МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

ДИСКРЕТНА МАТЕМАТИКА. Частина 2.

Булева алгебра.

ПРАКТИКУМ

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського

як навчальний посібник для студентів,

які навчаються за спеціальністю 113 «Прикладна математика»,

освітньою програмою «Наука про дані та математичне моделювання»

Київ

КПІ ім. Ігоря Сікорського

2024

Рецензенти: *Дичка І.А.*, д-р техн. наук., проф., професор кафедри програмного забезпечення комп'ютер-них систем ФПМ

Відповідальний редактор *Тарасенко-Клятченко О.В.,* канд. техн. наук, доц., доцент кафедри системного програмування і спеціалізованих комп'ютерних систем ФПМ

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського

(протокол № 06 від 28.03.2024р.)

за поданням Вченої ради факультету прикладної математики (протокол № 08 від 26.02.2024 р.)

Електронне мережне навчальне видання

Темнікова Олена Леонідівна

Тавров Данило Юрійович

ДИСКРЕТНА МАТЕМАТИКА. Частина 2. Булева алгебра. ПРАКТИКУМ

ДИСКРЕТНА МАТЕМАТИКА. Частина 2. Булева алгебра. ПРАКТИКУМ [Електронний ресурс]: навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані та математичне моделювання» / О. Л. Темнікова, Д. Ю. Тавров; КПІ ім. Ігоря Сікорського. — Електронні текстові дані (1 файл: 3,18 Мбайт). — Київ : КПІ ім. Ігоря Сікорського, 2024. — 185 с.

Навчальний посібник розроблено для оволодіння студентами, які навчаються за спеціальністю 113 «Прикладна математика», освітньою програмою «Наука про дані та математичне моделювання» факультету прикладної математики КПІ ім. Ігоря Сікорського, практичними навичками з кредитного модуля «Дискретна математика. Частина 2». Даний практикум з дискретної математики містить основні теоретичні відомості та практичні прийоми та широкий спектр застосування і розв'язання задач з булевої алгебри: дослідження булевих функцій та алгебри висловлювань і формального числення, як моделей булевої алгебри; теорія абстрактних автоматів, як розширення тривіальних автоматів (комбінаційних схем булевих функцій) і основи мереж Петрі.

ЗМІСТ

ВСТУП

Навчальне видання призначено для студентів, які вивчають дисципліну «Дискретна математика» з циклу загальної підготовки навчального плану спеціальності 113 «Прикладна математика», і є базовою в навчанні за освітньо-професійною програмою «Наука про дані та математичне моделювання», що спрямована на розроблення і використання нових інформаційних технологій, моделювання складних систем, систем штучного інтелекту, і для вирішення інших задач професійної діяльності.

Дисципліна «Дискретна математика» складається з двох кредитних модулів. Цей практикум розглядає теми й приклади, що належать до кредитного модуля «Дискретна математика. Частина 2. Булева алгебра». Модуль розраховано на 36 академічних годин лекційних та 36 академічних годин практичних занять; вивчається на факультеті прикладної математики в 2 семестрі; складається з трьох розділів: булева алгебра, висловлювання і абстрактні теорії, основи теорії абстрактних автоматів.

Метою вивчення кредитного модуля «Дискретна математика. Частина 2» є засвоєння основних понять і методів, потрібних для опанування подальших дисциплін спеціальності «113 Прикладна математика», формування у студентів компетенцій, необхідних для розв’язання складних спеціалізованих задач та практичних проблем прикладної математики у професійній діяльності, пов’язаної з аналізом та використанням математичних методів.

Предметом вивчення кредитного модуля є такі об’єкти, як булеві функції, висловлювання, аксіоматичні системи числення, абстрактні автомати та мережі Петрі.

Згідно з вимогами освітньо-професійної програми, студенти після засвоєння кредитного модуля «Дискретна математика. Частина 2» мають продемонструвати знання основних понять булевої алгебри; принципів дії абстрактих автоматів та мереж Петрі; математичних методів дискретного аналізу в прикладних науках і розв’язанні практичних задач; та уміння:

досліджувати системи булевих функцій на повноту, замкнутість, несуперечність, знаходити досконалі і мінімальні стандартні форми; синтезувати та аналізувати моделі, побудовані на базі абстрактних автоматів та мереж Петрі.

У цьому практикумі розглядаються основні теоретичні відомості та практичні прийоми дослідження булевих функцій, аналізу висловлювань і систем висловлювань, виявленню логічних законів та моделюванню процесів і об’єктів за допомогою абстрактних автоматів і мереж Петрі.

Кожній темі відповідає глава у практикумі; перед завданнями наведено короткі теоретичні відомості; приклад, що демонструє виконання та оформлення розв’язання задач за даною темою; наприкінці кожного розділу — відповіді й розв’язання деяких завдань. Практикум містить список рекомендованої літератури і додатки: із теоретичними відомостями про теорію комплектів (мультмножин); методичними вказівками щодо виконання перевірочних комплексних контрольних робіт і списком контрольних запитань та тестових завдань до кожної теми, які допомагають закріпити та поглибити розуміння матеріалу.

Це видання оновлює і значно розширює практикум із дисципліни «Дискретна математика» для студентів спеціальності 113 «Прикладна математика» / О. Л. Темнікова [Електронне видання] – К. : КПІ ім. Ігоря Сікорського, 2018. Видання має стати в пригоді під час проведення практичних занять, для підготовки до ректорського контролю та вступних іспитів на навчання за програмами магістерської підготовки.

1. БУЛЕВА АЛГЕБРА

1.1. Булеві формули. Диз’юнктивні та кон’юнктивні нормальні форми

Теоретичні відомості

Визначення 1.1. Будь-яку змінну, яка може приймати одне з двох можливих значень 0 або 1, назвемо булевою змінною.

Визначення 1.2. Функції, які визначені на множині {0, 1} і приймають значення з цієї множини, називаються булевими функціями.

Булева функція називається n-місною, якщо число її змінних дорівнює n. Занумеруємо змінні булевої функції. Сукупність значень змінних булевої функції назвемо набором (або кортежем). Набір довжини n будемо називати n-набором. Тоді областю визначення n-місної булевої функції є сукупність всіляких упорядкованих n-наборів, компонентами яких є значення булевих змінних 0 або 1.

Кожен n-набір можна розглядати як запис деякого числа в двійковій системі числення. Нехай n-набір має вигляд: a_1, a_2, …, a_n , де a_i=0 або 1.Тоді число, визначене як $a_1*2^{n-1}+a_2*2^{n-2}+\ldots+a_n$, назвемо номером даного набора. Зрозуміло, що номером n-набора може бути будь-яке натуральне число $0<=K<=2^{n-1}$.

Наприклад, для булевої функції $f(x, y, z)$ множина значень $x=1$, $y=0$, $z=1$ записується як набор 101. Булева функція може бути задана за допомогою таблиці – перерахування її значень на всіх наборах. Множину наборів прийнято записувати в лексикографічному порядку, тому кожен набір являє собою код двійкового числа. Відповідне йому десяткове число будемо називати номером набору. Наприклад, номер набору 101 дорівнює 5, номер набору 110 - 6.

Твердження 1.1. Кількість наборів булевої функції $f(x_1, \ldots, x_n)$ від n змінних дорівнює 2^n. Кількість булевих функцій від n змінних дорівнює 2^{2^n}.

Існує 4 булеві функції від однієї змінної та 16 булевих функцій від двох змінних, тримісних - вже 256, а 4-х-місних - близько 65 тисяч.

Всі набори розташуємо в порядку зростання їх номерів, а поруч з кожним набором розмістимо значення булевої функції на цьому наборі. Отриману таким чином таблицю назвемо таблицею значень, або таблицею істинності булевої функції. У таблиці 1.1 наведено чотири булеві функції від однієї змінної:

f_0 – константа 0,

f_1 – тотожна функція $f(x) = x$,

f_2 – заперечення $f(x) = \neg x$,

f_3 – константа 1.

Таблиця 1.1.

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

В табл. 1.2 приведені функції від двох змінних.

Таблица.1.2.

x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	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
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Ці функції:

f_0 – константа 0

f_1 – $x \wedge y$ (або x & y) – кон'юнкція

f_2 – Заперечення імплікації $x \rightarrow y$ $f(x,y) = \neg (x \rightarrow y)$

f_3 – Тотожна функція $f(x,y) = x$

f_4 – Заперечення імплікації $y \to x$ $f(x,y) = \neg(y \to x)$

f_5 – Тотожна функція $f(x,y) = y$

f_6 – $x \oplus y$ – сума за *mod* 2 (операція Жегалкіна)

f_7 – $x \vee y$ – диз'юнкція

f_8 – заперечення диз'юнкції $f(x,y) = \neg(x \vee y) = (x \downarrow y)$ – стрілка Пірса (іноді можна зустріти назву функція Даггера або функція Вебба)

f_9 – $x \equiv y$ (або $x \sim y$) – еквівалентність (еквіваленція)

f_{10} – заперечення $f(y) = \neg y$

f_{11} – $y \to x$ (або $y \supset x$) – імплікація

f_{12} – заперечення $f(x) = \neg x$

f_{13} – $x \to y$ (або $x \supset y$) – імплікація

f_{14} – заперечення кон'юнкції $f(x,y) = \neg(x \wedge y) = (x \mid y)$ – штрих Шеффера

f_{15} – константа 1

Твердження 1.2. Множина, що складається з двох значень 0 і 1, на якому визначені унарна операція заперечення $\neg$ згідно табл. 1.1 і бінарні операції диз'юнкції $\vee$ і кон'юнкції $\wedge$ відповідно до табл. 1.2, є булевою алгеброю.

Аксіоми і теореми булевої алгебри безпосередньо випливають з властивостей булевих решіток. Наведемо список аксіом і основних теорем булевої алгебри.

Аксіоми булевої алгебри:

(1.1) *комутативні закони:*

$a \vee b = b \vee a$;

$a \wedge b = b \wedge a$;

(1.2) *дистрибутивні закони:*

$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$;

$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$;

(1.3) *асоціативні закони:*

$a\vee(b\vee c) = (a\vee b)\vee c$;

$a\wedge(b\wedge c) = (a\wedge b)\wedge c$;

(1.4) *властивості 0 й 1:*

$a \vee 0 = a$, $a \wedge 1 = a$;

(1.5) *закон виключення третього:*

$a \vee \neg a = 1$; та *закон протиріччя:* $a \wedge \neg a = 0$.

Теореми:

(1.6) якщо для всіх a $a \vee b = a$, то $b = 0$;

якщо для всіх a $a \wedge b = a$, то $b = 1$;

(1.7) якщо $a \vee b = 1$ і $a \wedge b = 0$, то $b = \neg a$;

(1.8) $\neg\neg a = a$;

(1.9) $\neg 0 = 1$, $\neg 1 = 0$;

(1.10) *закон ідемпотентності:*

$a \vee a = a$;

$a \wedge a = a$;

(1.11) $a \vee 1 = 1$, $a \wedge 0 = 0$;

(1.12) *закони поглинання:*

$a \vee (a \wedge b) = a$;

$a \wedge (a \vee b) = a$;

(1.13) *закони де Моргана:*

$\neg(a \vee b) = \neg a \wedge \neg b$;

$\neg(a \wedge b) = \neg a \vee \neg b$;

(1.14) *закони склеювання:*

$(a \vee b) \wedge (a \vee \neg b) = a$;

$(a \wedge b) \vee (a \wedge \neg b) = a$;

(1.15) *закони поглинання доповнення:*

$a \vee (\neg a \wedge b) = a \vee b$;

$a \wedge (\neg a \vee b) = a \wedge b$.

Це найбільш проста модель булевої алгебри, разом з тим найбільш важлива для комп'ютерних наук. Інша поширена модель - модель обчислення висловлювань.

Якщо взяти деяку непорожню множину **A**, то її множина-ступінь **ρ(A)** буде моделлю булевої алгебри, якщо домовитися про наступне (встановити ізоморфізм між поняттям та визначенням):

- елементи цієї булевої алгебри – різні підмножини множини A;
- операція кон’юнкції визначається як перетин множин, диз’юнкції – об’єднання множин, операція заперечення – абсолютне доповнення;
- множина ʊ (універсум) і ø – відповідно одиниця й нуль алгебри.

Неважко переконатися, що всі аксіоми, наведені вище, будуть виконуватися.

Булеві формули

Визначення 1.3.

- Кожна змінна є формулою.
- Якщо x, y – формули, то формулами є $(\neg x)$, $(x \vee y)$, $(x \,\&\, y)$.
- Інших формул нема.

У зображенні формул прийняті наступні допущення: зовнішні дужки опускають; встановлюють пріоритети виконання операцій в наступному порядку:

$\neg$ – заперечення (найвищій пріоритет),

$\wedge$ – кон’юнкція**,**

$\vee$ – диз'юнкція,

$\rightarrow, \equiv$ – імплікація й еквіваленція (мають однаковий пріоритет, тим паче, що вони не є алгебраїчними операціями, а слугують для скорочення запису).

З урахуванням цих пріоритетів надлишкові дужки також опускаються.

Для кожної булевої формули можна побудувати її таблицю істинності, обчислюючи її значення на кожному наборі змінних. Звідси випливає, що кожній булевій формулі відповідає деяка булева функція, а формулу булевої алгебри можна розглядати як представлення булевої функції від деяких змінних.

Кожна булева формула єдиним чином визначає булеву функцію. Єдиність випливає з того, що для кожної формули булевої алгебри можна побудувати єдину таблицю істинності. Тому в подальшому поняття «булева формула» і «булева функція» будемо вживати як рівнозначні.

Протилежне твердження про відповідність булевих формул і функцій, не справедливо, тобто кожній булевій функції можна поставити у відповідність безліч булевих формул.

Булеві формули називаються еквівалентними, або рівносильними, якщо їх таблиці збігаються. Якщо дві рівносильні формули з'єднати за допомогою операції еквівалентності, то отримана формула буде тотожно дорівнювати одиниці. Такі тотожно істинні формули називаються тавтологіями або загальнозначущими. Формули, тотожно рівні нулю, тобто тотожно хибні, називаються суперечливими або суперечностями. Аксіоми і теореми булевої алгебри задають рівносильні формули.

Розглянемо деякі рівносильні формули (див. Табл. 1.3).

Таблиця 1.3.

x	y	$x\to y$	$\neg x\vee y$	$y\to x$	$(x\to y)(y\to x)$	$x\equiv y$	$x\oplus y$	$\neg(x\oplus y)$
0	0	1	1	1	1	1	0	1
0	1	1	1	0	0	0	1	0
1	0	0	0	1	0	0	1	0
1	1	1	1	1	1	1	0	1

Ми бачимо, що таблиці значення формул $x \to y$ й $\neg x \vee y$ збігаються. Отже, ці формули еквівалентні:

(1.16) $x \to y = \neg x \vee y = \neg(x \& \neg y)$.

Еквівалентні також наступні формули:

(1.17) $x \equiv y = \neg(x \oplus y)$,

(1.18) $x \equiv y = (x \to y)(y \to x)$.

В (1.18) ми можемо замінити імплікацію на рівносильну їй формулу (1.16). В результаті отримаємо:

(1.19) $x \equiv y = (x \to y)(y \to x) = (\neg x \vee y)(\neg y \vee x)$ або $(\neg x \& \neg y) \vee (x \& y)$.

(1.20) Тоді, $x \oplus y = \neg(x \equiv y) = (\neg x \& y) \vee (x \& \neg y) = (x \vee y) \& (\neg x \vee \neg y)$.

Ці равносильності пояснюють, чому при визначенні булевої формули використовуються тільки операції $\neg$, $\vee$, $\wedge$ – операції імплікації, еквіваленції й складання по модулю 2 можуть бути введені за допомогою рівносильних формул (1.16) - (1.20).

За допомогою рівносильних формул можна проводити еквівалентні перетворення булевих формул. Наприклад, перетворимо булеву формулу $xy \to (\neg y \oplus z)$.

$\mathbf{x\, y \to (\neg y \oplus z) =}$	$x \to y = \neg x \vee y$
$\mathbf{= \neg(x\, y) \vee (\neg y \neg z \vee y\, z) =}$	$x \oplus y = (\neg xy) \vee (x \neg y)$, $\neg\neg y = y$
$\mathbf{= \neg x \vee \neg y \vee \neg y \neg z \vee y\, z =}$	*закон де Моргана*
$\mathbf{= \neg x \vee \neg y \vee y\, z = \neg x \vee \neg y \vee z}$	*закони поглинання*
	закон поглинання доповнення

Побудуємо також для вихідної формули і перетвореної таблицю істинності (див. Табл.1.4) і переконаємося в їх равносильності.

Таблиця 1.4.

x	y	z	**xy**	$\neg y$	$\neg y \oplus z$	$xy \to (\neg y \oplus z)$	$\neg x$	$\neg x \vee \neg y$	$\neg x \vee \neg y \vee z$
0	***0***	***0***	0	1	1	1	1	1	1
0	***0***	***1***	0	1	0	1	1	1	1
0	***1***	***0***	0	0	0	1	1	1	1
0	***1***	***1***	0	0	1	1	1	1	1
1	***0***	***0***	0	1	1	1	0	1	1

1	***0***	***1***	0	1	0	1	0	1	1
1	***1***	***0***	1	0	0	0	0	0	0
1	***1***	***1***	1	0	1	1	0	0	1

Важливо вміти визначати, чи є формула тавтологією. Для цього можна скористатися одним з методів:

- побудувати таблицю істинності (формула повинна тотожно дорівнювати 1),
- виконати еквівалентні перетворення (і отримати 1),
- застосувати метод редукції (зведення до протиріччя – передбачається, що існує набір, на якому формула приймає значення 0, далі намагаються підібрати відповідні значення змінних або спростувати тезу).

Диз'юнктивні та кон'юнктивні нормальні форми

Оскільки одна і та сама булева функція може бути представлена різноманітними формулами, виникає питання про єдиність представлення булевих функцій, тобто знаходження такої універсальної форми, щоб кожна булева функція була представлена в ній. Такими формами в булевій алгебрі є *досконалі диз'юнктивна та кон'юнктивна нормальні форми*.

Визначення 1.4. Будь-яка булева формула, побудована за допомогою однієї тільки операції диз'юнкції над булевими змінними чи їхніми запереченнями, називається *елементарною диз'юнкцією* чи *диз'юнктом*. Наприклад, $x_1 \vee x_2 \vee \neg x_3 \vee x_4$ – елементарна диз'юнкція.

Визначення 1.5. Будь-яка булева формула, побудована за допомогою однієї тільки операції кон'юнкції над булевими змінними чи їхніми запереченнями, називається *елементарною кон'юнкцією,* чи *кон'юнктом*. Наприклад, $x_1 x_2 \neg x_3 \neg x_4$ – елементарна кон'юнкція.

Теорема 1.1. Для того, щоб елементарна диз'юнкція тотожно дорівнювала одиниці, необхідно та достатньо, щоб до неї входила деяка змінна разом із її запереченням.

Теорема 1.2. Для того, щоб елементарна кон'юнкція тотожно дорівнювала нулю, необхідно та достатньо, щоб у неї входила деяка змінна разом із її запереченням.

Справедливість теорем 1.1, 1.2 випливає з аксіоми (1.4) і визначення операцій диз'юнкції та кон'юнкції.

Визначення 1.6. Булева функція називається *конституентою одиниці* (мінтермом), якщо вона дорівнює одиниці тільки на одному наборі своїх аргументів.

Визначення 1.7. Булева функція називається *конституентою нуля* (макстермом), якщо вона дорівнює нулю тільки на одному наборі своїх аргументів.

Приклад. Серед булевих функцій двох аргументів кон'юнкція та стрілка Пірса є конституентами одиниці, а диз'юнкція, імплікація, штрих Шеффера є конституентами нуля.

Теорема 1.3. Не тотожно хибна елементарна кон'юнкція від n-змінних є конституентою одиниці від n-змінних.

Із теореми 1.3 випливає, що будь-яку конституенту одиниці можна представити у вигляді елементарної кон'юнкції. Для цього необхідно утворити кон'юнкцію всіх її аргументів та розставити заперечення над тими змінними, які рівні нулю на наборі, що перетворює функцію в одиницю.

Аналогічну теорему можна довести для елементарної диз'юнкції.

Теорема 1.4. Не тотожно істинна елементарна диз'юнкція від n-змінних є конституентою нуля від n-змінних.

Із цієї теореми випливає, що будь-яку конституенту нуля можна представити у вигляді елементарної диз'юнкції. Для цього необхідно утворити диз'юнкцію всіх змінних та розставити заперечення над тими змінними, які рівні одиниці на наборі, що перетворює функцію в нуль.

Приклад. Нехай функція $f(x, y, z, v)$ дорівнює одиниці на єдиному наборі 0110. Тоді вона є конституентою одиниці і її можна записати у вигляді: $f(x, y, z, v) = \neg xyz\neg v$.

Визначення 1.8. Диз'юнкція елементарних кон'юнкцій, що не містить двох однакових кон'юнкцій, називається *диз'юнктивною нормальною формою (ДНФ)*.

Визначення 1.9. Диз'юнктивна нормальна форма, всі кон'юнкції якої є конституенти одиниці, називається *досконалою диз'юнктивною нормальною формою (ДДНФ)*.

Визначення 1.10. Кон'юнкція елементарних диз'юнкцій, що не містить двох однакових диз'юнкцій, називається *кон'юнктивною нормальною формою (КНФ)*.

Визначення 1.11. Кон'юнктивна нормальна форма, всі диз'юнкції якої є конституенти нуля, називається *досконалою кон'юнктивною нормальною формою (ДКНФ)*.

Теорема 1.5. Будь-яку булеву функцію можна представити у вигляді ДДНФ і ДКНФ.

Приклад. Нехай булева функція $f(x, y, z)$ задана таблицею 1.5. Побудуємо ДДНФ і ДКНФ даної функції.

Таблиця 1.5.

x	0	0	0	0	1	1	1	1
y	0	0	1	1	0	0	1	1
z	0	1	0	1	0	1	0	1
$f(x, y, z)$	1	0	1	0	1	1	0	0

Для побудови ДДНФ булевої функції необхідно розглянути всі набори, на яких функція дорівнює одиниці, і утворити конституенти одиниці, що відповідають цим наборам. Отримані конституенти об'єднати знаками диз'юнкції. Отримаємо ДДНФ: $f(x, y, z) = \neg x\neg y\neg z \vee \neg xy\neg z \vee x\neg y\neg z \vee x\neg yz$. Двоїсте правило існує для побудови ДКНФ. Розглянемо всі набори, на яких булева функція дорівнює нулю. Утворимо всі конституенти нуля як елементарні диз'юнкції, в яких кожна змінна береться без заперечення, якщо вона дорівнює нулю, та з запереченням, якщо вона дорівнює одиниці на даному наборі. З’єднаємо всі конституенти нуля символами кон’юнкції, отримаємо ДКНФ:

$f(x, y, z) = (x \lor y \lor \neg z)(x \lor \neg y \lor \neg z)(\neg x \lor \neg y \lor z)(\neg x \lor \neg y \lor \neg z)$.

Інший спосіб полягає в тому, що будь-яку булеву формулу можна привести до вигляду ДНФ (КНФ) та ДДНФ (ДКНФ) за допомогою еквівалентних перетворень, використовуючи співвідношення (1.1) – (1.20).

Приклад. Приведемо до ДДНФ формулу $(x \oplus y) \to xz$. Отримаємо:

$(x \oplus y) \to xz = \neg(x \oplus y) \lor xz = (x \equiv y) \lor xz = \neg x \neg y \lor xy \lor xz$. Отримана формула представлена в ДНФ, і для того, щоб отримати її представлення в ДДНФ, необхідно в кожному «доданку» диз’юнкції (в кожному кон’юнкті) присутність всіх трьох змінних, від яких залежить функція. Отже, необхідно «помножити» кожний «доданок» на 1, утворену як диз’юнкцію змінної, якої не вистачає в даному кон’юнкті, і її заперечення. Таким чином, $\neg x \neg y$ та xy «помножуємо» на $1 = z \lor \neg z$, та xz – на $1 = y \lor \neg y$.

Тоді отримаємо: $\neg x \neg y(z \lor \neg z) \lor xy(z \lor \neg z) \lor xz(y \lor \neg y) = xyz \lor xy\neg z \lor \neg x \neg yz \lor \neg x \neg y \neg z \lor xyz \lor x \neg yz$. В силу ідемпотентності $xyz \lor xyz = xyz$, тому ДДНФ є $xyz \lor xy\neg z \lor \neg x \neg yz \lor \neg x \neg y \neg z \lor x \neg yz$. Аналогічно можна отримати ДКНФ.

Завдання 1.1.1. Перевірити еквівалентність формул *A* і *B*, використовуючи основні аксіоми та теореми булевої алгебри.

1) $A = xy \lor \neg x \neg z \lor x \neg z$, $B = x(\neg y \neg z) \lor (\neg x \lor \neg z)$;
2) $A = \neg x \neg y \neg z \lor \neg x \neg yz \lor \neg xy$, $B = \neg x$;
3) $A = xy \neg z \lor xy \lor z \neg z$, $B = xy$;
4) $A = x(y \lor z) \lor y(\neg x \neg z) \lor \neg z(\neg y \lor x)$, $B = x \lor y \lor \neg z$;
5) $A = (x \neg z) \lor (xy) \lor (y \neg z)$, $B = x \neg(yz) \lor \neg xz$;

6) $A = x \to (xy \to ((x \to y) \to y)\, z)$, $B = y \to (x \to z)$;

7) $A = \neg((x \to y) \vee ((x \to z)y))$, $B = (x\neg y)(\neg y \to x\neg z)$;

8) $A = (x \vee y)\,(y \vee z)\,(z \vee x)$, $B = xy \vee yz \vee zx$;

9) $A = ((x \vee y) \vee z) \to ((x \vee y)(x \vee z))$, $B = x \equiv z$;

10) $A = x(x \to y)$, $B = xy$;

11) $A = \neg(x \vee y) \vee (xy)$, $B = (x \vee \neg y)(\neg x \vee y)$;

12) $A = x \oplus y$, $B = (\neg xy) \vee (\neg yx)$;

13) $A = \neg(\neg xyz)$, $B = x \vee \neg z \vee \neg y$;

14) $A = (x \vee y)(x \vee z)(y \vee w)(z \vee w)$, $B = xw \vee yz$;

15) $A = ((x \oplus y) \to x \vee y))((\neg x \to y) \to (x \oplus y))$, $B = \neg(xy)$.

Завдання 1.1.2. Побудувати таблицю істинності булевої функції. Виразити булеву функцію у вигляді ДДНФ, ДКНФ. Спростити, використовуючи основні аксіоми та теореми булевої алгебри.

Приклад. $(x\to(y\vee z))\to xz$.

Метод еквівалентних перетворень булевих формул.

ДДНФ: $(x\to(y\vee z))\to xz =$ використовуємо еквівалентність $x\to y \equiv \neg x \vee y$

$=\neg(\neg x \vee y \vee z) \vee xz =$ використовуємо закон де Моргана

$=x\neg y\neg z \vee xz$ - ця форма представлення булевої функції є ДНФ.

Якщо виконувати спрощення, то далі використовуємо дистрибутивний закон

$x\neg y\neg z \vee xz = x(\neg y\neg z \vee z)=$ закон поглинання доповнення $\neg y\neg z \vee z=\neg y \vee z$

$= x(\neg y \vee z)= x\neg y \vee x z$.

Якщо метою є отримання ДДНФ алгебраїчним шляхом, то треба ввести 1 як $y \vee \neg y$ ($y \vee \neg y \equiv 1$).

$x\neg y\neg z \vee xz = (x\neg y\neg z) \vee (xz(y \vee \neg y)) =$ використовуємо закон дистрибутивності

$= x\neg y\neg z \vee xyz \vee x\neg yz$.

Таблиця істинності.

x	*Y*	*Z*	***f*(*x*, *y*, *z*)**	***конституенти***
0	0	0	0	$x \vee y \vee z$
0	0	1	0	$x \vee y \vee \neg z$
0	1	0	0	$x \vee \neg y \vee z$
0	1	1	0	$x \vee \neg y \vee \neg z$
1	0	0	1	$x \neg y \neg z$
1	0	1	1	$x \neg y z$
1	1	0	0	$\neg x \vee \neg y \vee z$
1	1	1	1	xyz

Для отримання ДКНФ: $(x \to (y \vee z)) \to xz = (\neg x \vee y \vee z) \to xz = \neg(\neg x \vee y \vee z) \vee xz = =$ $x \neg y \neg z \vee xz = =(x \vee xz)(\neg y \vee xz)(\neg z \vee xz) = x(\neg y \vee x)(\neg y \vee z)(\neg z \vee x)(\neg z \vee z) = =$ $x(\neg y \vee z)(\neg z \vee x) = x(\neg y \vee z) = (x \vee y \neg y)(\neg y \vee z \vee x \neg x) = ((x \vee y) \vee z \neg z) \&$ $\&((x \vee \neg y) \vee z \neg z)(\neg y \vee x \vee z)(\neg y \vee \neg x \vee z) = (x \vee y \vee z)(x \vee y \vee \neg z)(x \vee \neg y \vee z) \&$ $\&(x \vee \neg y \vee \neg z)(\neg x \vee \neg y \vee z)$.

Побудуємо ДДНФ та ДКНФ за допомогою таблиці істинності. Випишемо усі конституенти нулю та одиниці (див. таблицю). Диз'юнкція усіх конституент одиниці дає ДДНФ: $f(x, y, z) = x \neg y \neg z \vee xyz \vee x \neg y z$.

Кон'юнкція усіх конституент нулю дає ДКНФ:

$f(x, y, z) = (x \vee y \vee z)(x \vee y \vee \neg z)(x \vee \neg y \vee z)(x \vee \neg y \vee \neg z)(\neg x \vee \neg y \vee z)$.

1) $(x \vee \neg y \vee z)(\neg x \neg y \neg z)$;

2) $(\neg x \neg y \vee \neg y \neg z) \equiv (x \vee z \to y)$;

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

4) $x \to (x \to z)$;

5) $(x \to (x \to y)) \to z$;

6) $(x \equiv y) \equiv z$;

7) $(\neg x \to \neg y) \to (xy \to xz)$;

8) $((x \to y) \to \neg x) \to (x \to yx)$;

9) $\neg(x\, y \to \neg x)\, \neg(xy \to \neg y)$;

10) $(z \to x) \to (\neg(y \vee z) \to x)$;

11) $\neg(xy \to x) \vee (x(y \vee x))$;

12) $\neg(x(y \vee z)) \to (xy \vee z)$;

13) $((x \to y) \to (z \to \neg x)) \to (\neg y \to \neg z)$;

14) $((((x \to y) \to \neg x) \to \neg y) \to \neg z) \to z$;

15) $(x \to (y \to z)) \to ((x \to \neg z) \to (x \to \neg y))$.

Завдання 1.1.3. Визначити, чи є булева формула тавтологією.

1) $(y \downarrow \neg(x \neg z)) \mid ((\neg x \to \neg y) \to z)$;

2) $(y \downarrow (\neg z \oplus y)) \to ((((x \equiv y) \equiv z) \oplus y) \mid y)$;

3) $(y \downarrow (x \to z)) \mid (((y \to x) \equiv \neg z) \oplus (((x \vee y) \to y) \mid x))$;

4) $(x \oplus (x \to z)) \mid ((y \equiv z) \downarrow x)$

5) $(x \to (y \to z)) \to ((x \to \neg z) \to (x \to \neg y))$;

6) $\neg(x \to (\neg x \equiv z))\, (y \to \neg z) \vee \neg(x \to z)$;

7) $(x(y \vee z)) \to ((x \equiv y) \vee (z \to x))$;

8) $\neg(x \vee y \vee z) \to (\neg(xy) \equiv \neg(x \vee z))$;

9) $(b \to a \vee c) \to ((b \to c) \to ((d \to c) \to (b \vee d \to c)))$;

10) $(x \equiv \neg y) \vee (y \to xz) \vee (x \neg y \neg z)$.

Завдання 1.1.4. Розв'язати рівняння булевої алгебри.

Вказівка. *Для розв'язання рівнянь потрібно побудувати таблиці істинності для булевих формул з лівої і правої частин рівняння і визначити набори з однаковими значеннями функцій.*

1) $(x \vee \neg yz)(xz \to z) = 0$;

2) $(x \vee y \neg z)(x \vee z) = 1$;

3) $[(xy \equiv z) \rightarrow y] = [x\neg y \vee \neg xz]$;

4) $[xy \vee \neg yz] = [(x \oplus y) \rightarrow z]$;

5) $[(xy \equiv z) \rightarrow y] = [(x \oplus y) \rightarrow z]$;

6) $[y \equiv (\neg xz)] = 0$.

