах JK-типу в базисах І−Не та І−Або−Не за заданим графом (рис. 5.35).

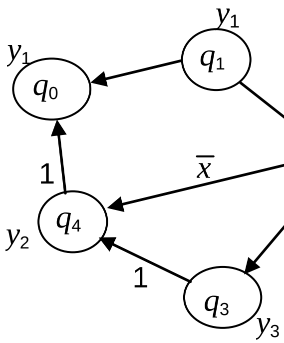


Рисунок 5.35 - Граф автомата Мура

Для управління переходами автомата тут використовується один двійковий вхідний сигнал $x \in \{0,1\}$ (відповідно $\overline{x}$ і x). Символ “1” на дузі графа позначає будь-яке значення вхідного сигналу (0 або 1), при цьому перехід до нового стану відбувається за синхросигналом.

Синтез виконаємо, дотримуючись алгоритму на рис. 5.25, реалізувавши наступні кроки.

1. Визначення кількості елементів пам’яті (тригерів).

У цьому прикладі кількість станів автомата M=5, тому кількість тригерів

$$n = \lceil \log_2 5 \rceil = \lceil 2{,}32 \rceil = 3,$$

де кутові дужки означають взяття найменшого цілого, більшого від виразу в дужках.

2. Визначення типу тригерів. Тип тригерів залежно від постановки задачі може задаватися або вибиратися, виходячи з міркувань доцільності. За завданням прикладу повинні використовуватися JK-тригери.

3. Кодування станів автомата. Закодуємо стани автомата трирозрядними двійковими числами в їх натуральному порядку (табл. 5.13).

Таблиця 5.13 - Кодування станів автомата

Стан q_i	Код стану		
	T_3	T_2	T_1
q_0	0	0	0
q_1	0	0	1
q_2	0	1	0
q_3	0	1	1
q_4	1	0	0

4. Складання таблиці переходів. Стрічки таблиці відповідають переходам ЦА із стану до стану, відзначених дугами на графі рис. 5.35, а в стовпчиках зазначаємо параметри цих переходів: коди попереднього $q_{i\text{п}}$ і наступного $q_{i\text{н}}$ станів, поданих логічними значеннями на виходах тригерів пам'яті (відповідно $T_{3\text{п}}$, $T_{2\text{п}}$, $T_{1\text{п}}$ і $T_{3\text{н}}$, $T_{2\text{н}}$, $T_{1\text{н}}$); вхідні сигнали, які обумовлюють ці переходи, і вихідні сигнали автомата, які цими переходами породжуються; логічні значення сигналів збудження тригерів J_i, K_i, необхідні для їх переведення до потрібного стану.

Таблиця 5.14 – Таблиця переходів автомата

Поч. стан $q_{i\text{п}}$	Код початкового стану			Наст. стан $q_{i\text{н}}$	Сигнали		Значення функцій збудження тригерів					
	T_3	T_2	T_1		Вх	Вих	J_3	K_3	J_2	K_2	J_1	K_1
q_0	0	0	0	q_1	1	y_1	0	*	0	*	1	*
q_1	0	0	1	q_2	1	y_1	0	*	1	*	*	1
q_2	0	1	0	q_3	x	y_2	0	*	*	0	1	*
q_2	0	1	0	q_4	$\bar{x}$	y_2	1	*	*	1	0	*
q_3	0	1	1	q_4	1	y_3	1	*	*	1	*	1
q_4	1	0	0	q_0	1	y_2	*	1	0	*	0	*

5. Одержання мінімізованих функцій збудження, що переключають тригери у новий стан:

$$J_i = J_i(T_{1n},\ T_{2n},\ T_{3n}, x);\ K_i = K_i(T_{1n},\ T_{2n},\ T_{3n}, x);\ \ (i = 1,\ 2,\ 3).$$

Для цього можна скористуватися картами Карно (рис. 5.36).

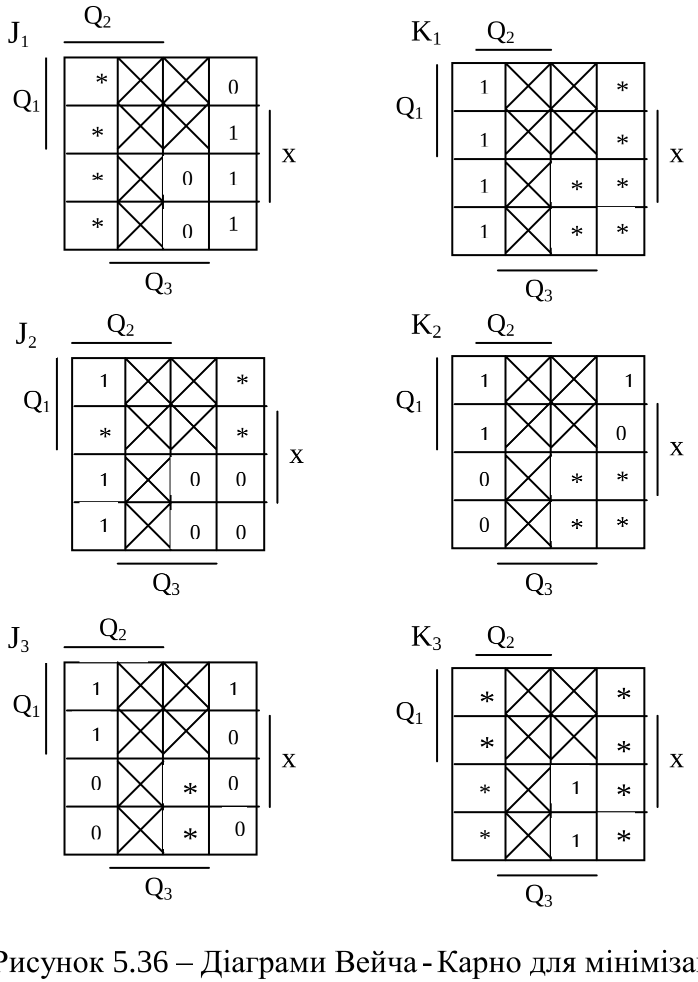


Рисунок 5.36 – Діаграми Вейча - Карно для мінімізації функцій збудження тригерів

На картах Карно (рис. 5.36) хрестиком позначені клітинки, що відповідають кодам станів, які не використовуються. Зрозуміло, що вони можуть включатися в контури поряд із зірочками, що позначають невизначені значення сигналів.

Результати мінімізації подані виразами:

$$J_1 = x\overline{T}_3 + \overline{T}_2\overline{T}_3, \; K_1 = 1;$$

$$J_2 = T_1, \;\; K_2 = \overline{x}T_2 + T_1T_2;$$

$$J_3 = K_2; \;\; K_3 = 1.$$

6. Перехід до базису І-НЕ, скориставшись законом подвійного заперечення і правилом де Моргана:

$$J_1 = \overline{\overline{x\overline{T}_3} \cdot \overline{\overline{T}_2\overline{T}_3}}, \;\; K_1 = 1; \;\; J_2 = T_1, \;\; K_2 = \overline{\overline{\overline{x}T_2} \cdot \overline{T_1T_2}};$$

$$J_3 = K_2, \;\; K_3 = 1.$$

7. Отримання мінімізованих функцій, що описують вихідні сигнали: $y_i = y_i\,(T_1, T_2, T_3)$; $(i = 1, 2, 3)$.

Вихідні сигнали виразимо в базисі І-Або-Не.

Оскільки в автоматі Мура вихідні сигнали залежать явно лише від стану, то вхідний сигнал x буде відсутнім у виразах, що описують y_i. Для цього в таблиці переходів знаходимо стрічки, в яких записаний вихідний сигнал y_i і записуємо його логічний вираз згідно із значеннями виходів тригерів T_{1n}, T_{2n}, T_{3n}. Наприклад, вихідний сигнал y_1 знаходиться в першій або другій стрічках, що відповідають станам q_0 і q_1, тому логічний вираз для нього запишеться через значення виходів тригерів таким чином:

$$y_1 = \overline{T}_1\overline{T}_2\overline{T}_3 + T_1\overline{T}_2\overline{T}_3 = \overline{T}_2\overline{T}_3,$$

де для отримання кінцевого виразу до першого і другого доданків було застосовано закон склеювання для диз'юнктивної форми.

Аналогічно

$$y_2 = q_2 + q_4 = \overline{T}_1T_2\overline{T}_3 + \overline{T}_1\overline{T}_2T_3.$$

Для мінімізації виразу використаємо діаграму Вейча-Карно для трьох змінних (рис. 5.37).

Із діаграми Вейча − Карно знаходимо мінімізований вираз для сигналу y_2:

$$y_2 = \overline{T}_1 T_2 + T_3 .$$

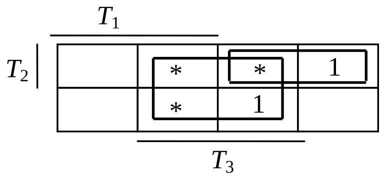


Рисунок 5.37 – Діаграма Вейча - Карно для мінімізації виразу логічного сигналу y_2

Із таблиці переходів записуємо логічний вираз для вихідного сигналу y_3 (5-та стрічка, стан q_3):

$$y_3 = T_1 T_2 \overline{T}_3 .$$

Мінімізуємо його за допомогою діаграми Вейча − Карно (рис. 5.38). Із діаграми отримуємо:

$$y_3 = T_1 T_2 .$$

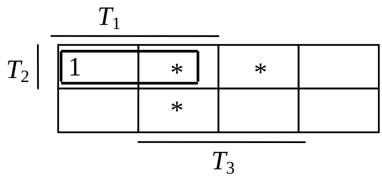


Рисунок 5.38 – Діаграма Вейча - Карно для мінімізації y_3

8. Побудуємо логічну схему автомата, використавши дозволені логічні елементи та тригери заданого типу (рис. 5.39).

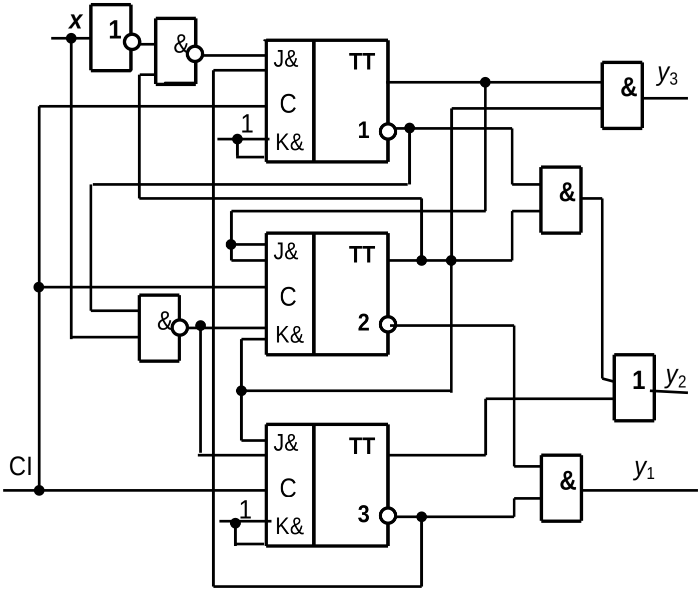


Рисунок 5.39 – Логічна схема синтезованого автомата

5.3.3 Синтез схеми автомата Мілі, заданого алгоритмом

Розглянемо синтез автомата Мілі на прикладі деякого керуючого автомата віртуального процесора для реалізації операції, заданої схемою алгоритму рис. 5.40. На ньому в умовних вершинах стоять вхідні сигнали автомата X1, X2, X3, X4, X5 (сигнали умов операційного блоку процесора), а в операційних вершинах – сигнали керування операційним блоком процесора У1, ..., У11.

Синтез ЦА проводимо, керуючись послідовністю етапів за схемою рис. 5.25.

1. Проводимо розмітку схеми алгоритму (див. розд 4, п. 4.3).

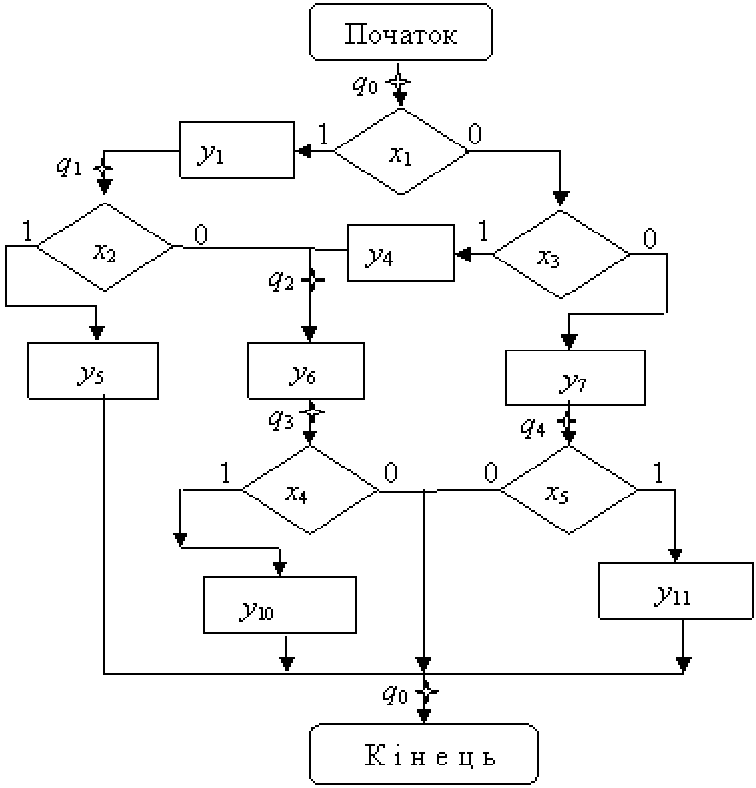


Рисунок 5.40 – Схема алгоритму функціонування автомата

Під час розмічування граф-схеми алгоритму автомата Мілі (рис. 5.40) виходи операторних вершин відзначаються символами внутрішніх станів автомата q_0, q_2, ..., q_n. Для ініціального автомата початкова і кінцева вершини позначаються одним і тим самим станом q_0.

Переходи автомата зі стану в стан здійснюються під впливом вхідних сигналів, відповідних виконанню або невиконанню умови x_i. Невиконання умови відповідає інверсне значення 0 змінної x_i, виконанню умови – пряме значення 1.

При виконанні чи невиконанні кількох умов одночасно перехід здійснюється під впливом сигналу, що подається кон'юнкцією відповідних прямих або інверсних значень змінних.

2. Виконаємо формалізований опис ЦА у вигляді орієнтованого графа (рис. 5.41). Вершинами графа є стани автомата, а його дуги описують всі можливі переходи з одного стану до іншого у послідовності виконання етапів алгоритму роботи автомата. Початок дуги позначається вхідним сигналом, який ініціює перехід з попереднього стану до наступного, а вихід дуги – вихідним керувальним сигналом, який виробляє автомат у цьому переході.

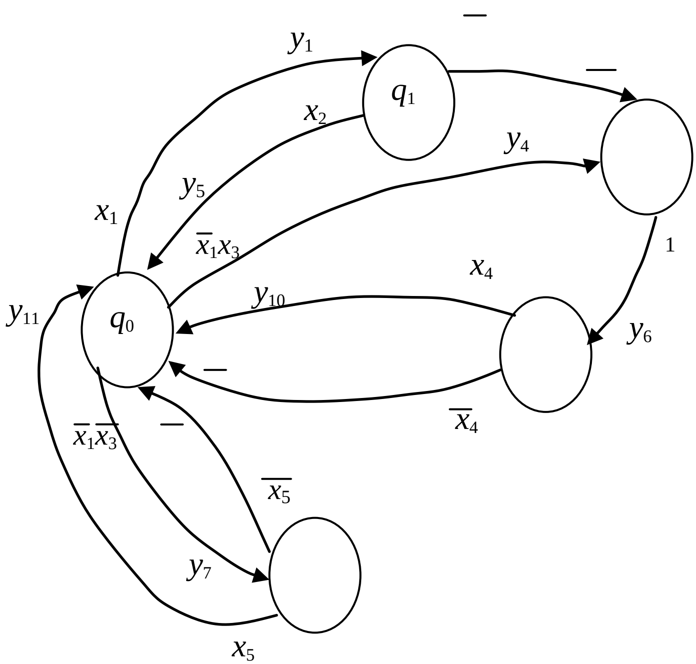


Рисунок 5.41 – Формальний опис ЦА у вигляді орграфа

3. На етапі мінімізації ЦА вибирають абстрактний автомат з мінімальною кількістю станів серед усіх еквівалентних авто-

матів. Для цього в множині станів знаходять еквівалентні класи та об'єднують їх у нові стани (див. розд. 4, п. 4.7).

4. На етапі розроблення схеми станів автомата спочатку визначають кількість елементів пам'яті за формулою

$$n = \lceil \log_2 M \rceil.$$

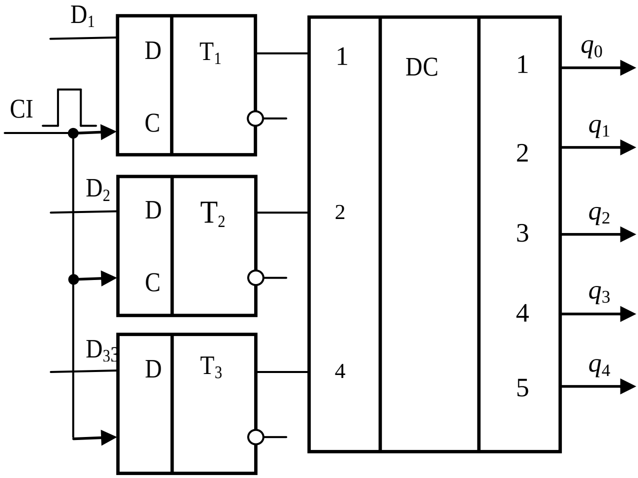
У нашому випадку $M = 5$. Отже, $n = \lceil \log_2 5 \rceil = \lceil 2.32 \rceil = 3$. За елементи пам'яті вибираємо синхронізовані тригери D-типу і задаємо таблицю кодів станів (табл. 5.15).