1.2. Алгебра Жегалкіна. Властивості булевих функцій

Теоретичні відомості

Визначення 1.12. Система елементів $\{0,1\}$, на якій визначені операції $\wedge$ (кон'юнкція) і $\oplus$ (сума за *mod* 2), для яких виконуються співвідношення:

(1.21) $x \oplus y = y \oplus x,$

(1.22) $x\,(y \oplus z) = xy \oplus xz,$

(1.23) $x \oplus x = 0,$

(1.24) $x \oplus 0 = x,$

а також співвідношення булевої алгебри, що відносяться до кон'юнкції і констант, називається *алгеброю Жегалкіна*.

З таблиці істинності операції суми за mod 2 слідує, що

(1.24) $\neg x = x \oplus 1$.

Операцію диз'юнкції можна виразити через $\oplus$ і $\wedge$ так:

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

Визначення 1.13. Будь-яка формула алгебри Жегалкіна, що має вигляд суми (за mod 2) кон'юнкції булевих змінних, називається *поліномом Жегалкіна*.

Якщо в кожен член полінома Жегалкіна кожна змінна входить один раз і поліном не містить однакових членів, то такий поліном Жегалкина називається *канонічним*.

Теорема 1.6. Будь-яка булева функція може бути єдиним способом предсталена у вигляді канонічного поліному Жегалкіна.

Правило для представлення булевої функції у вигляді полінома Жегалкіна: в булевої формулі, що задана у вигляді ДДНФ, досить замінити знак $\vee$ на знак $\oplus$, представити заперечення змінних як $\neg x = x\oplus 1$, розкрити дужки згідно із законом дистрибутивності (1.22) і привести подібні члени згідно (1.23, 1.24).

Приклад. Зведемо до канонічного поліному Жегалкіна булеву функцію: $f(x, y, z) = x\neg yz \vee x\neg y\neg z \vee xyz$. Оскільки функція знаходиться в ДДНФ, замінимо символи $\vee$ на $\oplus$, отримаємо:

$f(x, y, z) = x\neg yz \oplus x\neg y\neg z \oplus xyz = x(y \oplus 1)z \oplus x (y \oplus 1) (z \oplus 1) \oplus xyz =$
$= xyz \oplus xz \oplus xyz \oplus xz \oplus xy \oplus x \oplus xyz = xyz \oplus xy \oplus x.$

Привести функцію до канонічного поліному Жегалкіна (КПЖ) можна і через суперпозицію відповідних поліномів функцій, які складають вихідну.

На приклад, для $x\rightarrow y$ ДДНФ є $\neg x\neg y \vee \neg xy \vee xy$. Після перетворення ми отримаємо КПЖ $xy \oplus x \oplus 1$.

З іншої сторони, $x \rightarrow y = \neg x \vee y=(x\oplus 1) y \oplus (x \oplus 1) \oplus y = xy \oplus x \oplus 1$. Або $x \rightarrow y = \neg(x\neg y)== 1 \oplus x (1 \oplus y) = xy \oplus x \oplus 1$.

Тоді для функції з попереднього прикладу $(x\oplus y)\rightarrow xz$ по ДДНФ $xyz \vee xy\neg z \vee \neg x\neg yz \vee \neg x\neg y\neg z \vee x\neg yz$ побудуємо КПЖ $xyz \oplus xz \oplus x \oplus y \oplus 1$. Або, замінивши імплікацію на відповідний КПЖ ($\oplus$ і & допустимі операції в алгебрі Жегалкіна), отримаємо

$(x \oplus y)\rightarrow xz = (x \oplus y)xz \oplus (x \oplus y) \oplus 1 = xyz \oplus xz \oplus x \oplus y \oplus 1.$

Існують і інші способи побудови КПЖ: метод трикутника (трикутник Паскаля), перетворення Мебіуса.

Метод трикутника дозволяє перетворити таблицю істинності в поліном Жегалкіна шляхом побудови допоміжної трикутної таблиці згідно з такими правилами:

- Будується повна таблиця істинності.
- Будується допоміжна горизонтальна таблиця, в якій перший рядок збігається зі стовпцем значень функції в таблиці істинності.

- В кожному наступному рядку комірка отримується шляхом додавання за модулем 2 двох комірок попереднього рядка, які стоять в тому самому стовпчику та праворуч (перша комірка попереднього рядка додається до другої комірки, потім друга до третьої і т.д.).
- Рядки допоміжної таблиці нумеруються двійковими кодами в тому ж самому порядку, що й рядки таблиці істинності.
- Кожному двійковому коду ставиться у відповідність один із членів полінома Жегалкіна в залежності від позиції кодів-змінних, в яких стоять одиниці. Наприклад, комірці 111 відповідає член xyz, комірці 101 — член xz, комірці 010 — член y, комірці 000 — член const 1 і т. д.
- Якщо на першому місці будь-якого рядка стоїть одиниця, саме тоді відповідний член є в поліномі Жегалкіна.

Приклад на метод трикутника для функції $(x \oplus y) \to xz$ = **11000111.**

x	y	z									
0	0	0	**1**	***1***	*1*	*0*	*0*	*0*	*1*	*1*	*1*
0	0	1		0	1	0	0	1	0	0	
0	1	0	**y**	**1**	1	0	1	1	0		
0	1	1		0	1	1	0	1			
1	0	0	**x**	**1**	0	1	1				
1	0	1	**xz**	**1**	1	0					
1	1	0		0	1						
1	1	1	**xyz**	**1**							

Відповідь: $\boldsymbol{xyz \oplus xz \oplus x \oplus y \oplus 1}$

Властивості булевих функцій

Визначення 1.14. Функція, яка виражена поліномом Жегалкіна вигляду $\sum\alpha_\iota x_\iota \oplus \gamma$, де α_ι, γ є 0 або 1, називається *лінійною (L)*.

Всі функції однієї змінної лінійні. Лінійними функціями двох змінних є $x\oplus y$ і $x \equiv y$: $x \equiv y = \neg x\neg y \vee xy = (x\oplus 1)(y \oplus 1)\oplus xy = xy \oplus x\oplus y \oplus 1\oplus xy = x\oplus y \oplus 1$.

Визначення 1.15. Нехай дана деяка булева формула А. Формула А* називається *двоїстою* формулі А, якщо вона виходить з А шляхом заміни операцій кон'юнкції на диз'юнкцію, диз'юнкції на кон'юнкцію, 1 на 0, 0 на 1 усюди, де вони входять.

Наприклад: $A = xy \vee \neg xy$; $A^* = (x \vee y)(\neg x \vee y)$.

Теорема 1.7. Якщо $A(x_1,...,x_n)$ и $A^*(x_1,...,x_n)$ – дві взаємнодвоїсті формули, то $\neg A(x_1, ..., x_n) = A^*(\neg x_1, ..., \neg x_n)$.

Визначення 1.16. Функція називається *самодвоїстою (S)*, якщо вона двоїста сама собі, тобто $A(x_1, x_2, ..., x_n) = \neg A(\neg x_1, \neg x_2, ..., \neg x_n)$, або $\neg A(x_1, x_2, ..., x_n) = A(\neg x_1, \neg x_2, ..., \neg x_n)$.

Для самодвоїстої функції заперечення її аргументів призводить до заперечення самої функції, отже, самодвоїста функція на протилежних наборах приймає протилежні значення. Наприклад, функція $xy \vee xz \vee yz$ самодвоїста. Щоб показати це, візьмемо заперечення від кожної змінної і від всієї функції::

$\neg(\neg x\neg y \vee \neg x\neg z \vee\neg y\neg z) = (x\vee y)(x\vee z)(y\vee z) = (x \vee xy \vee xz \vee yz)(y\vee z) =$

$= xy\vee xy \vee xyz \vee yz\vee xz \vee xyz\vee xz \vee yz = xy \vee xz \vee yz$.

Відношення порядку на наборах булевих змінних визначається так. Розглянемо два набори: $A = (\alpha_1,..., \alpha_i, ..., \alpha_n)$ і $B = (\beta_1,..., \beta_i,..., \beta_n)$. Якщо для всіх i ($i = 1, ..., n$) $\alpha_i \le \beta_i$, і існує хоча б одне таке j, при якому $\alpha_j < \beta_j$, набір А передує (менше) набору В. Це позначається: $A \le B$. Якщо для деяких i $\alpha_i \le \beta_i$ і існує таке j, що $\alpha_j > \beta_j$, то набори A і B *непорівнювані*. Наприклад: набори (0101) і (0010) непорівнювані, а набір (0101) передує набору (0111).

Тоді булеву функцію можна задати на цій решітці, як показано на рис. 1.1. Причому серед булевих функцій можна виділити такі, які не зменшуються зі зростанням значень порівнянних наборів на булевої решітці. Такі функції називаються монотонними. Наприклад, на рис. 1.1. показана монотонна функція (00001111).

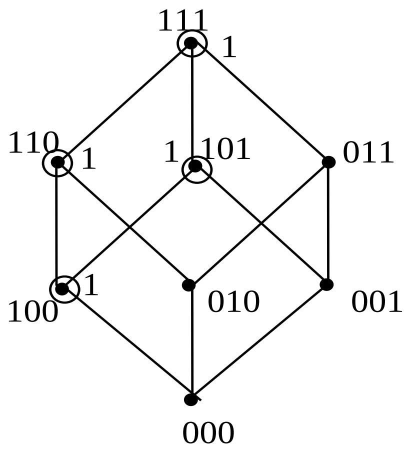


Рис. 1.1. Монотонна функція.

Визначення 1.17. Булева функція f називається *монотонною (M)*, якщо для будь-яких двох наборів A і B з її області визначення таких, що $A \le B$, $f(A) \le f(B)$. Якщо хоча б для однієї пари наборів таких, що $A \le B$, $f(A) > f(B)$, то функція немонотонна.

Перевірити на монотонність булеву функцію можна в наступний спосіб:

Задана таблиця істинності булевої функції.

Крок 1. Порівнюємо значення функції на наборах, сусідніх першої змінної, тобто верхню половину стовпця значень функції (вектор $\varphi 1$) з нижньою половиною (вектор $\psi 1$). Якщо умова $\varphi 1 \le \psi 1$ порушена, то функція не монотонна, йдемо на кінець.

Крок 2. Порівнюємо значення функції на наборах, сусідніх за другою змінною, тобто верхні четвертини стовпця значень функції (вектори $\varphi'2$, $\varphi''2$) із нижніми четвертинами (векторами $\psi'2$,$\psi''2$) у кожній половині. Якщо хоча б одна з умов $\varphi'2 \le \psi'2$ або $\varphi''2 \le \psi''2$ порушена, то функція не монотонна, йдемо на кінець.

Кроки 3-n. Аналогічно порівнюємо восьмі, шістнадцяті частини, і так далі. Якщо жодна з умов, що перевіряються, не порушена, то функція монотонна.

Розглянемо дві булеві функції.

x	*y*	*z*	**f(x,y,z)**	**g(x,y,z)**	
0	*0*	*0*	**0**	**0**	
0	*0*	*1*	**0**	**1**	φ'_2
0	*1*	*0*	**0**	**1**	ψ'_2
0	*1*	*1*	**1**	**0**	φ_1
1	*0*	*0*	**0**	**0**	ψ_1
1	*0*	*1*	**1**	**1**	φ''_2
1	*1*	*0*	**1**	**1**	ψ''_2
1	*1*	*1*	**1**	**1**	

Перевіримо на монотонність першу функцію. Порівнюємо половини стовпця значень: 0001≤0111. Порівнюємо четвертини: 00≤01, 01≤11. Порівнюємо восьмі частини: 0≤0, 0≤1 та 0≤1, 1≤1. Отже, функція **f(x,y,z) –** монотонна.

Перевіримо на монотонність функцію **g(x,y,z)**. Порівнюємо половини стовпця значень: 0110≤0111. Порівнюємо четвертини: 01≤10, 01≤11. Порівнюємо восьмі частини: 0≤1 але на другій позиції у першої четвертини 1, а у другій 0 – властивість ≤ не виконується, функція g(x,y,z) не монотонна.

Визначення 1.18. Булева функція називається *функцією, що зберігає нуль (*T_0*)*, якщо на нульовому наборі вона дорівнює нулю, тобто $f(0,\ldots,0) = 0$.

Визначення 1.19. Булева функція називається *функцією, що зберігає одиницю (*T_1*)*, якщо на одиничному наборі вона дорівнює одиниці, тобто $f(1,\ldots,1) = 1$.

Властивості основних функцій наведено в табл.1.6.

Таблиця 1.6.

		КПЖ	T_0	T_1	M	S	L
f_0 –	константа 0	0	+	-	+	-	+
f_1 –	$x \wedge y$ – кон'юнкція	xy	+	+	+	-	-
f_3 –	тотожна функція $f(x,y) = x$	x	+	+	+	+	+
f_6 –	$x \oplus y$ – сума за *mod* 2 (операція Жегалкіна)	$x \oplus y$	+	-	-	-	+
f_7 –	$x \vee y$ – диз'юнкція	$xy \oplus x \oplus y$	+	+	+	-	-
f_8 –	$x \downarrow y$ – стрілка Пірса	$xy \oplus x \oplus y \oplus 1$	-	-	-	-	-
f_9 –	$x \equiv y$ (або $x \sim y$) – еквівалентність	$x \oplus y \oplus 1$	-	+	-	-	+
f_{12} –	заперечення $f(x) = \neg x$	$x \oplus 1$	-	-	-	+	+
f_{13} –	$x \rightarrow y$ – імплікація	$xy \oplus x \oplus 1$	-	+	-	-	-
f_{14} –	$x \mid y$ – штрих Шеффера	$xy \oplus 1$	-	-	-	-	-
f_{15} –	константа 1	1	-	+	+	-	+

Завдання 1.2.1. Визначити властивості булевих функцій.

Приклад. $(x \rightarrow (y \vee z)) \rightarrow xz$.

Таблиця істинності.

x	***Y***	***Z***	***f*(*x*, *y*, *z*)**	***конституенти***
0	0	0	0	$x \vee y \vee z$
0	0	1	0	$x \vee y \vee \neg z$
0	1	0	0	$x \vee \neg y \vee z$
0	1	1	0	$x \vee \neg y \vee \neg z$
1	0	0	1	$x \neg y \neg z$
1	0	1	1	$x \neg y z$
1	1	0	0	$\neg x \vee \neg y \vee z$
1	1	1	1	xyz

ДДНФ: $f(x, y, z) = x\neg y\neg z \vee xyz \vee x\neg yz$.

Перевіримо властивості функції.

1. Функція зберігає нуль, так як $f(0,0,0) = 0$.

2. Функція зберігає одиницю, так як $f(1,1,1) = 1$.

3. Самодвоїстість. За визначенням, функція є самодвоїстою, якщо $f(\neg x, \neg y, \neg z) = \neg f(x, y, z)$, тобто на протилежних наборах функція приймає

протилежні значення. Набір 001 є протилежним до набору 110, але $f(0,0,1) = 0$ і $f(1,1,0) = 0$, отже, функція несамодвоїста. Взагалі, самодвоїста функція має однакову кількість 0 та 1 в таблиці істинності.

4. Монотонність. Функція монотонна тоді, коли на зростаючих порівняних наборах вона не спадає.

Набір 100 передує набору 110, але $f(1,0,0) = 1 > f(1,1,0) = 0$, отже, функція немонотонна.

5. Лінійність. Щоб перевірити, чи є функція лінійною, побудуємо поліном Жегалкіна. Для цього у ДДНФ даної функції замінимо усі символи $\vee$ на $\oplus$ и виразимо заперечення як $\neg x = x \oplus 1$. Отримаємо поліном Жегалкіна:

$f(x, y, z) = x\neg y\neg z \vee xyz \vee x\neg yz = (x(y \oplus 1)(z \oplus 1)) \oplus (xyz) \oplus (x\,(y \oplus 1)\,z)$.

Приведемо його до канонічного вигляду, використовуючи закон дистрибутивності: $x(y \oplus z) = xy \oplus xz$ і тотожності: $x \oplus x = 0$, $x \oplus 0 = x$. Тоді

$f(x, y, z) = (x(y \oplus 1)(z \oplus 1)) \oplus (xyz) \oplus (xz(y \oplus 1)) = ((xy \oplus x)(z \oplus 1)) \oplus xyz \oplus \oplus (xyz \oplus xz) = xyz \oplus xy \oplus xz \oplus x \oplus xyz \oplus xyz \oplus xz = xyz \oplus xy \oplus x.$

Функція нелінійна, так як у канонічному поліномі Жегалкіна є нелінійні елементи: xyz та xy. Даний спосіб побудови канонічного поліному Жегалкіна показує, що непарної кількості одиниць у таблиці істинності булева функція не буде лінійною, бо завжди залишиться елемент xyz.

Отримати поліном Жегалкіна можна й алгебраїчно через суперпозицію основних функцій і їх уявлення канонічним поліномом Жегалкіна:

$(x \to y = xy \oplus x \oplus 1;\ \ x \vee y = xy \oplus x \oplus y).$

$(x \to (y \vee z)) \to xz = (x \to (y \vee z))\, xz \oplus (x \to (y \vee z)) \oplus 1 =$

$= (x(yz \oplus y \oplus z) \oplus x \oplus 1)xz \oplus (x(yz \oplus y \oplus z) \oplus x \oplus 1) \oplus 1 =$

$= (xyz \oplus xy \oplus xz \oplus x \oplus 1)xz \oplus xyz \oplus xy \oplus xz \oplus x \oplus 1 \oplus 1 = xz \oplus xyz \oplus xy \oplus xz \oplus x =$

$= xyz \oplus xy \oplus x.$

Або за спрощеною формулою:

$x\neg y \vee x\,z = x(y \oplus 1)x\,z \oplus x(y \oplus 1) \oplus x\,z = xyz \oplus xz \oplus xy \oplus x \oplus xz = xyz \oplus xy \oplus x.$

1) $xy \vee \neg x\,(y \vee xz)\,\neg(x\,\neg y \vee yz)$

2) $x\,(xy \vee \neg(yz) \vee \neg(y \vee \neg zt))$

3) $x\neg(\neg yz) \vee \neg(x \vee z)$

4) $\neg(\neg(\neg x \vee \neg yz) \vee \neg x \vee \neg yz) \vee (\neg x \vee \neg yz)(\neg z \vee y)$

5) $(xy \equiv z) \rightarrow y$

6) f(x,y,z)=1 на наборах 1,2,4,6

7) f(x,y,z)=0 на наборах 1,4,5

8) $(x \oplus y) \rightarrow z$

9) $(y \vee z) \equiv \neg x$

10) $(x \rightarrow y) \oplus (y \rightarrow z) \oplus (z \rightarrow x)$

11) $(x \equiv \neg y) \vee (y \rightarrow xz) \vee (x\neg y\neg z)$

12) $(x \rightarrow (x \rightarrow y)) \rightarrow z$

13) $(x \vee \neg yz)(xz \rightarrow z)$

14) $((y \vee z) \equiv \neg x)(xz \rightarrow \neg y)$

15) $(x \vee y\neg z)(x \vee z)$

Завдання 1.2.2. Побудувати поліном Жегалкіна і визначити властивості булевих функцій із завдання 1.1.2.

1.3. Функціональна повнота систем булевих функцій

Теоретичні відомості

Замкнуті класи булевих функцій

Нехай функція f(x) представлена формулою F(x), а функція g представлена формулою G. Тоді формула F(G), яка виходить шляхом заміни кожного входження змінної x в формулі F(x) на формулу G, представляє суперпозицію булевих функцій f (g).

Визначення 1.20. Клас функцій називається *функціонально замкнутим*, якщо суперпозиція цих функцій належить даному класу.

Теорема 1.8. Суперпозиція лінійних функцій є функція лінійна.

Теорема 1.9. Суперпозиція монотонних функцій є функція монотонна. Отже, клас монотонних функцій є функціонально замкнутим.

Теорема 1.10. Клас функцій, що зберігають нуль, є *функціонально замкнутим.*

Теорема 1.11. Клас функцій, що зберігають одиницю, є функціонально замкнутим.

Теорема 1.12. Клас самодвоїстих функцій є функціонально замкнутим.

Визначення 1.21. Система булевих функцій називається *функціонально повною,* якщо будь-яка булева функція може бути представлена як суперпозиція функцій з цієї системи.

Позначимо: T_0 – клас функцій, що зберігають 0; T_1 – клас функцій, що зберігають 1; S – клас самодвоїстих функцій; M – клас монотонних функцій; L – клас лінійних функцій.

Теорема 1.13 (**Поста**). Для того, щоб система функцій була повна, необхідно і достатньо, щоб вона містила хоча б одну немонотонну, хоча б одну нелінійну, хоча б одну несамодвоїсту, хоча б одну, що не зберігає нуль, і хоча б одну, не зберігаючу одиницю, функцію.

Приклад. Системи функцій $\{\vee, \wedge, \neg\}$, $\{\oplus, \wedge, 0, 1\}$, $\{\rightarrow, \neg\}$ являються функціонально повними. Для прикладу перевіримо повноту системи функцій $\{\neg, \rightarrow\}$. Для досліджуваної системи складемо таблицю Поста – критерійну таблицю (табл.1.7). Якщо функція входить у функціонально замкнутий клас, то в таблиці Поста у відповідній чарунці ставиться знак ‘+’, інакше – знак ‘–’.

Таблиця 1.7

	T_0	T_1	S	M	L
$\neg$	–	–	+	–	+
$\rightarrow$	–	+	–	–	–

Система функцій $\{\neg, \rightarrow\}$ повна, так як в кожному стовпці таблиці Поста є хоч би один знак ‘–’.

Повну систему називають неприводимою або базисом, якщо з неї не можна виключити жодної функції без того, щоб система не втратила властивості повноти.

Теорема 1.14. Будь-яка неприводима функціональна система булевих функцій складається з не більш ніж чотирьох функцій.

Функціонально повна система називається незалежною, якщо жодна з функцій системи не може бути суперпозицією інших функцій цієї системи. Базис – незалежний. Наприклад система {⊕, ∧, 0, 1} є повною, але не є незалежною. Базисом буде система {⊕, ∧, 1}, так як $0=1\oplus 1$ (табл.1.8).

Таблиця 1.8

	T_0	T_1	S	M	L
⊕	+	–	–	–	+
∧	+	+	–	+	–
1	–	+	–	+	+
0	+	–	–	+	+

Тезу про повноту можна узагальнити. Якщо усі функції функціонально повної системи можна виразити через супрепозицію функцій іншої системи, то вона так само буде функціонально повною. Цей метод називається методом зведення. Наприклад, система алгебри Жегалкіна {⊕, ∧, 1} є функціонально повною. Система {≡, ∨, 0} також буде функціонально повною. Дійсно, можно виразити $x\oplus y = (x \equiv y) \equiv 0$, $x \wedge y = (x \vee y) \equiv (x \equiv y)$, $1 = 0 \equiv 0$**.** Та переконатися**,** побудувавши критеріальну таблицю (табл.1.9):

Таблиця 1.9

	T_0	T_1	S	M	L
≡	–	+	–	–	+
∨	+	+	–	+	–
0	+	–	–	+	+

Завдання 1.3.1.

1) Довести, чи можна виразити за допомогою суперпозицій:

a) ¬ через &, V, →, та ≡;

b) → через & і V;

c) ¬ через + і 1;

d) ¬ через → і 0;

e) ¬ через → і 1;

2) Виразити за допомогою суперпозицій, попередньо перевірив систему функцій на повноту:

a) → через ¬ і &;

b) → через ¬ і V;

c) & через ¬ і →;

d) V через ¬ і →;

e) ¬ , & та V через {→, 0};

f) & через {+, 1, V} .

Завдання 1.3.2. Дослідити системи функцій на повноту та знайти базис:

	Функції системи I	**Функції системи II**	**Функції системи III**
1	$x \to (x \to z)$	$x \to (y \to z)$	$(x \to y) \to z$
2	$yx \vee \neg yz$	$xy \oplus xz \oplus yz$	xyVxzVyz
3	$(x \oplus y) \oplus z$	$(x \equiv y) \oplus z$	$(x \equiv y) \equiv z$
4	$\neg y \vee yz$	$\neg x \neg y \vee \neg y \neg z \vee \neg x \neg z$	$y \vee yz$
5	$x \neg y \vee \neg xz$		

Завдання 1.3.3. Виразити основні функції булевої алгебри {¬, &, V, 0, 1} через функціонально повну систему, що складається з однієї функції {|} або {↓}.

1.4. Мінімізація булевих функцій

<u>Теоретичні відомості</u>

Визначення 1.22. *Імплікантою* булевої функції $f(x_1, ..., x_n)$ називається така булева функція $g(x_1, ..., x_n)$, яка дорівнює одиниці на деяких з тих наборів, на яких дорівнює одиниці дана функція, і дорівнює нулю на інших наборах (із визначення випливає, що функція g – *імпліканта* f, якщо $g \to f \equiv 1$.) Говориться, що імпліканта g покриває своїми одиницями деякі одиниці даної функції f.

Якщо функція задана у вигляді ДНФ, тобто у вигляді диз'юнкції елементарних кон'юнкцій, то кожний кон'юнкт являється імплікантою даної функції. Довжина деяких елементарних кон'юнкцій може бути зменшена за допомогою еквівалентних перетворень. Для цього застосовуються такі співвідношення:

(1.26) *закон неповного склеювання*:

$xy \vee \neg xy = y$;

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

(1.27) *закон повного склеювання*:

$xy \vee \neg xz = xy \vee \neg xz \vee yz$;

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

(1.28) *закони поглинання*:

$x \vee xy = x$;

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

(1.29) $x \vee\neg xy = x \vee y$;

$x (\neg x \vee y) = xy$.

Доведення законів повного склеювання:

$\boldsymbol{xy \vee \neg xz \vee yz} =$

$=xy \vee \neg xz \vee yz\,(x\vee \neg x)=$

$=xy \vee \neg xz \vee xyz\vee \neg xyz=$

$xy \vee xyz\vee \neg xz \vee \neg xyz =$

$=\boldsymbol{xy \vee \neg xz}.$

$\boldsymbol{xy \vee \neg xz} =$

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

$= 1(xy\vee x\neg x\vee zy\vee z\neg x)=$

$= xy\vee 0\vee zy\vee z\neg x=$

$\boldsymbol{xy \vee \neg xz \vee yz}.$

Визначення 1.23. Елементарна кон'юнкція, яка є імплікантою функції f, але жодна її власна частина імплікантою цієї функції не являється, називается *простою імплікантою* даної функції.

Довжина простої імпліканти вже не може бути зменшена шляхом склеювання її з іншими імплікантами даної функції.

Визначення 1.24. Диз'юнкція всіх простих імплікант булевої функції називаєтся *скороченою диз'юнктивною нормальною формою* булевої функції.

Приклад. Скоротимо формулу: $F(x, y, z) = xyz \vee x\neg yz \vee xy\neg z \vee \neg xy\neg z$. Склеїмо конституенти: $xyz \vee x\neg yz = xz$, $xy\neg z \vee \neg xy\neg z = y\neg z$, $xyz\vee xy\neg z = xy$. Отримаємо скорочену форму: $F(x, y, z) = xz \vee y\neg z \vee xy$.

Множина усіх простих імплікант покриває усі одиниці булевої функції. Але представлення булевої функції у вигляді скороченої ДНФ ще не являється найбільш економним. Серед множини простих імплікант можуть опинитись зайві, тобто такі, що покривають одиниці функції, вже покриті іншими імплікантами. Видаляючи зайві імпліканти, можна отримати *тупикову* диз'юнктивну нормальну форму. Тупикова ДНФ, яка містить найменшу кількість імплікант та змінних, називается *мінімальною* ДНФ.

В залежності від поставленої мети мінімізації, вибір мінімальної ДНФ можливий за двома критеріями:

- вміст найменшої кількості входження букв - загальна кількість букв у всіх елементарних кон'юнкціях, так звана ціна покриття C^{α};
- сума кількості входжень букв та кількості елементарних добутків являється найменшою – ціна покриття C^{β}.

В наведених прикладах при мінімізації застосовувався алгебраїчний метод Блейка, оснований на законах склеювания та поглинання.

Метод Квайна отримання скороченої диз'юнктивної нормальної форми

Теорема 1.15 (Квайна). Якщо в досконалій диз'юнктивній нормальній формі булевої функції зробити всі операції неповного склеювання (1.26), а потім всі операції поглинання (1.27, 1.28), то в результаті буде отримана скорочена ДНФ даної функції.

Для перетворення довільної ДНФ до виду ДДНФ необхідно застосувати до елементарних кон'юнкцій, що містять не всі змінні, операцію розгортання, наприклад: $xy(z \vee \neg z) = xyz \vee xy\neg z$, де z – недостатня змінна.

Приклад. Булеву функцію задано у вигляді: $f(x,y, z) = \neg xyz \vee x\neg y\neg z \vee \neg yz$. Приведемо формулу до ДДНФ:

$f(x, y, z) = \neg xyz \vee x\neg y\neg z \vee\neg yz(x \vee\neg x) = \neg xyz \vee x\neg y\neg z \vee x\neg yz \vee\neg x\neg yz$.

Випишемо всі конституенти одиниці і зробимо всі операції неповного склеювання:

* $\neg xyz$	$\neg yz$
* $x\neg yz$	$x\neg y$
* $x\neg y\neg z$	$\neg xz$
* $\neg x\neg yz$	

Відзначимо всі кон'юнкції, що склеюються символом «*». Вони поглинаються імплікантами, отриманими в результаті склеювання. Невідмічені кон'юнкції нічим не поглинаються і є простими імплікантами.

Скорочена ДНФ даної функції: $f(x,y,z)= \neg yz \vee x\neg y \vee \neg xz$. Для знаходження мінімальної ДНФ складемо імплікантну матрицю (таблиця 1.10), у якій по рядкам записуємо імпліканти, по стовпцям – конституенти одиниці.

Таблиця 1.10

	$\neg xyz$	$x \neg y \neg$	$x \neg yz$	$\neg x \neg yz$
$\neg yz$			×	×
$x\neg y$ *		×	×	
$\neg xz$ *	×			×
	*	*	+	+

У клітинах таблиці хрестиками відзначаємо імпліканти, що покривають одиниці даної функції. Внизу таблиці символом «*» відзначаємо ті стовпці, в яких стоїть тільки один хрестик, відповідні їм імпліканти також відзначаємо символом «*» – вони є обов'язковими. Відзначаємо також символом «+» ті стовпці, які покриваються обов'язковими імплікантами. Якщо всі стовпчики відзначені, то отриманий набір обов'язкових імплікант становить мінімальну

ДНФ. Якщо частина стовпців залишається непокритою, з решти імплікант вибирається найменше число найбільш коротких імплікант так, щоб всі стовпці були покриті.

У нашому прикладі мінімальна ДНФ: $f(x, y, z) = x\neg y \vee \neg xz$.

Метод Квайна містить незручності, пов'язані з необхідністю попарного порівняння імплікант на етапі визначення простих імплікант. Із зростанням кількості мінтерм в СДНФ збільшується кількість цих порівнянь, що характеризується факторіальною функцією.

У 1956 році Маккласки запропонував модернізацію першого етапу методу Квайна. Ідея полягає в тому, що, якщо записати мінтерми у вигляді двійкових чисел і розбити їх на непересічні групи за кількістю 1 в двійковому числі, то попарно порівнювати потрібно тільки елементи сусідніх груп. Для попереднього прикладу ми отримаємо 001, 010, 011, 100, 101, 110 і розіб'ємо на групи (001, 010, 100) і (011, 101, 110). В результаті порівняння ми отримаємо 0-1, -01, 01-, -10, 10, 1-0, що і відповідає $\neg$xz, $\neg$yz, $\neg$xy, y$\neg$z, x$\neg$y, x$\neg$z.

На підставі отриманого будується імплікантна матриця.

Мінімізація булевих функцій за допомогою діаграм Вейча

Булеві функції можуть бути представлені графічно у вигляді діаграм Вейча або карт Карно – вони відрізняються лише позначеннями клітинок таблиці.

Діаграма Вейча представляє собою прямокутник, для однієї змінної – поділений навпіл (одна половина відповіє змінній, друга - її запереченню). При введені кожної нової змінної відбувається повторне ділення діаграми навпіл. Розглянемо діаграми Вейча для функцій від трьох і чотирьох змінних (рис. 1.2).

Кожна клітинка діаграми відповідає одному набору змінних, номер клітинки – це двійковий код набору. При заданні булевої функції у відповідну до номера набору клітинку записується одиниця, і, таким чином, кожна клітинка діаграми з 1 представляє собою одну конституенту одиниці.

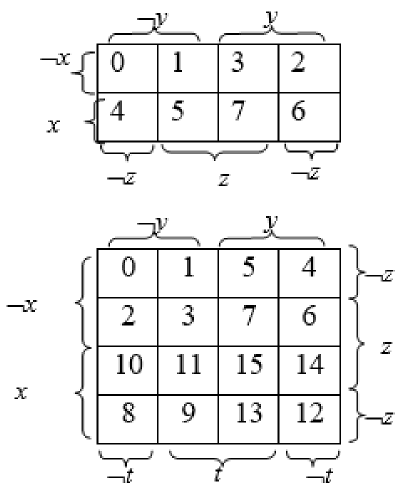


Рис. 1.2. Діаграми Вейча

Приклад. Нехай функція $f(x, y, z, t) = 1$ на наборах 0, 1, 2, 3, 6, 7, 8, 15. Діаграма Вейча для заданої функції зображена на рис. 1.3. Одиниці функції, записані в сусідніх клітинках, відрізняються значенням тільки однієї змінної, внаслідок, вони склеюються по цій змінній і утворюють імпліканту. У цьому випадку кажуть, що імпліканта *покриває* відповідні одиниці булевої функції. Наприклад, дві одиниці на наборах 7 і 15, покриваються імплікантою yzt, чотири одиниці на наборах 2,3,7,6 – імплікантою $\neg xz$. При цьому сусідніми вважаються також клітинки, записані вздовж лівого та правого краю діаграми (відрізняються значенням y) і вздовж верхнього і нижнього краю (відрізняються значенням x). Тобто, діаграму можно вважати ніби згорнутою в тор (а для трьох змінних – у циліндр).

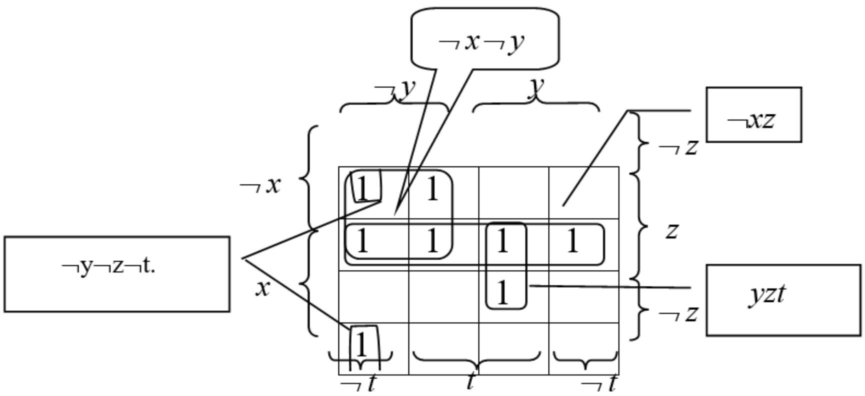


Рис. 1.3. Мінімальна ДНФ

При мінімізації булевої функції на діаграмі Вейча спочатку знаходять покриття, які містять максимальне число одиниць (8, 4, 2), а потім покриття, яке покриває решту одинить таким чином, щоб вони також були максимальні по величині, і при видаленні цього покриття хоча б одна одиниця функції залишилася непокритою. При цьому деякі одиниці можуть бути покриті неоднаразово. Для функції, зображеної на рис. 1.3, мінімальна ДНФ: $\neg x\neg y \vee \neg xz \vee yzt \vee \neg y\neg z\neg t$.

Мінімальна КНФ будується двоїсто за діаграмою Вейча, заповненої нулями в порожніх клітинках (для збереження номерів наборів при підписуванні частин діаграми змінні та заперечення міняються місцями, тому що при побудові ДКНФ заперечення ставиться над 1 в значеннях набору, а в ДДНФ – над 0). Для даної функції мінімальна КНФ: $(\neg y\vee z)(\neg x\vee\neg z\vee t)(\neg x\vee y\vee\neg t)$.

Карти Карно також служать для графічного представлення булевих функцій і відрізняються від діаграм Вейча тільки формою представлення змінних. Рядки і стовпці карти Карно відмічені наборами змінних, відповідних даному рядку або стовпцю.

Завдання 1.4.1. Визначити мінімальні ДНФ і КНФ булевих функцій, що задані.

Приклад. Булева функція $f(x, y, z, t) = 1$ на наборах 0, 3, 4, 6, 8, 9, 10, 11, 14, 15. Випишемо набори, на яких функція приймає значення 1 і проведемо склеювання:

№	двійковий код	елемент ДДНФ		11-15	xzt
0	0000	¬x¬y¬z¬t	перша	0-4	**¬x¬z¬t**
3	0011	¬x¬yzt		0-8	**¬y¬z¬t**
4	0100	¬xy¬z¬t	ітерація	8-10	x¬y¬t
6	0110	¬xyz¬t		9-11	x¬yt
8	1000	x¬y¬z¬t	склеювання	3-11	**¬yzt**
9	1001	x¬y¬zt		6-14	**yz¬t**
10	1010	x¬yz¬t	(номери	4-6	**¬xy¬t**
11	1011	x¬yzt	наборів -	10-14	xz¬t
14	1110	xyz¬t	результат)	10-11	x¬yz
15	1111	xyzt		8-9	x¬y¬z
				14-15	xyz

Проведемо другий крок скеювання:

11-15	xzt	набори	**результат**
0-4	**¬x¬z¬t**		
0-8	**¬y¬z¬t**	11-15:10-14	**xz**
8-10	x¬y¬t		
9-11	x¬yt	10-11:14-15	xz
3-11	**¬yzt**		
6-14	**yz¬t**	8-9:10-11	**x¬y**
4-6	**¬xy¬t**		
10-14	xz¬t	8-10:9-11	x¬y
10-11	x¬yz		
8-9	x¬y¬z		
14-15	xyz		

В результаті були отримані імпліканти:

xz	11-15-10-14
¬x¬z¬t	0-4
¬y¬z¬t	0-8
x¬y	8-10-9-11
¬yzt	3-11
yz¬t	6-14
¬xy¬t	4-6

Для знаходження мінімальної ДНФ складемо імплікантну матрицю

	0	**3**	**4**	**6**	**8**	**9**	**10**	**11**	**14**	**15**
¬x¬z¬t	+		+							
¬y¬z¬t	+				+					
¬yzt		+						+		
yz¬t				+					+	
¬xy¬t			+	+						
x¬y					+	+	+	+		
xz							+	+	+	+

До мінімальної ДНФ включаємо імпліканти, що оптимально покривають набори 3, 8, 9, 10, 11, 14, 15: **¬yzt, x¬y, xz.**

Непокриті набори 0, 4 та 6 можна покрити в різний спосіб: ¬x¬z¬t, ¬xy¬t або ¬xy¬t, ¬y¬z¬t або ¬x¬z¬ t, yz¬t. Але ціна покриття при кожному варіанті однакова. Тому, зупинимось на варіанті мінімальної ДНФ:

¬yzt **v** x¬y **v** xz **v** ¬x¬z¬ t **v** yz¬t.

Розглянемо модифікацію МакКласкі першого етапу методу Квайна.

Розміщуємо номери наборів, на яких булева функція дорівнює 1 рядкам за кількістю одиниць у двійковому запису набору.

I крок

0000

0100 1000

0011 0110 1001 1010

1011 1110

1111

Результат склеювання наборів в сусідніх рядках:

0_00 _000

01_0 ***100_*** ***10_0***

_011 _110 ***10_1*** ***101_*** ***1_10***

1_11 ***111_***

Проведемо ще можливі склеювання:

II крок - результат

10_ _ 10_ _

1_1_ 1_1_

Тепер виписуємо скорочення, отриманні в кінці II кроку і ті результати I кроку, що не підлягали склеюванню на наступному кроці:

0_00 _000 01_0 _011 _110 10_ _ 1_1_

Їм відповідають наступні імліканти (_ - немає змінної, 1 – змінна, 0 – заперечення відповідної змінної):

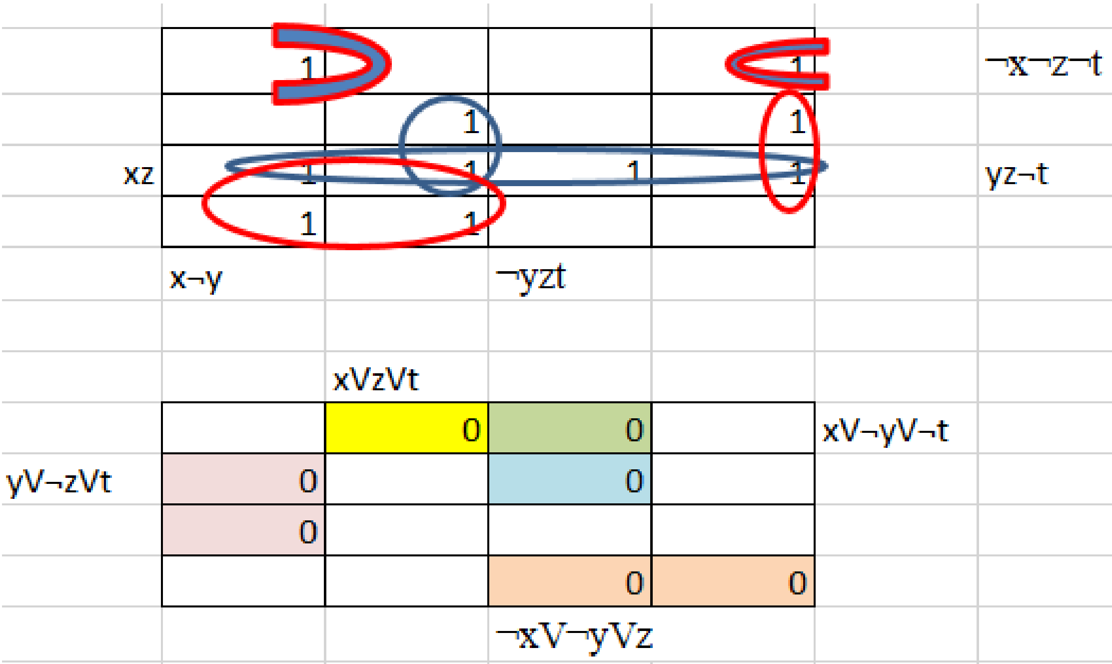
¬x¬z¬t ¬y¬z¬t ¬xy¬t ¬yzt yz¬t x¬y

xz

Етап побудови імплікантної матриці і вибір мінімальної ДНФ в методі МакКласкі не відрізняється від методу Квайна.

Варіант мінімальної ДНФ: ¬yzt **v** x¬y **v** xz **v** ¬x¬z¬ t **v** yz¬t.

Тепер розглянемо візуальний метод – діаграми Вейча (друга діаграма відповідає мінімізації КНФ).



До перших п'яти прикладів показано розв'язок.

1) f(x, y, z, t) = 1 на наборах 0, 3, 5, 7, 8, 9, 10, 11, 12, 14;
2) f(x, y, z, t) = 1 на наборах 1, 4, 5, 7, 9, 10, 12, 14;
3) f(x, y, z, t) = 1 на наборах 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 15;
4) f(x, y, z, t) = 1 на наборах 1, 3, 6, 7, 8, 9, 10, 13, 14, 15;
5) f(x, y, z, t) = 1 на наборах 0, 3, 5, 7, 8,10,11,12,14;
6) *f(x, y, z, t) = 1 на наборах 2, 4, 6, 7, 10, 12, 14, 15;*
7) *f(x, y, z, t) = 1 на наборах 0, 3, 4, 6, 8, 9, 10, 11, 13, 15;*
8) *f(x, y, z, t) = 1 на наборах 1, 2, 4, 6, 7, 9, 10, 13, 14;*
9) *f(x, y, z, t) = 1 на наборах 1, 3, 5, 7, 8, 10, 12, 14, 15;*
10) *f(x, y, z, t) = 1 на наборах 5, 7, 8, 9, 10, 11, 12, 14, 15;*
11) *f(x, y, z, t) = 1 на наборах 1, 8, 10, 11, 13, 14, 15;*
12) *f(x, y, z, t) = 1 на наборах 0, 1, 2, 3, 5, 6, 9, 10, 12, 14, 15;*
13) *f(x, y, z, t) = 1 на наборах 0, 3, 5, 7, 8, 9, 11, 12, 14, 15;*
14) *f(x, y, z, t) = 1 на наборах 2, 4, 6, 9, 11, 13, 15;*
15) *f(x, y, z, t) = 1 на наборах 0, 1, 2, 3, 8, 10, 11.*

Відповіді до розділу 1

1.1.1. Не еквівалентні формули: 1, 4, 5, 7. 9.

1.1.2.

№	спрощення	значення	КПЖ	T_0	T_1	S	M	L
1)	$\neg x\neg y\neg z$	10000000	$xyz\oplus xy\oplus xz\oplus$ $\oplus yz\oplus x\oplus y\oplus z\oplus 1$	-	-	-	-	-
2)	$\neg x\neg y\neg z\vee x\neg yz$	10000100	$xy\oplus yz\oplus x\oplus y\oplus z$ $\oplus 1$	-	-	-	-	-
3)	$xy \vee xz$	00000111	$xyz \oplus xy \oplus xz$	+	+	-	+	-
4)	$x \to z$	11110101	$xz \oplus x \oplus 1$	-	+	-	-	-
5)	$x\neg y \vee z$	01011101	$xyz \oplus xy \oplus xz \oplus$ $\oplus x \oplus z$	+	+	-	-	-
6)	$(x \equiv y) \equiv z$	01101001	$x \oplus y \oplus z$	+	+	+	-	+
7)	$\neg x\vee\neg y\vee z$	11111101	$xyz \oplus xy \oplus 1$	-	+	-	-	-
8)	$\neg x\vee y$	11110011	$xy \oplus x \oplus 1$	-	+	-	-	-
9)	xy	00000011	xy	+	+	-	+	-
10)	$x\vee y \vee z$	01111111	$xyz\oplus xy\oplus xz\oplus$ $\oplus yz\oplus x\oplus y\oplus z$	+	+	-	+	-
11)	x	00001111	x	+	+	+	+	+
12)	$xy \vee z$	01010111	$xyz \oplus xy \oplus z$	+	+	-	+	-
13)	$y \vee \neg z$	10111011	$yz \oplus z \oplus 1$	-	+	-	-	-
14)	z	01010101	z	+	+	+	+	+
15)	1	11111111	1	-	+	-	+	+

1.1.3. Не тавтологіями є формули 4, 6 і 10.

1.1.4. ***Підказка для формування відповіді:***

$(x\vee\neg yz)(xz \to z)$	01001111
$(x\vee y\neg z)(x\vee z)$	00001111
$(xy\equiv z) \to y$	01110111
$y\equiv(\neg xz)$	10011100
$(x\oplus y) \to z$	11010111
$x\neg y \vee\neg xz$	01011100
$xy \vee\neg yz$	01000111

1.2.1.

№	значення	КПЖ	T_0	T_1	S	M	L
1)	00100011	xyz+yz+y	+	+	-	-	-
2)	0000000011111111	x	+	+	+	+	+
3)	10101011	xyz+z+1	-	+	-	-	-
4)	10110000	xyz+xz+yz+x+z+1	-	-	-	-	-
5)	01110111	yz+y+z	+	+	-	+	-
6)	01101010	xy+x+y+z	+	-	-	-	-
7)	10110011	xyz+xy+zy+xz+x+z+1	-	+	-	-	-
8)	11010111	xz+yz+x+y+1	-	+	-	-	-
9)	01111000	yz+x+y+z	+	-	-	-	-
10)	10000001	xy+yz+xz+x+y+z+1	-	+	-	-	-
11)	11111101	xyz+xy+1	-	+	-	-	-
12)	01011101	xyz+xz+xy+x+z	+	+	-	-	-
13)	01001111	xyz+xz+yz+x+z	+	+	-	-	-
14)	01111000	yz+x+y+z	+	-	-	-	-
15)	00001111	x	+	+	+	+	+

1.2.2. Див. відповідь для завдання 1.1.2.

1.3.1.

1)

a) ні, всі $\boldsymbol{T_1}$;
b) ні, всі $\boldsymbol{T_0, M}$;
c) так, всі лінійні;
d) так, $\neg x = x \to 0$;
e) ні, обидві $\boldsymbol{T_1}$.

2)

a) $x \to y = \neg (x \& \neg y)$
b) $x \to y = \neg x V y$
c) $x \& y = \neg (x \to \neg y)$
d) $x V y = \neg x \to y$
e) $\neg x = x \to 0$, $x \& y = \neg (x \to \neg y) =$
$= \neg (x \to (y \to 0)) = (x \to (y \to 0)) \to 0$,
$x V y = \neg x \to y = (x \to 0) \to y$
f) $x \& y = \neg (\neg x V \neg y) = 1 + (1+x) V (1+y)$

1.3.2.

I) (1 або 4) та 5;

	Функції системи I	T_0	T_1	S	M	L	Таблиця іст.
1	$x\to(x\to z)$	-	+	-	-	-	11110101
2	$yx\vee\neg yz$	+	+	-	-	-	01000111
3	$(x\oplus y)\oplus z$	+	+	+	-	+	01101001
4	$\neg y\vee yz$	-	+	-	-	-	11011101
5	$x\neg y\vee\neg xz$	+	-	-	-	-	01011100

II) 1 та (3 або 4);

	Функції системи II	T_0	T_1	S	M	L	Таблиця іст.
1	$x\to(y\to z)$	-	+	-	-	-	11111101
2	$xy\oplus xz\oplus yz$	+	+	+	+	-	00010111
3	$(x\equiv y)\oplus z$	-	-	+	-	+	10010110
4	$\neg x\neg y\vee\neg y\neg z\vee\neg x\neg z$	-	-	+	-	-	11101000

II) не повна, базис не існує.

	Функції системи III	T_0	T_1	S	M	L	Таблиця іст.
1	$(x\to y)\to z$	+	+	-	-	-	01011101
2	$xyVxzVyz$	+	+	+	+	-	00010111
3	$(x\equiv y)\equiv z$	+	+	+	-	+	01101001
4	$y\vee yz$	+	+	+	+	+	00110011

1.3.3.

Для штриха Шеффера:

$\boldsymbol{\neg x=\neg(x\&x)=x|x}$

$x\&y=\neg\neg(x\&y)=\neg(x|y)=(x|y)\,|\,(x|y);$

$xVy=\neg\neg(xVy)=\neg(\neg x\&\neg y)=(\neg x)|(\neg y)=(x|x)|(y|y);$

$0=x\&\neg x=\neg\neg(x\&\neg x)=\underline{\neg(x|(\neg x))}=\neg(x|(x|x))=(x|(x|x))|(x|(x|x));$

$1=xV\neg x=(x|x)\,|\,(\neg x\,|\,\neg x)=(x|x)\,|((x|x)\,|\,(x|x))$ АЛЕ

$1=\neg 0=\neg\,\underline{\neg(x|(\neg x))}=x|(x|x)$

Для стрілки Пірса:

¬x= ¬ (xVx)=x↓x

xVy = ¬ ¬ (xVy) = ¬ (x↓y) = (x↓y) ↓ (x↓y);

x&y = ¬ ¬ (x&y) = ¬ (¬xV¬y) = (¬x) ↓ (¬y)= (x↓x) ↓ (y↓y);

1 = xV¬x = ¬¬ (xV¬x)= ¬(x↓(¬x)) = ¬ (x↓(x↓x)) = (x↓(x↓x)) ↓ (x↓(x↓x));

0 = x&¬x = (x↓x) | (¬x↓¬x) = (x↓x) ↓ ((x↓x) ↓ (x↓x))

АЛЕ 0= ¬1 = ¬ ¬(x↓ (¬x)) = x↓ (x↓x)

1.4.1.

1) f(x, y, z, t)=1 на наборах 0, 3, 5, 7, 8, 9, 10, 11, 12, 14.

Диз'юнктивна форма

	¬y	¬y	y	y	
¬x	**1** 0		**1** 5		¬z
¬x		**1** 3	**1** 7		z
x	**1** 10	**1** 11		**1** 14	z
x	**1** 8	**1** 9		**1** 12	¬z
	¬t	t	t	¬t	

x¬y	8-9-10-11
x¬t	8-10-12-14
¬xyt	5-7
¬y¬z¬t	0-8
¬yzt	3-11 або 3-7
¬xzt	

Всі можливі сусідні покриття:

3-7	¬xzt
10-11	x¬yz
14-10	xz¬t
8-9	x¬y¬z
12-8	x¬z¬t
5-7	¬xyt
3-11	¬yzt
10-8	x¬y¬t
11-9	x¬yt
14-12	xy¬t
0-8	¬y¬z¬t

Відповідь: мінімальна ДНФ: **x¬yV x¬tV¬xytV¬y¬z¬tV¬*yzt;*** $C^{\alpha} = 13$; $C^{\beta} = 21$.

Метод Квайна (для диз'юнктивної форми)

f(x, y, z, t)=1 на наборах 0, 3, 5, 7, 8, 9, 10, 11, 12, 14.

0	0000	**¬x¬y¬z¬t**	*3-7*	*¬xzt*		
3	0011	**¬x¬yzt**	*10-11*	*x¬yz*		
5	0101	**¬xy¬zt**	*14-10*	*xz¬t*	8-9 + 10-11	**x¬y**
7	0111	**¬xyzt**	*8-9*	*x¬y¬z*		
8	1000	**x¬y¬z¬t**	*12-8*	*x¬z¬t*	10-8 + 14-12	**x¬t**
9	1001	**x¬y¬zt**	*5-7*	*¬xyt*		
10	1010	**x¬yz¬t**	*3-11*	*¬yzt*	10-8 + 11-9	**x¬y**
11	1011	**x¬yzt**	*10-8*	*x¬y¬t*		
12	1100	**xy¬z¬t**	*11-9*	*x¬yt*	12-8 + 14-10	**x¬t**
14	1110	**xyz¬t**	*14-12*	*xy¬t*		
			0-8	*¬y¬z¬t*		

Побудуємо імплікативну матрицю

		0	3	5	7	8	9	10	11	12	14
3-7	¬xzt		+		+						
5-7	¬xyt			+	+						
3-11	¬yzt		+						+		
0-8	¬y¬z¬t	+				+					
8-9-10-11	x¬y					+	+	+	+		
8-10-12-14	x¬t					+		+		+	+
		I		**I**	5	0	**I**	9	9	**I**	**I**

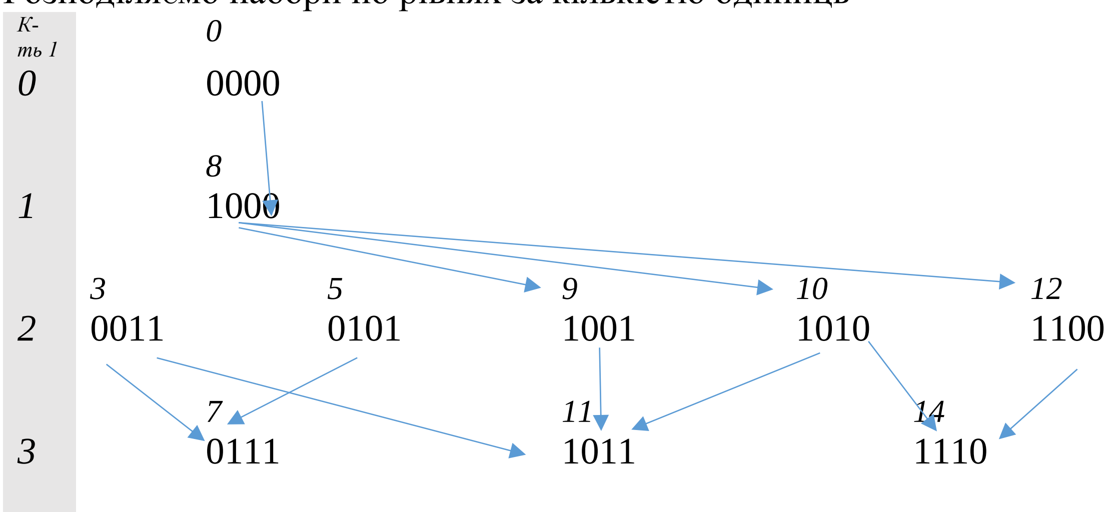
Мінімальна ДНФ: **x¬yV x¬tV¬xytV¬y¬z¬tV¬yzt**

Модифікація Маккласкі
(для першого етапу метода Квайна – отримання скорочених форм)

f(x, y, z, t)=1 на наборах 0, 3, 5, 7, 8, 9, 10, 11, 12, 14

0	**3**	**5**	**7**	**8**	**9**	**10**	**11**	**12**	**14**
0000	**0011**	**0101**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1110**

Розподіляємо набори по рівнях за кількістю одиниць



A) 0-8 **_000** (¬y¬z¬t)

B) 8-9 **100_** (x¬y¬z) 8-10 **10_0** (x¬y¬t) 8-12 **1_00** (x¬z¬t)

C) 3-7 **0_11** (¬xzt) 3-11 **_011** (¬yzt) 5-7 **01_1** (¬xyt) 9-11 **10_1** (x¬yt)

10-11 **101_** (x¬yz) 10-14 **1_10** (xz¬t) 12-14 **11_0** (xy¬t)

Друга ітерація – рядок А містить елементи без одиниць, В – з однією одиницею, С – з двома. Треба враховувати розташування «_» - повинно співпадати для виконання операції склеювання.

А та В не утворюють нові імпліканти.

В та С: 8-9 та 10-11 – **10_ _** (x¬y);

8-10 та 9-11 – **10_ _** (x¬y);

8-10 та 12-14 – **1_ _ 0** (x¬t);

8-12 та 10-14 – **1_ _ 0** (x¬t).

Збираємо останні імпліканти (крім тих, що червоні – вони використовувалися при другій ітерації) і переходимо до імплікативної матриці.

Далі все так, як в самому методі Квайна.

<u>Кон'юнктивна форма</u>

f(x, y, z, t)=1 на наборах 0, 3, 5, 7, 8, 9, 10, 11, 12, 14, відповідно

f(x, y, z, t)=0 на наборах 1, 2, 4, 6, 13, 15.

	y	y	¬y	¬y			
x	1 (0)	0 (1)	1 (5)	0 (4)	z	**xVyVzV¬t**	1
x	0 (2)	1 (3)	1 (7)	0 (6)	¬z	**xV¬zVt**	2-6
¬x	1 (10)	1 (11)	0 (15)	1 (14)	¬z	**¬xV¬yV¬t**	13-15
¬x	1 (8)	1 (9)	0 (13)	1 (12)	z	**xV¬yVt**	4-6
	t	¬t	¬t	t			

Відповідь: мінімальна КНФ:

(xVyVzV¬t)(xV¬zV¬t)(¬xV¬yV¬t)(xV¬yVt); $C^{\alpha} = 13$; $C^{\beta} = 22$.

2) f(x, y, z, t)=1 на наборах 1, 4, 5, 7, 9, 10, 12,14.

МАККЛАСКІ

1	4	5	7	9	10	12	14
0001	0100	0101	0111	1001	1010	1100	1110

0001 0100

0101 1001 1010 **1100**

0111 **1110**

Перша ітерація:

0_01 _001 010_ _100

01_1 1_10 **11_0**

Друга ітерація відсутня. Всі «одиниці» приймали участь у 1-ій ітерації.

Матриця:

		1	4	5	7	9	10	12	14
		0001	0100	0101	0111	1001	1010	1100	1110
¬x¬zt	0_01	+		+					
¬y¬zt	_001	+				+			
¬xy¬z	010_		+	+					
y¬z¬t	**_100**		+					+	
¬xyt	01_1			+	+				
xz¬t	1_10						+		+
xy¬t	11_0							+	+
		v		v	I	I	I		v

Залишилися непокритими два набори: 4 та 12, та незадіяними чотири імпліканти. Як раз одна з них покриває обидва набори - **y¬z¬t.**

Відповідь (мінімальна ДНФ): y¬z¬t V ¬y¬zt V ¬xyt V xz¬t.

Діаграма Вейча (синім окреслені «зайві» покриття):

	¬y	¬y	y	y			
¬x		1 1	1 5	1 4	¬z	y¬z¬t	4-12
¬x			1 7		z	¬y¬zt	1-9
x	1 10			1 14	z	¬xyt	5-7
x		1 9		1 12	¬z	xz¬t	10-14
	¬t	t	t	¬t			

Діаграма Вейча для КНФ:

	y	y	¬y	¬y			
x	**0** 0	1 1	1 5	1 4	z	**yVzVt**	1-8
x	**0** 2	**0** 3	1 7	**0** 6	¬z	**xV¬zVt**	2-6
¬x	1 10	**0** 11	**0** 15	1 14	¬z	**¬xV¬yV¬t**	13-15
¬x	**0** 8	1 9	**0** 13	1 12	z	**yV¬zV¬t**	3-11
	t	¬t	¬t	t			

- Запишіть мінімальну КНФ.
- Знайдіть три «зайвих» покриття на діаграмі для КНФ.
- Підрахуйте ціни покриття для мінімальних ДНФ та КНФ.
- Встановіть відповідність результатів, отриманих методом Квайна (Макккласкі) та за діаграмами Вейча.

3) $f(x, y, z, t) = 1$ на наборах 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 15;

КВАЙН

2	0010	¬x¬yz¬t		2-10	¬yz¬t			8-9:10-11	***x¬y***
4	0100	¬xy¬z¬t		***8-10***	***x¬y¬t***			8-10:9-11	***x¬y***

5	0101	¬xy¬zt		***9-11***	***x¬yt***	***8-9***	***x¬y¬z***	8-9:12-13	***x¬z***
7	0111	¬xyzt		***13-15***	***xyt***			8-12:9-13	***x¬z***
8	1000	x¬y¬z¬t		***7-15***	***yzt***	***11-15***	***xzt***	9-11:13-15	***xt***
9	1001	x¬y¬zt		***5-7***	***¬xyt***			9-13:11-15	***xt***
10	1010	x¬yz¬t		***5-13***	***y¬zt***	***9-13***	***x¬zt***	7-15:5-13	***yt***
11	1011	x¬yzt		***4-12***	***y¬z¬t***			5-7:13-15	***yt***
12	1100	xy¬z¬t		***4-5***	***¬xy¬z***	***12-13***	***xy¬z***	4-5:12-13	***y¬z***
13	1101	xy¬zt		***10-11***	***x¬yz***			4-12:5-13	***y¬z***
15	1111	xyzt		***8-12***	***x¬z¬t***				

Імплікантна матриця

	2	4	5	7	8	9	10	11	12	13	15
x¬y					+	+	+	+			
xt						+		+		+	+
x¬z					+	+			+	+	
yt			+	+						+	+
y¬z		+	+						+	+	
¬yz¬t	+						+				

Маккласкі

I

0010 0100 1000

0101 1001 1010 1100

0111 1011 1101

1111

_010 010_ _100 100_ 10_0 1_00

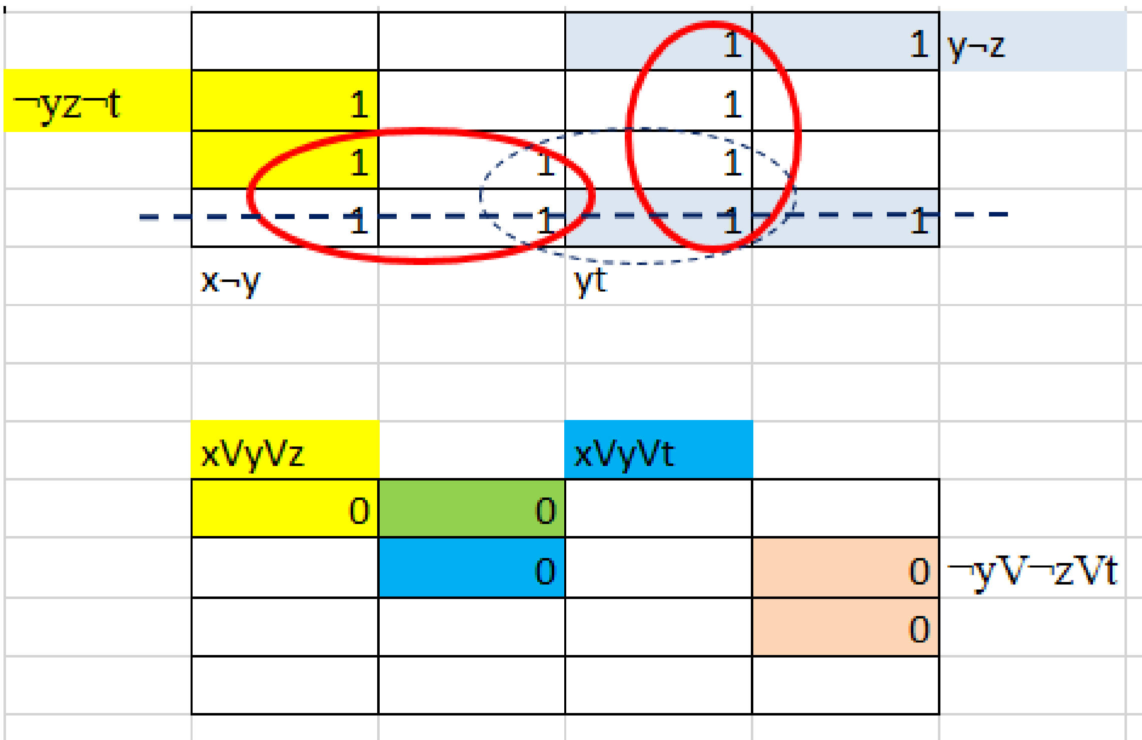
01_1 _101 10_1 1_01 101_ 110_

_111 1_11 11_1

II

_10_   _10_   10_ _   1_0_   10_ _   1_0_

_1_1   _1_1   1_ _1   1_ _1



4) *f*(*x*, *y*, *z*, *t*) = 1 на наборах 1, 3, 6, 7, 8, 9, 10, 13, 14, 15;

	1		
	1	1	1
1		1	1
1	1	1	

Мінімальна ДНФ

yz V x¬y¬t V ¬x¬yt V x¬zt

0		0	0
0			
	0		
			0

Мінімальна КНФ

(¬xVyV¬zV¬t)(xVyVt)

(xV¬yVz)(¬yVzVt)

6) $f(x, y, z, t) = 1$ на наборах 0, 3, 5, 7, 8,10,11,12,14;

1		1	
	1	1	
1	1		1
1			1

Мінімальна ДНФ

¬x¬t V ¬y¬z¬t V ¬yzt V ¬xyt

	0		0
0			0
		0	
	0	0	

Мінімальна КНФ

(¬xV¬zVt)(¬xV¬yV¬t)

(xV¬yVt)(yVzV¬t)

2. ТЕОРІЯ ВИСЛОВЛЮВАНЬ

2.1. Основні поняття

Теоретичні відомості

Логіка, вивчаючи правильні міркування, оперує поняттями істинності і хибності. Правильне міркування або висловлювання покладається істинним, неправильне – хибним; для формального апарату представлення знань нам досить двозначної логіки.