Таблиця 5.15 - Кодування станів автомата

Стан q_i	Код стану		
	T_3	T_2	T_1
q_0	0	0	0
q_1	0	0	1
q_2	0	1	0
q_3	0	1	1
q_4	1	0	0

Якщо виходи тригерів подати на входи дешифратора з M виходами, то одиничний сигнал на одному з виходів дешифратора покаже стан автомата. Схему станів розроблюваного автомата Мілі подано на рис. 5.42.

5. Цей етап структурного синтезу – розроблення таблиці переходів автомата, яку будують відповідно до графа останнього (рис. 5.41). Кількість рядків у таблиці переходів (табл. 5.16) дорівнює кількості дуг на графі (кількості різних переходів із стану до стану). Для кожного переходу записують початковий стан, його двійковий код (стани тригерів), стан після переходу та його двійковий код, кон'юнкції вхідних сигналів, що являють собою умову переходу (записані на початку дуг орієнтованого графа рис. 5.41), а також вихідні (керувальні) сигнали, що виробляються автоматом під час даного переходу (ними позначені кінці дуг графа).



Таблиця 5.16 – Таблиця переходів автомата

Початковий стан	Код стану			Наступний стан	Код стану			Сигнали		Функції збудження тригерів		
	$T_{3п}$	$T_{2п}$	$T_{1п}$		$T_{3н}$	$T_{2н}$	$T_{1н}$	Вх	Вих	D_3	D_2	D_1
q_0	0	0	0	q_1	0	0	1	x_1	y_1	0	0	1
q_0	0	0	0	q_2	0	1	0	$\bar{x}_1 x_3$	y_2	0	1	0
q_0	0	0	0	q_4	1	0	0	$\bar{x}_1 \bar{x}_3$	y_7	1	0	0
q_1	0	0	1	q_0	0	0	0	x_2	y_5	0	0	0
q_1	0	0	1	q_2	0	1	0	$\bar{x}_2$	–	0	1	0
q_2	0	1	0	q_3	0	1	1	1	y_6	0	1	1
q_3	0	1	1	q_0	0	0	0	x_4	y_{10}	0	0	0
q_3	0	1	1	q_0	0	0	0	$\bar{x}_4$	–	0	0	0
q_4	1	0	0	q_0	0	0	0	x_5	y_{11}	0	0	0
q_4	1	0	0	q_0	0	0	0	$\bar{x}_5$	–	0	0	0

Для тригерів інших типів значення функцій збудження одержують за таблицями переходу даних тригерів.

6. На етапі синтезу функцій переходів і виходів записуємо логічні вирази цих функцій згідно із табл. 5.16 у вигляді досконалих диз'юнктивних нормальних форм, при цьому їх аргументами вважаються початковий стан даного переходу і вхідний сигнал:

$$\begin{cases} D_1 = q_0 x_1 + q_2, \\ D_2 = q_0 \bar{x}_1 x_3 + q_1 \bar{x}_2 + q_2, \\ D_3 = q_0 \bar{x}_1 \bar{x}_3. \end{cases} \qquad (5.20)$$

$$\begin{cases} y_1 = q_0 x_1, \\ y_2 = q_0 \bar{x}_1 x_3, \\ y_5 = q_1 x_2, \\ y_6 = q_2, \\ y_7 = q_0 \bar{x}_1 \bar{x}_3, \\ y_{10} = q_3 x_4, \\ y_{11} = q_4 x_5. \end{cases} \qquad (5.21)$$

Набір функцій (5.20) використовується для побудови логічної схеми переходів автомата, а (5.21) – схеми виходів автомата.

7. Етап мінімізації логічних функцій (5.20) і (5.21) зручно здійснити за методом Вейча-Карно, оскільки число їх аргументів менше 7. Однак у цьому випадку мінімізація не проводиться, оскільки візуально можна бачити, що ні для одного виразу не можна застосувати закон склеювання чи поглинання. Кількість логічних елементів можна зменшити за рахунок того, що деякі вихідні сигнали одержуються на виході логічних елементів, які вже використовуються у функціях переходів (порівняйте, наприклад D_3 і y_7).

8. Етап переходу до заданого базису здійснюється так, як пояснено в попередньому прикладі.

9. На етапі побудови схеми автомата за логічними виразами функцій, приведеними до заданого логічного базису, встановлюють склад необхідних логічних елементів і вузлів, послідовність їх з’єднання в одну схему, і креслять її з дотриманням стандартів.

5.4 Елементарні автомати

Поряд із тригерами до елементарних автоматів можна віднести регістри та лічильники. Елементарними їх можна вважати тому, що вони реалізують найпростіші алгоритми обробки цифрових сигналів і використовуються як базові елементи під час побудови обчислювальних автоматів (мікропроцесорів), та автоматів (мікроконтролерів та спеціалізованих автоматів жорсткого типу), орієнтованих на автоматизоване управління технічними та технологічними об’єктами.

5.4.1 Регістри

Двійкові числа та інша двійково-кодована інформація у загальному випадку містять n біт. За пристрої, що забезпечують їх збереження, використовують регістри, які об'єднують n тригерів в один пристрій.

Регістрами називаються пристрої, що виконують функції прийому, збереження та передачі двійкових слів. Окремі тригери, що входять до складу регістра, називаються розрядами регістра. Як правило, в регістрах використовують тригери RS-, D-, JK-типу. Крім них, до складу регістра входить ряд схем, що забезпечують управління його роботою. За допомогою цих схем можна скинути регістр у стан “0”, взяти слово з іншого регістра, передати слово в інший регістр, перетворити послідовний код слова в паралельний та навпаки, здійснити зсув слова вправо чи вліво на задану кількість розрядів, виконати порозрядні логічні операції над словами. Основною класифікаційною ознакою, за якою розрізняють регістри, є спосіб запису інформації в регістр. За цією ознакою можна виділити такі типи регістрів:

- паралельні;
- послідовні (зсуву);
- паралельно-послідовні.

У *паралельних* регістрах прийом та видача слів виконується за всіма розрядами одночасно, їх основна функція – збереження слова. В них також можуть виконуватись порозрядні операції над словами.

Послідовні регістри характеризуються послідовним записом коду числа, починаючи з молодшого чи старшого розряду, шляхом послідовного зсуву коду числа тактовими імпульсами. Через це їх ще називають *зсувними* регістрами. Розрізняють реверсивні (можливий зсув як вліво, так і вправо) та нереверсивні (зсув можливий лише в одному напрямку) регістри.

Паралельно-послідовні регістри мають одночасно входи паралельного та послідовного прийому (видачі) слів, а також можуть виконувати перетворення паралельних кодів на послідовні та навпаки.

За кількістю каналів передачі даних розрізняють парафазні та однофазні регістри. У парафазних кожний розряд слів передається по двох ланцюгах – у вигляді прямого значення, наприклад, Q_i – одним ланцюгом та інверсного $\overline{Q_i}$ – іншим. В однофазних регістрах кожний розряд передається лише одним ланцюгом.

Схему найпростішого n-розрядного *паралельного* однофазного регістра можна скласти, об’єднавши n RS-тригерів (рис. 5.43). Входи $\overline{a_n}, ..., \overline{a_0}$ такого регістра призначені для паралельного введення розрядів двійкового слова в інверсному коді. Оскільки на входи S і R тригерів повинні завжди надходити протилежні сигнали, то на вході S кожного тригера використано допоміжні інвертори. Запис двійкового слова $a_n \ldots a_0$ дозволяє тактовий імпульс *C*, що одночасно надходить на синхровходи всіх тригерів регістра. Після закінчення цього імпульсу значення y_n, …, y_0 розрядів двійкового слова з'являться на виходах регістра.

Зсувні регістри (рис. 5.44) будують на синхронізованих тригерах, з'єднаних послідовно так, що вихідний сигнал одного з них надходить на інформаційний вхід наступного.

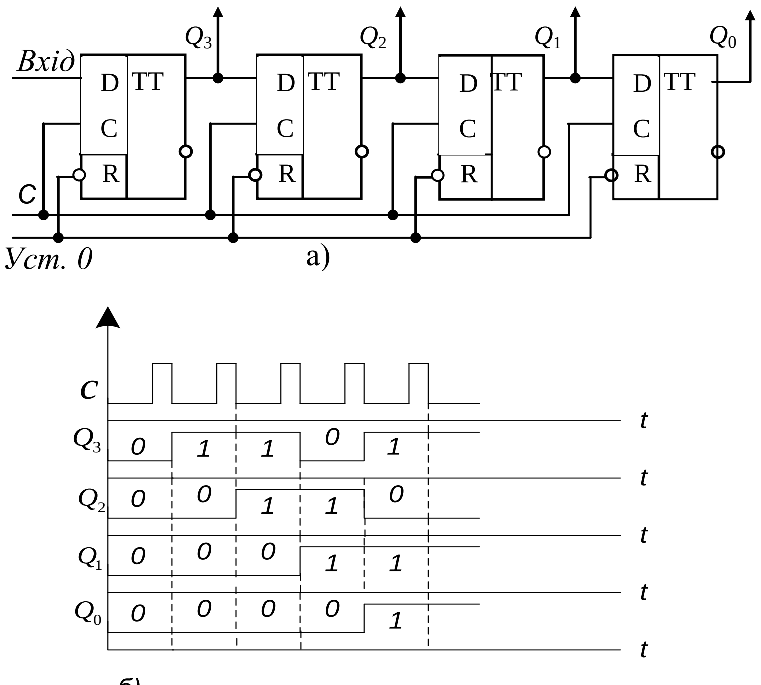


Рисунок 5.44 – Схема чотирирозрядного зсувного регістра (а) і часові діаграми його роботи (б)

Тактові входи всіх тригерів об'єднані. Для побудови зсувних регістрів використовують двотактні тригери, вихідні сигнали яких змінюються лище після закінчення спільного тактового сигналу синхронізації.

Сигналом скиду "Уст. 0" всі розряди регістра скидаються до нульового стану, після чого з допомогою синхронізуючого сигналу розпочинається приймання коду на вхід регістра, яким є вхід *D* тригера Т4. Із кожним тактом синхронізуючого сигналу на цей вхід надходить значення чергового розряду, тому його називають послідовним входом регістра. У кожному такті синхронізуючого імпульсу кожен наступний тригер переходить у стан, в якому раніше перебував попередній, а код зсувається на один розряд праворуч.

Часова діаграма запису в регістр коду 1101 шляхом послідовного зсуву показана на рис. 5.44, б.

Щоб зсунути код числа не лише праворуч, а й ліворуч, необхідно ввести в кожен розряд регістра додаткові схеми (рис. 5.45).

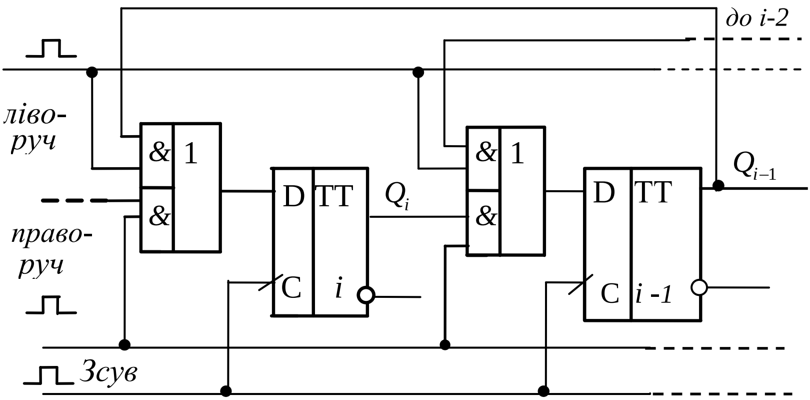


Рисунок 5.45 – Структура реверсивного зсувного регістра

Для зсуву коду слова праворуч одночасно з тактовим сигналом на шині “Зсув” потрібно подати одиничний сигнал на шину “Праворуч”, який відкриє нижню схему І схеми 2І-Або і підімкне вихід Q_i кожного лівого тригера до входу $D_{i\text{-}1}$ кожного правого.

5.4.2 Лічильники

Лічильником називається операційний елемент, що забезпечує зберігання слова інформації та виконання над ним мікрооперацій лічби. Мікрооперація лічби полягає в зміні (збільшенні або зменшенні) вмісту лічильника на одиницю. Крім того, в лічильнику виконуються такі допоміжні мікрооперації, як очищення, зберігання вмісту, видача слова.

У ЦА лічильники використовуються для утворення послідовності адрес команд, для лічби кількості циклів виконання операцій, ділення частоти імпульсів та інших цілей.

Лічильники можна класифікувати на основі таких ознак, як спосіб кодування, модуль лічби, напрямок лічби, спосіб організації міжрозрядних зв'язків, спосіб організації лічби.

За способом кодування розрізняють лічильники з позиційним та непозиційним кодуванням. У лічильниках з позиційним кодуванням числовий вираз i-го поточного стану визначається за формулою

$$S_i = \sum_{k=1}^{n} a_k y_k \ ,$$

де n – кількість розрядів лічильника; a_k – вага k-го розряду; y_k – логічне значення k-го розряду (0 чи 1).

Зразком такого лічильника є лічильник для лічби в двійковій системі числення.

У лічильниках із непозиційним кодуванням розряди не мають сталих ваг, і числового виразу стану лічильника надано кожному набору розрядів. Цей тип лічильників застосовують рідко.

Модуль лічби $M_{ліч}$ – це кількість різних стійких станів лічильника, в які він переходить у процесі одного циклу лічби. Інакше кажучи, це гранична кількість імпульсів, що їх може перелічити лічильник. Наприклад, лічильник, що може лічити від 000 до 111 (у двійковій системі), повинен мати вісім різних вихідних станів – лічильник за модулем 8. За значенням модуля лічби розрізняють *двійкові* лічильники ($M_{ліч} = 2^n$) та з *довільним коефіцієнтом лічби*, в яких модуль лічби не дорівнює цілому степеню числа 2. Останні кодують також двійковими кодами. Наприклад, лічильники з $M_{ліч}$=4, 16 чи 32 – двійкові, а з $M_{ліч}$=5, 10 чи 12 – з довільним коефіцієнтом лічби (двійкокодовані). Лічильник з $M_{ліч}$=10 часто називають декадним, або десятковим.

За напрямом лічби лічильники поділяються на *підсумовувальні*, *віднімальні* й *реверсивні*. У перших надходження на вхід одного імпульсу збільшує вміст лічильника на одиницю, у других – зменшує на одиницю, а в третіх можна реалізувати обидві мікрооперації.

За способом організації лічби розрізняють *асинхронні* й *синхронні* лічильники. В асинхронному лічильнику лічильні імпульси надходять лише на вхід першого тригера і кожен наступний тригер перемикається лише після зміни стану попереднього. Через послідовне перемикання тригерів час установлення в асинхронних лічильниках має значну величину:

$$t_{уст} = t_c + n\tau_T,$$

де t_c – тривалість синхроімпульсу; n – кількість розрядів лічильника; τ_T – час перемикання одного тригера.

У синхронному лічильнику лічильні імпульси надходять одночасно на входи всіх тригерів, і вони перемикаються синхронно з тактовим імпульсом.

Структури двійкових лічильників можна синтезувати формальними методами або одержати евристичним шляхом, тобто визначенням закономірностей змін двійкових чисел при послідовній лічбі.

Розглянемо зростаючу та спадну послідовності двійкових чотирирозрядних чисел.

Під час прямої лічби (підсумовування) сусідній старший розряд змінює стан на протилежний із переходом сусіднього молодшого розряду з 1 на 0. Отже, підсумовувальний лічильник може бути побудований із двотактних лічильних тригерів із управлінням за рівнем, при цьому прямий вихід кожного тригера повинен бути з’єднаний із входом керування кожного наступного тригера (рис. 5.46 а).

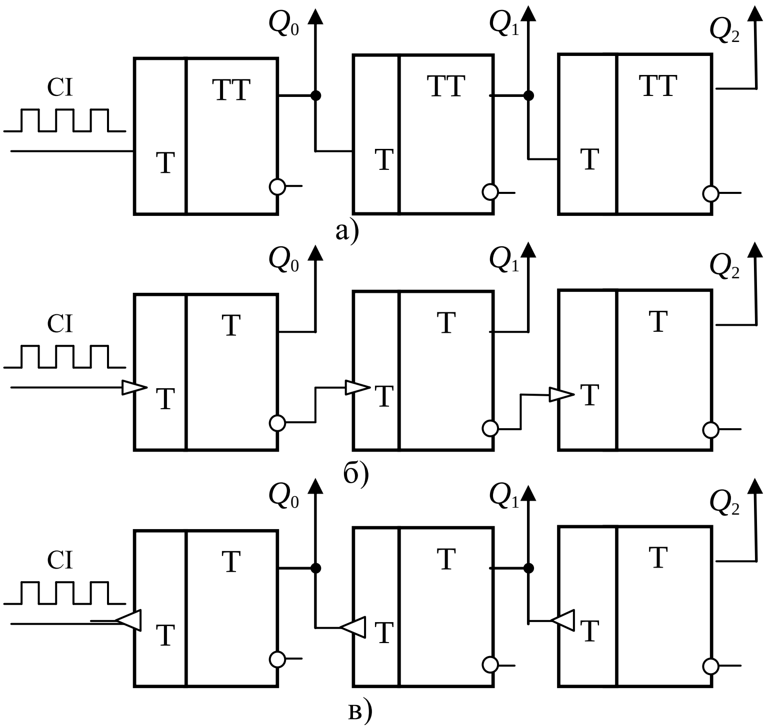


Рисунок 5.46 – Структури підсумовувальних лічильників

Якщо використовувати однотактні тригери з прямим динамічним керуванням, то необхідио інверсний вихід тригера кожного молодшого розряду з’єднати з лічильним входом тригера сусіднього старшого розряду (рис. 5.46 б).

Використовуючи однотактні тригери з інверсним динамічним керуванням, прямий вихід кожного тригера підключають до входу керування наступного тригера (рис. 5.46 в). Під час зворотної лічби (віднімання) наступні розряди змінюють свій стан з 0 на 1. Тому схема віднімального лічильника являє собою ланцюжок із тригерів з прямим динамічним керуванням (рис. 5.47 б). Інші варіанти схеми (рис. 5.47 а, в) одержують на підставі аналізу, аналогічного зробленому раніше для підсумовувального лічильника.

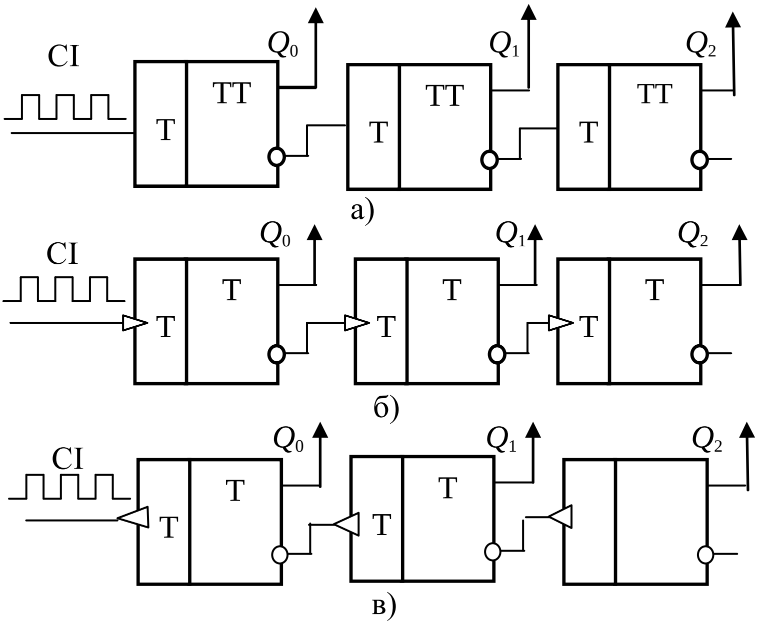


Рисунок 5.47 - Структури віднімальних послідовних лічильників

Оскільки в одержаних структурах лічильників кожен наступний тригер перемикається вихідним сигналом попереднього, то їх називають послідовними. Такі лічильники мають просту схему, але й низьку швидкодію. У послідовному лічильнику час установлення буде максимальним на переходах, які супроводжує перемикання всіх розрядів, наприклад, від 1111 до 0000 у чотирирозрядному підсумовувальному лічильнику.

Схеми двійкових послідовних лічильників будують на базі *RS*-, *D*- або *JK*-тригерів, увімкнених за схемою лічильного тригера (рис. 5.48).

З'єднання інверсного виходу кожного *D*–тригера з лічильним входом наступного тригера у зазначеному лічильнику дає можливість перетворити його на лічильний тригер. Завдяки входам передустановлення $a_0 - a_3$ можна заздалегідь занести в лічильник потрібне число, наприклад адресу чергової команди.

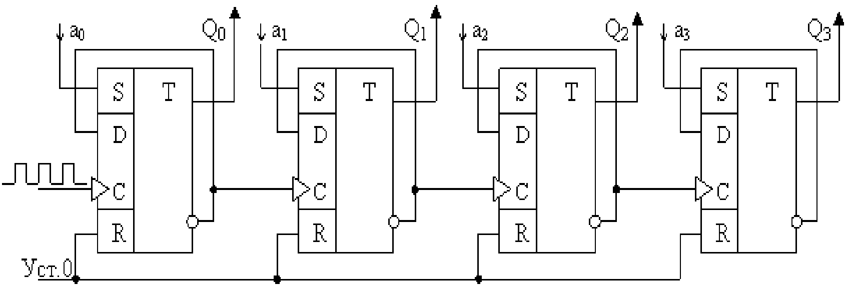


Рисунок 5.48 – Двійковий лічильник за модулем 16

Приклад 5.3. Визначити, який код установиться на виході 5-розрядного двійкового віднімального лічильника після підрахунку ним 25 імпульсів.

Розв'язування

Оскільки початковий стан лічильника відповідає коду 11111 (максимальне число, що відображається в цьому лічильнику), тобто числу 31, то після віднімання від нього числа 25 в ньому запишеться число $31-25=6$, тобто у двійковому вигляді 00110.

Відповідь: установиться код 00110.

На практиці часто потрібні лічильники, коефіцієнт лічби яких відрізняється від цілого степеня двійки, тобто $K_{ліч} \neq 2^n$. Щоб одержати такий коефіцієнт, застосовують вилучення зайвих станів у двійковому лічильнику з модулем лічби $M_{ліч} = 2^n$. Розрядність *n* цього лічильника потрібно вибирати так, щоб виконувалася умова

$$M_{ліч} = 2^n > K_{ліч} > 2^{n-1} \qquad (5.22)$$

Із (5.22) випливає, що коли задано коефіцієнт перерахунку, кількість тригерів лічильника

$$n = \lceil \log_2 K_{ліч} \rceil. \qquad (5.23)$$

Наприклад, $\lceil 4,2 \rceil = \lceil 4,53 \rceil = \lceil 4,97 \rceil = 5$.

Приклад 5.4. Визначити необхідну кількість розрядів в десятковому лічильнику.

Розв'язування

Згідно з виразом (5.23) записуємо:

$$n = \lceil \log_2 10 \rceil = \lceil 3,3 \rceil = 4.$$

Відповідь: для побудови десяткового лічильника потрібно взяти 4 розрядних тригери.

Щоб вилучити зайві стани лічильника з довільним коефіцієнтом лічби, в асинхронних лічильниках використовують найчастіше спосіб його примусового скиду у початковий стан і спосіб занесення кількості зайвих станів у лічильник на початку кожного циклу лічби.

Розглянемо застосування першого способу на прикладі побудови асинхронного десяткового лічильника (рис. 5.49).

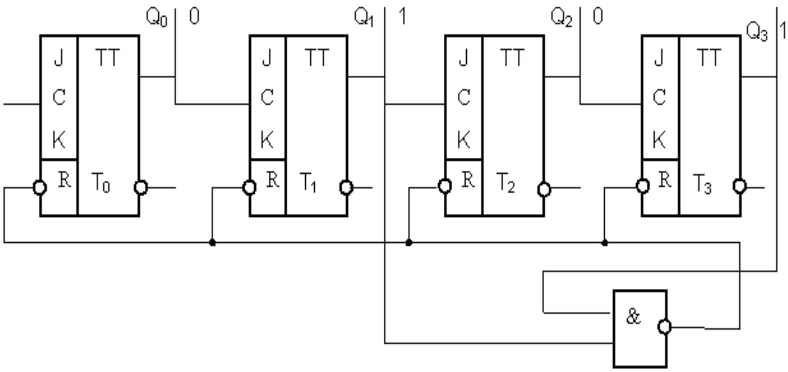


Рисунок 5.49 - Схема асинхронного лічильника з примусовим скидом для $K_{ліч} = 10$

Десятковий лічильник повинен відображати цифри від 0 до 9 (у двійковому вигляді від 0000 до 1001), а десятий імпульс (код 1010, або починаючи з молодшого розряду лічильника, як це зображено на рис. 5.49 – 0101) повинен скидати його до нуля. Схема І-Не на рис. 5.49 при появі на вході двох 1 з розрядів a_1 і a_3 лічильника виробляє на своєму виході 0, який скидає всі розрядні тригери лічильника у нульовий стан, і цикл лічби починається знову.

Приклад 5.5. Синтезувати схему синхронного десяткового лічильника на D-тригерах за допомогою формального методу синтезу ЦА, описаного в розділі 4, п. 7.2.

Розв'язування

Стосовно синтезу лічильника із заданим коефіцієнтом лічби $K_{ліч} = 10$ суть цього методу буде полягати в синтезі для кожного тригера лічильника окремої логічної схеми, яка аналізує в кожному такті лічби стан виходів усіх тригерів і подає на інформаційний вхід тригера такий сигнал, який перемикає даний розрядний тригер у стан, потрібний для відтворення наступного числа. Тому під час синтезу будемо використовувати за вихідні дані, крім заданого коефіцієнта лічби, ще й таблицю переходів лічильника - характеристичну таблицю D-тригера (табл. 5.12). Характеристична таблиця тригера свідчить про те, яке значення повинен мати інформаційний вхід тригера, щоб забезпечити зазначений перехід.

На першому кроці визначимо необхідну кількість тригерів лічильника: $n = \lceil \log_2 K_{ліч} \rceil = \lceil \log_2 10 \rceil = 4$.

На другому кроці складемо таблицю переходів десяткового лічильника (табл. 5.17), користуючись таблицею прямої лічби для чотирирозрядного двійкового числа.

Сумісний аналіз цієї таблиці і характеристичної таблиці D-тригера (табл. 5.11) свідчить, що сигнал на вході D_i відповідатиме значенню виходу тригера $Q_{ін}$ у наступному такті: $D_i = Q_{ін}$, де i – номер розряду лічильника.

Цей факт відбито в табл. 5.18 переходів тригера T_3 (старшого розряду десяткового лічильника) під час підрахунку десяткових чисел.

Такі самі таблиці можна скласти для інформаційних входів D_2, D_1, D_0 решти тригерів лічильника. Це дає підставу розглядати табл. 5.17 як таблицю істинності для логічних функцій D_3, D_2, D_1, D_0, що керують інформаційними входами відповідних тригерів: у лівій її частині записано логічні змінні $Q_{3j,} Q_{2j,} Q_{1j,} Q_{0j,}$ де $j = \overline{0,10}$ – номер рядка, а в стовпцях правої частини – значення логічних функцій, що відповідають цим значенням змінних. Базуючись на цих міркуваннях, можемо скласти формули (5.24) логічних функцій для логічних функцій D_3, D_2, D_1, D_0.

Таблиця 5.17 – Таблиця переходів десяткового лічильника

Десяткове число	Початковий стан				Наступний стан			
	$Q_{3П}$	$Q_{2П}$	$Q_{1П}$	$Q_{0П}$	$Q_{3Н}$	$Q_{2Н}$	$Q_{1Н}$	$Q_{0Н}$
0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	1	0	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	1	0
6	0	1	1	0	0	1	1	1
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	1	0	1	0
10	1	0	1	0	0	0	0	0

Таблиця 5.18 – Переходи тригера T_3

Десяткове число	Стан		Значення D-входу
	$Q_{3П}$	$Q_{3Н}$	D_3
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	1	1
8	1	1	1
9	1	1	1
10	1	0	0

$$\begin{cases} D_{3j} = f_3(Q_{3j}, Q_{2j}, Q_{1j}, Q_{0j}), \\ D_{2j} = f_2(Q_{3j}, Q_{2j}, Q_{1j}, Q_{0j}), \\ D_{1j} = f_1(Q_{3j}, Q_{2j}, Q_{1j}, Q_{0j}), \\ D_{0j} = f_0(Q_{3j}, Q_{2j}, Q_{1j}, Q_{0j}). \end{cases} \qquad (5.24)$$

На третьому кроці виконаємо мінімізацію логічних функцій, що описують закон керування логічними входами розрядних тригерів лічильника. Щоб одержати мінімізовані вирази функцій D_i за табл. 5.17, для кожної з них заповнимо діаграми Вейча – Карно (рис. 5.50).

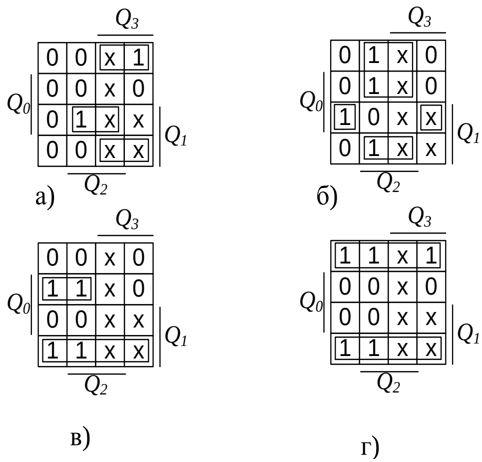


Рисунок 5.50 - Діаграми Вейча – Карно для логічних функцій: а) – D_3 ; б) – D_2; в) – D_1; г) – D_0

Значення функцій у зайвих станах (тобто станах 11–15) зображено знаком невизначеності “x”. Для синтезованої схеми логічного керування ці стани є байдужими, оскільки на її вході вони ніколи не з’являться. З одержаних діаграм можна записати систему логічних рівнянь :

$$\begin{cases} D_3 = Q_2 Q_1 Q_0 + Q_3 \bar{Q}_{0,} \\ D_2 = \bar{Q}_2 Q_1 Q_0 + Q_2 Q_1 Q_0 + Q_2 \bar{Q}_1, \\ D_1 = \bar{Q}_3 \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0, \\ D_0 = \bar{Q}_0. \end{cases} \qquad (5.25)$$

На четвертому кроці, знаючи функції керування інформаційними входами тригерів, можна побудувати схему десяткового синхронного лічильника на D-тригерах.

Контрольні питання

Якими моделями можна описати комбінаційну схему?

З яких етапів складається синтез КС?

Як визначається складність комбінаційної схеми?

У чому полягає аналіз комбінаційної схеми?

Назвіть методи усунення “гонок” в комбінаційних схемах.

Які типові вузли у вигляді комбінаційних схем використовуються в ЦА?

Що таке суматор і які типи суматорів вам відомі?

Наведіть схему півсуматора.

Наведіть схему повного одно розрядного суматора.

Накресліть схему 4-розрядного суматора з паралельним перенесенням.

Що таке дешифратор і де він використовується в ЦА?

Які переваги і недоліки лінійних, каскадних і прямокутних дешифраторів?

Дайте визначення шифратора.

Де використовуються шифратори?

Чим відрізняються мультиплексор і демультиплексор? Де вони використовуються?

Що називають структурним синтезом ЦА?

Назвіть етапи структурного синтезу ЦА.

Яка різниця між структурним і абстрактним автоматами?

Накресліть узагальнену функціональну схему ЦА.

Опишіть послідовність дій під час синтезу схеми станів ЦА.

Назвіть відомі вам елементарні автомати.

Дайте визначення тригера.

Як можна класифікувати тригери.

Запишіть логічне рівняння і таблицю істинності RS-тригера.

Накресліть схему синхронізованого RS-тригера на елементах І-Не.

Накресліть схему двоступеневого RS-тригера і часову діаграму його роботи.

Запишіть логічне рівняння D-тригера і наведіть таблицю істинності його роботи.

Запишіть логічне рівняння Т-тригера і наведіть таблицю істинності його роботи.

Дайте визначення JK-тригера.

Наведіть характеристичні таблиці тригерів.

Що називають регістром?

Які типи регістрів вам відомі?

Накресліть схему 3-розрядного паралельного регістра.

Наведіть приклад схеми послідовного регістра.

Дайте визначення лічильника.

Які типи лічильників розрізняють?

Запишіть формулу визначення кількості тригерів у лічильнику.

Накресліть схему 3-розрядного двійкового лічильника.

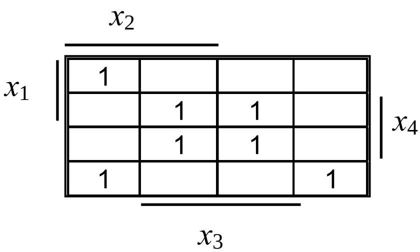
Назвіть методи побудови лічильників із довільним коефіцієнтом лічби.

5.5 Вправи

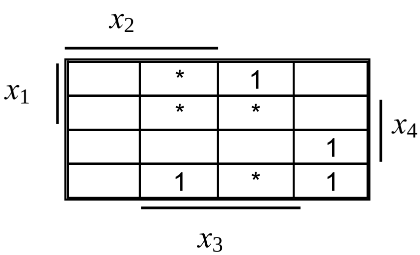
1. Синтезувати схему в базисі “І - Не” для такої логічної функції:

$$F(X) = \overline{(x_0 + x_1 + x_2)(\overline{x_0} + x_1 + x_2)} + x_0 \overline{x_1} x_2 .$$

2. Мінімізувати логічну функцію, задану діаграмою Вейча - Карно, і синтезувати для неї комбінаційну схему в базисі “Або - Не”.



3. Синтезувати в базисі “І - Не” комбінаційну схему для логічної функції, заданої діаграмою Вейча – Карно:



4. Синтезувати схему повного однорозрядного суматора на логічних елементах І - Або - Не.

5. Розробити схему трирозрядного асинхронного підсумовувального лічильника на JK-тригерах і накреслити часову діаграму його роботи.

6. Скласти схему трирозрядного асинхронного віднімального двійкового лічильника на D-тригерах і накреслити часову діаграму його роботи.

7. Синтезувати асинхронний лічильник із коефіцієнтом лічби $K_{\text{ліч}} = 12$ за схемою з примусовим скиданням.