Методологічні основи формальної логіки полягають у виконанні декількох принципів.

Принцип тотожності. Істинність фактів, що лежать в основі висловлювань і міркувань, встановлюється на підставі відомих законів, спостережень. Якщо істинність будь-якого факту встановлено, то вона не піддається сумніву і не змінюється в процесі міркування. Це означає також, що один і той же термін використовується завжди в одному і тому ж сенсі.

Принцип несуперечності означає, що, стверджуючи що-небудь, не можна заперечувати те ж саме. Один і той же факт (вислів) не може бути одночасно істинним і хибним.

Принцип виключення третього. Не можна одночасно відкидати висловлювання і його заперечення. Будь-яке висловлювання може бути або істинним, або хибним, – третього не дано.

Принцип достатньої підстави. Будь-яке висловлювання має бути обгрунтовано, тобто істинність твердження не можна приймати на віру. Якщо твердження виводиться з будь-яких суджень, даних, фактів – *підстав,* то їх повинно бути досить для встановлення істинності твердження.

Можна вказати ще на одну дуже важливу властивість **- монотонність достовірних міркувань.** Якщо істинність деякого висловлювання *(А)* вже встановлена, то додавання нових фактів *(В)* не змінює істинності *А*.

Визначення 2.1. *Просте висловлювання* – це просте розповідне речення, відносно якого можна однозначно сказати, істинне воно чи хибне.

Питальні або окличні речення висловлюваннями не є.

Логічні значення *Істинно (True)* чи *Хибно (False)* будемо позначати відповідно T и F.

Кожне просте висловлювання позначають символами латинського алфавіту (з індексами або без індексів), які називають *пропозиціональними символами:* $\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{A}_1, \mathrm{A}_2 \ldots$

Складні висловлювання складаються з простих за допомогою сполучників «ні», «і», «або», «якщо…, то…», «тоді і тільки тоді». Цим сполучникам відповідають логічні операції: унарна операція заперечення $\neg$ («ні»), бінарні операції кон'юнкції & («і»), диз'юнкція $\vee$ («або»), імплікація $\rightarrow$ («якщо…, то…»), еквівалентність $\equiv$ («тоді і тільки тоді»). Символи операцій називаються *пропозиціональними зв'язками.*

Приклади:

«Лондон – столиця Англії» $\Leftrightarrow$ *A*. «Лондон не є столицею Англії» $\Leftrightarrow$ $\neg A$. «Найбільше місто Англії, Лондон (А), є її столицею (В)» $\Leftrightarrow$ $A\&B$. «Населення Канади говорить англійською (А) або французькою мовою (В)» $\Leftrightarrow$ $A \vee B$. «Якщо горобець – птаха (А), то у неї є крила (В)» $\Leftrightarrow$ $A \rightarrow B$. «Тварина є птахою (А) тоді і тільки тоді, коли у неї є крила (В)» $\Leftrightarrow$ $A \equiv B$.

Істинність або хибність складних висловлювань залежить від істинності або хибності простих висловлювань, які входять до складних, а також тим способом, якими вони комбінуються, тобто зв'язкою, яку використовують для побудови складного висловлювання. Кожна логічна зв'язка визначається своєю *таблицею істинності* (табл. 2.1, 2.2).

Таблиця 2.1

A	$\neg A$
F	T
T	F

Таблиця 2.2

A	B	$A \& B$	$A \vee B$	$A \to B$	$A \equiv B$
F	F	F	F	T	T
F	T	F	T	T	F
T	F	F	T	F	F
T	T	T	T	T	T

За допомогою зв’язки **заперечення** $\neg$ із стверджувальних утворюються заперечні висловлювання.

Кон’юнкція $A\&B$, що відповідає сполучникам «і», «а», істинна в тому і тільки в тому випадку, якщо істинні обоє з висловлювань, що входять в неї. Наприклад, висловлювання «на вулиці йде дощ (A) із сильним вітром (B)» виражається формулою $A\&B$. Кон’юнкція комутативна, тому висловлювання «на вулиці сильний вітер (B) і дощ (A)», яке можна висловити формулою $B\&A$, еквівалентне першому.

За допомогою кон’юнкції формалізується принцип несуперечливості: $A\&\neg A \equiv F$ – **закон протиріччя.**

Диз’юнкція $A \vee B$, що відповідає сполучнику «або», є істинною в будь-якому випадку, коли істинне хоча б одне висловлювання, що входить в неї, і хибна тільки у випадку, коли обидва простих висловлювання хибні. Диз’юнкція формалізує **закон виключення третього**: $A \vee \neg A \equiv T$.

Імплікація $A \to B$ виражає логічний (часто – причинно-наслідковий) зв’язок між висловлюваннями A і B та формалізує звичайний умовивід, в якому з посилки (антецедента) A випливає висновок (консеквент) B. Таблиця істинності імплікації відображає правильні (достовірні) умовиводи.

Формальне висловлювання **принципу тотожності** можна записати так: $\boldsymbol{A \to A}$ *(А є А).* Формальне висловлювання принципу монотонності достовірних міркувань: $\boldsymbol{A \to (B \to A)}$.

Еквівалентність $A \equiv B$ стверджує рівнозначність (рівносильність, тотожність) двох тверджень A і B; вона істинна тоді і тільки тоді, коли істинностні значення A і B співпадають.

Еквівалентність $A \equiv B$ стверджує не тільки необхідність умови A для того, щоб було істинно B, але і достатність цієї умови, тобто $A \equiv B = (A \to B)\ \&\ (B \to A)$, где $A \to B$ виражає *необхідність* A, а $B \to A$ – *достатність* A.

За допомогою логічних зв'язок складні висловлювання можна записати у вигляді формули, яку називають *пропозиціональною формулою.*

Визначення 2.2.

- Кожна пропозиціональна буква є формула.
- Якщо A і B – формули, то формулами є: $(\neg A)$, $(A\ \&\ B)$, $(A \vee B)$.
- Інших формул немає.

При записі формул прийнято наступне: зовнішні дужки можна опускати; встановлено пріоритет операцій: $\neg$, $\&$, $\vee$, $\to$, $\equiv$. Логічні зв'язки для імплікації $\to$ та еквівалентності $\equiv$ вводяться для скорочення запису формул:

$A \to B = \neg A \vee B = \neg(A\ \&\ \neg B)$, $A \equiv B = (A \to B)\ \&\ (B \to A)$.

Побудувавши формулу алгебри висловлювань, ми відволікаємося від її змістовного сенсу і оперуємо тільки поняттями істинності та хибності.

Приписування пропозиціональним літерам їх істинностних значень називається *інтерпретацією формули*. Множина всіх інтерпретацій формули утворює її таблицю істинності. Якщо виконати відображення $0 \Leftrightarrow F$, $1 \Leftrightarrow T$, то кожній пропозиціональній зв'язці буде відповідати булева операція, а кожній формулі алгебри висловлювань – булева формула, з чого випливає, що алгебра висловлювань є інтерпретацією булевої алгебри. В зв'язку з цим в ній зберігаються всі аксіоми і теореми булевої алгебри, зокрема й зображуваність формул алгебри висловлювань у вигляді ДДНФ і ДКНФ.

Визначення 2.3. Тотожно істинна формула називається *тавтологією*. Тотожно хибна формула називається *протиріччям*. Формула, яка приймає істинне значення хоча б на одній своїй інтерпретації, називається *здійсненною.* Формула, яка приймає на одних наборах істинні значення, а на якихось – хибні, називається *нейтральною.* Таким чином, формули розпадаються на три класи.

Тавтології є *виділенними* формулами логіки висловлювань; саме вони представляють найбільший інтерес. Тавтологія істинна при будь-яких значеннях простих висловлювань, що входять в неї; таблиця істинності тавтології в кожному рядку містить значення «істинне».

В формальній логіці поняття тавтології включає в себе всі тотожно істинні твердження. Вони формалізують правильні схеми міркувань.

Проблема можливості розв'язання в алгебрі висловлювань полягає в тому, щоб відшукати ефективну процедуру (алгоритм), за допомогою якої для кожної формули логіки висловлювань можна встановити, чи є вона тавтологією, чи ні.

Очевидно, що така процедура для формул логіки висловлювань існує: це побудова таблиць істинності.

Приклади.

1. Як приклад розглянемо закон ствердження консеквента: |= $A \to (B \to A)$. (***принцип монотонності достовірних міркувань*)**. Оскільки кожен рядок таблиці істинності (табл. 2.3) цієї формули містить значення Т, формула є тавтоогією. Також, $A \to (B \to A) = \neg A \vee \neg B \vee A = T \vee \neg B = T$.

Таблиця 2.3

A	B	$B \to A$	$A \to (B \to A)$
F	F	T	T
F	T	F	T
T	F	T	T
T	T	T	T

2**. Метод редукції (зведення до протиріччя)** – це спосіб скорочення переборів при складанні таблиці істинності. В якості приклада доведемо, що формула $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ – тавтологія.

Припустимо, що це не так, тобто існує така інтерпретація, на якій формула приймає хибне значення:

$$|\overbrace{(A \to (B \to C))}^{T} \to (\overbrace{\underbrace{(A \to B)}_{T} \to \underbrace{(A \to C)}_{F}}^{F})| = F.$$

Маємо систему рівнянь :

1) $|A \to (B \to C)| = T,$

2) $|A \to B| = T,$

3) $|A \to C| = F$.

З 3) випливає: $|A| = T$, $|C| = F$. Підставимо ці значення в 2): $|T \to B| = T$, звідки $|B| = T$. Підставимо знайдені значення $|A| = T$, $|C| = F$, $|B| = T$ в 1): $|T \to (T \to F)| = |T \to F| = F$. Отримане значення суперечить умові 1), а отже, не існує такої інтерпретації, на якій формула приймає хибне значення, тобто вона є тавтологією.

2. Перевіримо, чи є така формула тавтологією:

$(A \to (B \to C)) \to (A \vee B \to C)$.

Припустимо, що $|(A \to (B \to C)) \to (A \vee B \to C)| = F$. Тоді

1) $|A \to (B \to C)| = T$,

2) $|A \vee B| = T$,

3) $|C| = F$.

Із 2) випливає: а) $|A| = T$, $|B| = T$; б) $|A| = T$, $|B| = F$; с) $|A| = F$, $|B| = T$. Підставляючи значення $|A| = T$, $|B| = T$ в 1), отримаємо протиріччя: $|T \to (T \to F)| = F$. Але це ще не доводить, що формула є тавтологією. Розглянемо інші значення: б) $|A| = T$, $|B| = F$. Підставляючи ці значення в 1), маємо: $|T \to (F \to F)| = T$. Таким чином, при інтерпретації $|A| = T$, $|B| = F$, $|C| = F$ формула приймає хибне значення, а отже, вона не є тавтологією.

Логічний наслідок

Визначення 2.4. Якщо A і B – формули, то кажуть, що B *логічно випливає з* A, чи A *логічно призводить* B, якщо на всіх інтерпретаціях, де A приймає істинне значення, B також приймає істинне значення. Це позначають як $A \models B$ чи $A \Rightarrow B$.

Теорема 2.1. Логічний наслідок А |= В виконан тоді і тільки тоді, коли формула $A \to B$ – тавтологія.

Визначення 2.5. Формула B логічно випливає з формул *A1, A2, ..., An*, якщо на всіх тих інтерпретаціях, на *яких A1, ..., An* приймають істинні значення одночасно, формула *B* також приймає істинне значення. Це позначається так: *A1, ... An* |= *B*.

Теорема 2.2. *A1, A2, ..., An* |= *B* тоді і тільки тоді, коли $|= A_1 \,\&\, A_2 \,\&\, \ldots \,\&\, A_n \to B$.

Визначення 2.5. A |= B і B |= A, то формула A логічно еквівалентна формулі B. Це позначається як $A \Leftrightarrow B$, чи $A \equiv B$.

Якщо формула A логічно еквівалентна формулі B, то $A \equiv B$ – тавтологія.

Приклад. Перевіримо логічний наслідок: $A \to B,\ \ A \to \neg B \models \neg A$ (правило доведення до абсурду) і тавтологію: $\models (A \to B) \,\&\, (A \to \neg B) \to \neg A$. Позначимо тавтологію через *E*. Побудуємо таблицю істинності (табл. 2.4).

Таблиця 2.4

A	B	$A \to B$	$A \to \neg B$	$(A \to B) \,\&\, (A \to \neg B)$	$\neg A$	E
F	*F*	*T*	*T*	*T*	*T*	*T*
F	*T*	*T*	*T*	*T*	*T*	*T*
T	*F*	*F*	*T*	*F*	*F*	*T*
T	*T*	*T*	*F*	*F*	*F*	*T*

Як бачимо, на тих наборах, на яких посилки $A \to B, A \to \neg B$ приймають істинне значення одночасно, формула ¬A також приймає істинне значення. Отже, логічний наслідок виконаний. З іншого боку, формула $(A \to B) \,\&\, (A \to \neg B) \to \neg A$ є тавтологією, що також є доказом дійсності логічного наслідку.

Логічні наслідки встановлюють *правила виведення* істинних висновків з істинних посилок. Кажуть, що логічне слідування (правило виведення) *зберігає істинність.* Правила виведення, що зберігають істинність, називаються *достовірними (дедуктивним) міркуваннями.*

Метатеореми про тавтології

Теорема 2.3 (*правило modus ponens*). Якщо А – тавтологія і $A \to B$ – тавтологія, то В – тавтологія, тобто якщо $|= A$ *і* $|= A \to B,$ то $|= B$.

Правило *modus ponens* (скорочено МР) встановлює логічне слідування А, А $\to$ В |= В і називається ще правилом *відділення*.

Правило МР висловлює елементарний акт дедукції. Імплікації $A \to B$, яка за визначенням має сенс «якщо *A*, то *B*», можна інтерпретувати як правило, в якому А є «причиною», а В - «наслідком». Тоді правило МР говорить про те, що наслідок В настає при виконанні умови А, тобто при істинності посилки.

Теорема 2.4 (*правило підстановки*). Якщо *А* – тавтологія, яка містить пропозіційні змінні $a_1, a_2, ..., a_n$, то формула *В*, отримана з *А* заміною кожного входження a_i на деяку формулу A_i, також буде тавтологією.

Приклад. Формула $A \to (B \to A)$ - тавтологія. Замінимо *A* на $A \vee B$. Отримаємо нову тавтологію: $| = A \vee B \to (B \to A \vee B)$. Таким чином, кожну тавтологію можна розглядати як схему, з якої за допомогою підстановки можна отримати безліч тавтологій.

Теорема 2.5 (*правило еквівалентної заміни*). Якщо *В* можна отримати з *А* підстановкою формули B_1 замість одного або декількох входжень підформули A_1 в *А*, то $((A_1 \equiv B_1) \to (A \equiv B))$ є тавтологія, і, отже, якщо A_1 *і* B_1 логічно еквівалентні, то *А* і *В* також логічно еквівалентні.

Приклад. У тавтології $A \to (B \to A)$ замінимо підформулу $B \to A$ на еквівалентну їй формулу $\neg B \vee A$, отримаємо нову тавтологію $A \to \neg B \vee A$. Тавтологією також буде формула: $A \to \neg (B \,\&\, \neg A)$, так як $B \to A$

еквівалентна $\neg$ (B & $\neg$A). Розглянуті метатеореми дають можливість отримувати нові тавтології і нові правила виведення з уже наявних.

Будь-яка система посилок має цінність тільки тоді, коли існує хоча б одна інтерпретація (тобто істиннісний розподіл літер, що входять в кожне висловлювання системи), на якій всі посилки істинні одночасно. Якщо такої інтерпретації не існує, то система посилок суперечлива, і з неї разом з деякою формулою А виводиться і її заперечення $\neg$A.

Приклад. Розглянемо систему висловлювань: $\{A\to B,\ B\to C\vee\neg A,\ \neg D\to A\&\neg C,\ D\to A\}$. Для перевірки системи на несуперечлівість, потрібно прирівняти кожну гіпотезу до T. Якщо взяти значення для кожної пропозиціональної літери, що входять до системи як «істина», то всі гіпотези будуть T - ситема несуперечлива.

Через алгебраїчні перетворення формули, що відповідає системі, можна отримати:

$(A\to B)\ \&\ (B\to C\vee\neg A)\ \&\ (\neg D\to A\&\neg C)\ \&\ (D\to A) =$

$= (\neg A\vee B)\ \&\ (\neg B\vee C\vee\neg A)\ \&\ (D\vee(A\&\neg C))\ \&\ (\neg D\vee A) =$

$= (\neg A\vee(B\&\ (\neg B\vee C))\ \&\ (D\vee A)\&(D\vee\neg C)\ \&\ (\neg D\vee A) =$

$= (\neg A\vee(B\&\ C))\&A\&(\ D\vee\neg C) = (B\&\ C)\ \&A\&(D\vee\neg C) = A\ \&\ B\ \&\ C\ \&\ (D\vee\neg C) =$

$= A\ \&\ B\ \&\ C\ \&\ D.$

Для несуперечливості системи посилок отримана формула має допівнювати T. Це можливо тільки при входженні кожної пропозиціональної літери із значенням T.

Завдання 2.1.1. Визначити, чи є наступні формули тавтологіями.

Приклад 1. Чи є формула $((A\to B)\to B)\to B$ *тавтологією*?

1 спосіб – побудова таблиці істинності.

A	B	$A \to B$	$(A \to B) \to B$	$((A \to B) \to B) \to B$
F	F	T	F	T
F	T	T	T	T
T	F	F	T	F
T	T	T	T	T

Формула не є тавтологією, тому що існує інтерпретація $|A| = T$, $|B| = F$, на якій вона набуває хибне значення.

2 спосіб. Дослідження формули методом редукції.

Припустимо, що існує набір, на якому формула набуває хибне значення:

$|((A \to B) \to B) \to B| = F$.

Тоді $|((A \to B) \to B)| = T$, $|B| = F$. Підставимо знайдене значення $|B| = F$ в першу рівність: $|((A \to F) \to F) | = T$. Розв'яжемо це рівняння відносно A: якщо $A = T$, то $|((T \to F) \to F)| = T$. Отже, при $|\boldsymbol{A}| = \boldsymbol{T}$, $|\boldsymbol{B}| = \boldsymbol{F}$ формула набуває значення, яке дорівнює F. Таким чином, вона не є тавтологією.

3 спосіб. Алгебраїчне перетворення.

$((A \to B) \to B) \to B = \neg (\neg (\neg \mathrm{A} \vee \mathrm{B}) \vee \mathrm{B}) \vee \mathrm{B} = ((\neg \mathrm{A} \vee \mathrm{B}) \,\&\, \neg \mathrm{B}) \vee \mathrm{B} =$

$= (\neg \mathrm{A} \,\&\, \neg \mathrm{B}) \vee \mathrm{B} = \neg \mathrm{A} \vee \mathrm{B}$.

Як це корелюється з висновком, отриманим в попередньому пункті:

$|\boldsymbol{A}| = \boldsymbol{T}$, $|\boldsymbol{B}| = \boldsymbol{F}$**?** Так, звісно, формула не є тавтологією. Спрощення формули привело нас до розуміння, що формула отримує значення $\boldsymbol{F}$ тільки на одному наборі ($|\boldsymbol{A}| = \boldsymbol{T}$, $|\boldsymbol{B}| = \boldsymbol{F}$), тобто вона є конституентою нуля (макстермом), спрощення – її ДКНФ.

Приклад 2. Чи є формула $(A \to B) \,\&\, (B \to C) \,\&\, A \to C$ тавтологією?

Дослідження формули методом редукції.

Припустимо, що існує така інтерпретація, на якій формула набуває хибне значення.

Тоді $|A \to B| = T, |B \to C| = T, |A| = T, |C| = F.$

Тоді:

$|B \to C| = |B \to F| = T \Rightarrow |B| = F;$

$|A \to B| = |A \to F| = T \Rightarrow |A| = F.$

Звідси $|A| = F$, що суперечить третій рівності $|A| = T$. Отримане протиріччя демонструє, що формула не може набувати хибних значень. Отже, формула $(A \to B)$ & $(B \to C)$ & $A \to C$ є тавтологією.

Можемо побудувати і таблицю істинності і довести алгебраїчно.

$(A \to B) \mathbin{\&} (B \to C) \mathbin{\&} A \to C = \neg ((A \to B) \mathbin{\&} (B \to C) \mathbin{\&} A) \vee C =$

$= \neg (\neg A \vee B) \vee \neg (\neg B \vee C) \vee \neg A \vee C = A \mathbin{\&} \neg B \vee B \mathbin{\&} \neg C \vee \neg A \vee C =$

$= \underline{\mathbf{\neg B \vee B}} \vee \neg A \vee C = \mathbf{T} \vee \neg A \vee C = T.$

$E = (A \to B) \mathbin{\&} (B \to C) \mathbin{\&} A \to C$

A	B	C	$A \to B$	$B \to C$	$(A \to B) \& (B \to C) \& A$	**E**
F	F	F	T	T	F	T
F	F	T	T	T	F	T
F	T	F	T	F	F	T
F	T	T	T	T	F	T
T	F	F	F	T	F	T
T	F	T	F	T	F	T
T	T	F	T	F	F	T
T	T	T	T	T	T	T

Будь-який спосіб доводить, що формула є тавтологією.

1) $(A \to B) \to ((B \to C) \to (A \to \mathbf{C}));$

2) $(A \to B) \to ((C \to B) \to (A \lor C \to B));$

3) $(A \to (B \to C)) \to ((A \to C) \to (A \to B));$

4) $((A \to B) \to (A \to C)) \to ((A \to (B \to C));$

5) $(B \to A \lor C) \to ((B \to C) \to ((D \to C) \to (B \lor D \to C)));$

6) $(B \to C) \to (A \lor C \to (B \to C));$

7) $(A \to (B \to C)) \to (C \to A \lor B).$

Завдання 2.1.2. Перевірити, чи є формули тавтологіями, протиріччями або нейтральними. Які формули є здійсненними?

Для перевірки, до якого класу належить формула, можна звернутися до побудови таблиць істинності або до алгебраїчних перетворень. Спробуємо визначити клас формули через міркування – через метод редукції (*в логіці – пошук такого судження, з якого випливає судження, визнане нами за істинне; зведення складного до простішого в інтересах наукового аналізу).* В загальному випадку рекомендується намагатися визначити. чи є формула тавтологією, бо такі міркування нами вже відпрацьовані. Якщо формула виявиться не тавтологією, то підбиранням значень пропозиційних літер визначитися, чи буде вона здійсненною.

1) $(A \,\&\, B) \lor (C \,\&\, D) \to (A \lor B) \,\&\, (C \lor D);$

2) $A\& \, (A\&\neg B \lor \neg A\&B)\&B$

3) $(A \to B) \,\&\, (C \to D) \to (A \lor C \to B \lor D);$

4) $(A \to (B \to C)) \to ((A \to B) \to (A \to C));$

5) $\neg((A \lor B \to C) \to \neg(\neg A \lor C \to B\&\neg C));$

6) $(A \lor B \to C) \to (A \to C) \lor (B \to C);$

7) $(A \,\&\, B \to C) \to ((A \to C) \to (B \to C));$

8) $(A \to B) \& (C \to D) \to (A \& C \to B \& D)$;

9) $A\&(\neg A\vee \neg B)\&B$;

10) $(A\&\neg B \vee \neg A\&B) \to B$.

Завдання 2.1.3. Дослідити систему посилок (гіпотез) на несуперечливість.

Приклад 1.

$\{F \to E\vee L, E \vee \neg F\&U, U \to E \vee L, \neg F\to \neg L, E \to \neg E\}$

Система посилок несуперечлива, якщо можна знайти таку інтерпретацію літер, при який кожна посилка є істинною, тобто дорівнює *T*.

1. $|F \to E\vee L|=T$;
2. $|E \vee \neg F\&U|=T$;
3. $|U \to E \vee L|=T$;
4. $|\neg F\to \neg L|=T$;
5. $|E \to \neg E|=T$.

З 5-ої посилки слідує, що $|E|=F$. Тоді, з 2-ої : $|F|=F$ *і* $|U|=T$.

Для того, щоб 3-я посилка дорівнювала *T* , при вже отриманих значеннях пропозиційних літер, має бути $|L|=T$. Але при таких значеннях *F* та *L* 4-а посилка не буде дорівнювати *T*. Отже, дана система посилок є суперечливою.

Приклад 2.

$\{A \to B,\ C \vee A,\ C\to E,\ \neg E,\ \neg E\to\neg C\}$

Розглянемо систему:

1. $|A \to B|=T$;
2. $|C \vee A|=T$;
3. $|C\to E|=T$;

4. $|\neg E| = T$;

5. співпадає з Г3

З 4-ої посилки отримуємо $|E|=F$. Тоді при $|C|=F$, $|A|=T$, $|B|=T$ кожна посилка є істинною.

Система – несуперечлива.

1) $\{ A \vee B \to C \& D,\ D \vee E \to G,\ A \vee \neg G \}$;

2) $\{(A \to B)\&(C \to D),\ (B \to D)\&(\neg C \to A),\ (E \to G)\&(G \to \neg D), \neg E \to E\}$;

3) $\{(A \to B\&C)\&(D \to B\&E),\ (G \to \neg A)\&H \to I,\ (H \to I) \to G\&D,\ \neg(\neg C \to E)\}$;

4) $\{(A \to B\&C)\&(D \to B \& E),\ (G \to \neg A),\ (H \to I) \to G \& D,\ \neg(\neg C \to E)\}$;

5) $\{ A \to C \& D,\ D \vee E \to C,\ A \vee C \to E \}$;

6) $\{ A \to B ,\ B \to D ,\ C \to A ,\ (E \to C) \& (G \to \neg D) \}$;

2.2. Формалізація і рішення логічних задач

<u>Теоретичні відомості</u>

Мова алгебри висловлювань використовується для формалізації речень природної мови і доказів логічних слідувань.

Двозначна класична логіка корисна в повсякденному житті, навіть, якщо доводиться визнати, що в припущеннях, наприклад, про двозначність, є елемент свавілля. Для вирішення будь-якої інтелектуальної проблеми не слід нехтувати логікою. Наприклад, якщо розглядається якась ймовірнісна задача, то перш за все можна знайти наслідки, що випливають з набору гіпотез, без розгляду їх ймовірнісного (змістовного) сенсу. Посилання на модальну або інтуїционістську логіку дають зрозуміти, що в деяких ситуаціях потрібна інша, не класична логіка

Зв'язкою «еквівалентність» замінюються речення природної мови наступної природи: *якщо - і - тільки - якщо, тоді і тільки тоді, рівносильно.* «Імплікація» замінює: *якщо - то, в разі A має місце B, A – тільки, якщо B, B - якщо A, A тягне B, для B досить A, для A необхідно B*. Заміна на «кон'юнкцію» - *A і B , не тільки A - а й B, B - хоча і A, B - незважаючи на A, як A - так і B, A разом з B, A - в той же час як B.*

Переклад ***A & B*** найбільш простий, якщо в ньому не бере участь причина часу (про що йшлося вище). Багато смислових відтінків (незважаючи, хоча, але і) зникають при формалізації.

«*Диз'юнкція*» - A чи B, A - якщо не B, A чи B або обидва. Труднощі перекладу «*диз'юнкції*» полягають в двозначності певних термінів. Якщо в меню ресторану вказано «*чай або кава - безкоштовно*», то ми не здивуємося, якщо, зажадавши, і те й інше, отримаємо збільшений рахунок. У той же час, коли оголошення говорить, що «*пожертвування приймаються в церкві або школі*», то ми не думаємо, що принесене в церкву відкинуть через те, що ми вже пожертвували в школі.

Якщо хочуть висловити виняткове «*або»*, то повинні додати до «*або*», наприклад, фразу «*але не те й інше разом*» - формалізується $\neg (A \& B) \& (A \vee B)$.

Виразам «*A, якщо не B*», «*A, крім випадку, коли B*» надається сенс такого, що наявність *B* звільняє від відповідальності стверджувати істинність *A*. Тобто, якщо не $\neg B$, то *A*, а якщо *B*, то про *A* нічого не говориться ($\neg B \rightarrow A = \neg A \rightarrow B$).

У звичайній мові не вживаються дужки для вказівки сполучуваності частин складного речення. «*Якщо Джонс присутній або якщо Вільямс висловиться за нашу пропозицію і Старк не стане заперечувати, то наша пропозиція буде прийнята*» («стратити не можна помилувати» - на листі необхідно розставляти потрібні акценти знаками пунктуації). Формалізація цього речення наступна:

$\boldsymbol{D \vee W \,\&\, \neg S \rightarrow P}$ – формулі відсутні дужки, що не дозволяє однозначно сприймати інформацію відповідно з вихідним текстом (слід зазначити, що і правила побудови формул теж порушуються).

Приклад 2.1. Розглянемо логічне слідування: *Якщо не буде дощу, то ми поїдемо на пікнік. Якщо ми поїдемо на пікнік, то ми добре проведемо час. Дощу немає. Отже, ми добре проведемо час.*

Позначимо пропозіціональнимі буквами прості висловлювання: P - «*не буде дощу*»; S - *«поїдемо на пікнік»;* R - «*добре проведемо час*». Необхідно довести логічне слідування : $P \rightarrow S, S \rightarrow R, P \mid = R$.

Доведення.

1 спосіб. Довести тавтологію: $\mid = (P \rightarrow S) \,\&\, (S \rightarrow R) \,\&\, P \rightarrow R$. Для цього досить побудувати таблицю істинності.

2 спосіб. Доведення від протилежного (метод редукції).

Припустимо, що існує така інтерпретація, на якій всі посилки приймають істинне значення, а наслідок - хибне, тобто $\mid P \rightarrow S \mid = T, \mid S \rightarrow R \mid = T, \mid P \mid = T$. Припускаємо $\mid R \mid = F$. Тоді з $\mid S \rightarrow R \mid = \mid S \rightarrow F \mid = T$ випливає, що $\mid S \mid = F$; *з* $\mid P \rightarrow S \mid = \mid P \rightarrow F \mid = T$ випливає, *що* $\mid P \mid = F$, що суперечить третьої посилці | P | = T. Це протиріччя доводить логічне слідування.