8. Синтезувати синхронний лічильник на D-тригерах із коефіцієнтом лічби $K_{\text{ліч}} = 11$.

9. Скласти схему чотирирозрядного паралельно-послідовного регістра і накреслити часову діаграму його роботи під час зсуву прийнятого коду 1101.

10. Синтезувати трирозрядний регістр зсуву згідно з циклом 0-1-3-5-2-4-0.

11. Синтезувати схему перетворювача двійково-десяткового коду 8-4-2-1 в код із вагами 6-3-1-1.

12. Синтезувати схему шифратора коду “1 з 8” в код 4-2-1.

13. Синтезувати дешифратор коду 4-2-1 в код “1 з 8”.

14. Реалізувати на мультиплексорі логічну функцію

$$F(X) = x_1 x_2 x_3 + \overline{x}_1 x_2 \overline{x}_3 + \overline{x}_1 \overline{x}_2 x_3 + x_1 \overline{x}_2 \overline{x}_3 .$$

15. Синтезувати ЦА, заданого таблично.

Таблиця виходів автомата

$x(t)$ \ $q(t)$	q_0	q_1	q_2	q_3	q_4
x_1	y_3	y_2	y_5	y_6	y_2
x_2	-	y_2	y_6	y_2	y_3
x_3	y_6	-	y_6	y_1	-
x_4	y_2	-	y_4	y_3	y_5

Таблиця переходів автомата

$x(t)$ \ $q(t)$	q_0	q_1	q_2	q_3	q_4
x_1	q_3	q_4	q_1	q_0	q_2
x_2	q_0		q_1	q_2	q_2
x_3	q_3	q_1	q_4	q_1	q_4
x_4	q_2	q_0	q_2	q_2	q_1

ПРЕДМЕТНИЙ ПОКАЖЧИК

Д

Е

М

Н

О

Список рекомендованої літератури

1. Бондаренко М. Ф. Комп’ютерна дискретна математика : підручник / М. Ф. Бондаренко, Н. В. Білоус, А. Г. Руткас. – Харків : Компанія СМІТ, 2004. – 480 с.

2. Нікольський Ю. Т. Дискретна математика : підручник / Ю. Т. Нікольський, В. А. Пасічник, Ю. Р. Щербина. – Львів : Магнолія плюс, 2006. – 608 с.

3. Борисенко О. А. Лекції з дискретної математики (множини і логіка) : навч. посібник / О. А. Борисенко. – 3-тє вид., випр. і допов. – Суми : ВТД «Університетська книга», 2002. – 180 с.

4. Теорія цифрових автоматів та формальних мов. Вступний курс : навч. посібник / С. Ю. Гавриленко, А. М. Клименко, Н. Ю. Любченко та ін. – Харків : НТУ ХПІ, 2011. – 176 с.

5. Прикладна теорія цифрових автоматів : навч. посібник / І. В. Жабін, І. А. Жуков, І. А. Клименко, В. В. Ткаченко. – К. : Книжкове вид-во НАУ, 2007. – 364 с.

6. Карпов Ю. Г. Теория автоматов : учебник для ВУЗов / Ю. Г. Карпов. – СПб. : Питер, 2003. – 208 с.

7. Бабич М. П. Комп´ютерна схемотехніка / М. П. Бабич, І. А. Жуков. – Київ : МК-Пресс, 2004. – 478 с.

8. Зайцев Е. И. Прикладная теория цифровых автоматов : учебное пособие / Е. И. Зайцев. – М. : МГУПИ, 2007. – 76 с.

9. Пєтух А. М. Прикладна теорія цифрових автоматів / А. М. Пєтух, В. В. Войтко. – Вінниця : ВДТУ, 2003. – 67 с.

Додаток А
(довідковий)
Формули комбінаторики

А1. Перестановки

Упорядкована вибірка обсягом n із n елементів, в якій всі елементи різні, називається перестановкою із n елементів. Число всіх можливих перестановок із n елементів

$$P = 1 \cdot 2 \cdots n = n!$$

А2. Розміщення

Упорядкована вибырка обсягом m із n елементів, в якій всі елементи різні, називається розміщенням. Число розміщень із n елементів за m

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

Упорядкована вибірка обсягом m із n елементів, в якій деякі елементи можуть повторюватися, називається розміщенням із повтореннями. Число розміщень із n елементів за m із повтореннями

$$A_n^m(n) = n^m.$$

А3. Сполучення

Неупорядкована вибірка обсягом m із n елементів, $m < n$, в якій всі елементи різні, називається сполученням. Число сполучень із n елементів до m

$$C_n^m = \binom{m}{n} = \frac{m!}{m!(n-m)!},$$

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

Числа C_n^m називаються биноміальними коефіцієнтами.

Неупорядкована вибірка обсягом m із n елементів, $m < n$, в якій деякі елементи можуть повторюватись, називається сполученням із повтореннями. Число сполучень із n елементів до m із повтореннями

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

Додаток Б

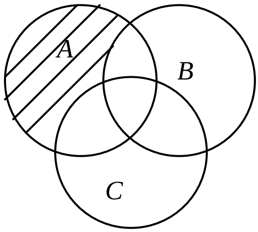
(обов'язковий)

Контрольна робота

Наводиться зразковий варіант набору типових задач контрольної роботи за матеріалами навчального посібника з еталонами викладення розв'язувань.

Задача 1

За діаграмою Венна виразити через операції алгебри множин заштриховану множину.



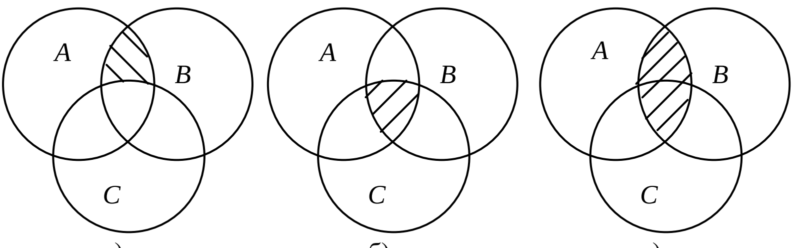
Відповідь: заштрихована множина $D = A \setminus (B \cup C)$.

Задача 2

Довести за допомогою діаграм Венна тотожність

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

Розв'язування



а) б) в)

$\overline{C} \cap A \cap B = D$ – заштрихована множина на рис. а);

$A \cap B \cap C = E$ – заштрихована множина на рис. б);

$D \cup E$ – заштрихована множина на рис. в).

Очевидно, $D \cup E = A \cap B$.

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

Задача 3

Нехай A ={1, 2, 3, 4}, B ={5, 6, 7, 8}, C ={11, 22, 33}. Відповідності $F \subseteq A \times B$ і $G \subseteq B \times C$ задані так:
F ={(1, 6), (1, 8), (2, 5), (3, 6)}, G ={(5, 11), (6, 11), (7, 22), (8, 23)}.

1. Задати відповідності F і G матрицями та стрілковими діаграмами.

2. Дослідити відповідності F і G на рефлексівність, антирефлексивність, симетричність, антисиметричність, транзитивність.

3. Визначити відповідності:

F^{-1} і G^{-1}, $G \circ F$, $G \circ G^{-1}$, $G^{-1} \circ G$, $F^{-1} \circ G^{-1}$.

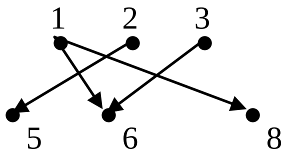
Розв'язування.

1. Задання відповідності $F \subset A \times B$:

матрицею

(A,B,F)	5	6	8
1	0	1	1
2	1	0	0
3	0	1	0

стрілковою діаграмою

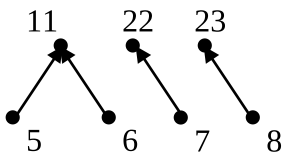


Задання відповідності $G \subset B \times C$:

матрицею

(B,C,G)	11	22	23
5	1	0	0
6	1	0	0
7	0	1	0

стрілковою діаграмою



2. Множини $A=\{1, 2, 3, 4\}$, $B=\{5, 6, 7, 8\}$ і $C=\{11, 22, 33\}$ є підмножинами множини N натуральних чисел. Тому множини $F=\{(1, 6), (1, 8), (2, 5), (3, 6)\}$ і $G=\{(5, 11), (6, 11), (7, 22), (8, 23)\}$ є відповідностями на множині N: $F\subset N^2$, $G\subset N^2$, а це означає, що вони можуть володіти властивостями рефлексивності, антирефлексивності, симетричності, антисиметричності, транзитивності. Перевіримо наявність перелічених властивостей у відповідностях F і G.

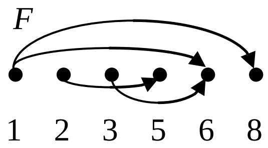
Відношення F і G не рефлексивні, тому що вони не містять, відповідно, пар виду (a, b), $a, b\in N$, де $b=a$, для всіх $a\in A$ і пар виду (b, c), $b, c\in N$, де $c=b$, для всіх $b\in B$.

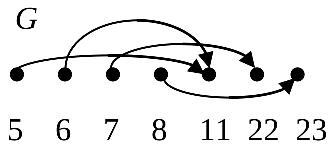
Відповідності F і G антирефлексивні, оскільки вони не містять жодної пари виду (a, a), $a\in N$ однакових елементів.

Умовою симетричності відповідності Q на множині (у нашому випадку на множині N) є наявність в ній сполучень пар $(b, a)\in Q$, де $a\neq b$, обернених за впорядкуванням до кожної пари $(a,b)\in Q$. Оскільки у кожній із відповідностей F і G немає двох пар різних елементів з оберненим упорядкуванням (наприклад, ((1, 6) і (6, 1), (5, 11) і (11, 5)), то вони не є симетричними.

Умовою антисиметричності відповідності Q на множині є відсутність у ній сполучень двох пар $(a, b)\in Q$ і $(b, a)\in Q$, де $a\neq b$, з можливою наявністю пар виду $(a, b)\in Q$, де $a=b$. Відповідності F і G задовольняють цю умову, тому вони є антисиметричними.

Відповідність F не транзитивна, оскільки множина $F=\{(1, 6), (1, 8), (2, 5), (3, 6)\}$ не містить сполучень трійок пар (a, b), (b, c) і (a, c), де $a\neq b\neq c$, для всіх a, b і c, що беруть участь у відношенні F. Таких трійок упорядкованих пар не містить і відповідність $G=\{(5, 11), (6, 11), (7, 22), (8, 23)\}$, тому вона також не є транзитивною. Нетранзитивність F і G як відношень на N ілюструють їх графи:





Як бачимо, на цих графах немає сполучень двох стрілок одного спрямування з примиканням кінця однієї до початку іншої, чим підтверджується нетранзитивність F і G.

3) Відповідності F^{-1} і G^{-1}, обернені до відповідностей F і G, мають вигляд:

$$F^{-1} = \{(6, 1), (8, 1), (5, 2), (6, 3)\},$$
$$G^{-1} = \{(11, 5), (11, 6), (22, 7), (23, 8)\}.$$

Композиції $G \circ F$, $G \circ G^{-1}$, $G^{-1} \circ G$, $F^{-1} \circ G^{-1}$ мають вигляд:

$G \circ F = \{(1, 11), (1, 23), (2, 11)\}$;

$G \circ G^{-1} = \{(11,11), (22,22), (23,23)\}$;

$G^{-1} \circ G = \{(5, 5), (5, 6), (6, 5), (6, 6), (7, 7), (8, 8)\}$;

$F^{-1} \circ G^{-1} = \{(11, 2), (11, 1), (11, 3), (23, 1)\}$.

Задача 4

Які з функцій f: $X \to Y$ на множині R дійсних чисел є сюр'єктивними, ін'єктивними, бієктивними, взаємно однозначними, якщо $x \in [-3, 3]$, $y \in [-27, 27]$:

а) $f_1(x, y) = x^3$; б) $f_2(x, y) = |\sin x|$; в) $f_3(x, y) = 3^x$; г) $f_4(x, y) = 10$?

Розв'язування

Функції $f_1 - f_4$ є повними та однозначними, тобто функціональними відображенням множини $X = \{x$: $x \in [-3, 3] \subset R\}$ у множину $Y = \{y$: $y \in [-27, 27] \subset R\}$, тому що будь-якому $x \in X$ відповідає єдине значення $y \in Y$. Множини X і Y є відповідно областями визначення та значень цих функцій.

Функція $f_1(x, y) = x^3$ є:

- сюр'єктивною, тому що для кожного $y \in [-27, 27]$ існує елемент $x = y^{1/3} \in [-3, 3]$ такий, що має місце $y = x^3$;

- ін’єктивною, тому що функція $f_1(x, y) = x^3$ монотонна, у зв’язку з чим різним елементам $y \in Y$, які беруть участь у відображенні (у даному випадку всім $y \in Y$) відповідають різні елементи $x \in X$;

- бієктивною, тому що вона одночасно є сюр’єктивною та ін’єктивною;

- взаємно однозначною, тому що вона є бієктивною, внаслідок чого обернена функція $x = y^{1/3}$ також є однозначною.

Функція $f_2(x, y) = |\sin x|$:

- не сюр’єктивна, тому що частина елементів області значень , а саме – $y \in [-27, 0)$ та $y \in (1, 27]$ не беруть участі у функціональному відображенні;

- не ін’єктивна, тому що кожному образу $y \in (0, 1]$ відповідають не один, а два прообрази із підобластей визначення, а саме – $x \in [-3, 0)$ та $x \in (0, 3]$;

- не бієктивна, тому що вона не є одночасно сюр’єктивною та ін’єктивною;

- не є взаємно однозначною, тому що вона не бієктивна.

Функція $f_3(x, y) = 3^x$:

- не сюр’єктивна, тому що частина елементів області значень, а саме – $y \in [-27, 1/27)$ не беруть участі у функціональному відображенні ($3^{-3} = 1/27$);

- ін’єктивна, тому що кожному образу $y \in [1/27, 27]$ відповідає єдиний прообраз із області визначення $x \in [-3, 3]$, внаслідок монотонності цієї функції;

- не бієктивна, тому що вона не є одночасно сюр’єктивною та ін’єктивною;

- не є взаємно однозначною, тому що вона не бієктивна.

Функція $f_4(x, y) = 10$ – це функція-константа. Участь у відображенні бере лише один образ $y = 10$, $10 \in Y$ нескінченної множини прообразів $x \in X$. Тому функція $f_4(x, y)$ ані сюр’єктивна, ані ін’єктивна. Отже, вона не бієктивна і не взаємно однозначна.

Задача 5

1. Скласти таблицю істинності для логічної функції $f(x, y) = (x\overline{y} + \overline{x}y)\,x + xy$.

Відповідь:

x	y	$x\overline{y}$	$\overline{x}y$	$(x\overline{y} + \overline{x}y)\,x$	xy	$f(x, y)$
0	0	0	0	0	0	0
0	1	0	1	0	0	0
1	0	1	0	1	0	1
1	1	0	0	0	1	1

2. На підставі таблиці істинності записати логічну функцію.

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

Відповідь: $f(x, y) = x\overline{y} + xy = x(\overline{y} + y) = x$.

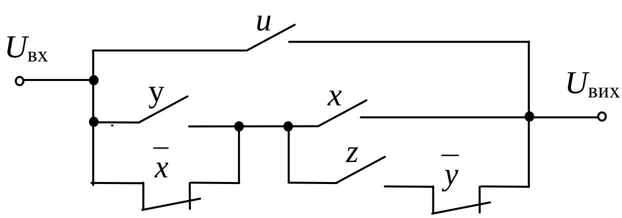
Задача 6

1. Скласти контактну схему, яка технічно реалізує задану логічну функцію $f = (\overline{x} \vee y) \wedge (x \vee z \wedge \overline{y}) \vee u$.

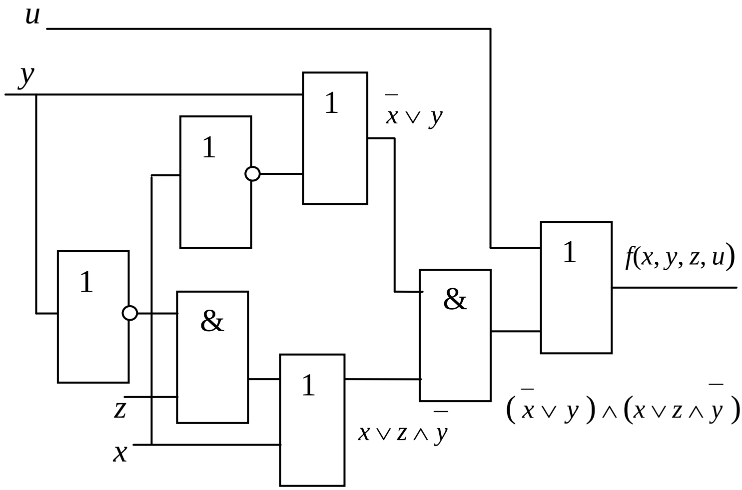
2. Замінити контактну схему комбінаційною схемою на логічних елементах у базисі І-Або-Не.

Розв'язування

1. Контактна схема:



2. Логічна схема:



Задача 7

1. Скласти контактну схему за заданими умовами її роботи (умови роботи зазначені наборами, на яких логічна функція, яку реалізує контактна схема, має значення 1):

$$f(111) = f(110) = f(001) = 1.$$

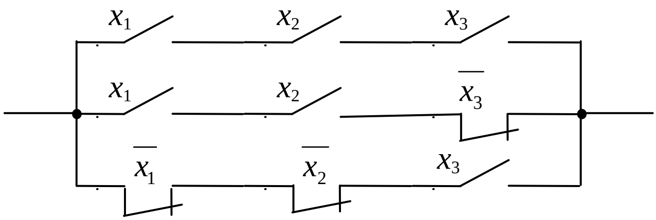
2. Замінити контактну схему комбінаційною схемою на логічних елементах в базисі Або – Не або І – Не.