3 спосіб. Побудова формального логічного висновку.

Побудуємо логічний висновок, використовуючи відомі правила виведення. Логічний висновок записується зазвичай як пронумерована послідовність формул, праворуч від кожної формули записується коментар, який вказує, на якій підставі формула включена в цю послідовність. Посилки виведення зазвичай позначаються літерою Г (гіпотеза).

1. $P \to S$ Г1

2. $S \to R$ Г2

3. P Г3

4. S правило MP (3,1)

5. R правило MP (4,2)

Остання формула в цьому висновку є логічним наслідком посилок *Г1, Г2, Г3*.

В силу того, що правило MP зберігає істинність, кожна формула, яка бере участь у виведенні, істинна при істинності посилок Г1, Г2, Г3. Тому, якщо з'єднати символом $\to$ формули, так, щоб посилка імплікації передувала висновку в цьому виведенні, то отримана формула також буде істинною, і, отже, вона є логічним наслідком посилок.

Тому з даного висновку ми можемо отримати логічне виведення:

$P \to S, S \to R \mid = P \to R.$

Це логічне слідування відповідає правилу виведення, яке називають правилом силогізму:

$A \to B, B \to C \mid = A \to C.$

Приклад 2.2. *Через снігопад (А) машина потрапить в затор (В): А→В. У затор машина не потрапила (¬В). Отже, снігопаду не було (¬А). Доведемо* $A \to B, \neg B \mid = \neg A.$

Припустимо, що $\mid A \to B \mid = T, \mid \neg B \mid = T$ *і* $\mid \neg A \mid = F$, тоді $\mid A \to B \mid = \mid T \to F \mid = F$. Отримане протиріччя доводить логічне слідування.

Це логічне слідування відповідає правилу виведення ***modus tollens (MT):***

$A \to B, \neg B \mid = \neg A.$

Приклад 2.3. *Решітка є модулярною (B) в тому випадку, якщо вона дистрибутивною (A): $A \to B$. Але ми знаємо, що якщо решітка немодулярна ($\neg B$), то вона і недистрибутивна ($\neg A$): $\neg B \to \neg A$.*

Припустимо, що $| A \to B | = T$, $| \neg B \to \neg A | = F$, тоді $| \neg B | = T$, $| \neg A | = F$ і $| A \to B | = | T \to F | = F$. Отримане протиріччя доводить логічне слідування.

Це логічне слідування відповідає правилу виведення, яке називають правилом контрапозиції:

$A \to B \mid = \neg B \to \neg A$

Приклад 2.4. *Якщо долар росте (B), то ціни не зменшуються ($\neg A$). Якщо долар падає ($\neg B$), то ціни теж не зменшуються ($\neg A$): $B \to \neg A$ і $\neg B \to \neg A$. Тобто, ціни не зменшуватися в будь-якому випадку.* Згідно із законом контрапозиції ми можемо отримати з умов задачі: $A \to \neg B$ і $A \to B$.

$\boldsymbol{A \to \neg B, A \to B \mid = \neg A.}$

Цей закон відображає правило приведення до абсурду. Якщо з посилки слідує одночасно істино і хибно, то сама посилка – хибна.

Приклад 2.5. Я не зможу заснути (*A*) тільки в тому випадку, якщо вип'ю каву (*B*). Як формалізувати це речення? Здається, що «якщо» знаходиться при висловлюванні *B*, то має бути $\boldsymbol{B \to A}$. Але сенс фрази полягає в наступному: «Якщо я не заснув, то з цього випливає тільки одне - я випив каву». Тобто, слід проводити формалізацію таким чином: $\boldsymbol{A \to B}$. Вживання «тільки», «крім» тощо змінює, напрямок імплікації.

Випишемо розглянуті логічні закони:

$\boldsymbol{A \to B, A \mid= B}$	***modus ponens***
$\boldsymbol{A \to B, B \to C \mid= A \to C}$	***правило силогізму***
$\boldsymbol{A \to B \mid= \neg B \to \neg A}$	***правило контрапозиції***

$\boldsymbol{A \to B, \neg B \models \neg A}$	***modus tollens***
$\boldsymbol{A \to B, A \to \neg B \models \neg A}$	***правило приведення до абсурду***

Приклад 2.6. Отримати можливі логічні наслідки з даної множини посилок: *Якщо в городі - бузина (A), то в Києві - дядько (B). Коні їдять овес (C) або в городі - бузина. Якщо коні їдять овес, то Дніпро впадає в Чорне море (E). Днпро не впадає в Чорне море.*

Побудуємо дедуктивний формальний висновок:

1. $A \to B$ Г1

2. $C \vee A$ Г2

3. $C \to E$ Г3

4. $\neg E$ Г4

5. $\neg E \to \neg C$ правило контрапозиції (3)

(*Якщо Дніпро не впадає в Чорне море, то коні не їдять овес)*

6. $\neg C \to A$ *еквівалентна заміна (2)*

7. $\neg E \to A$ правило силогізму (5, 6)

(Якщо Дніпро не впадає в Чорне море, то в городі - бузина)

8. $\neg E \to B$ правило силогізму (1, 7)

(Якщо Дніпро не впадає в Чорне море, то в Києві дядько)

9. B МР (4,8)

(В Києві - дядько)

Таким чином, з даного набору посилок ми крок за кроком отримали всі можливі висновки.

В одному з оповідань про Шерлока Холмса склалася така ситуація: «Нам відомо, що злочинець не міг потрапити в кімнату ні через двері (A), ні через камін (B). Ми знаємо також, що він не міг сховатися в кімнаті (C), оскільки в ній ховатися ніде. Як же тоді він проник сюди? - Через дах (D)! - Без сумніву. Він міг проникнути в цю кімнату тільки через дах.»

Це міркування можна формалізувати так: $A \vee B \vee C \vee D, \neg A, \neg B, \neg C \models D$, що рівносильно: $\neg A \to (\neg B \to (\neg C \to D)), \neg A, \neg B, \neg C \models D$. Триразове застосування правила MP доводить цей логічний висновок.

Задачі (за теоремою 2.2.) відповідає формула:

$(A \vee B \vee C \vee D) \& \neg A \& \neg B \& \neg C \to D$, яка має бути тавтологією у випадку, коли висновок з цієї множини посилок є дійсним.

Не слід забувати, що логічний наслідок виконується тільки тоді, коли з істинних висловлювань слідує істинний висновок. Тому повинна існувати хоча б одна інтерпретація, на якій всі посилки істинні одночасно. Якщо такої інтерпретації не існує, то система посилок суперечлива.

Приклад 2.7.

Які висновки можна зробити з наступних висловлювань:

" Повернувшись додому, Мегре подзвонив на набережну Орфевр:

- Є новини?

- Так, шеф. Надійшли повідомлення від інспекторів. Торранс встановив, що, якщо Франсуа був п'яний (F), то або Етьєн вбивця (E), або Франсуа бреше (L). Жуссе вважає, що або Етьєн вбивця, або Франсуа не був п'яний і вбивство сталося після опівночі (U). Інспектор Люка просив передати Вам, що якщо вбивство сталося після опівночі, то або Етьєн вбивця, або Франсуа бреше. Потім дзвонила ...

- Усе. Дякую. Цього достатньо. - Комісар поклав трубку. Він знав, що тверезий Франсуа ніколи не бреше. Тепер він знав все. "

Для побудови формального виводу необхідне знання деяких прийомів і теорем. Приведемо формальний вивід для вищевикладеної задачі:

1. $F \to E \vee L$ гіпотеза 1

2. $E \vee \neg F\&U$ гіпотеза 2

3. $U \to E \vee L$ гіпотеза 3

4. $\neg F \to \neg L$ гіпотеза 4

5. $\neg E \to \neg F$ гіпотеза 2 (видалення кон'юнкції)

6. $\neg E \to U$ гіпотеза 2 видалення кон'юнкції)

7. $\neg E \to E \vee L$ силогізм 3, 8

8. $\neg E \to L$ з 7 еквів.перетв. і ідемпотентність

9. $L \to F$ контрапозиція гіпотези 4

10. $\neg E \to F$ силогізм 8, 11

11. E довед.до абсурда 5, 10

Ми бачимо, що гіпотеза 1 є зайвою, вона не бере участь в побудові формального виводу.

Якщо ми використаємо гіпотезу 1, наприклад, зробимо силогізм з контрапозицією гипотези 4 ($L \to F$ та $F \to E \vee L$), то отримаємо $L \to E \vee L =$ $= \neg L \vee E \vee L = T \vee E = T$, що є даремним кроком.

Завдання 2.2.1. Перевірити логічне слідування.

Приклад. *Джон або втомився, або він хворий. Якщо він втомився, то він роздратований. Він не роздратований. Отже, Джон – хворий.*

Формалізуємо в логіці висловлювань гіпотези.

Нехай *A* – «Джон втомився»,

B – «Джон хворий»,

C – «Джон роздратований».

Побудуємо дедуктивний вивід, використовуючи наступні правила виводу:

$A, A \to B \models B$ **– *modus ponens* (MP);**

$A \to B, B \to C \models A \to C$ **– правило силогізму (ПС);**

$\boldsymbol{A \to B \models \neg B \to \neg A}$ **– правило контрапозиції (ПК);**

$\boldsymbol{A \to B\ ,\ A \to \neg B \models \neg A}$ **– правило доведення до абсурду (ПА);**

$\boldsymbol{A \to B,\ \neg B \models \neg A}$ **– *modus tollens* (МТ).**

Побудуємо формальний вивід:

1. $A \vee B$ — Γ_1
2. $A \to C$ — Γ_2
3. $\neg C$ — Γ_3
4. $\neg A$ — МТ (2,3)
5. $\neg A \to B$ — еквівалентна заміна (1)
6. B — МР (4,5)

Приклад 2. *Малі діти нерозумні. Той, хто може приборкувати крокодилів, заслуговує на повагу. Нерозумні люди не заслуговують на повагу. Отже, малі діти не приборкують крокодилів.*

Нехай *A* – «малі діти», *B* – «розумні», *C* – «заслуговують на повагу», *D* - «можуть приборкувати крокодилів». Побудуємо дедуктивний вивід, використовуючи наступні правила виводу:

Формальний вивід:

1. $A \to \neg B$ — Γ_1
2. $D \to C$ — Γ_2
3. $\neg B \to \neg C$ — Γ_3
4. $A \to \neg C$ — правило силогізму (1,3)
5. $\neg C \to \neg D$ — правило контрапозиції (2)
6. $A \to \neg D$ — правило силогізму (4,5)

1) Якщо студент пропускає заняття (Z), він погано засвоює предмет (P). Якщо студент не навчається вдома (D), він також погано засвоює предмет. Якщо студент погано засвоїв предмет, він не складе залік (A). Студент не допускається до іспиту (I), якщо він не склав залік. Недопущений до іспиту студент має менше часу на відпочинок (R). Отже, якщо студент пропускає заняття або не навчається вдома, у нього менше часу на відпочинок.

2) Якщо Джонс <u>не зустрічав</u> цієї ночі Сміта (V), то або Сміт був убивцею (S), або Джонс бреше (D). Якщо Сміт був убивцею, то Джонс <u>бачив</u> Сміта цієї ночі (¬V), і вбивство мало місце після опівночі (P). Якщо вбивство сталося після опівночі, то або Сміт був убивцею, або Джонс бреше. <u>Джонс не зустрічав цієї ночі Сміта</u> (V). Отже, <u>якщо Джон говорить правду</u> (¬D), то Сміт - убивця.

3) Якщо Сміт переможе на виборах (V), він буде задоволений (Z), а якщо він буде задоволений, то він поганий борець в передвиборній кампанії (B). Але якщо він завалить вибори (¬V), то втратить довіру партії (D). Якщо він поганий борець у передвиборній кампанії, йому слід вийти з партії (E). Він поганий борець, якщо втратить довіру партії. Сміт або переможе в передвиборній кампанії, або завалить її. Отже, йому треба вийти з партії в будь якому випадку.

4) Якщо Петров є членом нашої команди (P), він обов'язково хоробрий (H) і добре володіє технікою удару (T). Але він не входить у склад нашої команди. Отже, він не хоробрий.

5) Зарплата збільшується **тільки** (Z), якщо буде інфляція (I). Якщо буде інфляція, то збільшиться вартість життя (V). Якщо вартість життя зростає, то люди нещасливі (¬H). Отже, при збільшенні зарплати люди будуть нещасними?

6) Якщо конгрес відмовляється прийняти нові закони (¬K), то страйк не буде завершено (¬Z), якщо тільки він не триває більше року (Y). Страйк також буде завершено (Z), якщо президент фірми піде у відставку (P). Конгрес відмовляється діяти, страйк завершено, і президент фірми не йде у відставку. Отже, страйк тривав більше року.

Вказівка: 1 та 2 речення можна разом трактувати так, що, якщо страйк завершено, то або конгрес прийняв певні рішення або президент пішов у відставку або страйк тривав більше року.

7) Або Сергій і Борис – одного віку (A), або Сергій старше за Бориса (B). Якщо Сергій і Борис – одного віку, то Микола і Борис – різного (C). Якщо Сергій старше Бориса, то або Борис старше Віталія (D), або Микола і Борис – різного віку. Микола і Борис – однолітки. Отже, Борис старше Віталія.

8) Якщо Петро не бачив Миколу на вулиці (¬B), то або Микола ходив в кіно (C), або Петро сказав правду (P). Якщо Микола не ходив в кіно, то Петро не бачив Миколу на вулиці, і Микола сказав правду (M). Якщо Микола сказав правду, то або він ходив в кіно, або Петро збрехав. Перевірити, чи ходив Микола в кіно.

2.3. Метод резолюцій

Теоретичні відомості

В основі методу резолюцій лежить процедура пошуку спростування, тобто замість доведення тавтології, що відповідає логічній задачі, доводиться те, що заперечення формули є протиріччям. Метод спростування для доказу логічного слідування полягає в наступному. Нехай виконується логічне слідування: $F1$, $F2$ |= G. Тоді |= $F1\&F2 \to G$ є тавтологією, і, отже, $|\neg(F1\&F2 \to G)| = |F1\&F2\&\neg G| \equiv F$. Оскільки за визначенням посилки $F1$, $F2$ істинні, формула $F1\&F2\&\neg G$ може обернутися в хибну тільки, якщо $|\neg G| = F$, тобто якщо $|G| = T$. Тоді логічне слідування виконано. В принципі процедура спростування формалізує метод редукції (так званий неформальний вивід).

Введемо декілька визначень.

Визначення 2.6. Диз'юнкція літер називається диз'юнктом, або clause (клаузою, клозом). Однолітерний диз'юнкт називається *одиничним диз'юнктом (фактом)*. Коли диз'юнкт не містить ніяких літер, його називають *пустим* диз'юнктом. Так як пустий диз'юнкт не містить літер, які могли б бути істинними за будь-яких інтерпретацій, то пустий диз'юнкт завжди хибний. Пустий диз'юнкт позначається символом $\square$.

Правило резолюцій Робінсона. Якщо для будь-яких двох диз'юнктів C_1 і C_2 існує літера $L_1 \in C_1$ і контрарна їй літера $L_2 \in C_2$ ($L_2 = \neg L_1$), то викресливши L_1 з C_1 і L_2 з C_2 і побудувавши диз'юнкт з решти літер, отримаємо так звану *резольвенту* C_1 і C_2 : $C'_1 \vee C'_2$, де $C'_1 = C_1 \backslash L_1$, $C'_2 = C_2 \backslash L_2$.

Теорема 2.6. *Резольвента* C є логічним слідуванням C_1 і C_2, що містять контрарні літери L і $\neg L$: $L \vee C'_1, \neg L_1 \vee C'_2 \models C'_1 \vee C'_2$.

Доведення. Припустимо, що $|L \vee C'_1| = T$, $|\neg L \vee C'_2| = T$, $|C'_1 \vee C'_2| = F$. Тоді $|C'_1| = F$, $|C'_2| = F$. Якщо $|L \vee C'_1| = T$, то $|L| = T$, але $|\neg L \vee C'_2| = T$, отже, $|L| = F$. Отримане протиріччя доводить теорему.◊

Правило резолюцій є узагальненням багатьох відомих нам правил виводу. Наприклад, правило силогізму: $A \to B, B \to C \models A \to C$ може бути переписано у вигляді: $\neg A \vee B, \neg B \vee C \models \neg A \vee C$, що відповідає правилу резолюцій. Правило MP: $A, A \to B \models B$ може бути переписано у вигляді: A $\neg A \vee B \models B$, що також відповідає правилу резолюцій. Нарешті, закон протиріччя $A \& \neg A \equiv F$ рівнозначний правилу: $A, \neg A \models \square$, згідно якому, резольвента двох контрарних однолітерних диз'юнктів є пустий диз'юнкт.

Метод резолюцій стосовно вирішення задачі полягає в тому, щоб перевірити, чи містить множина диз'юнктів S, що складається з перетворених в КНФ посилок і заперечення висновка, порожній диз'юнкт □. Якщо S містить □, то множина S нездійсненна, якщо ні, то треба перевірити, чи може він бути отриманий з даної множини диз'юнктів. Іншими словами, необхідно знайти множину основних прикладів, які спростовують вихідну множину диз'юнктів. Ця процедура полягає в побудові резолютивного виведення і ґрунтується на правилі резолюцій Робінсона.

Визначення 2.7. *Резолютивний вивід* з множини диз'юнктів S є послідовність $C_1, C_2, ..., C_k$, така, що кожне C_i або належить S, або є резольвентою попередніх C_i. Якщо останній диз'юнкт C_k = □, то множина диз'юнктів S є нездійсненною, а весь вивід називається спростуванням S. Якщо C_k не є пустим диз'юнктом і подальше застосування правила резолюцій неможливо, то множина S є здійсненною (але висновок в цьому разі не є наслідком з множини посилок, тобто, задача не розв’язується).

Приклад 2.8. Розглянемо задачу з прикладу 2.1. Необхідно перевірити логічне слідування в логіці висловлювань: $P \to S$, $S \to R$, $P \models R$. Складемо множину диз'юнктів S, для чого кожну формулу приведемо до КНФ, а від виснвка R візьмемо заперечення. Отримаємо:

1. $\neg P \vee S$
2. $\neg S \vee R$
3. P
4. $\neg R$
5. $\neg S$ резольвента 4, 2
6. $\neg P$ резольвента 5, 1
7. □ резольвента 3, 6

Приклад 2.9.

Повернемось до умови задачі 2.7. Припустимо, що вбивця - Етьєн. Можна формалізувати задачу і застосувати метод резолюції для з'ясування істинності нашого припущення.

1.	$F \to E \vee L$	*Умова задачі - список гіпотез*
2.	$E \vee \neg F \& U$	
3.	$U \to E \vee L$	
4.	$\neg F \to \neg L$	
		Приведемо гіпотези в КНФ
1.	$\neg E$	заперечення виводу
2.	$\neg F \vee E \vee L$	гіпотеза 1
3.	$E \vee \neg F$	гіпотеза 2 (видалення кон'юнкції)
4.	$E \vee U$	гіпотеза 2 (видалення кон'юнкції)
5.	$\neg U \vee E \vee L$	гіпотеза 3
6.	$F \vee \neg L$	гіпотеза 4
7.	$\neg F$	резольвента 1,3
8.	U	резольвента 1,4
9.	$\neg U \vee L$	резольвента 1,5
10.	$\neg L$	резольвента 6,7
11.	$\neg U$	резольвента 9,10
12.	$\square$	резольвента 8,11

Правило резолюцій – дуже потужний і зручний засіб логічного доведення.

Гіпотеза 1 не задіяна в процедурі виведення, так само, як при формальному способі.

Приклад 2.10.

Побудуємо резолютивний вивід для задачі з прикладу 2.6. Але трохи змінемо умову, щоб розглянути приклад отримання заперечення висновку для іншої форми, ніж в попередній задачі.

Якщо в городі - бузина (A), то в Києві - дядько (B). Коні їдять овес (C) або в городі - бузина. Якщо коні їдять овес, то Дніпро впадає в Чорне море (E). ~~Дніпро не впадає в Чорне море~~. *(останню посилку викреслюємо). Висновок для перевірки:* <u>Якщо Дніпро не впадає в Чорне море, то в Києві – дядько</u> *(формальний вивід зупинився би на п.8 – див вище розв'язок прикладу 2.6).*

Формалізація:

1. $A \to B$	Г1	$(\neg A \vee B)$
2. $C \vee A$	Г2	так вже і є диз'юнкт
3. $C \to E$	Г3	$(\neg C \vee E)$
$\neg E \to B$	висновок	

Візьмемо заперечення висновку:

$\neg(\neg E \to B) = \neg(\neg\neg E \vee B) = \neg(E \vee B) = \neg E \& \neg B.$

Ми отримали два однолітерних негативних диз'юнкта: $\neg E$ *та* $\neg B$.

Запишемо множину диз'юнктів, яку будемо перевіряти на несуперечлівість: $\{\neg A \vee B, C \vee A, \neg C \vee E, \neg E, \neg B\}$.

1.	$\neg E$	*заперечення висновку*
2.	$\neg B$	*заперечення висновку*
3.	$\neg A \vee B$	*Г1*
4.	$C \vee A$	*Г*
5.	$\neg C \vee E$	*Г3*
6.	$\neg C$	*резольвента 1,5*
7.	A	*резольвента 4,6*

8. B   *резольвента 3,7*

9. □   резольвента 2,8

Пустий диз'юнкт отримано, система, що складається з гіпотез-умов задачі і заперечення висновку, суперечлива; висновок – дійсний.

Системи з формалізованих гіпотез мають цінність, якщо вони не є суперечливими. Це означає, що існує така інтерпретація пропозиційних літер, при якої всі гіпотези-посилки можуть одночасно дорівнювати TRUE. Несуперечливість системи можна перевірити і методом резолюцій, застосовуючи правило резолюцій Робінсона для пошуку пустого диз'юнкта (якщо він є – система суперечлива).
Але системи можуть розглядатися не тільки в якості опису умов задачі. Використати метод резолюцій для визначення суперечності чи ні системи посилок можна для перевірки логічного слідування (також визначення чи є формула тавтологією). В такому випадку система містить умови (які є несуперечливі) і заперечення висновку, який перевіряється на логічне слідування з посилок. В такому разі суперечливість системи (наявність пустого диз'юнкту) означає, що заперечення висновку - хибне, тобто сам висновок є логічним наслідком з умови задачі.

Завдання 2.3.1. Визначити, чи є наступні формули тавтологіями методом резолюції.

Приклад 1. Чи є формула $((A \to B) \to B) \to B$ *тавтологією*?

Формула складається з антецедента $(A \to B) \to B$ та консеквента B.

Перетворимо антецедент в кон'юнктивну форму, тобто в кон'юнкцію диз'юнктів для запису множини диз'юнктів:

$(A \to B) \to B = \neg(\neg A \vee B) \vee B = (A \& \neg B) \vee B = (A \vee B) \& (\neg B \vee B) =$

$= (A \vee B) \& T^{true} = A \vee B$, та візьмемо заперечення висновку.

Отримаємо множину *{ $A \vee B$, $\neg B$}*. Пустий диз'юнкт отримати не можливо, тому формула не є тавтологією.

Приклад 2. Чи є формула ($A \to B$) & ($B \to C$) & $A \to C$ тавтологією?
Складемо множину диз'юнктів з посилок і заперечення висновку (це можливо тоді, коли зовнішня формула є імплікацією, тоді висновок – останній консеквент). Посилки *($A \to B$) , ($B \to C$) , A,* та консеквент – C.

Тоді множина диз'юнктів буде наступною: *{ $\neg A \vee B$, $\neg B \vee C$, A , $\neg C$}.*

Ми бачимо, що пустий диз'юнкт отримати можна, отже, система є суперечливою (за рахунок додавання заперечення висновку), тобто формула – тавтологія.

Зауваження *. *$A \vee B$ та $\neg A \vee \neg B$ не дають пустий диз'юнкт. Вони не є запереченням один до одного.*

Якщо діяти за правилом Робінсона, то, обравши за основу для контрарних літер будь-яку з двох A або B, наприклад, $\neg A$ та A, отримаємо в якості резольвенти диз'юнкт, що побудований з літер, що залишилися після видалення контрарних, $B \vee \neg B$=True, що робить його зовсім непотрібним, але не приводить до пустого диз'юнкта.

1) (B → C) → (A ∨ C → (B → C));
2) (A → (B → C)) → (A ∨ B → C);
3) (A → B) → ((B → C) → (A → **C**));
4) (A → B) → ((C → B) → (A ∨ C → B));
5) (A → (B → C)) → ((A → B) → (A → C)).

Завдання 2.3.2. Дослідити систему посилок (гіпотез) на несуперечливість методом резолюції.

Приклад 1.

$\{F \to E \vee L,\ E \vee \neg F \& U,\ U \to E \vee L,\ \neg F \to \neg L,\ E \to \neg E\}$

Перетворимо кожну гіпотезу (посилки) у диз'юнкти і перевіримо на несуперечливість отриману множину диз'юнктів, використовуючи правило резолюції Робінсона по вилученню контрарних літер.

1. $F \to E \vee L = \neg F \vee E \vee L$ Г1
2. $E \vee \neg F \& U = (E \vee \neg F) \& (E \vee U)$:

   2.а $E \vee \neg F$

   2.б $E \vee U$ Г2
3. $U \to E \vee L = \neg U \vee E \vee L$ Г3
4. $\neg F \to \neg L = F \vee \neg L$ Г4
5. $E \to \neg E = \neg E \vee \neg E = \neg E$ Г5
6. $\neg F \vee L$ резольвента (1,5)
7. $\neg F$ резольвента (2.а,5)
8. U резольвента (2.б,5)
9. $\neg U \vee L$ резольвента (3,5)
10. $\neg L$ резольвента (4,7)
11. $\neg U$ резольвента (9,10)
12. $\square$ резольвента (8,11)

Отримано пустий диз`юнкт, отже система є суперечлівою.

Приклад 2.

$\{A \to B,\ C \vee A,\ C \to E,\ \neg E,\ \neg E \to \neg C\}$

Метод резолюції:

1. ~~$\neg A \vee B$ Г1~~

2. ~~$C \vee A$ Г2~~

3. ~~$\neg C \vee E$ Г3~~

4. $\neg E$ Г4

5. ~~$E \vee \neg C$ Г5 співпадає з Г3~~

6. $\neg C$ резольвента (3,4)

7. A резольвента (2,6)

8. B резольвента (1,7)

Пустий диз'юнкт отримати не можна, система не є суперечливою (викресленні диз'юнкти з унікальними літерами).

1) $\{ A \vee B \to C \& D,\ D \vee E \to G,\ A \vee \neg G$;

2) $\{(A \to B)\&(C \to D),\ (B \to D)\&(\neg C \to A),\ (E \to G)\&(G \to \neg D), \neg E \to E\}$;

3) $\{(A \to B\&C)\&(D \to B\&E),\ (G \to \neg A)\&H \to I,\ (H \to I) \to G\&D,\ \neg(\neg C \to E)\}$

Завдання 2.3.3. Перевірити логічне слідування методом резолюції.

Приклад. *Джон або втомився, або він хворий. Якщо він втомився, то він роздратований. Він не роздратований. Отже, Джон – хворий.*

Формалізуємо в логіці висловлювань гіпотези.

Нехай *A* – «Джон втомився», *B* – «Джон хворий», *C* – «Джон роздратований».

Формалізуємо умову:

1. $A \vee B$ $Г_1$

2. $A \to C$ $Г_2$

3. $\neg C$ $Г_3$

Можна перевірити логічне слідування **методом резолюції.** Для цього перетворимо кожну гіпотезу (посилку) у диз'юнкт, візьмемо заперечення висновку і складемо множину диз'юнктів, яку перевіримо на несуперечливість за правилом резолюції Робінсона.

Складемо систему: { $A \vee B$, $\neg A \vee C$, $\neg C$, $\neg B$}.

1. $A \vee B$	$Г_1$
2. $\neg A \vee C$	$Г_2$
3. $\neg C$	$Г_3$
4. $\neg B$	заперечення висновку
5. A	резольвента 1, 4
6. C	резольвента 2, 5
7. □	резольвента 3,6

Отримано пустий диз`юнкт, отже система, що складається з гіпотез і заперечення висновку, суперечлива; тобто логічне слідування є істинним.

Зауваження. Порівняйте з використанням методу резолюції в попередньому завданні – 2.3.2.*. Метод резолюції методично перевіряє систему на суперечливість шляхом пошуку за правилом Робінсона пустого диз'юнкта (якщо знайде □, то суперечлива; якщо ні – то несуперечлива). Глобальні висновки залежать від того, ЩО ми перевіряємо на суперечливість. Ми можемо перевіряти будь яку систему посилок (як в завданні 2.3.2.) – наприклад, систему правил бази знань, систему гіпотез для постановки задачі. В такому випадку наявність □ робить неможливим подальше використання системи, тому що вона є суперечливою. Інша справа – перевірка логічного слідування з множини гіпотез (поточна задача) – множина диз'юнктів, що перевіряється на суперечливість, складається з диз'юнктів гіпотез та з диз'юнктів, отриманих від заперечення запропонованого висновку. Тоді наявність □ – це добре, це означає, що система стала суперечливою при додаванні до неї заперечення* *висновку, отже, висновок – дійсний. А, якщо не отримано □, то висновок не є слідуванням з системи гіпотез.*

Приклад 2. *Малі діти нерозумні. Той, хто може приборкувати крокодилів, заслуговує на повагу. Нерозумні люди не заслуговують на повагу. Отже, малі діти не приборкують крокодилів.*

Нехай A – «малі діти», B – «розумні», C – «заслуговують на повагу», D - «можуть приборкувати крокодилів». Формалізуємо задачу:

1. $A \to \neg B$ Γ_1
2. $D \to C$ Γ_2
3. $\neg B \to \neg C$ Γ_3

Для перевірки висновка методом резолюції перетворимо кожну гіпотезу (посилку) у диз'юнкт, візьмемо заперечення висновку і складемо множину диз'юнктів, яку перевіримо на несуперечливість за правилом резолюції Робінсона.

Складемо систему: $\{ \neg A \vee \neg B, \neg D \vee C, B \vee \neg C, \boldsymbol{A}, \boldsymbol{D} \}$.

1. $\neg A \vee \neg B$	Γ_1
2. $\neg D \vee C$	Γ_2
3. $B \vee \neg C$	Γ_3
4. $\boldsymbol{A}$	**заперечення висновку*:***
5. $\boldsymbol{D}$	$\boldsymbol{\neg (A \to \neg D) = \neg (\neg A \vee \neg D) = A \& D}$
6. $\neg B$	*резольвента 1,4*
7. $\neg C$	*резольвента 3,6*
8. $\neg D$	*резольвента 2,7*
9. □	резольвента 5,8

Отримано пустий диз`юнкт, отже система, що складається з гіпотез і заперечення висновку, суперечлива; тобто логічне слідування є істинним.

1-8) Перевірити логічний висновок методом резолюції задач із **завдання 2.2.1.**

9) Кульки синього кольору (B) легко проколоти (P). Фірмові кульки (F) відрізняються особливою міцністю (¬P). Кульки у формі "серце" (H) бувають тільки фірмовими. Кульки у формі "серце" - завжди синього або сріблястого кольору (S). Зараз у нас тут (Z) всі кульки мають форму "серце". Отже, зараз у нас тут кульки сріблясті.

10) Дерева, які ростуть в цьому саду (D), плодоносять (P). Дерева, які плодоносять, дають хороший урожай (U). Дерева, що дають хороший урожай, отримують ретельний догляд (R). Жодне дерево в цьому саду не отримує ретельного догляду. Отже, в цьому саду не ростуть дерева (¬D).

11) Якщо результат перегонів буде вирішений змовою (Z), або в будинках будуть орудувати шулери (S), то доходи від туризму впадуть (T) і місто постраждає (M). Якщо доходи від туризму впадуть, то поліція буде задоволена (P). Поліція ніколи не буває задоволена. Отже, змови на перегонах не буде.

Для задач 9-11 провести також формальний висновок.

Відповіді до розділу 2

Відповіді 2.1.1. Тавтології 1,2,4,5,6; НІ – 3,7.

Відповіді 2.1.2.

1) нейтральна; 6) тавтологія;

2) протиріччя;

3) тавтологія;

4) тавтологія;

5) протиріччя;

7) нейтральна;

8) тавтологія;

9) протиріччя;

10) нейтральна.

Відповіді 2.1.3.

1) несуперечлива;

2) суперечлива;

3) суперечлива;

4) несуперечлива;

5) несуперечлива;

6) несуперечлива.

Розв'язок 2.1.3. 3)

$\{(A\rightarrow B\&C)\&(D\rightarrow B\&E), (G\rightarrow\neg A)\&H\rightarrow I, (H\rightarrow I)\rightarrow G\&D, \neg(\neg C\rightarrow E)\}$

Видалімо кон'юнкцію в першій гіпотезі; перетворювати посилки не потрібно.

Всі посилки повинні дорівнювати **T**.

1 $A\rightarrow B\&C$

2 $D\rightarrow B\&E$

3 $(G\rightarrow\neg A)\&H\rightarrow I$

4 $(H\rightarrow I)\rightarrow G\&D$

Розглянемо останню посилку $|\neg(\neg C\rightarrow E|=T$. Тоді $|\neg C\rightarrow E|=F$, і з цього слідує те, що записано в п. 5 і 6.

5 $|\neg C|=T \quad => |C|=F$

6 $|E|=F$

З того, що в 5 та 6 визначено значення C і E, з 1 та 2 => $|A|=F$, $|D|=F$, а B може приймати будь-яке значення.

З 4 через хибне значення D консеквент вже буде хибний, незалежно від значення G. Тоді для істинності посилки $|H \to I|=F \Rightarrow |H|=T, |I|=F$.

Підставляємо отримані значення в 3: $|(G\to\neg A)\&H\to I|=|(G\to\neg A)\&\boldsymbol{T}\to\boldsymbol{F}|=T$

Для виконання умови істинності посилки 3 треба мати $|G\to\neg A| = F$. Але у нас вже відомо значення A і $|\neg A|=T$, що робить одночасну здійсненність системи неможливою Система – суперечлива.

2.2.1.

1) ФОРМАЛІЗАЦІЯ і формальний висновок

1. $Z \to P$	Г1
2. $D \to P$	Г2
3. $P \to A$	Г3
4. $A \to I$	Г4
5. $I \to R$	Г5

Висновок: $Z \lor D \to R$

6. $P \to I$	силогізм 3, 4
7. $P \to R$	силогізм 6, 5

8. $(Z \to P)\&(D \to P)=(\neg Z \lor P)\&(\neg D \lor P)= (\neg Z \&\neg D) \lor P = \neg (Z \lor D) \lor P =$

$= (Z \lor D) \to P$ введення & та екв.перетворення 1 і 2

9. $Z \lor D \to R$ силогізм 8, 7

2) ФОРМАЛІЗАЦІЯ і формальний висновок

1. $V \to S \lor D$
2. $S \to \neg V \& P$
3. $P \to S \lor D$
4. V

Висновок : $\neg D \to S$

5. SvD =DvS = ¬D → S MP 1, 4 та екв.перетворення

Гіпотези 2 та 3 – зайві для отримання даного висновку.

3) ФОРМАЛІЗАЦІЯ і формальний висновок

1. V → Z
2. Z → B
3. ¬V → D
4. B → E
5. D → B
6. V v ¬V

Висновок : E

7. V → B силогізм 1, 2
8. V → E силогізм 7, 4
9. ¬V → B силогізм 3, 5
10. ¬V → E силогізм 9, 4
11. ¬E → ¬V контрапозиція 8
12. ¬E → V контрапозиція 10
13. ¬¬E=E доведення до абсурду і зняття подвійного заперечення 11,12

4) ФОРМАЛІЗАЦІЯ і формальний висновок (не виконується)

1. P → H&T = *¬Pv(H&T)=(¬PvH)&(¬PvT)= (P→H) (P→T)*
2. ¬P

Висновок : ¬H

3. P→H видалення & з 1
4. P→T видалення & з 1

5) ФОРМАЛІЗАЦІЯ і формальний висновок

1. Z→ I (тільки)

2. $I \rightarrow V$

3. $V \rightarrow \neg H$

Висновок: $Z \rightarrow \neg H$

4. $Z \rightarrow V$ силогізм 1, 2

5. $Z \rightarrow \neg H$ силогізм 4, 3

6) ФОРМАЛІЗАЦІЯ і формальний висновок

1. $Z \rightarrow KvYvP$ *якщо страйк завершено, то або конгрес прийняв певні рішення або президент пішов у відставку або страйк тривав більше року*

2. $\neg K$

3. $\neg P$

4. Z

Висновок: Y

5. $KvYvP$ MP 1, 4

6. $\neg K \rightarrow (\neg P \rightarrow Y)$ еквівал.перетворення 5

7. $\neg P \rightarrow Y$ MP 2, 6

8. Y MP 3, 7

7) ФОРМАЛІЗАЦІЯ і формальний висновок

1. AvB

2. $A \rightarrow C$

3. $B \rightarrow DvC$

4. $\neg C$

Висновок: D

5. $\neg A \rightarrow B$ екв.перетв.1

6. $\neg A$ MT 2, 4

7. $\neg A \rightarrow D \vee C$ силогізм 3, 5

8. $D \vee C$ MP 6, 7

9. $\neg D \rightarrow C$ екв.перетв.8

10. $\neg C \rightarrow D$ контрапозиція 9

11. D MP 4, 10

8) ФОРМАЛІЗАЦІЯ і формальний висновок

1. $\neg B \to (C \vee P)$ Г1

2. $\neg C \to (\neg B \& M)$ $= C \vee (\neg B \& M) = (C \vee \neg B) \& (C \vee M) =$

$= (\neg C \to \neg B) \& (\neg C \to M)$ Г2

3. $M \to (C \vee \neg P)$ Г3

Висновок: C

4. $\neg C \to \neg B$ екв. та видалення & 2

5. $\neg C \to M$ екв. та видалення & 2

6. $\neg C \to C \vee P = C \vee C \vee P = C \vee P = \neg C \to P$ сил. 4, 1, екв. та ідемпот.

7. $\neg C \to C \vee \neg P = C \vee C \vee \neg P =$

$= C \vee \neg P = \neg C \to \neg P$ сил. 5, 3, екв. та ідемпот.

8. $\neg\neg C = C$ абсурд 6,7 та зняття подвійного заперечення

2.3.1.

1) $(B \to C) \to (A \vee C \to (B \to C))$:

Якщо ми допускаємо, що формула – тавтологія, то можемо «розібрати» її за правилами ОМТД: $B \to C, A \vee C, B \ \vdash C$

Отримаємо множину диз'юнктів з посилок та заперечення висновку:

$\{ \neg B \vee C , A \vee C , B , \neg C\}$.

Ми бачимо, що пустий диз'юнкт отримати можна (навіть без другої посилки, вона – зайва), отже, система є суперечливою (за рахунок додавання заперечення висновку), тобто формула – тавтологія.

2) $(A \to (B \to C)) \to (A \vee B \to C)$:

Розберемо формулу так, як би вона була тавтологією (відокремлюємо комами подформули, рухаючись зліва направо по черзі, починаючи з самої зовнішньої операції):

1) $(A \to (B \to C)) \to (A \vee B \to C)$;

2) $A \to (B \to C)$, $(A \vee B \to C)$;

3) $A \to (B \to C)$, $A \vee B$

та висновок, що перевіряється в нашому випадку C.

Таким чином перетворюючи першу посилку та взявши заперечення висновку отримаємо множину диз'юнктів $\{\neg A \vee \neg B \vee C, A \vee B, \neg C\}$.

Спробуємо застосувати правило Робінсона і отримати резольвенти:

1. $\neg C$ – заперечення висновку
2. $\neg A \vee \neg B \vee C$ – посилка (гіпотеза) 1
3. $A \vee B$ – посилка (гіпотеза) 2
4. $\neg A \vee \neg B$ – резольвента 1,2
5. $B \vee \neg B$ – резольвента 3,4 *

Пустий диз'юнкт отримати не можна, тобто, формула не є тавтологією.

3) $(A \to B) \to ((B \to C) \to (A \to \mathbf{C}))$:

$\{ A \to B, B \to C, A, –C\}$

$\{ –A \text{ v } B, –B \text{ v } C, A, –C\}$

1	–A v B	Г1
2	–B v C	Г2
3	A	Г3

4 –C заперечення висн.

5 B резольвента 1,3

6 C резольвента 2,5

7 □ резольвента 4,6

4) $(A \to B) \to ((C \to B) \to (A \vee C \to B))$:

1 $A \to B = \neg A \vee B$ Г1

2 $C \to B = \neg C \vee B$ Г2

3 $A \vee C$ Г3

4 $\neg B$ заперечення висновку

5 $\neg A$ резольвента 1, 4

6 $\neg C$ резольвента 2, 4

7 C резольвента 3, 5

8 □ резольвента 6, 7

5) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$:

$A \to (B \to C),\ A \to B \models A \to C$

$\neg (A \to C) = \neg (\neg A \vee C) = A \& \neg C$

$\{\neg A \vee \neg B \vee C,\ \neg A \vee B,\ A,\ \neg C\}$

Пустий диз’юнкт отримати можна, формула – тавтологія.

2.3.2.

2) $\{ A \vee B \to C \& D,\ D \vee E \to G,\ A \vee \neg G\}$

~~1 –AvC~~

2 –AvD

~~3 –BvC~~

~~4 –BvD~~

5 –DvG

~~6 –EvG~~

7 Av–G

8 –AvG резольвента 2,5

9 –AvA або –GvG (резольвента 7,8) = тавтології

Пустий диз'юнкт не отримано, система не є суперечливою.

3) {(A →B)&(C →D), (B→ D)&(¬C →A), (E →G)&(G →¬D),¬E →E}

1	–AvB	
2	–CvD	
3	–BvD	
4	CvA	
5	–EvG	
6	–Gv–D	
7	E	
8	G	резольвента 7,5
9	–D	резольвента 8,6
10	–C	резольвента 9,2
11	–B	резольвента 9,3
12	–A	резольвента 11,1
13	C	резольвента 12,4
14	□	резольвента 10,13

Система – суперечлива, вона не має цінності для розв'язку задачі.

3) {(A→ B&C)&(D→ B&E), (G→¬A)&H→I, (H →I)→ G&D, ¬(¬C→ E)}

Перетворення:

1 A→ B = ¬A v B

2 A→ C = ¬A v C

3 D→ B = ¬D v B

4 D→ E = ¬D v E

5 (G→¬A)&H→I =¬ ((¬G v ¬A)&H) v I= ¬ (¬ G v¬ A) v¬ H vI =

= (G&A) v ¬ H v I = (G v ¬ H v I) & (A v ¬ H v I)

6 (H →I)→ G&D = ¬ (¬ H v I) v G&D = H&¬I v G&D =

=(H v G)&(H v D)&(¬I v G)&(¬I v D)

7 ¬(¬C→ E) = ¬ (C v E) = ¬ C & ¬ E

РЕЗОЛЮЦІЯ

~~1 ¬AvB~~

2 ¬AvC

~~3 ¬DvB~~

4 ¬DvE

~~5 G v ¬ H v I~~

6 A v ¬ H v I

~~7 H v G~~

8 H v D

~~9 ¬I v G~~

10 ¬I v D

11 ¬ C

12 ¬ E

Викреслено унікальні літери, але порожній диз'юнкт отримати можна – система суперечлива.

13 ¬A	резольвента 2, 11
14 ¬H v I	резольвента 6, 13
15 ¬D	резольвента 4, 12
16 H	резольвента 8, 15
17 ¬I	резольвента 8, 10
19 ¬H	резольвента. 14, 17
20 □	резольвента 16, 19

2.3.3.

1) 1. ¬Z v P Г1

2. ¬D v P Г2

3. ¬P v A Г3

4. ¬A v I Г4

5. ¬I v R Г5

Отримаємо заперечення висновку

¬ (Z v D → R) = ¬ (¬ (Z v D) v R) = (Z v D) & ¬R

6. Z v D запереч.висновку

7. ¬R запереч.висновку

8. ¬I резольвента 7, 5

9. ¬A резольвента 4, 8

10. ¬P резольвента 3, 9

11. ¬Z резольвента 1, 10

12. ¬D резольвента 2, 10

13. D резольвента 6, 11

14. □ резольвента 12, 13

2) 1. ¬VvSvD

2. V

¬ (¬D →S)= ¬ (DvS)= ¬D&¬S заперечення висновку

3. ¬D

4.¬S

Далі, використовуючи правило Робінсона, можна отримати пустий диз'юнкт без застосування гіпотез 2 і 3.

3) 1. ¬E

2. ¬V v Z

3. ¬Z v B

4. V v D

5. ¬B v E

6. ¬D v B

~~7. V v V~~

8. ¬B резольвента 1, 5

9. ¬Z резольвента 3, 8

10. ¬V резольвента 2, 9

11. D резольвента 4, 10

12. B резольвента 6, 11

13. □ резольвента 8, 12

4) 1. H

2. ¬P

3. ¬P v H

4. ¬P v T

Жодної резольвенти отримати не можна.

5) 1. Z заперч.висн.

2. H запереч.висн.

3. ¬Z v I Г1

4. ¬I v V Г2

5. ¬V v ¬H Г3

6. I резольвента 1, 3

7. V резольвента 4, 6

8. ¬H резольвента 5, 7

9. □ резольвента 2, 8

6) 1. ¬Y запереч.висновку

2. ¬Zv KvYvP Г1

3. ¬K Г2

4. ¬P Г3

5. Z Г4

6. KvYvP резольвента 2, 5

7. YvP резольвента 3, 6

8. P резольвента 1, 7

9. □ резольвента 4, 8

7) 1. ¬D заперечення висновку

2. A v B Г1

3. ¬A v C Г2

4. ¬B v D v C Г3

5. ¬C Г4

6. ¬B v C резольвента 1, 4

7. A v C резольвента 2, 6

8. C резольвента 3, 7

9. □ резольвента 5, 8

8) 1. ¬C заперечення висновку

2. B v C v P Г1

3. C v ¬B Г2

4. C v M Г2

5. ¬M v C v ¬P Г3

6. B v P резольвента 1, 2

7. ¬B резольвента 1, 3

8. M резольвента 1, 4

9. ¬M v ¬P резольвента 1, 5

10. P резольвента 6, 7

11. ¬M резольвента 10, 9

12. □ резольвента 8, 11

9) 1. B → P

2. F → ¬P

3. H → F

4. H → B v S

5. Z → H ВИСНОВОК: Z → S

Формальний висновок:

6. $Z \rightarrow F$ сил. 5, 3

7. $Z \rightarrow \neg P$ сил. 6, 2

8. $\neg P \rightarrow \neg B$ контр. 1

9. $Z \rightarrow \neg B$ сил. 7, 8

10. $Z \rightarrow B \vee S$ сил. 4, 5

11. $\neg Z \vee B \vee S = \neg B \rightarrow (Z \rightarrow S)$ екв.перетв. 10

12. $Z \rightarrow (Z \rightarrow S) = \neg Z \vee \neg Z \vee S = \neg Z \vee S = \mathbf{Z \rightarrow S}$

сил. 9, 10+екв.перетв.

Резолюція [*заперечення висновку* $\neg (Z \rightarrow S) = \neg (\neg Z \vee S) = \mathbf{Z \,\&\, \neg S}$]:

1. $\neg B \vee P$ Г1
2. $\neg F \vee \neg P$ Г2
3. $\neg H \vee F$ Г3
4. $\neg H \vee B \vee S$ Г4
5. $\neg Z \vee H$ Г5
6. Z запереч.висновку
7. $\neg S$ запереч.висновку
8. H резольвента 5, 6
9. F резольвента 3, 8
10. $B \vee S$ резольвента 4,8
11. B резольвента 7,10
12. P резольвента 1, 11
13. $\neg F$ резольвента 2, 12
14. $\square$ резольвента 9, 13

10) Формалізація:

1. $D \to P$

2. $P \to U$

3. $U \to R$

4. $D \to \neg R$

Висновок: $\neg D$

Вказівка для формального виведення: З гіпотез 1-3 отримаємо через силогізми $D \to R$. Разом з гіпотезою 4 через правило доведення до абсурду отримаємо висновок.

11) Формалізація:

1. $Z \vee S \to T\&M$

2. $T \to P$

3. $\neg P$

Висновок: $\neg Z$

Вказівка для формального виведення: перша гіпотеза через еквівалентні перетворення розбивається на чотири дволітерні диз'юнкти.

$Z \vee S \to T\&M = \neg (Z \vee S) \vee (T\&M) = (\neg Z \& \neg S) \vee (T\&M) =$

$= (\neg Z \vee T) \& (\neg Z \vee M) \& (\neg S \vee T) \& (\neg S \vee M) =$

$= (Z \to T) \& (Z \to M) \& (S \to T) \& (S \to M)/$

3. ФОРМАЛЬНІ ТЕОРІЇ

Теоретичні відомості

Теорія називається *змістовною,* якщо існує яка-небудь інтерпретація об'єктів, операцій, символів теорії. Алгебра висловлювань є змістовною теорією, оскільки кожен символ в ній інтерпретується як просте висловлювання, істинне або хибне.

Формальні теорії будуються за наступними правилами.

Задається зчисленна множина символів, що називається *алфавітом* теорії. З цієї множини символів будуються зчисленні послідовності, які називаються *словами* або *виразами* даної теорії. З множини виразів виділяють *правильно побудовані вирази – формули*. З множини формул виділяють підмножину *аксіом*. Між формулами теорії визначають відношення – *правила виведення* теорії. Правила виведення дозволяють з множини аксіом виводити *теореми*. Таким чином, формальна теорія є аксіоматичною теорією.

Визначення 3.1. *Доведенням* теореми називають послідовність формул $A_1,\ \ldots,\ A_n$, кожна з яких або є аксіомою, або отримана з попередніх за правилами виведення цієї теорії. Остання формула називається *теоремою* цієї теорії. Іншими словами, *теорема* - це формула, що виводиться з множини аксіом за правилами виведення, визначеними в цій теорії. Запис $|-_J A_n$ означає, що формула A_n виводиться в теорії J, тобто є теоремою теорії J.

Визначення 3.2. *Виведенням* формули A_n з множини гіпотез Г називається послідовність формул $A_1,\ \ldots,\ A_n$, кожна з яких є або аксіомою, або гіпотезою з множини Г, або отриманою з попередніх по *правилах виведення*. Формула $B = A_n$ називається такою, що виводиться з множини гіпотез Г. Означення Г $|-_J A_n$ означає, що формула A_n виводиться з множини гіпотез Г в теорії J.

Доведення відрізняється від виведення тим, що в ньому не використовуються гіпотези, тому теорему можна визначити як формулу, що виводиться з порожньої множини гіпотез.

Визначення 3.3.

1. *Символами алфавіту* теорії L є пропозиціональні літери $A, B, C, \ldots$ з індексами або без індексів, пропозиціональні зв'язки $\neg, \rightarrow$, та допоміжні символи – дужки: (та).

2. Визначення формули.

Будь-яка пропозиціональна літера є формула.

Якщо A та B є формулами, то формулами є $(\neg A), (A \rightarrow B)$.

Інших формул немає.

3. У формальній теорії L визначена нескінченна кількість аксіом, що задаються трьома схемами аксіом:

A1: $A \rightarrow (B \rightarrow A)$;

A2: $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$;

A3: $(\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B)$.

Конкретні аксіоми виходять підстановкою формули замість пропозиціональної літери.

4. В теорії L визначено єдине правило виводу MP:

$A, A \rightarrow B \ \vdash B$.

Для скорочування запису формул вводяться метавизначення:

MB1: $\neg(A \rightarrow \neg B) = A \& B$.

MB2: $\neg A \rightarrow B = A \vee B$.

MB3: $(A \rightarrow B) \& (B \rightarrow A) = A \equiv B$.

Теорема 1. $\vdash A \rightarrow A$ (закон тотожності - висловлювання є логічним наслідком самого себе)

Теорема 2. $\vdash (A \to A) \to A$

В1. Правило силогізму.

$A \to B, B \to C \vdash A \to C$

В2. Правило видалення середнього посилання.

$A \to (B \to C), B \vdash A \to C$

В3. Правило видалення крайнього посилання.

$(A \to B) \to C \vdash B \to C$.

Метатеорема про дедукцію (МТД).

Якщо з множини формул Г і формули A виводима формула B, то з множини Г виводима формула $A \to B$, тобто якщо Г, $A \vdash B$, то Г $\vdash A \to B$.

Обернена метатеорема про дедукцію.

Якщо існує вивід Г $\vdash A \to B$, то формула B виводиться з Г і A, т.е. якщо Г $\vdash A \to B$, то Г, $A \vdash B$

Теорема 3 (зняття подвійного заперечення). $\vdash \neg\neg A \to A$

Теорема 4 (введення подвійного заперечення). $\vdash A \to \neg\neg A$

Теорема 5 (протиріччя/суперечність). $\vdash \neg A \to (A \to B)$

Теорема 6 (контрапозиції). $\vdash (\neg A \to \neg B) \to (B \to A)$

Теорема 7 (контрапозиції). $\vdash (B \to A) \to (\neg A \to \neg B)$

Теорема 8. $\vdash A \to (\neg B \to \neg(A \to B))$

Теорема 9. $\vdash (A \to B) \to ((\neg A \to B) \to B)$

Приклад 1.Метавизначення $\vdash (\mathbf{A} \to \mathbf{B}) \to \neg \boldsymbol{A} \cup \boldsymbol{B}$

ОМТД 2 рази і МВ 2 $(\neg A \to B = A \cup B)$ $A \to B, \neg\neg A \vdash B$

1.	$A \to B$	Г1
2.	$\neg\neg A$	Г2
3.	$\neg\neg A \to A$	Т3
4.	A	МР(2,3)
5.	B	МР(1,4)

Приклад 2. $\vdash \neg(\boldsymbol{A} \cup \boldsymbol{B}) \to \neg\boldsymbol{A}\&\neg\boldsymbol{B}$ **– Закон де Моргана**

ОМТД $\quad \neg(A \cup B) \vdash \neg A\&\neg B$

МО1, МО2 $\quad \neg(\neg A \to \mathrm{B}) \vdash (\neg A \to \neg\neg\mathrm{B})$

Доведемо допоміжну лему

ЛЕМА $\quad \vdash (\neg A \to \neg\neg\mathrm{B}) \to (\neg A \to \mathrm{B})$

ОМТД $(\neg A \to \neg\neg\mathrm{B}), \mathrm{A} \vdash B$

1. $\neg A \to \neg\neg\mathrm{B}$ Г1
2. $\neg A$ Г2
3. $\neg\neg\mathrm{B}$ МР(1,2)
4. $\neg\neg\mathrm{B} \to \mathrm{B}$ Т3
5. В МР(3,4)

Продовжимо доведення початкової теореми

1. $\neg(\neg A \to \mathrm{B})$ Г1
2. $\vdash (\neg A \to \neg\neg\mathrm{B}) \to (\neg A \to \mathrm{B})$ Лема
3. $\big((\neg A \to \neg\neg\mathrm{B}) \to (\neg A \to \mathrm{B})\big) \to \big(\neg(\neg A \to \mathrm{B}) \to \neg(\neg A \to \neg\neg\mathrm{B})\big)$ Т7
4. $\big(\neg(\neg A \to \mathrm{B}) \to \neg(\neg A \to \neg\neg\mathrm{B})\big)$ МР(2,3)
5. $\neg(\neg A \to \neg\neg\mathrm{B})$ МР(1,4)

Приклад 3. $\vdash ((\boldsymbol{A} \to \mathbf{B}) \to \mathbf{A}) \to \mathbf{A}$ **– Закон Пірса**

ОМТД $\quad (A \to \mathrm{B}) \to \mathrm{A} \vdash A;$

1. $(A \to \mathrm{B}) \to \mathrm{A}$ Г1
2. $\vdash \neg\mathrm{A} \to (A \to \mathrm{B})$ Т5
3. $\neg\mathrm{A} \to \mathrm{A}$ В1 (1,2)
4. $(\neg\mathrm{A} \to \mathrm{A}) \to \mathrm{A}$ Т2
5. А МР 3,4

Завдання 3.1. Довести теореми теорії L.

1) $\vdash ((\mathrm{A} \to \mathrm{B}) \to \mathrm{A}) \to ((\mathrm{A} \to \mathrm{B}) \to \mathrm{B});$

2) $\vdash (A \to C) \to (A \to B \vee C)$;

3) $\vdash (\neg A \to B) \to (\neg B \to A)$;

4) $\vdash (B \to C) \to (B \to (A \to C))$;

5) $\vdash (B \to A) \to (A \vee B \to A)$;

6) $\vdash (A \to (B \to C)) \to (B \to (A \to C))$;

7) $\vdash A \,\&\, B \to (C \to B)$;

8) $\vdash ((B \to A) \to C) \to (A \to C)$;

9) $\vdash (A \to B) \,\&(B \to C) \to (A \to C)$;

10) $\vdash (A \to B) \to ((B \to C) \to (A \to C))$;

11) $\vdash (A \to B) \,\&\, (C \to D) \to (A \vee C \to B \vee D)$;

12) $\vdash (A \vee B \to C) \to (A \to C) \vee (B \to C)$;

13) $\vdash (A \to B) \,\&\, (C \to D) \to (A \,\&\, C \to B \,\&\, D)$;

14) $\vdash \neg(A \,\&\, B) \to (A \,\&\, B \to B)$;

15) $\vdash (B \to C) \to (A \vee C \to (B \to C))$;

16) $\vdash (A \to C) \,\&\, (B \to C) \to (A \vee B \to C)$;

17) $\vdash (A\to B \vee C) \to (\neg A \vee B \vee C)$;

18) $\vdash (A \vee B \to C) \to (B \to C)\&(A \to C)$;

19) $\vdash (A\to B\&C) \to (A \to B)\&(A \to C)$;

20) $\vdash (A \to (B \to C)) \to (A \,\&\, B \to C)$.

Відповіді до розділу 3

3.1.

1) $\vdash ((A \to B) \to A) \to ((A \to B) \to B)$;

$(A \to B) \to A,\ A \to B \vdash B$ 2 рази ОМТД

1 $(A \to B) \to A$ Г1

2 $A \to B$ Г2

3 A MP 1,2

4 B MP 2,3

2) $\vdash (A \to C) \to (A \to B \lor C)$;

$A \to C, A \vdash B \lor C$ 2 рази ОМТД

$A \to C, A \vdash \neg B \to C$ МВ

$A \to C, A, \neg B \vdash C$ ОМТД

1 $A \to C$ Г1

2 A Г2

3 $\neg B$ Г3

4 C MP 1,2

3) $\vdash (\neg A \to B) \to (\neg B \to A)$;

$\neg A \to B, \neg B \vdash A$ 2 РАЗИ ОМТД

1 $\neg A \to B$ Г1

2 $\neg B$ Г2

3 $(\neg A \to B) \to (\neg B \to \neg\neg A)$ T7

4 $\neg B \to \neg\neg A$ MP 1,3

5 $\neg\neg A \to A$ T3

6 $\neg B \to A$ В1(сил) 4,5 **або** *6 $\neg\neg A$ MP 2,4*

7 A MP 2,6 *7 A MP 5,6*

4) $\vdash (B \to C) \to (B \to (A \to C))$;

$B \to C, B, A \vdash C$ 3 рази ОМТД

1 $B \to C$ Г1

2 B Г2

3 A Г3

4 C MP 1, 2

5) $\vdash (B \to A) \to (A \lor B \to A)$;

$B \to A, A \lor B \vdash A$ 2 рази ОМТД

1 $B \to A$ Г1

2 $A \lor B$ Г2

3 $\neg A \to B$ МВ 2

4 $\neg A \to A$ В1(сил) 1,3

5 $(\neg A \to A) \to A$ Т2

6 A МР 4,5

або

4 $(B \to A) \to (\neg A \to \neg B)$ Т7

5 $\neg A \to \neg B$ МР 1,4

6 $(\neg A \to \neg B) \to ((\neg A \to B) \to A)$ А3

7 $(\neg A \to B) \to A$ МР 5, 6

8 A МР 3, 7

11) $\vdash (A \to B) \& (C \to D) \to (A \lor C \to B \lor D)$;

$A \to B, C \to D, A \lor C \vdash B \lor D$ 2 рази ОМТД, видалення &

$A \to B, C \to D, \neg A \to C \vdash \neg B \to D$ МВ

$A \to B, C \to D, \neg A \to C, \neg B \vdash D$ ОМТД

1 $A \to B$ Г1

2 $C \to D$ Г2

3 $\neg A \to C$ Г3

4 $\neg B$ Г4

5 $\neg A \to D$ В1 (сил) 2,3

6 $(A \to B) \to (\neg B \to \neg A)$ Т7

7 $\neg B \to \neg A$ МР 1, 6

або два МР 4-7 та 8-5 // або В1 7-5 та МР 4-8

8 $\neg A$ МР 4, 7 // 8 $\neg B \to D$ В1 7, 5

9 D MP 5, 8 // 9 D MP 4, 8

12) $\vdash (A \vee B \to C) \to (A \to C) \vee (B \to C)$;

$A \vee B \to C \vdash (A \to C) \vee (B \to C)$ ОМТД

$(\neg A \to B) \to C \vdash \neg (A \to C) \to (B \to C)$ МВ

$(\neg A \to B) \to C, \neg (A \to C) \vdash (B \to C)$ ОМТД

1 $(\neg A \to B) \to C$ Г1

2 $\neg (A \to C)$ Г2

3 $B \to C$ В3 (крайн) 1

13) $\vdash (A \to B) \& (C \to D) \to (A \& C \to B \& D)$;

$A \to B, C \to D, A, C \vdash B \& D$ 2 рази ОМТД

1 $A \to B$ Г1

2 $C \to D$ Г2

3 A Г3

4 C Г4

5 B MP 1,3

6 D MP 2,4

7 B & D введення & 5,6

16) $\vdash (A \to C) \& (B \to C) \to (A \vee B \to C)$;

$A \to C, B \to C, A \vee B \vdash C$ 2 рази ОМТД

1 $A \to C$ Г1

2 $B \to C$ Г2

3 $A \vee B$ Г3

4 $\neg A \to B$ МВ 3

5 $\neg A \to C$ В1 4, 2

6 $(\neg A \to C) \to ((A \to C) \to C)$ Т9

7 $(A \to C) \to C$ MP 5, 6

8 C MP 1, 7

17) $\vdash (A \to B \vee C) \to (\neg A \vee B \vee C)$;

$A \to B \vee C \vdash \neg A \vee B \vee C$ ОМТД

1 $A \to B \vee C$ Г1

2 $\neg\neg A \to A$ Т3

3 $\neg\neg A \to B \vee C$ В1 1, 2

4 $\neg A \vee B \vee C$ МВ 3

або

$A \to B \vee C \vdash \neg A \vee B \vee C$ ОМТД

$A \to (\neg B \to C) \vdash \neg\neg A \to (\neg B \to C)$ МВ

$A \to (\neg B \to C), \neg\neg A, \neg B \vdash C$ 2 рази ОМТД

1 $A \to (\neg B \to C)$ г1

2 $\neg\neg A$ Г2

3 $\neg B$ Г3

4 $\neg\neg A \to A$ Т3

5 $\neg\neg A \to (\neg B \to C)$ В1 1, 4

6 $\neg B \to C$ MP2, 5

7 C MP 3,6

18) $\vdash (A \vee B \to C) \to (B \to C) \& (A \to C)$;

Доводимо окремо, як у 18, для $B \to C$ та $A \to C$.

Для $B \to C$ – через метавизначення та видалення крайньої посилки

Доведемо для:

$A \vee B \to C \vdash A \to C$

1 $A \vee B \to C$ Г1

2 $(\neg A \to B) \to C$ МВ 1

3 $(\neg B \to A) \to (\neg A \to B)$ правило контрапоз.

(або можна використати комутативний закон –

треба його теж довести $A \vee B \to B \vee A$)

4 $(\neg B \to A) \to C$ В1 2, 3

5 $A \to C$ В3 4

4. АБСТРАКТНІ АВТОМАТИ

Теоретичні відомості

Теорія автоматів відноситься до числа ключових розділів сучасної кібернетики. Вона безпосередньо пов'язана з математичною логікою, теорією алгоритмів і теорією формальних граматик.

У техніці з поняттям автомату пов'язаний деякий пристрій, здатний виконувати деякі функції без втручання людини або з обмеженням його участі. Проте таке розуміння автомату є занадто вузьким. У загальному розумінні скінченний автомат – це математична модель, що відображає фізичні або абстрактні явища найрізноманітнішої природи.