Розв'язування

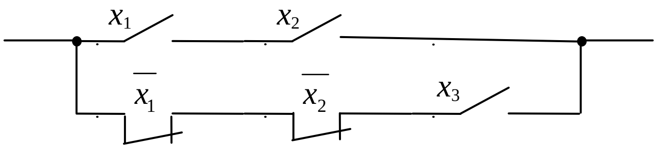
1. На підставі заданих умов роботи комбінаційної схеми складемо функцію провідності:

$$F = x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3. \qquad (1)$$

Функція провідності (1) реалізується контактною схемою, яка являє собою паралельне з'єднання трьох гілок:



Оскільки завжди замкнений один з контактів x_3 і $\overline{x_3}$ у двох верхніх гілках, то схему можемо спростити, замінивши ци гілки однією з умовою її роботи $f_{12}(11) = 1$:



Спрощеній контактній схемі відповідає функція провідності

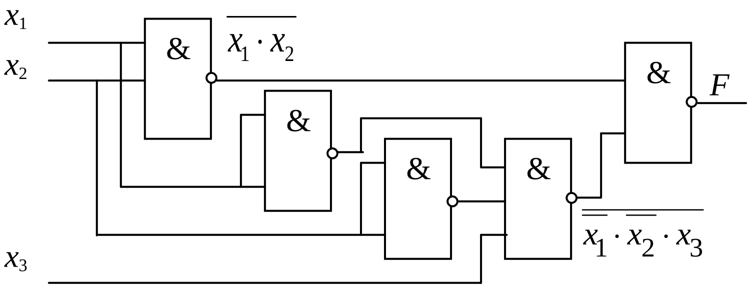
$$F = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2} \cdot x_3. \qquad (2)$$

Зазначимо, що функцію (2) можна також одержати із (1) шляхом здійснення операції повного склеювання двох перших доданків за змінною x_3.

2. Логічні елементи в базисах І – Не та Або – Не виконують логічні операції за законами де Моргана. На виході логічної схеми повинне формуватися значення функції (2) за законом де Моргана:

$$x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2} \cdot x_3 = \overline{(\overline{x_1 x_2})(\overline{\overline{x_1}\,\overline{x_2} x_3})}\,. \qquad (3)$$

Формула правої частини (3) свідчить про те, що контактну схему, яка реалізує функцію провідності (2), доцільно замінити логічною схемою в базисі І – Не:



Задача 8

За заданими умовами роботи комбінаційної схеми

$f(000) = f(010) = f(011) = f(100) = f(101) = 1$

визначити логічну функцію, яку вона реалізує, в ДДНФ та ДКНФ.

Розв'язування

За умовами роботи КС

$$f(000)_0 = f(010)_2 = f(011)_3 = f(100)_4 = f(101)_5 = 1,$$

де значення нижніх індексів дорівнюють десятковим номерам наборів, на яких логічна функція набуває значення 1, складемо її формулу в ДДНФ:

$$f_{ДДНФ}(x_1, x_2, x_3) = (\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}) + (\overline{x_1} \cdot x_2 \cdot \overline{x_3}) + (\overline{x_1} \cdot x_2 \cdot x_3) +$$
$$+ (x_1 \cdot \overline{x_2} \cdot \overline{x_3}) + (x_1 \cdot \overline{x_2} \cdot x_3).$$

Повна кількість наборів функції f дорівнює $2^3 = 8$. Визначимо умови роботи на наборах 1, 6, 7:

$$f(001)_1 = f(110)_6 = f(111)_7 = 1.$$

За цими умовами роботи складемо логічну функцію в ДКНФ:

$$f_{ДКНФ}(x_1, x_2, x_3) = (x_1 + x_2 + \overline{x_3}) \cdot (\overline{x_1} + \overline{x_2} + x_3) \cdot (x_1 + x_2 + x_3).$$

Задача 9

Логічну функцію $f(x, y, z) = (\overline{x} + y\cdot z)\cdot(x + \overline{y}\cdot z) + x\cdot z$ подати в ДДНФ і ДКНФ.

Розв'язування

Формулу будь-якої логічної функції можна перетворити в ДНФ, користуючись правилами перетворення формул та законами алгебри логіки, а потім перетворити її в ДДНФ шляхом розгортання елементарних кон'юнкцій та видалення дублюючих членів.

Перетворимо формулу функції f в ДНФ:

$$f = (\overline{x} + y\cdot z)\cdot(x + \overline{y}\cdot z) + x\cdot u = \overline{x}\cdot x + \overline{x}\cdot\overline{y}\cdot z + x\cdot y\cdot z + y\cdot\overline{y}\cdot z\cdot z + x\cdot z =$$
$$= \overline{x}\cdot\overline{y}\cdot z + x\cdot y\cdot z + x\cdot z = \overline{x}\cdot\overline{y}\cdot z + x\cdot z\cdot(y+1) = \overline{x}\cdot\overline{y}\cdot z + x\cdot z.$$

Отриману ДНФ перетворимо у ДДНФ:

$$f_{ДДНФ} = \overline{x}\cdot\overline{y}\cdot z + x\cdot z = \overline{x}\cdot\overline{y}\cdot z + x\cdot z\cdot(y + \overline{y}) = \overline{x}\cdot\overline{y}\cdot z + x\cdot y\cdot z + x\cdot\overline{y}\cdot z.$$

Номери наборів, на яких $f_{ДДНФ}=1$: $(001)_1$; $(101)_5$; $(111)_7$.

Повна кількість наборів функції f дорівнює $2^3 = 8$. На наборах за номерами $(000)_0$, $(010)_2$, $(011)_3$, $(100)_4$, $(110)_6$ маємо $f = 0$. Отже, ДКНФ заданої логічної функції має вигляд

$$f_{ДКНФ} = (x + y + z)\cdot(x + \overline{y} + z)\cdot(x + \overline{y} + \overline{z})\cdot(\overline{x} + y + z)\cdot(\overline{x} + \overline{y} + z).$$

Задача 10

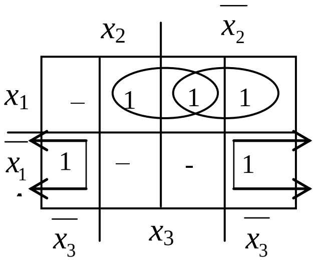
Мінімізувати в ДНФ логічну функцію

$$f(x_1, x_2, x_3) = \overline{x_1}\,\overline{x_2}\,\overline{x_3} + x_1\overline{x_2}\,\overline{x_3} + x_1\overline{x_2}\,x_3 + \overline{x_1}\,x_2\overline{x_3} + x_1 x_2 x_3.$$

Розв'язування

Функція задана в ДДНФ. Побудуємо карту Карно.

Варіант1



Отримано 3 контури з 2 клітинок (в наборах контурів виділені спільні значення змінних):

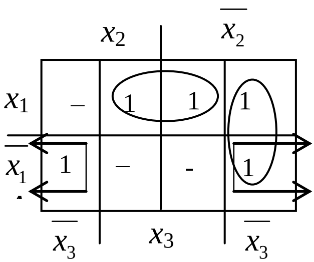
1) **111**   2) **10**1   3) **0**1**0**

   10**1**   **10**0   **0**0**0**

Цим контурам відповідають конституенти одиниці, диз'юнктивне сполучення яких являє собою тупикову ДНФ:

$$f_{\text{ДНФ}}^{туп}(x_1, x_2, x_3) = x_1 \cdot x_3 + x_1 \cdot \overline{x_2} + \overline{x_1} \cdot \overline{x_3}\,. \quad (1)$$

Варіант 2



Отримано 3 контури з 2 клітинок :

1) **111**   2) 1**00**   3) **0**1**0**

   10**1**   0**00**   **0**0**0**

Цим контурам відповідають конституенти одиниці, диз'юнктивне сполучення яких являє собою тупикову ДНФ:

$$f_{\text{ДНФ}}^{\,туп}(x_1, x_2, x_3) = x_1\cdot x_3 + \overline{x_2}\cdot\overline{x_3} + \overline{x_1}\cdot\overline{x_3}\,. \qquad (2)$$

За мінімальну ДНФ слід прийняти функцію (1) з кількістю символів на 1 менше, ніж у функції (2).

Задача 11

Мінімізувати в КНФ логічну функцію

$$f(x_1, x_2, x_3, x_4) = \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,x_4 + x_1\,\overline{x_2}\,\overline{x_3}\,x_4 + x_1\,x_2\,x_3\,\overline{x_4} + x_1\,\overline{x_2}\,x_3\,x_4 +$$

$$+ x_1\,x_2\,x_3\,x_4 + \overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4}\,.$$

Розв’язування

Функція задана в ДДНФ. Побудуємо карту Карно.

Варіант 1

$x_1x_2 \backslash x_3x_4$	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	0	0	1	1
10	0	1	1	0

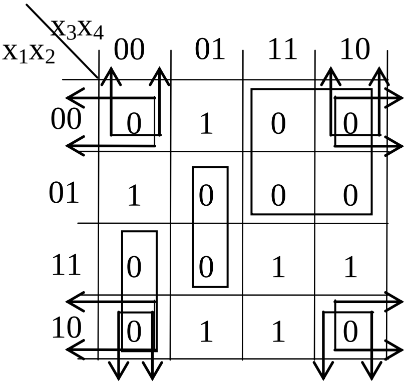
Отримано 2 контури з 4 клітинок і 2 контури з 2 клітинок (у наборах контурів виділені спільні значення змінних):

1) 0**0**0**0** 2) **0**0**1**1 3) **01**0**1** 4) **110**0
 0**0**1**0** **0**0**1**0 **01**1**1** **110**1
 1**0**0**0** **0**1**1**1
 1**0**1**0** **0**1**1**0

Цим контурам відповідають конституенти нуля, кон'юнктивне сполучення яких являє собою тупикову КНФ:

$$f_{\text{КНФ}}^{\,myn}(x_1, x_2, x_3, x_4) = (x_2 + x_4)\cdot(x_1 + \overline{x_3})\cdot(x_1 + \overline{x_2} + \overline{x_4})\cdot(\overline{x_1} + \overline{x_2} + x_3). \quad (1)$$

Варіант 2



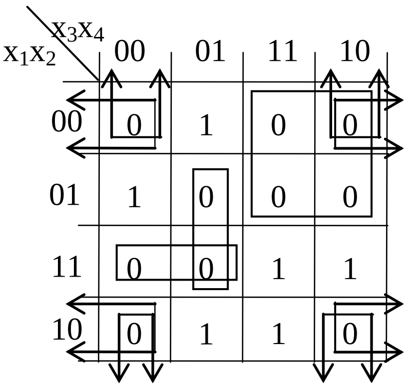
Отримано 2 контури з 4 клітинок і 2 контури з 2 клітинок (у наборах контурів виділені спільні значення змінних):

1) 0**0**0**0**	2) **0**0**1**1	3) **1**1**00**	4) 0**101**
0**0**1**0**	**0**0**1**0	**1**0**00**	1**101**
1**0**0**0**	**0**1**1**1		
1**0**1**0**	**0**1**1**0		

Цим контурам відповідають конституенти нуля, кон'юнктивне сполучення яких являє собою тупикову КНФ:

$$f_{\text{КНФ}}^{\,myn}(x_1, x_2, x_3, x_4) = (x_2 + x_4)\cdot(x_1 + \overline{x_3})\cdot(\overline{x_1} + x_3 + x_4)\cdot(\overline{x_2} + x_3 + \overline{x_4}). \quad (2)$$

Варіант 3



Отримані 2 контури з 4 клітинок і 2 контури з 2х клітинок (у наборах контурів виділені спільні значення змінних):

1) 0**0**0**0**   2) **0**0**1**1   3) **110**0   4) 0**101**
0**0**1**0**   **0**0**1**0   **110**1   1**101**
1**0**0**0**   **0**1**1**1
1**0**1**0**   **0**1**1**0

Цим контурам відповідають конституенти нуля, кон’юнктивне сполучення яких являє собою тупикову КНФ:

$$f_{\text{КНФ}}^{\,\text{туп}}(x_1, x_2, x_3, x_4) = (x_2+x_4)\cdot(x_1+\overline{x_3})\cdot(\overline{x_1}+\overline{x_2}+x_3)\cdot(\overline{x_2}+x_3+\overline{x_4}). \quad (3)$$

За мінімальну КНФ слід прийняти функцію (2) з кількістю символів на 1 менше ніж у формулах функцій (1) і (3).

Задача 12

Граф (неорієнтований) $G(V, E)$ заданий множинами вершин $V=\{v_1, v_2, v_3, v_4, v_5\}$ та ребер $E=\{e_{11}, e_{12}, e_{13}, e_{14}, e_{23}, e_{24}, e_{34}, e_{35}, e_{45}\}$, де $e_{ij}=(v_i, v_j)$ – ребро, що з’єднує вершини v_i та v_j, $i=\overline{1,5}$, $j=\overline{1,5}$. Ребра графа мають вагові параметри: $p(e_{11})$, $p(e_{12})$, $p(e_{13})$,

$p(e_{14})$, $p(e_{23})$, $p(e_{24})$, $p(e_{34})$, $p(e_{35})$, $p(e_{34})$, $p(e_{45})$, що наведені в таблиці:

e_{ij}	e_{11}	e_{12}	e_{13}	e_{14}	e_{23}	e_{24}	e_{34}	e_{35}	e_{45}
$p(e_{ij})$	2	5	7	3	8	4	7	9	11

Потрібно:

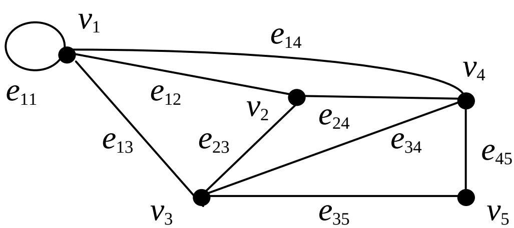
1) побудувати діаграму графа;

2) скласти матриці суміжностей, інциденцій та ваг ребер;

3) визначити: всі прості шляхи $v_1 \to v_5$ між вершинами v_1 і v_5, вагу кожного з них, найкоротший шлях $v_1 \to v_5$, шлях найменшої ваги.

Розв'язування

1. Діаграма графа:



2. Матриця суміжностей

$$A=\begin{Vmatrix}1 & 1 & 1 & 1 & 0\\ 1 & 0 & 1 & 1 & 0\\ 1 & 1 & 0 & 1 & 1\\ 1 & 1 & 1 & 0 & 1\\ 0 & 0 & 1 & 1 & 0\end{Vmatrix}$$

Матриця інциденцій

$$\begin{array}{cc} & \begin{matrix}e_{11} & e_{12} & e_{13} & e_{14} & e_{23} & e_{24} & e_{34} & e_{35} & e_{45}\end{matrix}\\ B= & \begin{Vmatrix}2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0\\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1\end{Vmatrix}\end{array}$$

Оскільки граф не є простим (містить петлю), то матрицю ваг складемо за структурою матриці інциденцій:

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} e_{11} & e_{12} & e_{13} & e_{14} & e_{23} & e_{24} & e_{34} & e_{35} & e_{45} \end{array} \\
P= & \left\|\begin{array}{ccccccccc}
2 & 5 & 7 & 3 & \infty & \infty & \infty & \infty & \infty \\
\infty & 5 & \infty & \infty & 8 & 4 & \infty & \infty & \infty \\
\infty & \infty & 8 & \infty & 4 & \infty & 7 & 9 & \infty \\
\infty & \infty & \infty & 3 & \infty & 4 & 7 & 0 & 11 \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & 9 & 11
\end{array}\right\|
\end{array}
$$

Вибір значень ∞ ваг неіснуючих ребер пов’язаний із тим, що надалі вирішується завдання пошуку шляху з найменшою вагою.

3. Прості шляхи $v_1 \to v_5$ та їх ваги:

$v_1 \to v_4 \to v_5$, P =3+11=14;

$v_1 \to v_4 \to v_2 \to v_3 \to v_5$, P =3+4+8+9 =24;

$v_1 \to v_4 \to v_3 \to v_5$; P =3+7+9 =19;

$v_1 \to v_2 \to v_4 \to v_5$; P =5+4+11 =20;

$v_1 \to v_2 \to v_3 \to v_5$; P =5+8+9 =22;

$v_1 \to v_2 \to v_3 \to v_4 \to v_5$; P =5+8+7 =20;

$v_1 \to v_3 \to v_5$; P=7+9 =16;

$v_1 \to v_3 \to v_4 \to v_5$; P =7+7+11=25;

$v_1 \to v_3 \to v_2 \to v_4 \to v_5$; P =7+8+4+11=30.

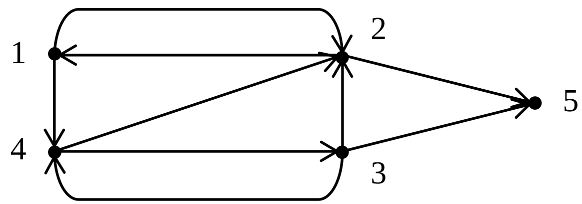
Найкоротшими є 2 шляхи: $v_1 \to v_4 \to v_5$, $v_1 \to v_3 \to v_5$, які містять по 2 ребра. Довжина цих шляхів $d(v_1, v_5)$=2/

Найменшу вагу має шлях $v_1 \to v_4 \to v_5$, його вага P =3+11=14.