Основні визначення теорії скінченних автоматів:

– кінцева сукупність різних символів називається *алфавітом*;

– кожен окремий символ алфавіту - *буквою*;

– послідовність літер, що має кінцеву довжину - *словом*;

– число букв в слові - *довжиною*;

– два алфавіти називаються *рівнозначними*, якщо між буквами цих алфавітів можна встановити взаємно-однозначну відповідність.

Будемо розглядати абстрактний скінченний автомат у вигляді «чорного ящика» з одним входом і одним виходом, тобто аналізований пристрій описується на рівні завдання залежності значень на його виході від значень на його вході.

Робота автомата розглядається в дискретні моменти часу ti, звані автоматними тактами (i змінюється від 0 до n, де n - довжина вхідного слова). У загальному випадку скінченний автомат визначає пристрій, який реалізує відображення множини слів вхідного алфавіту в множину слів вихідного алфавіту.

Отже, абстрактними автоматами ми будемо називати дискретні перетворювачі інформації, що видають деякий сигнал виходу (літеру алфавіту виходу) у відповідь на деякий вхідний сигнал (літеру вхідного алфавіту). Внутрішня структура автомату не розглядається.

Розрізняють два класи абстрактних автоматів:

– автомат без вихідного перетворювача;

– автомат з вихідним перетворювачем.

Автомат з вихідним перетворювачем. Автомати Мілі і Мура

Автомат з вихідним перетворювачем це є сукупність наступних шести об'єктів: $\boldsymbol{A} = \langle X, A, A0, \sigma, Y, \lambda \rangle$, де

X = { x1, x2, ..., xn} - вхідний алфавіт;

A = {A0, A1, A2, ..., Ak} - множина станів; A0 - початковий стан автомата;

σ - функція переходу автомата;

Y = { y1, y2, ..., ym} - вихідний алфавіт;

λ - функція виходу.

Визначення об'єктів X , A , A0, σ збігається з визначенням тих же об'єктів в автоматі без вихідного перетворювача.

В загальному вигляді закони функціонування автоматів з перетворювачем задаються рівняннями:

а) для автоматів першого роду (автоматів Мілі)

$$a(t) = \sigma(a(t-1), x(t)),$$

$$y(t) = \lambda(a(t-1), x(t));$$

б) для автоматів другого роду

$$a(t) = \sigma(a(t-1), x(t)),$$

$$y(t) = \lambda(a(t), x(t));$$

в) для правильних автоматів другого роду (автоматів Мура) вихідні сигнали залежать тільки від стану автомату

$$a(t) = \sigma(a(t-1), x(t)),$$
$$y(t) = \lambda(a(t)).$$

Розглянемо дві найбільш поширені математичні моделі автомата з вихідним перетворювачем: автомат Мура і автомат Мілі. Визначення функції виходу λ є різним в кожній з цих моделей.

В *автоматі Мура* функція виходу λ здійснює відображення множини A на множину Y, тобто A→Y. Це означає що кожному стану Ai ставиться у відповідність вихідний символ yi. Функція переходу записується в такий спосіб: yi = λ (Ai), де Ai - поточний стан автомата; yi - вихідний символ.

Істотним у визначенні функції виходу автомата Мура є те, що новий вихідний символ yi (t+1) в такті (t+1) визначається наступним станом Ai (t+1), в який переходить автомат. Тобто, yi (t+1) = λ(Ai (t+1)).

В *автоматі Мілі* функція виходу λ здійснює відображення декартова добутку множин X і A на множину Y, тобто X × A →Y.

Це означає, що кожній парі вхідних символів xi і стану Aj ставиться у відповідність вихідний символ yk. Функція виходу записується в такий спосіб: yk = λ (xi, Aj),), де xi - вхідний символ; Aj - поточний стан автомата; yk - вихідний символ.

У функції виходу автомата Мілі новий вихідний символ yk (t+1) в такті (t+1) визначається вхідним символом xi (t) і станом Ai (t) в поточному такті (t): yk (t+1) = λ (xi (t), Aj (t)).

Способи завдання автомата Мура.

Функції переходу і функції виходу можуть бути представлені трьома способами:

– перерахуванням;

– табличним;

– графічним.

При перерахуванні наводяться всі функції переходу і виходу:

$A1 = \sigma (A0, x1)$, $y1 = \lambda (A1)$,

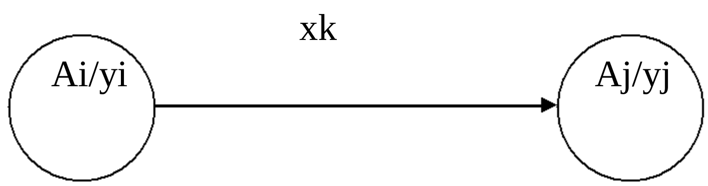
$A2 = \sigma (A1, x1)$, $y2 = \lambda (A2)$,

....

$Am = \sigma (Ak , xn)$. $ym = \lambda (Am)$.

У *табличному способі* будується таблиця переходів, яка зверху доповнюється рядком, в якому для кожного стану Aj вказується відповідний йому вихідний символ yj.

При *графічному способі* кожному станому автомата ставиться в відповідність вершина графа, яка позначається символом цього стану Ai і вихідним символом yi, відповідним цього стану. Якщо зі стану Ai існує перехід в стан Aj під впливом вхідного символу xk, то вершини Ai і Aj з'єднуються дугою що виходить з Ai , а сама дуга позначається символом xk під впливом, якої має реалізуватись перехід.



У момент t0 значення y0 не враховується, так як не є реакцією на вхідний символ, а визначається A0 - початковим станом автомата Мура ще до подачі на вхід першого символу вхідної послідовності.

Способи завдання автомата Мілі

Функції переходу і функції виходу можуть бути представлені трьома способами, як і в автоматі Мура:

– перерахуванням;

– табличним;

– графічним.

При перерахуванні наводяться всі функції переходу і виходу:

A1 = σ (A0, x1),   y1 = λ (A0, x1),

A2 = σ (A1, x1),   y2 = λ (A1, x1),

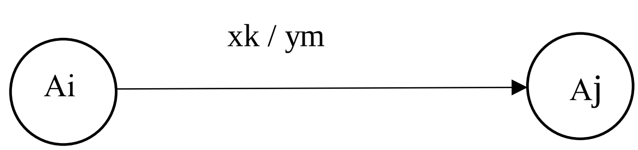
....

Am = σ (Ak , xn).   ym = λ (Ak , xn).

Функція переходу задається як в автоматі Мура. Відмінність полягає в завданні функції виходу.

При *табличному способі* будується таблиця переходів і аналогічна їй таблиця виходів, на перетині і - того рядка *і* j - того стовпця якій, вказується вихідний символ yk , який видає автомат при переході зі стану Aj під впливом вхідного символу xi. Таблиці переходів і виходів можна об'єднувати, і виходить поєднана таблиця переходів і виходу, де на перетині рядка та стовпця ставиться через будь-який розділювач два значення – новий стан і вихідний сигнал, наприклад, A2/y3.

При графічному способі кожному стану автомата ставиться в відповідність вершина графа, яка позначається символом цього стану Ai . Якщо зі стану Ai існує перехід в стан Aj під впливом вхідного символу xk, то вершини Ai і Aj з'єднуються дугою, що виходить із Ai, а сама дуга позначається символом xk, під впливом якої має реалізуватись перехід, і вихідним символом ym, який виробляється при цьому переході.



Робота автомата Мілі полягає в перетворенні вхідного слова в вихідний. Відмінність в роботі автоматів Мура і автоматів Мілі в тому, що в автоматі Мура вихідний символ можна зчитувати відразу після установки його в початковий стан і подачі на його вхід першого вхідного символу, але при цьому треба враховувати, що цей символ не є реакцією на вхідний символ, а визначається тільки початковим станом автомата Мура.

Еквівалентні перетворення автоматів

Два автомата називаються еквівалентними, якщо вони мають однакові вхідні і вихідні алфавіти і на однакові вхідні слова видають однакові вихідні слова.

Перехід від автомата Мура до еквівалентного автомата Милі:

З аналізу переходів автоматів Мілі і Мура слідує, що для переходу від автомата Мілі до автомата Мура необхідно як би віднести кожен вихідний сигнал до попереднього стану і вхідного сигналу, який перевів автомат Мура в даний стан. Іншими словами, на графі автомата вихідні сигнали, раніше приписані вершинам, необхідно віднести до відповідних дуг, що заходять в цю вершину.

Нехай заданий автомат Мура: $A^{\circ} = < X^{\circ}, A^{\circ}, A0^{\circ}, \sigma^{\circ}, Y^{\circ}, \lambda^{\circ} >$.

Знайти автомат Мілі: $A = <X, A, A0, \sigma, Y, \lambda>$.

З визначення еквівалентності слідує, що $X = X^{\circ}$, $Y = Y^{\circ}$, $A0 = A0^{\circ}$.

Як в автоматі Мура, так і в автоматі Мілі наступний стан залежить від поточного стану і поточного вхідного символу:

в автоматі Мура - $Ak^{\circ}(t+1) = \sigma^{\circ}(Ai^{\circ}(t), xj^{\circ}(t))$;

в автоматі Мілі - $Ak (t+1) = \sigma (Ai (t), xj (t))$.

З цього випливає, що функція переходу автомата Мілі збігається з функцією переходу автомата Мура при одному і тому ж вхідному символі. При цьому число станів і функцій переходів не зміниться, тобто:

$A = A^{\circ}$ і $\sigma (Ai, xj) = \sigma^{\circ}(Ai^{\circ}, xj^{\circ})$.

В автоматі Мілі вихідний символ $yk (t + 1)$ визначається поточним станом $Ai(t)$ і вхідним символом $xj(t)$: $yk (t+1) = \lambda(Ai (t), xj (t))$

В автоматі Мура вихідний символ $yk^{\circ}(t+1)$ визначається наступним станом $Ak^{\circ}(t+1)$: $yk^{\circ}(t+1) = \lambda^{\circ}(Ak^{\circ}(t + 1))$, а це стан: $Ak^{\circ}(t + 1) = \sigma^{\circ}(Ai^{\circ}(t), xj^{\circ}(t))$.

Якщо підставити цей вираз для $A^{\circ}(t+1)$ в функцію виходу автомата Мура, отримаємо його визначення через старий стан:

$yk^{\circ}(t+1) = \lambda^{\circ}(Ak^{\circ}(t+1)) = \lambda^{\circ}(\sigma^{\circ}(Ai^{\circ}(t),\ xj^{\circ}(t))$, а так як $\sigma(Ai, xj) = \sigma^{\circ}(Ai^{\circ}, xj^{\circ})$, то для забезпечення умов еквівалентності необхідно вихідні символи автомата Мілі, одержувані на переходах в нові стани, зробити рівними вихідним символам автомата Мура цих станів, і в результаті отримуємо:

$yk\ (t+1) = \lambda(Ai\ (t), xj\ (t))$.

Перетворення автомата Мура в автомат Мілі зручно робити при графічному поданні автоматів. Для цього необхідно вихідні символи, якими позначені вершини графа автомата, перенести на всі дуги, що входять в кожну вершину.

Перехід від автомата Мілі до еквівалентного автомата Мура:

Закони функціонування автоматів Мілі і Мура відрізняються функцією виходів. Оскільки кожній парі "стан - вхідний сигнал" автомата Мілі може відповідати свій вихідний сигнал, а в автоматі Мура вихідний сигнал приписується станам, то кожній парі $(a_i;x_j)$ автомата Мілі ставиться у відповідність стан синтезованого автомата Мура b_{ij}. Таким чином, кожній клітинці таблиці переходів автомата Мілі буде відповідати новий стан автомата Мура. Крім того, оскільки перший вихідний сигнал автомат видасть тоді, коли з початкового стану під впливом першого вхідного сигналу він перейде в якийсь стан, який і визначить перший вихідний сигнал, то необхідно для автомата Мура ввести також початковий стан b_0, якому може бути приписаний будь-який допустимий вихідний сигнал. Оскільки функції переходів у автоматів Мілі і Мура однакові, кожному стану автомата Мілі ставиться у відповідність клас ізоморфних станів автомата Мура.

Нехай заданий автомат Мілі: $A = \langle X, A, A0, \sigma, Y, \lambda \rangle$.

Знайти автомат Мура: $A^{\circ} = \langle X^{\circ}, A^{\circ}, A0^{\circ}, \sigma^{\circ}, Y^{\circ}, \lambda^{\circ} \rangle$.

З визначення еквівалентності слідує: $X^{\circ} = X$, $Y^{\circ} = Y$, $A0^{\circ} = A0$. Як в автоматі Мілі, так і в автоматі Мура наступний стан залежить від поточного стану і поточного вхідного символу:

в автоматі Мілі - $Ak\ (t+1) = \sigma(Ai\ (t), xj\ (t))$;

в автоматі Мура - $Ak^{\circ}(t+1) = \sigma^{\circ}(\ Ai^{\circ}(t), xj^{\circ}(t))$.

З цього випливає, що функція переходу автомата Мура аналогічна функції переходу автомата Мілі при одному і тому ж вхідному символі. Однак, так як в автоматі Мура вихідний символ $yk^{\circ}(t+1)$ визначається наступним станом $Ak^{\circ}(t+1)$, то кожному стану може відповідати тільки один вихідний символ. Таким чином, якщо в вихідному автоматі Мілі існують переходи в один і той же стан з видачею різних вихідних символів, то в еквівалентному йому автоматі Мура число станів і відповідно число функцій переходів збільшиться.

Перетворення автомата Мілі в автомат Мура зручно проводити при графічному представленні автоматів. Для цього необхідно виконати дії, зворотні діям при перетворенні автомата Мура в автомат Мілі, тобто вихідний символ, яким позначена дуга графа автомата, перенести в вершину, в яку ця дуга входить.

Наприклад, для фрагмента графа автомата Мілі (рис. 4.1), зробивши таке перенесення, отримаємо фрагмент графа автомата Мура (рис. 4.2).

Рис. 4.1. Фрагмент автомата Мілі.

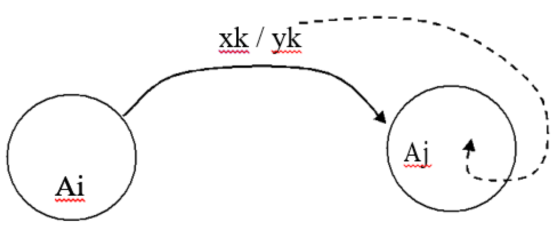
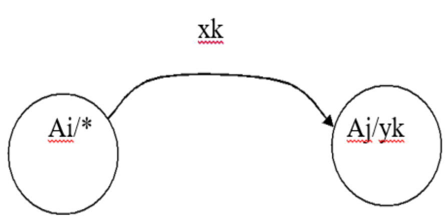


Рис. 4.2. Фрагмент автомата Мура



Для фрагмента графа автомата Мілі на рис. 4.3 таке перенесення здійснити не можливо, так як на кожному переході відбувається видача різних вихідних символів. У цьому випадку стан Am необхідно розщепити (продублювати) на таку кількість станів, скільки різних вхідних символів є на всіх вхідних ребрах цього стану, і зіставити кожному з них свій вихідний символ (рис.4.4).

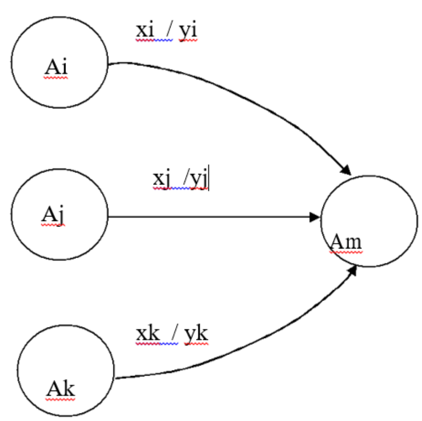


Рис.4.3. Автомат Мілі

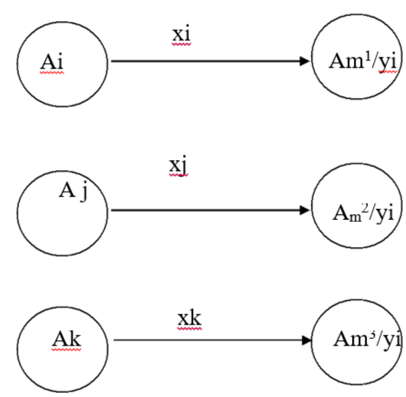


Рис. 4.4. Автомат Мура (фрагмент; автомат відповідний автомату Мілі з рис. 4.3)

Таким чином, існує стандартний прийом, за допомогою якого можна перетворити автомат Мілі в еквівалентний йому автомат Мура. Причому, якщо в автоматі Мілі було *n* внутрішніх станів і кількість вхідних сигналів *m*, то в отриманому автоматі Мура буде *(n * m) +1* станів.

Мінімізація автоматів

Мінімальний автомат – це автомат, який має найменшу можливу кількість станів, і який реалізує задану функцію виходів.

Два стани автомата називаються *1-евівалентними*, якщо, перебуваючи в будь-якому із цих станів, автомат на один і той же вхідний сигнал (вхідне слово довжиною в 1) видає один і той же вихідний сигнал. Два стани автомата називаються *К-еквівалентними*, якщо, починаючи з будь-якого із цих станів, автомат на будь-які однакові слова довжини *К* видає однакові вихідні слова (також довжини *К*). Якщо два стани *К*-еквіваленті для будь-яких *К*, то їх називають (просто) еквівалентними. Множина попарно еквівалентних станів формує клас еквівалентності.

Зміст мінімізації полягає у виявленні класів еквівалентності і заміні кожного класу одним станом. Процедура мінімізації полягає у наступному:

1) за таблицею виходів автомата Мілі (або за першою строфою відміченої таблиці переходів автомата Мура) знаходяться стани, які мають однакові стовпчики відмічені однаковими вихідними сигналами – це 1-еквівалентні стани.

2) далі використовується таблиця переходів. Всі стани, що входять до 1-еквівалентного класу, і, що під дією першого сигналу перейшли в стани, які у свою чергу належать до 1-еквівалентного класу, формують 2-еквівалентний клас.

3) процедура розподілу на класи еквівалентності продовжується до того часу, доки при черговому кроці К-еквівалентні класи не співпадуть із (К-1) - еквівалентними, тобто поки не отримаємо еквівалентні класи.

4) всі стани, що входять в один клас еквівалентності, замінюються одним станом.

Розпізнавальні автомати

Розпізнавальний автомат – це автомат Мура, в якому фіксуються початковий стан і підмножина станів F×Q, яка називається множиною кінцевих станів. Кажуть, що автомат допускає (приймає, розпізнає, представляє) дане слово, якщо реакцією на це слово може бути перехід автомата в один із кінцевих станів.

Одним із найбільш використовуваних на практиці типів розпізнавальних автоматів є частковий недетермінований автомат. Недетермінізм проявляється в тому, що із одного стану за одним із тим самим вхідним сигналом можливі переходи в різні стани, тобто функція переходів заміняється відношенням переходів. Недетермінований автомат приймає, наприклад, слова ab, aa, bb, bba і т.п. Тут початковий стан A, кінцевий – F. Таблиця переходів даного автомата буде мати вигляд:

	A	**B**	**C**	**F**
a	B,C	-	F	-
b	B	C,F	-	-

Порожні клітинки свідчать про те, що автомат частковий, а наявність одразу декількох літер в одній клітинці – про те, що автомат недетермінований.

Для таких автоматів зазвичай надають перевагу не табличному або графічному представленню, а запису у вигляді, так званих продукцій, чиї граматичні правила представлені у двох стовпчиках відповідно:

$A \rightarrow aB$	$A ::= aB / bB / aC$
$A \rightarrow bB$	$B ::= bC / bF$
$A \rightarrow aC$	$C ::= aF$
$B \rightarrow bC$	
$B \rightarrow bF$	
$C \rightarrow aF$	

Граматики такого роду називають регулярними або автоматними. Автомати такого типу будуть розглядатися у зв'язку із вивченням теорії формальних мов і граматик.

Приклади побудови скінчених автоматів за функціональним описом задачі

Абстрактна теорія автоматів виділяє способи використання автоматів: перетворення вхідної послідовності символів у вихідну (синтез дискретних пристроїв) і перевірка правильності вхідної послідовності символів (синтез програмних аналізаторів); аналіз поведінки автомата або видати у якості відповіді на вхідне слово вихідне слово у заданому алфавіті при наявності опису дії автомата у вигляді таблиць виходів та переходів.

1. Автомат заданий множиною станів Q = {A, B, C}, множиною вхідних і вихідних сигналів X = Y = {0,1} і таблицями переходів(4.1) і виходів(4.2). Початковий стан – A.

Табл.4.1

	A	**B**	**C**
0	B	A	A
1	C	A	A

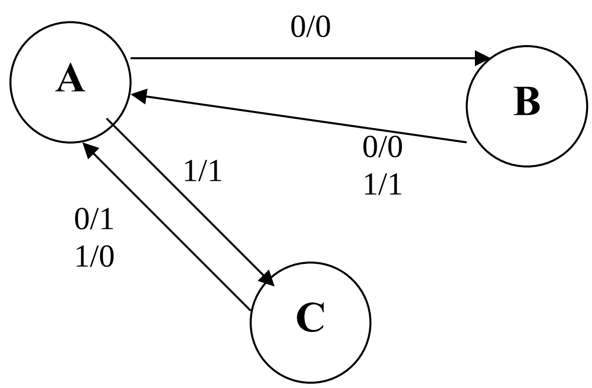
Табл.4.2

	A	**B**	**C**
0	0	0	1
1	1	1	0

Сформувати вихідне слово, якщо на вхід подано 10110001.

Цей автомат реалізує булеву функцію «додавання за модулем 2» (x⊕y) двох двійкових чисел, які подаються на вхід за розрядами по черзі. У прикладі вхідне слово *1*__0__*1*__1__*0*__0__*0*__1__ сформовано з двох двійкових чисел *1100* та **0101**. Відповіддю має бути «додавання за модулем 2» за розрядами – число 1001. *Але насправді, автомат – «чорний ящик», і в задачах аналізу його дії ми не можемо встановити, яку функцію він реалізує.*

такт	1	2	3	4	5	6	7	8	
стан	A	C	A	C	A	B	A	B	A
вхід	*1*	**0**	*1*	**1**	*0*	**0**	*0*	**1**	
вихід	1	1	1	0	0	0	0	1	
перехід в стан	C	A	C	A	B	A	B	A	

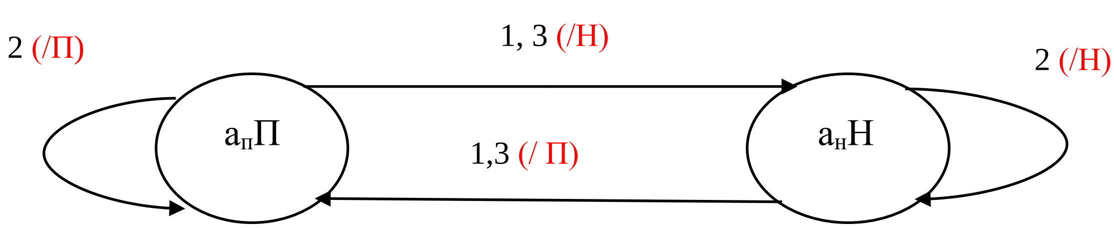


При перетворюванні інформації за допомогою абстрактного автомату вихідний сигнал повинен формуватися у відповідь на кожний вхідний, тобто, кожен непарний символ вхідного слова (розряд першого числа) дублюється, а кожен парний (розряд другого числа) перетворюється на відповідь з урахуванням попереднього символу (відповідного розряду першого числа).

Відповідь (вихідне слово): **11100001.**

Побудувати (синтезувати) автомати за змістовим описом.

2. Побудувати автомат, на вхід якого можуть надходити монети 1, 2 і 3 коп. Автомат видає сигнал "парна" (П), якщо сума опущених монет парна, і "непарна" (Н), якщо непарна.



Сума 0 вважається парною. Це автомат Мура, тобто його вихідний сигнал однозначно визначається станом, в який автомат перейшов, тому вихідні сигнали приписані безпосередньо до станів.

Цей же автомат можна представити відміченою таблицею переходів.

y	Н	П
x/A	$\mathbf{a_н}$	$\mathbf{a_п}$
1	$a_п$	$a_н$
2	$a_н$	$a_п$
3	$a_п$	$a_н$

3. Побудувати (синтезувати) автомат, на вхід якого можуть надходити в будь-якій послідовності і, можливості, з будь-яким числом повторів монети 1, 2 і 3 коп. Автомат продає квиток, якщо сума опущених монет дорівнює 3. У разі перевищення суми автомат повертає гроші.

Вхідний алфавіт в описі заданий явно: X = {1,2,3}.

Вихідний алфавіт буде складатись з літер: К - видає квиток, П - повертає гроші, Н-нічого не видає, тобто Y = {Н, К , П}.

Внутрішній стан будемо ототожнювати з сумою, яку пам'ятає автомат. Будемо мати на увазі, що після продажу квитка або повернення грошей автомат пам'ятає нульову суму: A = {a0, a1, a2}, тут індекс відповідає сумі.

Це автомат Мілі. Надпис *1 / К ; 2,3 / П* на стрілки **a2→a0** означає, що якщо в стані **a2** буде прийнятий сигнал *1*, то вихідний сигнал буде *К*, а для вхідних сигналів *2 і 3* вихідних буде **П**. Цей же автомат можна представити у вигляді двох таблиць: таблиці переходів (табл.4.3) і таблиці виходів (табл.4.4):

Табл.4.3 (σ)

x/A	$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$
1	a_1	a_2	a_0
2	a_2	a_0	a_0
3	a_0	a_0	a_0

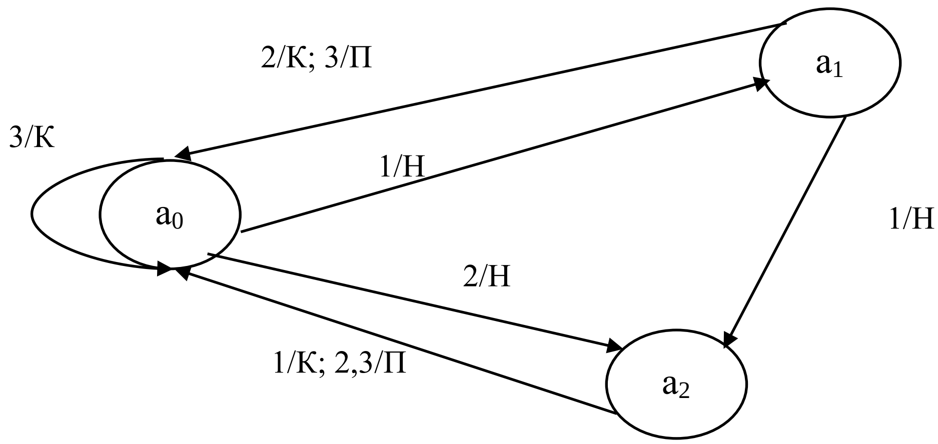
Табл.4.4 (λ)

x/y	$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$
1	Н	Н	К
2	Н	К	П
3	К	П	П

або у вигляді об'єднаної таблиці переходів (σ) і виходів(λ):

x/A	a_0	a_1	a_2
1	a_1/Н	a_2/Н	a_0/К
2	a_2/Н	a_0/К	a_0/П
3	a_0/К	a_0/П	a_0/П

Автомат у вигляді графа:



4. Автомат (розпізнавальний) з початковим станом S(start) допускає слова, у яких є лише парні входження літер a і b, наприклад, a a, a a a a a a, b b a a і т. п. Ознакою кінця слова є сигнал R. Тобто у випадку вхідного слова aaaabbaaR видається вихідне слово aaaabbaaY (тобто, так, yes), а у випадку aaaabaaR – aaaabaaN (тобто, ні, no).

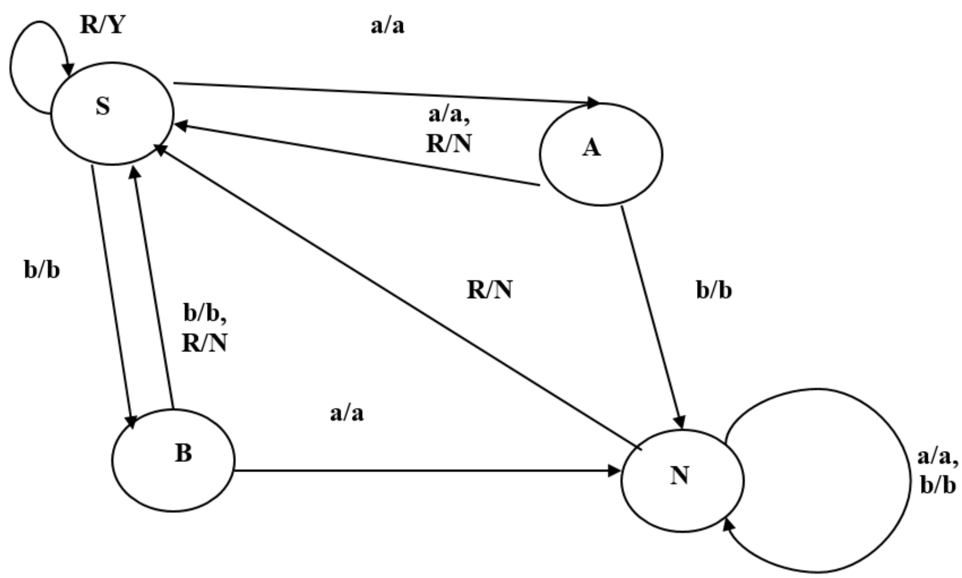
Приклад:

aaaabbaaR

такт		1	2	3	4	5	6	7	8	9	
стан		S	A	S	A	S	B	S	A	S	S
вхід		a	a	a	a	b	b	a	a	R	
вихід		a	a	a	a	b	b	a	a	Y	
перехід в стан	S	A	S	A	S	B	S	A	S	S	

aaaabaaR

такт		1	2	3	4	5	6	7	8	
стан		S	A	S	A	S	B	N	N	S
вхід		a	a	a	a	b	a	a	R	
вихід		a	a	a	a	b	a	a	N	
перехід в стан	S	A	S	A	S	B	N	N	S	



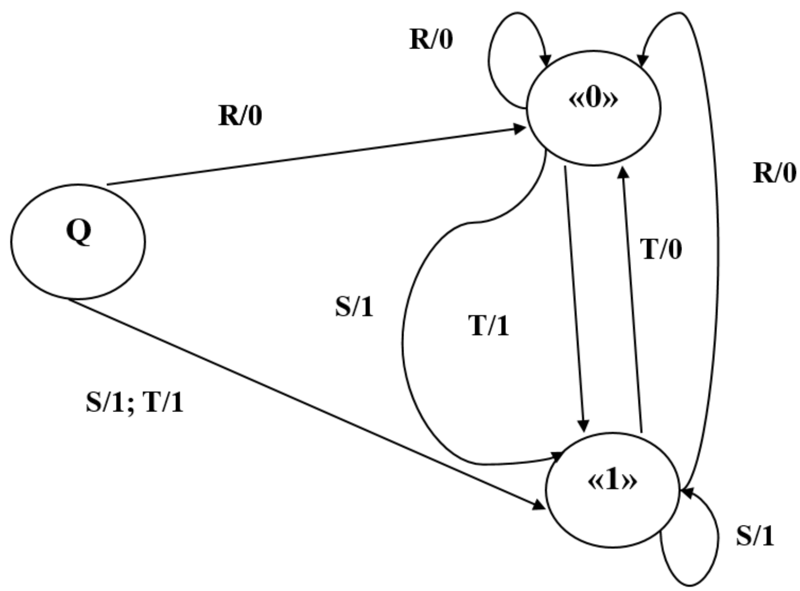
Табличний вигляд:

x/A	**S**	**A**	**B**	**N**
a	A/a	S/a	N/a	N/a
b	B/b	N/b	S/b	N/b
R	S/yes	S/no	S/no	S/no

5. На вхід автомата можуть поступати сигнали R, S і T. На вхідний сигнал R автомат видає вихідний сигнал 0, на S – вихідний сигнал 1, і на T – вихідний сигнал, протилежний попередньому вихідному сигналу. Для визначеності вважаємо, що в початковому стані автомат пам’ятає «попередній» вихідний сигнал 0:

Приклад:

такт		1	2	3	4	5	6	7	8	9	10	11	12	13	
стан		Q	«0»	«1»	«1»	«0»	«0»	«1»	«0»	«1»	«0»	«1»	«1»	«0»	«0»
вхід		R	T	S	T	R	S	T	T	R	S	S	T	R	
вихід		0	1	1	0	0	1	0	1	0	1	1	0	0	
перехід в стан	Q	«0»	«1»	«1»	«0»	«0»	«1»	«0»	«1»	«0»	«1»	«1»	«0»	«0»	

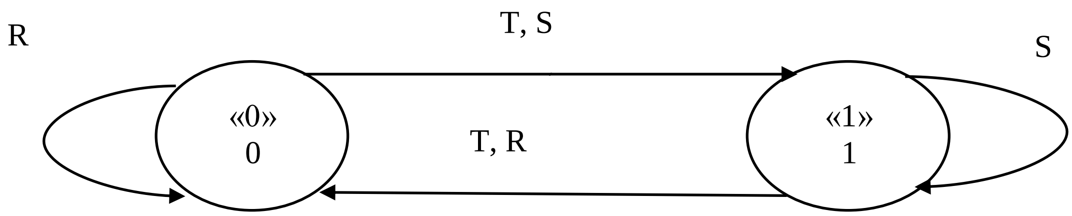


Представимо автомат у вигляді об’єднаної таблиці переходів і виходів. Можна побачити, що стовпці для станів «0» і Q – ідентичні. Автомат можна мінімізувати, вилучив стан Q, та отримати еквівалентний автомат на 2 станах – «0» та «1»:

x/A	***Q***	**«0»**	**«1»**
S	*«1»/1*	«1»/1	«1»/1
T	*«1»/1*	«1»/1	«0»/0
R	*«0»/0*	«0»/0	«0»/0

Стан Q був ініціальним, тому стартовим треба призначити стан «0». Також вочевидь, що ми маємо автомат Мура. Тому будемо використовувати відмічену таблицю переходів для автомата Мура.

y	***0***	***1***
x/A	**«0»**	**«1»**
S	«1»	«1»
T	«1»	«0»
R	«0»	«0»



Завдання 4.1.