Задача 13

Для зваженого орграфа, наведеного на рисунку, скласти матриці суміжностей, інциденцій та ваг. Знайти простий оршлях з найменшою вагою від вершини 1 до вершини 5. Скласти матрицю досяжності. Ваги дуг задані в таблиці:

Дуга	e_{12}	e_{21}	e_{32}	e_{14}	e_{42}	e_{43}	e_{34}	e_{25}	e_{35}
Номер дуги	1	2	3	4	5	6	7	8	9
Вага дуги	2	4	2	6	3	5	3	8	7



Розв'язування

Матриця суміжностей

$$A=\begin{Vmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{Vmatrix}$$

Матриця інциденцій

$$B=\begin{Vmatrix} -1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{Vmatrix}$$

Заданий орграф не є простим, тому матрицю ваг слід задати за структурою матриці інциденцій, замінівши одиниці значеннями ваг відповідних дуг:

$$C=\begin{Vmatrix} -2 & 4 & 0 & -6 & 0 & 0 & 0 & 0 & 0 \\ 2 & -4 & 2 & 0 & 3 & 0 & 0 & -4 & 0 \\ 0 & 0 & -2 & 0 & 0 & 5 & -3 & 0 & -3 \\ 0 & 0 & 0 & 6 & -3 & -5 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 3 \end{Vmatrix}$$

Із вершини 1 у вершину 5 існують 4 прості оршляхи:

1, L_{12}, 2, L_{25}, 5 вагою $P_1 = 2+8 = 10$;
1, L_{14}, 4, L_{43}, 3, L_{35}, 5 вагою $P_2 = 6+5+7 = 19$;
1, L_{14}, 4, L_{43}, 3, L_{32}, 2, L_{25}, 5 вагою $P_3 = 6+5+2+8 = 21$;
1, L_{14}, 4, L_{42}, 2, L_{25}, 5 вагою $P_4 = 6+5+2+8 = 21$.

Найкоротшим є оршлях 1, L_{12}, 2, L_{25}, 5 вагою $P_1 = 10$.

Матриця досяжності

$$R=\begin{Vmatrix}1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 0 & 0 & 0 & 0 & 1\end{Vmatrix}$$

Із матриці досяжності випливає, що вершини 1 – 4 досяжні одна з іншої, а вершина 5 є стоком, тому вершини 1 – 4 із неї не досяжні.

Задача 14

Автомат Мілі заданий таблицями переходів і виходів. Подати автомат абстрактною моделлю, суміщеною таблицею та графом.

Таблиця переходів δ

	q_1	q_2	q_3	q_4
z_1	q_2	q_1	q_2	q_3
z_2	q_3	q_4	q_1	q_1

Таблиця виходів λ

	q_1	q_2	q_3	q_4
z_1	w_1	w_2	w_1	w_2
z_2	w_2	w_3	w_4	w_5

Розв'язування

Із таблиць автомата бачимо, що алфавіт вхідних сигналів $Z=(z_1, z_2)$, множина станів $Q=(q_1, q_2, q_3, q_4)$, алфавіт вихідних сигналів $W=(w_1, w_2, w_3, w_4, w_5)$. Функція переходів δ задана таблицею переходів, функція виходів λ задана таблицею виходів. Для ініціального автомата початковим станом можна вважати q_1, тоді абстрактною моделлю автомата є кортеж

$$A=(Z,\ W,\ Q,\ \delta,\ \lambda,\ q_1).$$

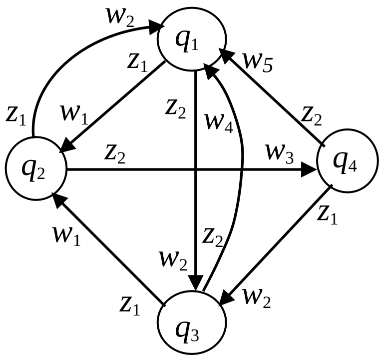
Подамо автомат Мілі однією суміщеною таблицею переходів і виходів, в якій на перетині стовпця q_m і рядка z_f запишемо елемент q_s/w_g, який визначається таким чином:

$$q_s=\delta(q_m,\ z_f);\quad w_g=\lambda(q_m,\ z_f).$$

Суміщена таблиця переходів-виходів:

	q_1	q_2	q_3	q_4
z_1	q_2/w_1	q_1/w_2	q_2/w_1	q_3/w_2
z_2	q_3/w_2	q_4/w_3	q_1/w_4	q_1/w_5

Графічний спосіб задання автомата Мілі передбачає його подання у вигляді орграфа. Вершини графа відповідають станам автомата q_m і ними позначаються, а дуги графа відповідають усім можливим переходам автомата $q_m \to q_s$ із одного стану в інший, відображені в таблиці переходів. Початок дуги позначаємо вхідним сигналом z_f, кінець дуги – вихідним сигналом w_g.



Граф автомата

Задача 15

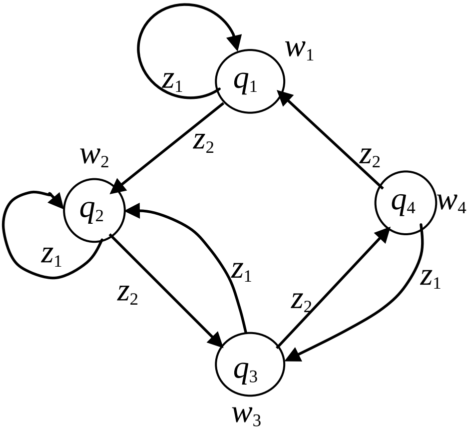
Автомат Мура заданий позначеною таблицею переходів.

	w_1	w_2	w_3	w_4
	q_1	q_2	q_3	q_4
z_1	q_1	q_2	q_2	q_3
z_2	q_2	q_3	q_4	q_1

Подати автомат графічним способом.

Розв'язування

Графічний спосіб задання автомата Мура подає його у вигляді орграфа. У вершинах графа проставляються символи станів q_1, q_2, …, q_M, а біля них – символи вихідних сигналів w_g, що відповідають їм у позначеній таблиці переходів. Дуги графа відповідають усім можливим переходам автомата $q_m \to q_s$ з одного стану в інший, відображеним у таблиці пере-

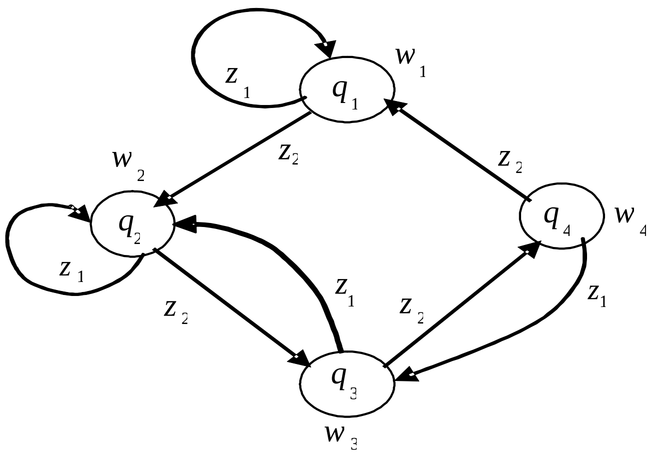


Граф автомата

ходів. Дуги позначаємо вхідним сигналом z_f. Одержаний для даної таблиці граф автомата зображений на рисунку.

Задача 16

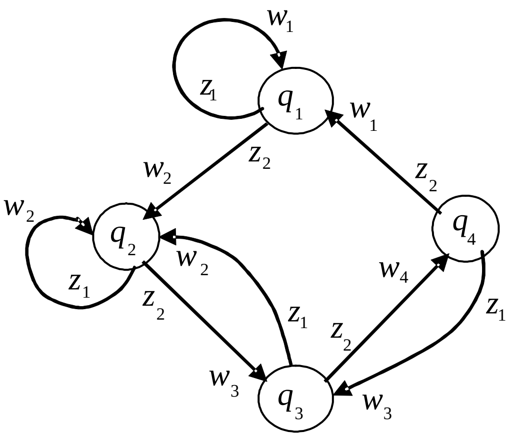
Автомат Мура заданий графічно, як показано на рисунку.



Перетворити автомат до еквівалентного йому автомата Мілі, і одержаний автомат подати табличним способом.

Розв'язування

Коли задано граф автомата Мура, то за ним можна перейти до графа еквівалентного автомата Мілі таким чином: вихідний сигнал w_g, записаний поруч із вершиною q_s, переносять на всі дуги, що входять у цю вершину.



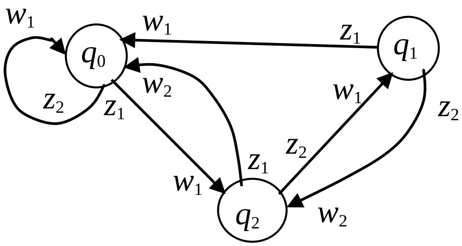
Під час такого перетворення функції переходів в автоматі A Мура та автоматі A' Мілі залишаються однаковими. Одержимо граф автомата Мілі, зображений на рисунку.

Подамо автомат Мілі таблично за допомогою суміщеної таблиці. Кількість стовпців M таблиці дорівнює кількості станів $M=4$; кількість стрічок F – кількості вхідних сигналів $F=2$. Всередині клітинок таблиці ставимо символи q_s/w_g, що відповідають дузі, яка описує перехід $q_m \rightarrow q_s$. Наприклад, в одержаному графі автомата Мілі з дуги, яка описує перехід із стану q_1 у стан q_2, ми бачимо, що цей перехід ініціюється вхідним сигналом z_2 і супроводжується виробленням вихідного сигналу w_2, тому в клітинці на перетині рядка z_2 і стовпця q_1 записуємо символи q_2/w_2. У результаті одержимо суміщену таблицю автомата Мілі:

	q_1	q_2	q_3	q_4
z_1	q_1/w_1	q_2/w_2	q_2/w_2	q_3/w_3
z_2	q_2/w_2	q_3/w_3	q_4/w_4	q_1/w_1

Задача 17

Перетворити автомат Мілі, заданий графічно, на еквівалентний автомат Мура.



Граф автомата Мілі

Розв'язування

Із заданого графа знаходимо, що в автоматі Мілі A алфавіт вхідних сигналів $Z_A=\{z_1, z_2\}$, алфавіт вихідних сигналів $W_A=\{w_1, w_2\}$, множина станів $Q_A=\{q_0, q_1, q_2\}$.

В еквівалентному автоматі Мура алфавіти вхідних і вихідних сигналів повинні бути такими самими, як у заданому автоматі Мілі:

$$Z_A=Z_B=\{z_1, z_2\}, \quad W_A=W_B=\{w_1, w_2\}.$$

Побудуємо множину станів Q_B автомата Мура таким чином, щоб він був еквівалентним заданому автомату Мілі. Для цього знайдемо множини пар, породжуваних кожним станом q_m автомата A:

Стан	Породжувані пари
q_2	$\{(q_2, w_1), (q_2, w_2)\}=\{b_4, b_5\}$
q_1	$\{(q_1, w_1)\}=\{b_3\}$
q_0	$\{(q_0, w_1), (q_0, w_2)\}=\{b_1, b_2\}$

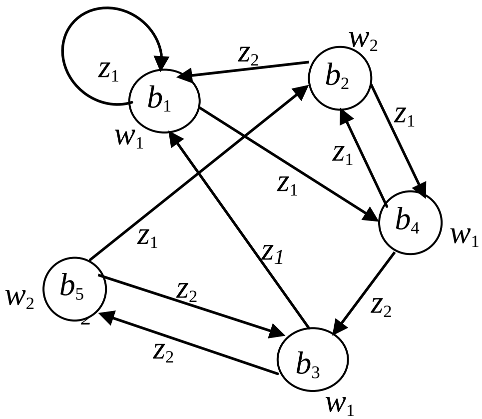
Звідси одержуємо множину станів автомата Мура: $Q_B=\{b_1, b_2, b_3, b_4, b_5\}$. Для знаходження функції виходів λ_B із кожним станом, що являє собою пару виду (q_m, w_g), ототожнимо вихідний сигнал, який є другим елементом цієї пари. У результаті маємо: $\lambda_B(b_1)=\lambda_B(b_3)=\lambda_B(b_4)=w_1$; $\lambda_B(b_2)=\lambda_B(b_5)=w_2$.

Побудуємо функцію переходів δ_B. Оскільки в автоматі A зі стану q_0 є перехід під дією сигналу z_1 у стан q_2 з видачею w_1, то з підмножини станів $\{b_1, b_2\}$, породжуваних q_0, в автоматі B повинен здійснитись перехід у стан $(q_2, w_1)=b_4$ під дією сигналу z_1.

Аналогічно, з $\{b_1, b_2\}$ під дією z_2 повинен здійснитись перехід у $(q_0, w_1) = b_1$, з $(q_1, w_1) = b_3$ під дією z_1 – перехід у $(q_0, w_1) = b_1$, а під дією z_2 – е $(q_2, w_2) = b_5$. Нарешті, зі станів $\{(q_2, w_1), (q_2, w_2)\} = \{b_4, b_5\}$ під дією z_1 – у $(q_0, w_2) = b_2$, а під дією z_2 – у $(q_1, w_1) = b_3$. У результаті одержуємо таблицю переходів і граф еквівалентного автомата Мура.

Позначена таблиця переходів автомата Мура

w_g	w_1	w_2	w_1	w_1	w_2
z_j / b_i	b_1	b_2	b_3	b_4	b_5
z_1	b_4	b_4	b_1	b_2	b_2
z_2	b_1	b_1	b_5	b_3	b_3



Граф автомата Мура, еквівалентного заданому автомату

Задача 18

Підйомник переміщується з одного рівня на інший реверсивним приводом, що вмикається виконавчими елементами – контакторами КМ1 і КМ2. Пуск підйомника здійснюється за командою від поверхових кнопок SB1, SB2, SB3, SB4. Зупинення підйомника забезпечують нижній і верхній кінцеві вимикачі SQ1 і SQ2. Аварійним ситуаціям запобігає контроль закриття дверей шахти на першому і другому поверхах вимикачами SQ3, SQ4. Розробити абстрактний автомат керування підйомником у вигляді графа.

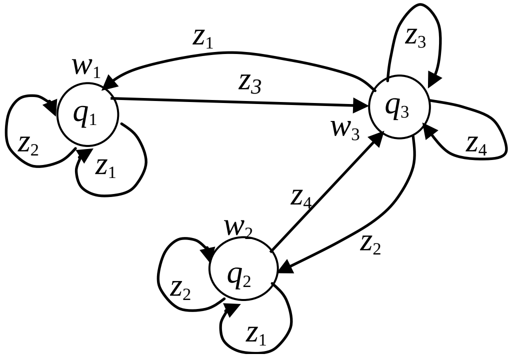
Розв'язування

Із фізичного принципу роботи підйомника зрозуміло, що він може перебувати в трьох станах – стан нерухомості q_3 (зупинення на 1-му або на 2-му поверсі), стан руху вгору q_1, стан руху вниз q_2. Тому зручно вважати, що автомат, який ним керує, має теж три стани: $Q=\{q_1, q_2, q_3\}$. Автомат не є ініціальним, оскільки на початку керування він може бути в будь-якому стані. Рух угору забезпечується увімкненням контактора КМ1 реверсивного привода, для чого на цей контактор потрібно подавати вихідний (керуючий) сигнал автомата w_1, для руху вниз потрібно подавати керуючийсигнал w_2 на контактор КМ2, для забезпечення спокою потрібно вимкнути обидва сигнали, для цього використовуємо керуючий сигнал $w_3=\overline{w_1+w_2}$, який діє тоді, коли не увімкнений хоча б один із сигналів w_1, w_2. Таким чином, алфавіт вихідних сигналів автомата $W=\{w_1, w_2, w_3\}$. Автомат керування підйомником є автоматом Мура, тому що його вихідні сигнали w_3, w_1, w_2 діють відповідно під час зупинення, руху вгору і руху вниз.

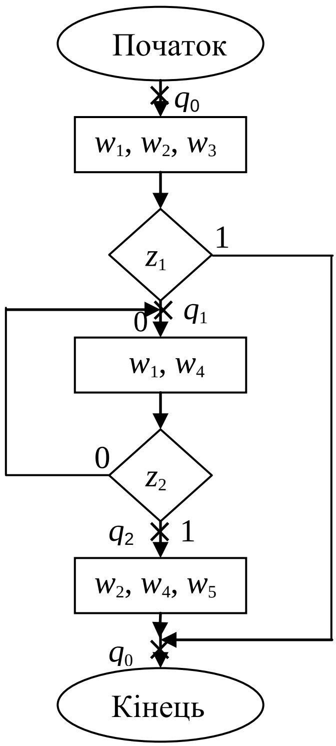
Усі командні змінні (вхідні сигнали автомата) зведемо до чотирьох: пуск вгору (ПВ) z_1 – натиснені кнопки "Вгору" SB1 або SB3 на першому або другому поверсі, $z_1=$ (SB1+SB3)·SQ3, де знак "+" є символом логічного додавання (диз'юнкція), а знак "·" є символом логічного множення (кон'юнкція); пуск вниз (ПН) z_2 – натиснуті кнопки "Вниз" SB2 або SB4 на першому або другому поверсі, $z_2=$ (SB2+SB4)·SQ4; зупинення руху вгору z_3 - замкнено кінцевий вимикач SQ2, $z_3=$ SQ2; зупинення руху вниз z_4 - кінцевий вимикач SQ1 замкнено, $z_4=$ SQ1. Таким чином, алфавіт вхідних сигналів $Z=\{z_1, z_2, z_3, z_4\}$. Позначена таблиця автомата Мура і його граф зображені нижче.

	w_1	w_2	w_3
	q_1	q_2	q_3
z_1	q_1	q_2	q_1
z_2	q_1	q_2	q_2
z_3	q_3	–	q_3
z_4	–	q_3	q_3

Таблиця автомата



Граф керуючого автомата



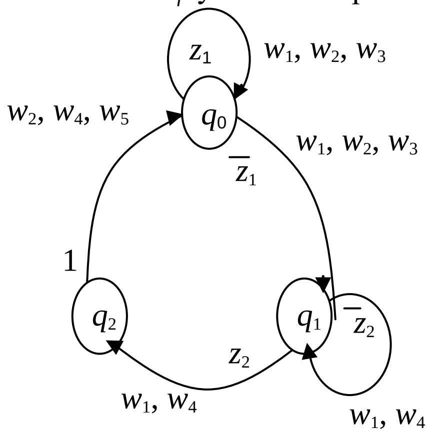
Задача 19

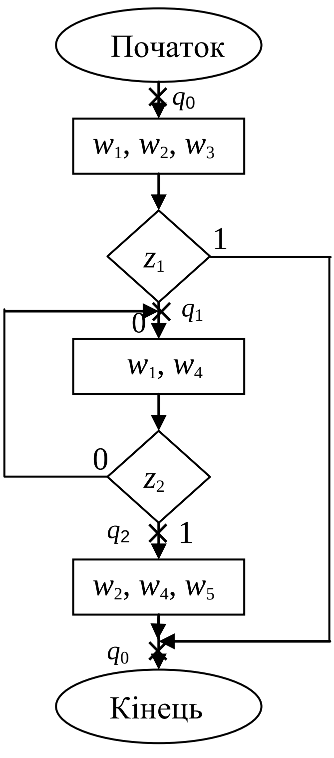
Граф автомата Мілі подано граф-схемою алгоритму його роботи. Подати автомат у вигляді графа.

Розв'язування

Спочатку виконаємо розмітку станів для автомата Мілі. Символами внутрішніх станів автомата q_0, q_1, ..., q_M позначаються виходи операторних вершин граф-схеми алгоритму. У випадку, коли після операторної вершини стоїть умовна вершина, то символ стану проставляється після такої умовної вершини. З одержаної розмітки будуємо граф автомата Мілі, для чого креслимо три вершини q_0, q_1, q_2, позначені на граф-схемі, після чого креслимо всі дуги, які відображують усі переходи в граф-схемі між станами.

Початок дуги позначаємо сигналом z_f умовної вершини, який ініціює даний перехід, а вихід дуги – сигналами w_f, що стоять в операторній вершині. Результати розмітки і граф автомата представлені на рисунку.



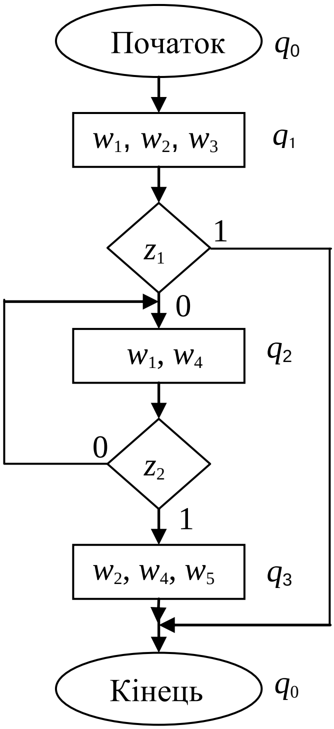


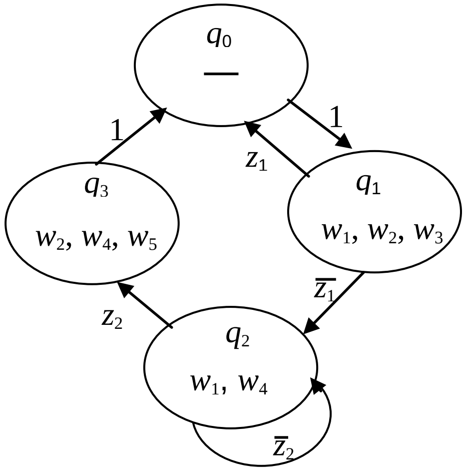
Задача 20

Граф автомата Мура подано граф-схемою алгоритму його роботи. Подати автомат у вигляді графа.

Розв'язування

Спочатку виконаємо розмітку станів для автомата Мура. Згідно п. 4.3.5, символами внутрішніх станів автомата q_0, q_1, ..., q_M позначаються операторні вершини граф-схеми алгоритму. З одержаної розмітки будуємо граф автомата Мура, для чого креслимо три вершини q_0, q_1, q_2, позначені на граф-схемі хрестиками, і біля них проставляємо вихідні сигнали автомата w_g. Після цього креслимо дуги, що відображують всі переходи в граф-схемі між станами. Дуги позначаємо сигналами умовних вершин z_f, які викликають відповідний перехід. Результати розмітки і граф автомата подані на рисунку.





Задача 21

Автомат Мілі заданий суміщеною таблицею переходів і виходів. Визначити його реакцію на вхідне слово $S_{вх} = z_1 z_2 z_2 z_2 z_1$. Вважати, що автомат установлений у початковий стан q_0.

$z_f \backslash q_m$	q_0	q_1	q_2	q_3
z_1	q_1/w_1	q_2/w_3	q_3/w_2	q_0/w_1
z_2	q_0/w_2	q_0/w_1	q_3/w_1	q_2/w_3

Розв'язування

Для автомата Мілі вихідний сигнал $w_g=\lambda(z_f, q_m)$, тому реакція автомата $S_{вих}=\lambda(q_0, S_{вх})$ визначається послідовністю станів, які автомат A_1 проходить, сприймаючи вхідне слово $S_{вх}$, і вихідним сигналам w_g, які відповідають цим переходам у суміщеній таблиці переходів і виходів автомата.

Отже, в початковому стані q_0 при надходженні вхідного сигналу z_1 слова $S_{вх}$, з суміщеної таблиці переходів знаходимо, що автомат виробить вихідний сигнал w_1 і перейде у стан q_1. Тепер у стані q_1 на вхід автомата надходить вхідний сигнал z_2 слова $S_{вх}$, і з суміщеної таблиці автомата бачимо, що автомат виробить вихідний сигнал w_1 і перейде у стан q_0. Аналогічно знаходимо реакцію автомата на наступні символи слова $S_{вх}$, і заносимо їх у таблицю реакції автомата:

z_f	z_1	z_2	z_2	z_2	z_1	
$w_g=\delta(z_f,q_m)$	w_1	w_1	w_2	w_2	w_1	
$q_s=\delta(z_f,q_m)$	q_0	q_1	q_0	q_0	q_0	q_1

Таким чином, реакція автомата $S_{вих}= w_1\ w_1\ w_2\ w_2\ w_1$.

Задача 22

Автомат Мілі заданий таблично. Знайти еквівалентний йому мінімальний автомат.

Таблиця переходів (δ)

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
Z_1	q_1	q_2	q_8	q_2	q_1	q_5	q_6	q_4
Z_2	q_5	q_4	q_2	q_6	q_6	q_3	q_2	q_1

Таблиця виходів (λ)

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
z_1	w_1	–	w_1	w_2	w_2	w_2	–	w_2
z_2	w_2	w_2	–	w_1	–	w_1	w_1	w_1

Розв'язування

1. Знаходимо розбиття π_1 множини станів $Q=\{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$ на 1-еквівалентні класи. За таблицею виходів знаходимо, що таких класів два:

$B_1=\{q_1, q_2, q_3\}$, $B_2=\{q_4, q_5, q_6, q_7, q_8\}$; $\pi_1=\{B_1, B_2\}$.

Підставимо у початкову таблицю переходів замість станів $q_1 \div q_8$ значення відповідних класів B_1, B_2. Після підстановки одержимо таблицю розбиття на 1-еквівалентні класи.

	B_1			B_2				
	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
z_1	B_1	B_1	B_2	B_1	B_1	B_2	B_2	B_2
z_2	B_2	B_2	B_1	B_2	B_2	B_1	B_1	B_1

2. Знаходимо розбиття π_2 множини станів Q на 2-еквівалентні класи (2-еквівалентні класи знаходяться всередині 1-еквівалентних класів). Виділяють їх за наступним загальним правилом. Якщо в таблиці переходів автомата, в якому стани позначені найменуваннями k-еквівалентних класів, декілька стовпців, що належать до одного й того самого класу, однакові, то стани автомата, що позначають ці стовпці, утворюють k+1- еквівалентні стани. Відповідно до цього правила за таблицею розбиття на 1-еквівалентні класи, знаходимо 2-еквівалентні класи: $C_1=\{q_1, q_2\}$, $C_2=\{q_3\}$, $C_3=\{q_4, q_5\}$, $C_4=\{q_6, q_7, q_8\}$, $\pi_2=\{C_1, C_2, C_3, C_4\}$. Виконавши перевірку, бачимо, що $\pi_2 \neq \pi_1$, тому продовжуємо.

Підставимо в початкову таблицю переходів замість станів $q_1 \div q_8$ відповідні значення C_1, C_2, C_3, C_4. Після підстановки отримаємо таблицю розбиття на 2-еквівалентні класи.

	C_1		C_2	C_3		C_4		
	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
z_1	B_1	B_1	B_2	B_1	B_1	B_2	B_2	B_2
z_2	B_2	B_2	B_1	B_2	B_2	B_1	B_1	B_1

3. Знаходимо розбиття π_3 множини станів $Q=\{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$ на 3-еквівалентні класи з одержаної таблиці розбиття на 2-еквівалентні класи аналогічно до того, як були знайдені 2-еквівалентні класи з таблиці розбиття на 1-еквівалентні класи: $D_1=\{q_1, q_2\}$, $D_2=\{q_3\}$, $D_3=\{q_4, q_5\}$, $D_4=\{q_6\}$, $D_5=\{q_7\}$, $D_6=\{q_8)$; $\pi_3=\{D_1, D_2, D_3, D_4, D_5, D_6\}$. Виконуємо перевірку, бачимо, що $\pi_3 \neq \pi_2$, тому продовжуємо.

Підставимо в початкову таблицю переходів замість станів $q_1 \div q_8$ відповідні значення D_1, D_2, D_3, D_4, D_5, D_6. Після підстановки отримаємо:

	D_1		D_2	D_3		D_4	D_5	D_6
	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
z_1	D_1	D_1	D_6	D_1	D_1	D_3	D_4	D_3
z_2	D_3	D_3	D_1	D_4	D_4	D_2	D_1	D_1

4. Знаходимо розбиття π_4 множини станів $Q=\{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}$ на 4-еквівалентні стани: $E_1=\{q_1, q_2\}$, $E_2=\{q_3\}$, $E_3=\{q_4, q_5\}$, $E_4=\{q_6\}$, $E_5=\{q_7\}$, $E_6=\{q_8)$; $\pi_4=\{E_1, E_2, E_3, E_4, E_5, E_6\}$. Виконуємо перевірку на рівність π_4 і π_3. Бачимо, що потужність їх однакова, а також рівні відповідні підмножини (класи): $E_1=D_1=\{q_1,q_2\}$, $E_2=D_2=\{q_3\}$, $E_3=D_3=\{q_4,q_5\}$, $E_4=D_4=\{q_6\}$, $E_5=D_5=\{q_7\}$, $E_6=D_6=\{q_8)$). Розбиття π_4 збігається з розбиттям π_3, отже, розбиття π_3 граничне. Тому пошук еквівалентних класів припиняємо.

5. Знаходимо множину Q' станів мінімального автомата A', для чого в класах D_m, що містять більше одного стану, залишаємо лише один стан q_s:

$$Q' = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}.$$

6. Із початкової таблиці переходів автомата викреслюємо стани, що не увійшли до множини Q', а в тих стовпцях, що залишилися, всі стани замінюємо на еквівалентні з множини Q': q_2 на q_1, а q_5 на q_4. Причому, якщо в таблиці виходів викреслюється стовпець, що має на перетині з i-м рядком пев-

ний сигнал w_j, а в залишеному стовпці з станом, еквівалентним викресленому, на перетині з i-м рядком проставлений прочерк, то в ньому замість прочерку необхідно записати сигнал w_j.

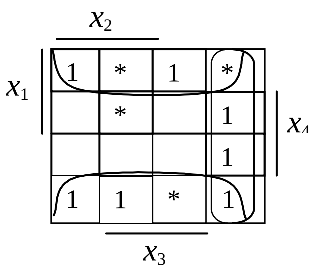
У результаті одержимо такі таблиці переходів і виходів мінімального автомата, еквівалентного заданому.

	q_1	q_3	q_4	q_6	q_7	q_8
Z_1	q_1	q_8	q_1	q_4	q_6	q_4
Z_2	q_4	q_1	q_6	q_3	q_1	q_1

	q_1	q_3	q_4	q_6	q_7	q_8
z_1	w_1	w_1	w_2	w_2	–	w_2
z_2	w_2	–	w_1	w_1	w_1	w_1

Задача 23

Синтезувати в базисі “І-Не” логічну функцію, заданою діаграмою Вейча - Карно.



Розв’язування

Синтез логічної схеми проводимо двома етапами – спочатку мінімізуємо логічну функцію за допомогою діаграми Вейча - Карно, а потім зводимо одержаний вираз до заданого логічного базису.

На діаграмі Вейча – Карно проводимо два контури, у перший із них входять верхній та нижній рядки таблиці, а у другий – правий крайній стовпець. При цьому ми врахували, що символ “*” в клітинках діаграми означає байдужий стан, тобто в них можна ставити 1, якщо кількість одиниць в цьому контурі збільшується (це приводить до спрощення логічної функції), або 0, якщо це не так. Крім того, верхня і нижня стрічки діаграми вважаються сусідніми, тому їх можна об’єднувати в один контур.

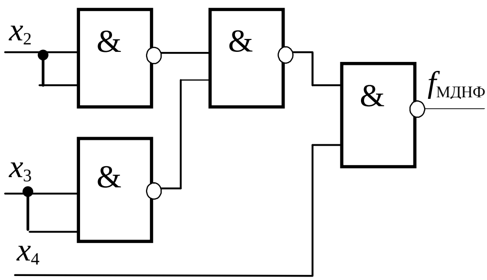
Згідно з одержаними контурами записуємо вираз мінімізованої логічної функції, користуючись правилом: кількість кон’юнкцій в мінімальній диз’юнктивній формі (МДНФ) логічної функції дорівнює кількості контурів, а в кон’юнкціях залишаються тільки ті змінні, які перетинаються контуром тільки один раз із запереченням або без заперечення. Оскільки першим контуром змінні x_1, x_2 і x_3 перетинаються двічі – один раз з запереченням і один раз без нього, то вони вилучаються з кон’юнкції, і в ній залишається лише змінна $\overline{x}_4$. З аналогічних міркувань в кон’юнкції, що відповідає другому контуру, залишаються змінні $\overline{x}_2 \cdot \overline{x}_3$. Отже, мінімізована логічна функція має вигляд

$$f_{\text{МДНФ}} = \overline{x_4} + \overline{x}_2 \cdot \overline{x}_3 .$$

Перетворюємо одержану логічну функцію до базису І-НЕ:

$$f_{\text{МДНФ}} = \overline{\overline{\overline{x_4} + \overline{x_2} + \overline{x_3}}} = \overline{x_4\,\overline{x_2}\;\overline{x_3}}.$$

Логічну схему, що реалізує дану функцію, подано на рисунку.



Задача 24

Синтезувати логічну схему в базисі І-Не для такої логічної функції: $F(X) = \overline{x_1}\,\overline{x_2} + \overline{x}_1 x_2 + x_0 x_1 x_3$.

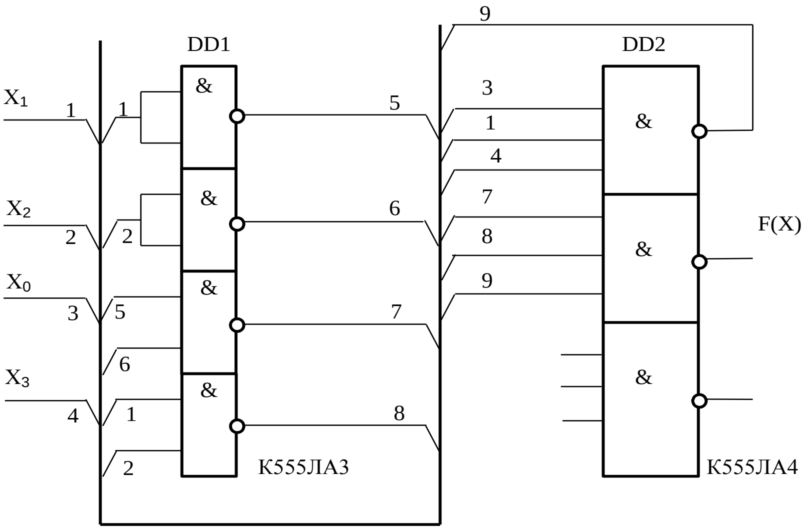
Розв’язування

Функцію $F(X)$ перетворюємо до такого вигляду, щоб вона містила лише операції кон’юнкції та заперечення:

$$F(X) = \overline{x_1}\,\overline{x_2} + \bar{x}_1 x_2 + x_0 x_1 x_3 = \overline{\overline{\bar{x}_1 \bar{x}_2}\;\;\overline{\bar{x}_1 x_2}\;\;\overline{x_0 x_1 x_3}}\,.$$

Перетворення виконано на основі використання закону подвійного заперечення: $\overline{\overline{F(X)}} = F(X)$ та правила де Моргана: $\overline{x + y + z} = \bar{x}\;\;\bar{y}\;\;\bar{z}$.

Логічна схема може бути реалізована за допомогою двох мікросхем: К555ЛА3 та К555ЛА4



Задача 25

На рисунку зображено узагальнений варіант алгоритму функціонування керуючого автомата процесора для реалізації деякої операції. На ньому в умовних вершинах стоять вхідні сигнали автомата X1, X2, X3, X4, X5, а в операційних вершинах – сигнали керування операційним блоком процесора Y1, …, Y10. За індивідуальним варіантом алгоритму синтезувати схему керуючого автомата жорсткого типу, виконуючи послідовно наступні етапи процедури синтезу.

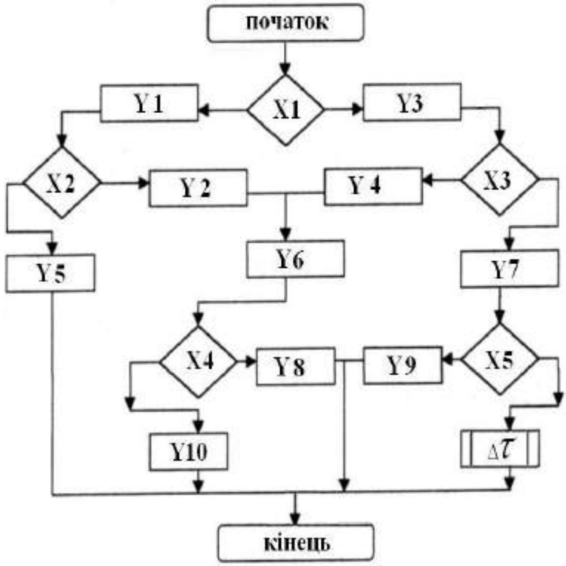


Рисунок 1 – Узагальнений варіант алгоритму функціонування керуючого автомата процесора

1. Вибрати індивідуальний варіант алгоритму роботи автомата, для чого записати номер варіанта (заданого викладачем) п'ятирозрядним двійковим числом у порядку зменшення ваг розрядів на лівих виходах умовних вершин X1, X2, X3, X4, X5 відповідно. Значення правих виходів умовних вершин вибирають інверсними відносно записаних. Нарисувати блок-схему індивідуального варіанта алгоритму, залишивши в ньому обов'язково операційні вершини Y6, Y7, Y10, а з інших – ті операційні вершини, які стоять в гілках умовних вершин.

2. Розробити алгоритм синтезу автомата.

3. Обґрунтувати вибір типу автомата (Мілі чи Мура).

4. Провести структурний синтез автомата.

5. Обґрунтувати вибір методу мінімізації логічних функцій

для процедури структурного синтезу.

6. Накреслити функціональну схему керуючого автомата.

7. Записати вирази для вихідних сигналів і функції збудження елементів пам’яті.

8. Провести сумісну мінімізацію цих функцій.

9. Функції звести до заданого базису.

10. Побудувати функціональну схему автомата.

Розв’язування

Синтезуємо автомат Мілі з використанням D-тригерів в базисі І-НЕ.

Варіант $5_{10} = 00101_2$.

Задану варіативно граф-схему алгоритма функціонування керуючого автомата подано на рис. 2 з розміткою станів q_i.

Згідно з алгоритмом синтезу цифрового автомата виконаємо формалізований опис ЦА у вигляді орієнтованого графа (рис. 3). Вершинами графа є стани автомата, а його дуги описують всі можливі переходи із одного стану до іншого в послідовності виконання етапів алгоритму роботи автомата. Початок дуги позначається вхідним сигналом, який ініціює перехід із попереднього стану в наступний, а вихід дуги – вихідним керувальним сигналом, який виробляє автомат у цьому переході.

На етапі розроблення схеми станів автомата спочатку визначимо кількість елементів пам’яті за формулою $n = \lceil \log_2 M \rceil$, де M- кількість станів автомата (у нашому випадку M=5). Отже, $n = \left| \log_2 5 \right| = \lceil 2.32 \rceil = 3$. За елементи пам’яті вибираємо синхронізовані тригери D-типу і задаємо таблицю кодів станів (табл. 1).

Якщо виходи тригерів подати на входи дешифратора з M виходами, то одиничний сигнал на одному з виходів дешифратора покаже стан автомата. Отже, схема станів автомата має такий вигляд (рис. 4).

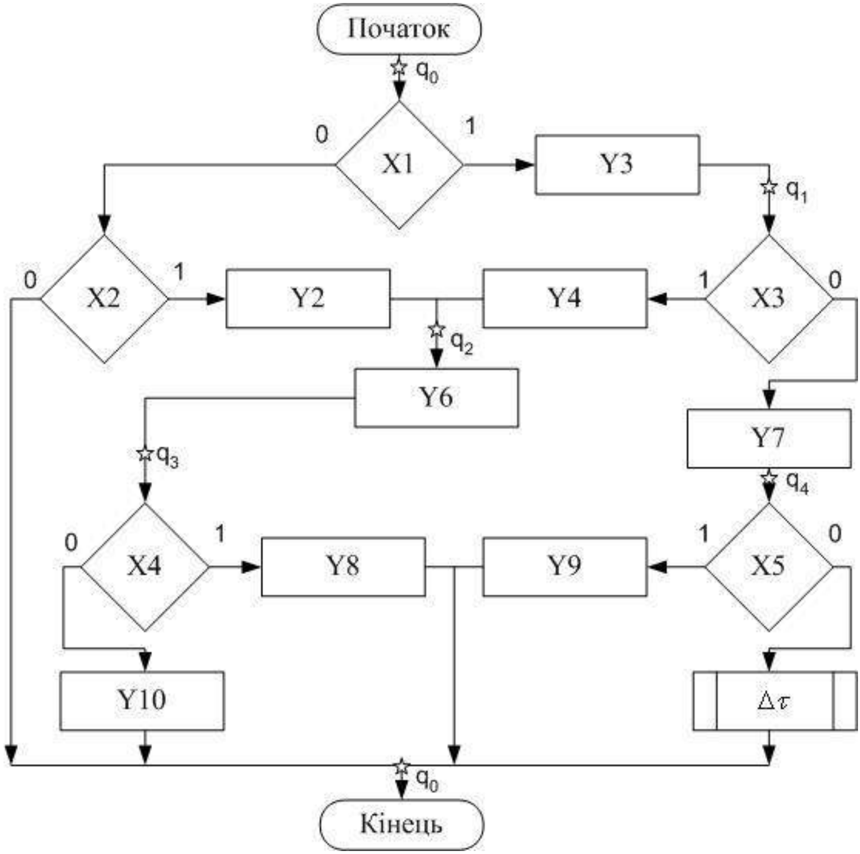


Рисунок 2 – Граф-схема алгоритму цифрового автомата

Розробимо таблицю переходів автомата, яку побудуємо відповідно до графа. Кількість рядків у таблиці переходів (табл. 2) дорівнює кількості дуг на графі (кількості різних переходів із стану в стан).

Для кожного переходу записують початковий стан, його двійковий код (стани тригерів), стан після переходу та його двійковий код, кон'юнкції вхідних сигналів, що являють собою умову переходу, а також вихідні (керувальні) сигнали, що виробляються автоматом під час цього переходу (ними позначені кінці дуг графа).

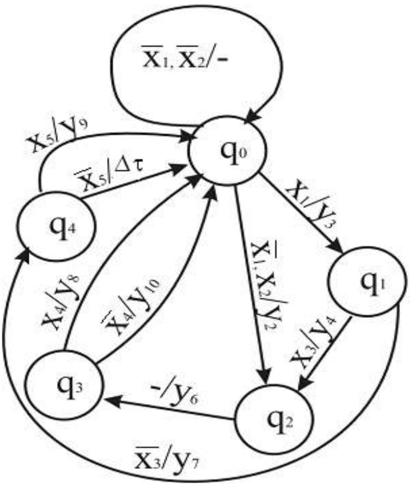


Рисунок 3 – Граф автомата

Таблиця 1 – Кодування станів автомата

Стан	Код стану		
	T_3	T_2	T_1
q_0	0	0	0
q_1	0	0	1
q_2	0	1	0
q_3	0	1	1
q_4	1	0	0

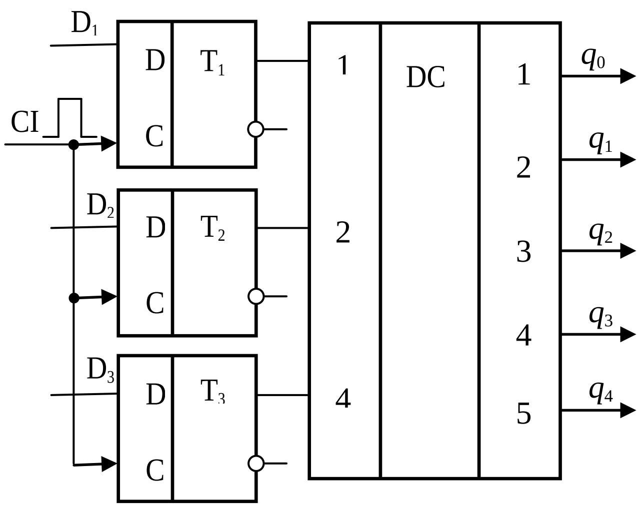


Рисунок 4 – Схема станів розроблюваного автомата

Таблиця 2 – Таблиця переходів автомата

Початковий стан автомата	**Код стану**			**Наступний стан автомата**	**Код стану**			**Умова переходу**	**Вихідні сигнали**	**Функції збудження тригерів**		
	$\mathbf{T_{3п}}$	$\mathbf{T_{2п}}$	$\mathbf{T_{1п}}$		$\mathbf{T_{3н}}$	$\mathbf{T_{2н}}$	$\mathbf{T_{1н}}$			$\mathbf{D_3}$	$\mathbf{D_2}$	$\mathbf{D_1}$
q_0	0	0	0	q_1	0	0	1	x_1	y_3	0	0	1
q_0	0	0	0	q_2	0	1	0	$\bar{x}_1 x_2$	y_2	0	1	0
q_0	0	0	0	q_0	0	0	0	$\bar{x}_1 \bar{x}_2$		0	0	0
q_1	0	0	1	q_2	0	2	0	x_3	y_4	0	2	0
q_1	0	0	1	q_4	1	0	0	$\bar{x}_3$	y_7	1	0	0
q_2	0	1	0	q_3	0	1	1	1	y_6	0	1	1
q_3	0	1	1	q_0	0	0	0	x_4	y_8	0	0	0
q_3	0	1	1	q_0	0	0	0	$\bar{x}_4$	y_{10}	0	0	0
q_4	1	0	0	q_0	0	0	0	x_5	y_9	0	0	0
q_4	1	0	0	q_0	0	0	0	$\bar{x}_5$	$\Delta\tau$	0	0	0

В останній колонці табл. 2 наведено значення функцій збудження (функцій переходів) входів тригерів D_1, D_2, D_3, які потрібно подати на D-входи відповідних тригерів, щоб спричинити заданий перехід. Із закону функціонування D-тригерів випливає, що їх значення відповідають значенням виходів тригерів після переходу, тобто $D_3 = T_{3н}$, $D_2 = T_{2н}$, $D_1 = T_{1н}$. Для тригерів інших типів значення функцій збудження одержують за таблицями переходу даних тригерів.

На етапі синтезу функцій переходів і виходів записуємо логічні вирази цих функцій згідно з табл. 2 у вигляді досконалих диз’юнктивних нормальних форм, при цьому їх аргументами вважаються початковий стан даного переходу і вхідний сигнал.

В останній колонці табл. 2 наведено значення функцій збудження (функцій переходів) входів тригерів D_1, D_2, D_3, які потрібно подати на D-входи відповідних тригерів, щоб спри-

чинити заданий перехід. Із закону функціонування D-тригерів зрозуміло, що їх значення відповідають значенням виходів тригерів після переходу, тобто $D_3 = T_{3н}$, $D_2 = T_{2н}$, $D_1 = T_{1н}$. Для тригерів інших типів значення функцій збудження одержують за таблицями переходу даних тригерів.

На етапі синтезу функцій переходів і виходів записуємо логічні вирази цих функцій згідно з табл. 2 у вигляді досконалих диз’юнктивних нормальних форм, при цьому їх аргументами вважаються початковий стан даного переходу і вхідний сигнал.

Для входів тригерів система рівнянь буде мати вигляд:

$$\left.\begin{array}{c} D_1 = q_0 x_1 + q_2, \\ D_2 = q_0 \overline{x_1} x_2 + q_1 x_3 + x_2, \\ D_3 = q \overline{x_3}. \end{array}\right\} \qquad (1)$$

Для функцій виходів система рівнянь буде мати вигляд:

$$\left.\begin{array}{c} y_2 = q_0 \overline{x_1} x_2, \\ y_3 = q_0 x_1, \\ y_4 = q_1 x_3, \\ y_6 = q_2, \\ y_7 = q_1 \overline{x_3}, \\ y_8 = q_3 x_4, \\ y_9 = q_4 x_5, \\ y_{10} = q_3 \overline{x_4}, \\ \Delta\tau = q_4 \overline{x_5}. \end{array}\right\} \qquad (2)$$

Набір функцій (1) використовується для побудови логічної схеми переходів автомата, а (2) – схеми виходів автомата.

Мінімізацію логічних функцій (1) і (2) зручно здійснити за методом діаграм Вейча - Карно, оскільки число їх аргументів менше 7. Однак в цьому випадку мінімізація не проводиться, оскільки візуально можна бачити, що ні для одного виразу не можна застосувати закон склеювання чи поглинання. Кількість логічних елементів можна зменшити за рахунок того, що деякі вихідні сигнали одержуються на виході логічних елементів, які вже використовуються у функціях переходів (порівняйте, наприклад, D_3 і y_7).

Переведемо складені системи рівнянь у базис І-Не:

$$D_1 = \overline{\overline{q_0 x_1}}\,\overline{\overline{q_2}},$$
$$D_2 = \overline{\overline{q_0 \overline{x_1} x_2}\;\overline{q_1 x_3}\;\overline{q_2}}$$
$$D_3 = \overline{\overline{q_1 \overline{x_3}}}.$$
$$y_2 = \overline{\overline{q_0 \overline{x_1} x_2}},$$
$$y_3 = \overline{\overline{q_0 x_1}},$$
$$y_4 = \overline{\overline{q_1 x_3}},$$
$$y_6 = q_2,$$
$$y_7 = D_3,$$
$$y_8 = \overline{\overline{q_3 x_4}},$$
$$y_9 = \overline{\overline{q_4 x_5}},$$
$$y_{10} = \overline{\overline{q_3 \overline{x_4}}},$$
$$\Delta\tau = q\overline{x_5}.$$

На етапі побудови схеми автомата за логічними виразами функцій, зведеними до заданого логічного базису, встановлюють склад необхідних логічних елементів і вузлів, послідовність їх з´єднання в одну схему, і креслять її з дотриманням відповідних стандартів. Функціональна схема автомата зображена на рис. 5.

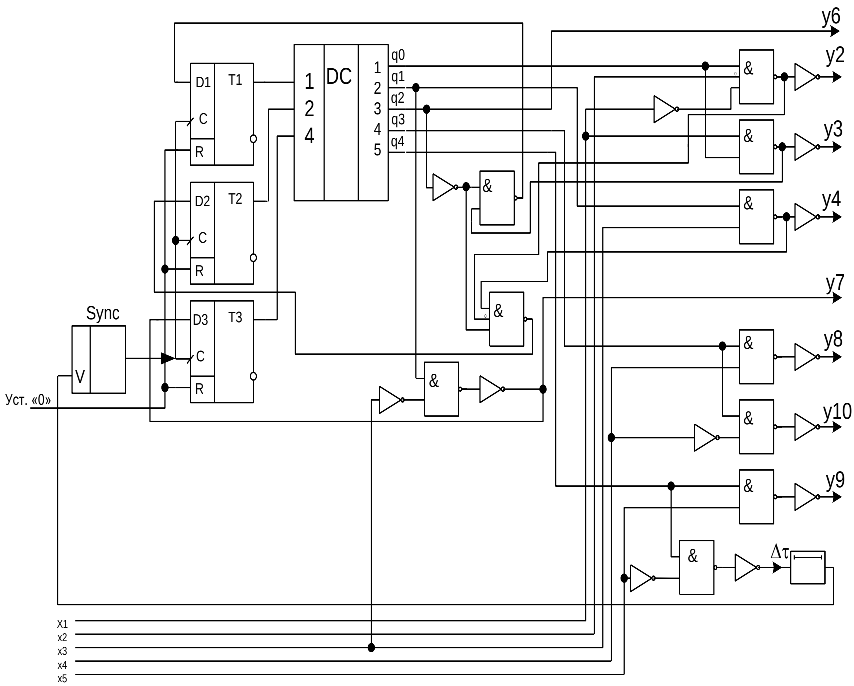


Рисунок 5 – Функціональна схема автомата

Навчальне видання

Биков Микола Максимович,
Черв’яков Володимир Дмитрович

ДИСКРЕТНИЙ АНАЛІЗ І ТЕОРІЯ АВТОМАТІВ

Навчальний посібник

Художнє оформлення обкладинки А. В. Павлова
Редактори: Н. А. Гавриленко, Н. З. Клочко, Н. В. Лисогуб, С. М. Симоненко
Комп’ютерне верстання І. В. Щокотової

Формат 60×84/16. Ум. друк. арк. 20,69. Обл.-вид. арк. 17,26. Тираж 300 пр. Зам. №

Видавець і виготовлювач
Сумський державний університет,
вул. Римського-Корсакова, 2, м. Суми, 40007
Свідоцтво суб’єкта видавничої справи ДК № 3062 від 17.12.2007.