I Автомат заданий множиною станів Q = {A, B, C}, початковим станом – A; множиною вхідних і вихідних сигналів X = Y = {0,1} і таблицями переходів і виходів:

Табл. переходів

	A	**B**	**C**
0	B	B	C
1	C	C	B

Табл.виходів

	A	**B**	**C**
0	0	0	1
1	0	0	1

Сформувати вихідне слово, якщо на вхід подано

1) 00011100;

2) 00011101.

II Автомат заданий множиною станів Q = {A, B, C}, початковим станом – A; множиною вхідних і вихідних сигналів X = Y = {0,1} і таблицями переходів і виходів:

Табл. переходів

	A	**B**	**C**
0	B	A	A
1	C	A	A

Табл.виходів

	A	**B**	**C**
0	0	0	0
1	1	0	1

Сформувати вихідне слово, якщо на вхід подано

1) 00101111;
2) 00110101.

III Автомат заданий множиною станів Q = {A, B}, початковим станом – A; множиною вхідних і вихідних сигналів X = Y = {0,1,R} і таблицями переходів і виходів:

Табл. переходів

	A	**B**
0	A	B
1	B	A
R	A	A

Табл.виходів

	A	**B**
0	0	0
1	1	1
R	1	0

Сформувати вихідне слово, якщо на вхід подано

1) 00101101R;
2) 00101111R;
3) 00110101R.

Завдання 4.2. Побудувати (синтезувати) автомати за змістовим описом.

1) Накопичувальний лічильник, на вхід якого подаються двійкові цифри 0 і 1, підраховує за модулем 3 загальне число одиниць, що надійшли на вхід.

Вхід	*0*	*1*	*1*	*0*	*1*	*0*	*0*	*1*	*1*
Вихід	*0*	*0*	*0*	*0*	***1***	*0*	*0*	*0*	*0*

2) Побудувати автомат для *X=Y={0,1},* на виході якого з’являється сигнал 1 тоді і тільки тоді, коли чотирма останніми вхідними сигналами є 1010 (два варіанти – без перетину та з ним).

I без перетину

Вхід	*0*	*1*	*1*	*0*	*1*	*0*	*1*	*0*	*1*
Вихід	*0*	*0*	*0*	*0*	*0*	***1***	*0*	*0*	*0*

II з перетином

Вхід	*0*	*1*	*1*	*0*	*1*	*0*	*1*	*0*	*1*
Вихід	*0*	*0*	*0*	*0*	*0*	***1***	*0*	***1***	*0*

3) Текст, складений з 33 букв алфавіту і пропусків між словами, аналізується з метою розпізнавання слів, що починаються з літери 'а' і закінчуються на 'ія' (наприклад, арія, армія). Всі букви, крім 'а', 'і', 'я', позначимо через α, а пропуск - через β. Вхідний алфавіт X, вихідний алфавіт У і множина станів А визначаються наступним чином: X = {'а', 'і', 'я', 'α', 'β'}, У = {'а', 'і', 'я', 'α', Y, N} – так та ні, А = { S - старт, а – поява літери а, і – поява "а..і", я – поява "а..ія", F – очікування нового слова}.

<u>Зауваження до таблиці:</u> вихідні сигнали – ***дублюємо входні сигнали****; тільки на пробіл* ***(β)*** *– видамо відповідь відмінну, від входного пробілу- Y/N.*

В будь-якій задачі, якщо не прописано в умові детально про вихідні сигнали, то краще їх дублювати, щоб на виході мати повну картину слова; але можно і видаляти – записувати вихід або про нього попередити треба обов'язково, тобто, x/y – такий запис повинен бути присутним і в таблиці і на графі.

4) Побудувати автомат для *X=Y={0,1},* на виході якого з'являється сигнал 1 тоді і тільки тоді, коли чотирма останніми вхідними сигналами є 0110 (без перетина та з ним).

Вхід	*1*	*0*	*1*	*1*	*0*	*1*	*1*	*0*	*1*	*1*	*0*	*1*
Вихід	*0*	*0*	*0*	*0*	*1*	*0*	*0*	*0*	*0*	*0*	*1*	*0*
Вихід з перетином	***0***	***0***	***0***	***0***	***1***	***0***	***0***	***1***	***0***	***0***	***1***	***0***

5) Побудувати автомат затримки на два такти, в якому вихідний сигнал у момент часу t+2 збігається зі вхідним сигналом у момент часу t, X=Y={0,1}, y(0)=y(1)=0.

Вхід	*0*	*1*	*1*	*0*	*1*	*0*	*1*	*0*	*1*
Вихід	***0***	***0***	*0*	*1*	*1*	*0*	*1*	*0*	*1*

6) Побудувати (синтезувати) автомат, на вхід якого можуть надходити гроші 1, 2, 5 та інші грн. Автомат продає квиток, якщо дорівнює 8. У разі перевищення суми автомат видає квіток та повертає решту.

Вхідний алфавіт в описі заданий явно: X = {1,2,5, x (інші купюри)}.

Вихідний алфавіт буде складатись з літер: К - видає квиток, Р - квіток та повертає решту, Н – накопичує гроши і нічого не видає, тобто Y = {Н, К, Р}.

Внутрішній стан будемо ототожнювати з сумою, яку пам'ятає автомат. Будемо мати на увазі, що після продажу квитка або повернення грошей автомат пам'ятає нульову суму: A = {A0, A1, A2, A3, A4, A5, A6, A7}, тут індекс відповідає сумі.

7) Текст, складений з 33 букв алфавіту і пропусків між словами, аналізується з метою розпізнавання слів, що закінчуються на ТОР.

Всі букви, крім 'т', 'о', 'р', позначимо через α, а пропуск - через β.

Вхідний алфавіт X, вихідний алфавіт У і множина станів A визначаються наступним чином: X={'т', 'о', 'р', 'α', 'β'}, **У = {'т', 'о', 'р', 'α', Y, N}**, A = { S - старт, т – поява літери т, то – наявність "то", тор – наявність "тор", F – очікування нового слова}. Слова повинні закінчуватися на ТОР, тобто само слово ТОР – не входить до списку (входять: авіатор, затор, екватор).

Вихідні сигнали дублюємо, виводимо в таблицю тільки остаточну відповідь на пробіл – ***Y/N.***

8) Побудувати автомати, що реалізують одну з двох операції зсуву трирозрядного двійкового коду на одну позицію. Вхідні сигнали визначають характер операції зсуву:

- логічний зсув уліво (або вправо); 001 L_l → 010 ; 001 R_l → 000

- циклічний зсув управо (або вліво). 001 L_c → 010 ; 001 R_c → 100

Вихідним є сигнал, що виходить за межі розрядної сітки.

Стани відповідають вісьмом можливим варіантам кодів.

вхідні сигнали R_c, R_l, L_c , L_l ; вихідні 0 та 1; стани від 000 до 111.

Відповіді до розділу 4

4.1.

I. **1)** 00001010;

2) 00001011;

II. **1)** 00101111;

2) 00110000;

III. **1)** 001011011;

2) 001011110;

3) 001101011;

4.2.

1)

	S	**a1**	**a2**
0	S/0	a1/0	a2/0
1	a1/0	a2/0	S/1

2)

I – без перетину

	S	**a1**	**a10**	**a101**
0	S/0	a10/0	S/0	S/1
1	a1/0	a1/0	a101/0	a1/0

II – з перетином

	S	**a1**	**a10**	**a101**	**a1010**
0	S/0	a10/0	S/0	a1010/1	S/0
1	a1/0	a1/0	a101/0	a1/0	a101/0

Мінімізований –

	S	**a1**	**a10**	**a101**	~~**a1010**~~
0	S/0	a10/0	S/0	**a10**/1	~~S/0~~
1	a1/0	a1/0	a101/0	a1/0	~~a101/0~~

3) *Зауваження до таблиці:* *вихідні сигнали –* ***дублюємо входні сигнали****; тільки на пробіл (****β****) – видамо відповідь відмінну, від входного пробілу- Y/N. В таблиці буду відображати тільки такий вихід. Повтори вихідних за вхідними записувати не будемо, щоб не захаращувати таблицю.*

	S	***a***	***a..i***	***a..ія***	***F***
a	a/a	a	a/a	a	F
i	F/i	a..i/i	a..i	a..i	F
я	F	a/я	a..ія	a	F
α	F	a	a	a	F
β	***S/N***	***S/N***	***S/N***	***S/Y***	***S/N***

6)

	A0	***A1***	***A2***	***A3***	***A4***	***A5***	***A6***	***A7***
1	A1/Н	A2/Н	A3/Н	A4/Н	A5/Н	A6/Н	A7/Н	**A0/К**
2	A2/Н	A3/Н	A4/Н	A5/Н	A6/Н	A7/Н	**A0/К**	**A0/Р**
5	A5/Н	A6/Н	A7/Н	**A0/К**	**A0/Р**	**A0/Р**	**A0/Р**	**A0/Р**
x	***A0/Р***	***A0/Р***	***A0/Р***	***A0/Р***	***A0/Р***	***A0/Р***	***A0/Р***	***A0/Р***

7) Зауваження: *F – очікування нового слова* – це стан саме очікування слова, не помилка. Тут обигрується те, що слово не повинно починатися з ТОР.
Вихідні сигнали дублюємо, виводимо в таблицю тільки остаточну відповідь на пробіл – ***Y/N.***

	S	**Слово (F)**	**T**	**TO**	**TOP**
т	Слово (F)	T	T	T	T
o	Слово (F)	Слово (F)	TO	Слово (F)	Слово (F)
p	Слово (F)	Слово (F)	Слово (F)	TOP	Слово (F)
α	Слово (F)	Слово (F)	Слово (F)	Слово (F)	Слово (F)
β	S/N	S/N	S/N	S/N	S/Y

8)

	000	**001**	**010**	**011**	**100**	**101**	**110**	**111**
R_c	000/**0**	100/**1**	001/**0**	101/**1**	010/**0**	110/**1**	011/**0**	111/**1**
R_l	000/**0**	000/**1**	001/**0**	001/**1**	010/**0**	010/**1**	011/**0**	011/**1**
L_c	000/**0**	010/**0**	100/**0**	110/**0**	001/**1**	011/**1**	101/**1**	111/**1**
L_l	000/**0**	010/**0**	100/**0**	110/**0**	000/**1**	010/**1**	100/**1**	110/**1**

Зі стану **000** ми не виходимо в інші. але він не є відокремленим, через те, що ми в нього входимо з **001** та **100**. Стан **111** – більш різноманітний. В ньому є не тільки петлі, а і переходи.
Подивиться на рядки з циклічним зсувом – автомат перебігає всі стани.
Такого не спостерігається при логічних зсувах. Чому?

5. МЕРЕЖІ ПЕТРІ

5.1. Моделювання об'єктів мережами Петрі

Теоретичні відомості

Мережі Петрі - математичний апарат для моделювання динамічних дискретних систем. Вперше описані Карлом Петрі в 1962 році.

Мережа Петрі це дводольний орієнтований граф, що складається з вершин двох типів - позицій і переходів, з'єднаних між собою дугами, вершини одного типу не можуть бути з'єднані безпосередньо. У позиціях можуть розміщуватися мітки (маркери), здатні переміщатися по мережі.

Подією називають спрацьовування переходу, при якому мітки з вхідних позицій цього переходу переміщаються в вихідні позиції. Події відбуваються миттєво, різночасно при виконанні деяких умов.

Моделювання в мережах Петрі здійснюється на подієвому рівні. Визначаються, які дії відбуваються в системі, які стани передували цим діям і які стани прийме система після виконання дії. Виконання подієвої моделі в мережах Петрі описує поведінку системи.

Мережа Петрі, як вже було сказано вище, являє собою деякий різновид орієнтованого графа із заданим початковим станом, який називається початковим маркуванням (або розміткою) M_0. Граф n мережі Петрі є орієнтованим і включає вузли (або вершини) двох типів, звані позиціями (або місцями) і переходами, дуги в якому ведуть або з позиції в перехід, або з переходу в позицію. У графічному поданні позиції зображуються кружками, а переходи жирними рисками або прямокутниками. Дуги позначаються відповідними вагами (цілими позитивними числами), і дугу з вагою k можна вважати еквівалентної k паралельним дуг. Вказівка одиничної ваги зазвичай опускається. Маркування (стан) приписує кожній позиції ціле невід'ємне число. Якщо маркування приписує позиції p ціле невід'ємне число k, то говорять, що p марковано k фішками; в графічному зображенні всередині позиції-кружка в цьому випадку поміщають k чорних точок (фішок). Маркування позначається вектором M довжиною m, де m- загальне число позицій. Компонента вектора M з номером p, що позначається $M\ (p)$, дорівнює числу фішок в позиції p.

В задачах моделювання, де застосовуються поняття умов (станів) і подій, позиції відповідають умовам, а переходи - подіям. Кожен перехід (подія) пов'язаний з певним числом вхідних і вихідних позицій - аналогів відповідно передумов та постумов цієї події. Наявність фішки в деякій позиції інтерпретується як істинність умови, що відповідає даній позиції. В іншому трактуванні позиція позначається k фішками, з тим, щоб вказати на наявність k елементів даних або відповідної кількості ресурсів. **Формальне визначення мережі Петри (P_n)**

Мережа Петрі – це п'ятірка $P_n = (P, T, F, W, M_0)$, де

$P = (p_1, p_2, \ldots, p_m)$ – кінцева множина позицій;

$T = (t_1, t_2, \ldots, t_n)$ – кінцева множина переходів;

$F \subseteq (p \times t) \cup (t \times p)$ – множина дуг;

$W : f \rightarrow (1, 2, 3, \ldots)$ – вагова функція;

$M_0 : p \rightarrow (0, 1, 2, 3,..)$ – початкова маркування;

$P \cap t = \varnothing$.

Структура мережі Петрі $N=(P,T,F,W)$ без заданого початкового маркування позначається однією літерою N, із заданим – двома (N, M_0).

При моделюванні динаміки системи стан, або маркування мережі Петрі, змінюється відповідно до правила (запуску) переходу:

1. Перехід дозволено, якщо все вхідні позиції р переходу помічені не менше ніж $w\ (p, t)$ фішками, де $w\ (p, t)$ - вага дуги, що веде з p в t;

2. Запуск дозволеного переходу носить випадковий характер (в залежності від настання або ненастання відповідної події);

3. Запущений перехід *t* вилучає *w(p, t)* фішок з кожної своєї вхідної позиції і додає *w (t, p)* фішок в кожну свою вихідну позицію; тут *w (t, p)* - вага дуги, що веде з *t* в *p*.

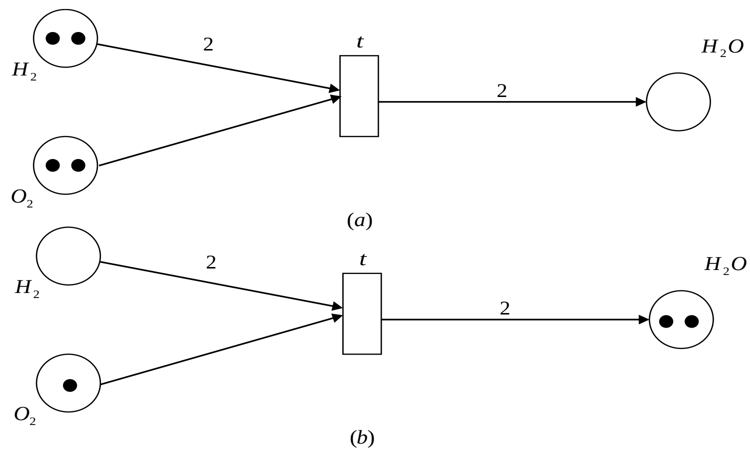


Рис. 5.1. Пояснення до правила запуску переходу: (a) маркування перед спрацьовуванням дозволеного переходу t; (b) маркування після спрацьовування цього переходу, після чого він стає забороненим переходом.

ПОДІЇ І УМОВИ.

Просте уявлення системи мережею Петрі засноване на двох основних поняттях: подіях та умовах. Події - це дії, що мають місце в системі. Виникненням подій керує стан системи. Стан системи може бути описаний безліччю умов. Умова є предикат або логічний опис стану системи. Умова може набувати або значення «істина», або значення «хибно».

Оскільки події є діями, вони можуть відбуватися. Щоб подія відбулася, необхідно виконання відповідних умов. Ці умови називаються передумовами події. Виникнення події може викликати порушення передумов і може призвести до інших умов, постумов.

Як приклад, розглянемо завдання моделювання простого автомата-продавця. Автомат-продавець знаходиться в стані очікування доти, доки не з'явиться замовлення, яке він виконує та надсилає на доставку. Умовами для такої системи є:

а) автомат-продавець чекає; б) замовлення прибуло і очікує; в) автомат-продавець виконує замовлення; г) замовлення виконано.

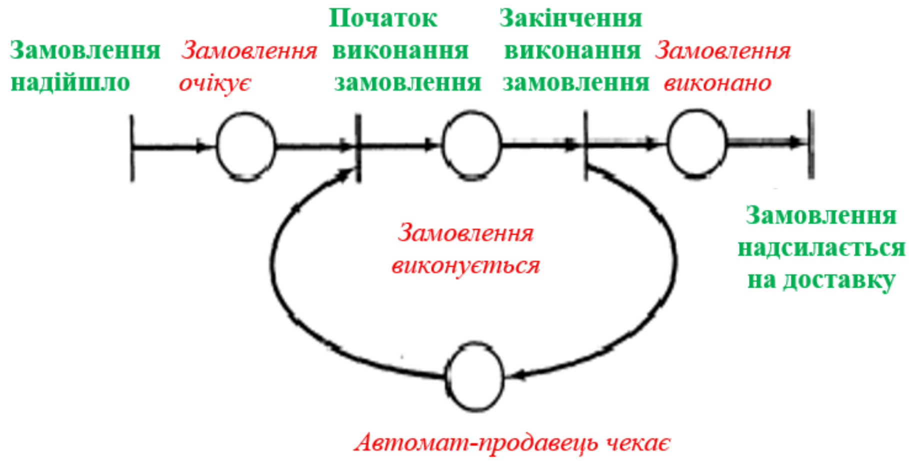
Подіями будуть: 1. Замовлення надійшло. 2. Автомат-продавець починає виконання замовлення. 3. Автомат-продавець закінчує виконання замовлення. 4. Замовлення надсилається на доставку.

Передумови події 2 (автомат-продавець починає виконання замовлення) очевидні: (а) автомат-продавець чекає; (б) замовлення прибуло і очікує. Постумови для події 2: (в) автомат-продавець виконує замовлення. Аналогічно можемо визначити передумови та постумови для інших подій та скласти наступну таблицю подій та їх перед- та постумов:

Передумова	*ПОДІЯ*	*Постумова*
немає	**1**	б
а, б	**2**	в
в	**3**	г, а
г	**4**	немає

Таке уявлення системи легко моделювати мережею Петрі. У мережі Петрі умови моделюються позиціями, події – переходами. При цьому входи переходу є передумовами відповідного події; виходи - постумовами. Виникнення події рівносильне запуску відповідного переходу. Виконання умови представляється фішкою у позиції, що відповідає цій умові. Запуск переходу видаляє фішки, що представляють виконання передумов і утворює нові фішки, які представляють виконання постумов.

Мережа Петрі на малюнку нижче ілюструє модель наведеного вище автомата-продавця. Для кожного переходу та позиції вказані відповідні події та умови.



Фрагмент з позицією P_1, яка передує двом (або більше) вихідним переходам t_1 і t_2 (рис.5.2), називається, в залежності від концепта, конфліктом або вузлом прийняття рішення.

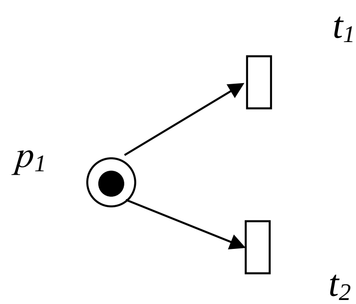


Рис. 5.2. Фрагмент мережі Петрі, званий конфліктом, вибором або вузлом прийняття рішення. Ця структура має властивість недетермінованості.

За допомогою мереж Петрі зручно відображати паралельні процеси, або паралелізм. Наприклад, в мережі на рис.5.3 паралельні процеси, яким відповідають переходи t_2 і t_3, починаються при запуску переходу t_1 і завершуються при запуску переходу t_4.

У загальному випадку два переходи вважаються паралельними, якщо між ними відсутній причинно-наслідковий зв'язок, тобто один перехід може спрацювати раніше іншого, після нього або одночасно з ним, як, наприклад, в разі переходів t_2 і t_3 на рис.5.3.

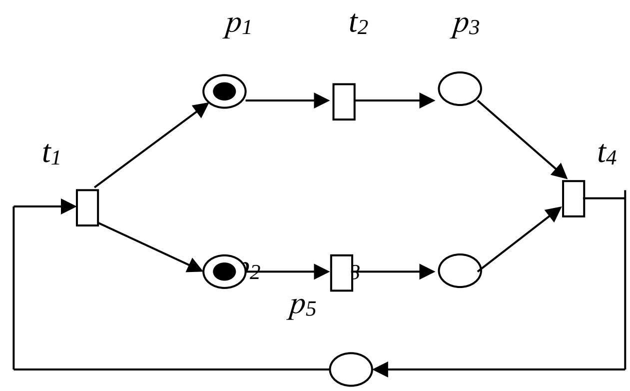


Рис. 5.3. Мережа Петрі (маркований граф), що моделює детерміновані паралельні процеси.

Паралелізм можна розглядати як бінарне відношення (позначається символом ρ і діє на множині $A = \{e_1, e_2, \ldots\}$), яке є рефлексивним, тобто $e_i \rho e_i$, і симетричним, тобто з $e_1 \rho e_2$ слідує $e_2 \rho e_1$, але не володіє транзитивністю, тобто з $e_1 \rho e_2$ і $e_2 \rho e_3$ не обов'язково слідує $e_1 \rho e_3$. Наприклад, можна вести машину (подія e_1) або йти пішки (подія e_3) і при цьому співати (подія e_2), проте не можливо одночасно (паралельно) вести машину і йти пішки.

Події e_1 і e_2 є конфліктуючими, якщо вони обидва можливі, але не можуть наступити одночасно. Події e_1 і e_2 паралельні, якщо вони можуть наступити в будь-якій послідовності, не приводячи до конфліктів. Поєднання конфліктної ситуації з паралелізмом називається змішанням.

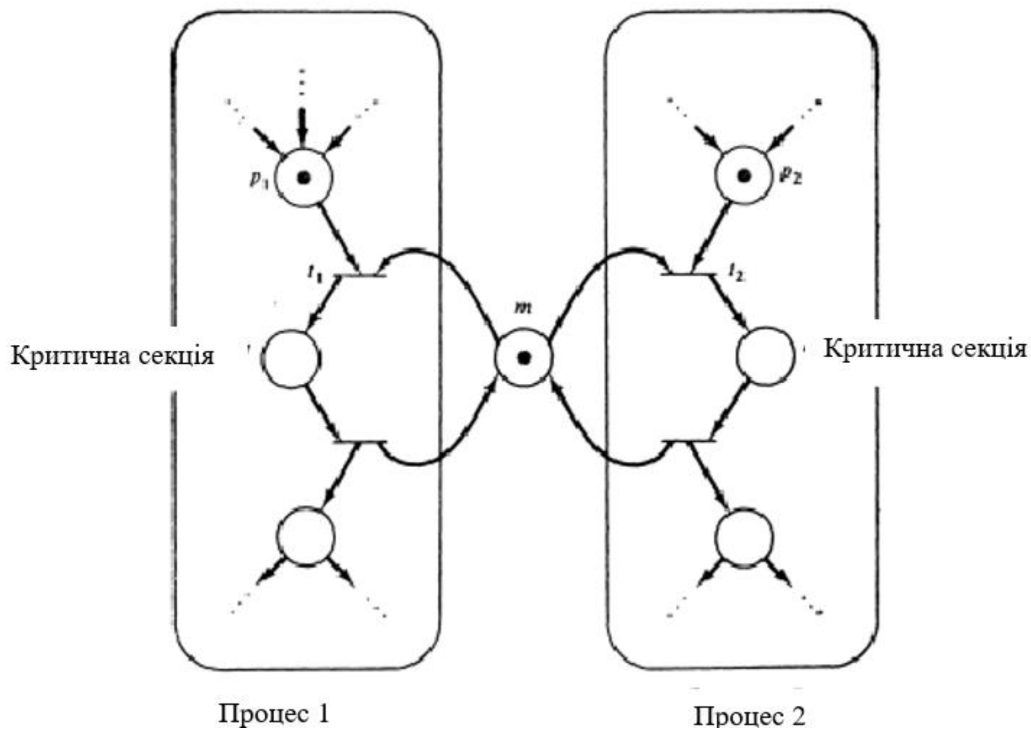
Введення паралелізму корисне лише у тому випадку, коли компоненти процесів можуть взаємодіяти при отриманні розв'язання задачі. Така взаємодія потребує розподілу ресурсів між процесами. Для гарантії правильності роботи системи загалом розподілом необхідно керувати. Проблеми синхронізації, що виникають при взаємодії процесів, ілюструються численними прикладами: завдання про взаємне виключення, виробника/споживача, мудреців, що обідають, читання/записи та ін. І хоча мережі Петрі є схемою моделювання, а не механізмом синхронізації, вони безумовно здатні моделювати механізми синхронізації.

Припустимо, кілька процесів поділяють загальну змінну, запис, файл чи інший елемент даних. Цей елемент даних, що розділяється, може використовуватися процесами різними способами, спрощено їх можна класифікувати як читання значення елемента даних або запис нового значення. Ці дві операції часто є єдиними примітивними операціями. Це означає, що для оновлення елемента даних, що розділяється, процес повинен спочатку вважати старе значення, потім обчислити нове і, нарешті, записати його на те ж місце. Якщо два процесу одночасно і намагаються виконати таку послідовність дій, можуть виникнути труднощі. Можлива наступна послідовність:

1. Перший процес зчитує значення x з об'єкта, що розділяється;
2. Другий процес зчитує значення x з об'єкта, що розділяється;
3. Перший процес обчислює нове значення f(x);
4. Другий процес обчислює нове значення g(x);
5. Перший процес записує f(x) в об'єкт, що розділяється;
6. Другий процес записує g(x) в об'єкт, що розділяється, знищуючи значення f(x) (наприклад, процеси 1 і 2 - банківські операції над одним і тим самим рахунком).

Для запобігання таким проблемам необхідно забезпечити механізм взаємного виключення. *Взаємне виключення* – це метод створення таких програм, що одночасно не більше ніж один процес має доступ до об'єкта даних, що розділяється. Ділянка коду, в якому здійснюється доступ до об'єкта, що розділяється і який вимагає захисту від втручання інших процесів, називається *критичною секцією*. Ідея полягає в тому, що коли процес готовий виконати свою критичну секцію, він спочатку чекає, поки інший процес не виконає власну критичну секцію. Потім він блокує доступ до критичної секції, не даючи можливості ніякому іншому процесу увійти в свою критичну секцію. Він входить у критичну секцію, виконує її та, вийшовши з неї, звільняє її для доступу з боку інших процесів.

Це завдання можна вирішити мережею Петрі, яка демонструється нижче. Позиція **m** представляє дозвіл для входу в критичну секцію. Для того щоб якийсь процес увійшов у критичну секцію, він повинен мати фішку **p_1** або **p_2** відповідно, що свідчить про бажання потрапити в критичну секцію, а також повинна існувати фішка **m**, що дає дозвіл на вхід. У результаті, управління доступом до критичних секцій двох процесів здійснюється таким чином, що обидва процеси не можуть одночасно виконувати свої критичні секції.



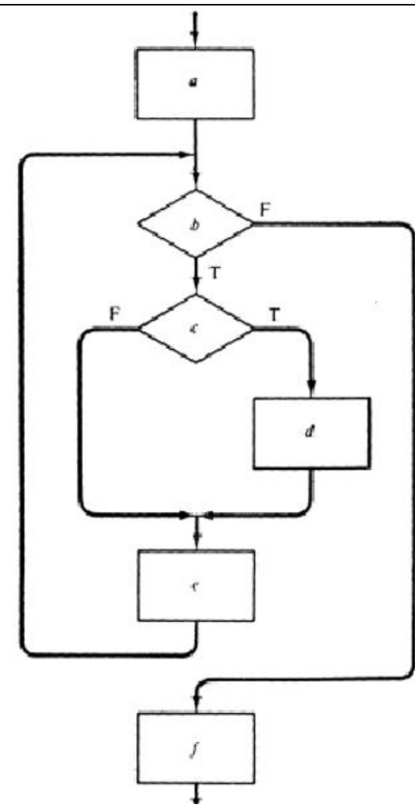
Вироджений випадок паралельної системи обробки є система з одним процесом. Окремий процес описується програмою. Ця програма може бути написана багатьма мовами, але для зручності приймемо загальноцільову гіпотетичну мову. Програма представляє два різні аспекти процесу: обчислення та управління. Обчислення пов'язане з поточними арифметичними та логічними операціями, введенням та висновком, звичайними маніпуляціями над ділянками пам'яті та їх вмістом. Управління ж пов'язане не зі значеннями або обчисленнями, що виконуються, а тільки з порядком їх виконання.

Мережі Петрі успішно представляють структуру управління програм. Мережі Петрі призначені для моделювання впорядкування інструкцій та потоку інформації, але не для дійсного обчислення самих значень. Модель системи за своєю природою є абстракцією системи, що моделюється. Тому вона ігнорує всі можливі специфічні деталі. Якби моделювалися всі деталі, тоді модель була б дублікатом системи, що моделюється, а не абстракцією.

Стандартний спосіб представлення структури управління програм – блок-схеми. Блок-схема представляє потік управління програмою.

```
begin
     Input( y1);
     Input( y2);
     y3  := 1;
     while y1 > 0
          do begin
               if odd ( y1)
                    then begin
                         y3 := y3 * y2;
                         y1 := y1 − 1;
                    end;
               y2 := y2 * y2;
               y1 := y1 − 2;
          end;
     Output ( y3);
end;
```

Дія	Інтерпретація
a	Input(y_1); Input (y_2); $y_3 := 1$;
b	$y_1 > 0$?
c	odd(y_1) ?
d	$y_3 := y_3 * y_2$; $y_1 := y_1 - 1$;
e	$y_2 := y_2 * y_2$; $y_1 := y_1 / 2$;
f	Output (y_3);



Зауважимо, що наведена до програми блок-схема не вказує конкретні обчислення, які треба зробити, а лише визначає структуру програми. Така блок-схема є абстрактною.

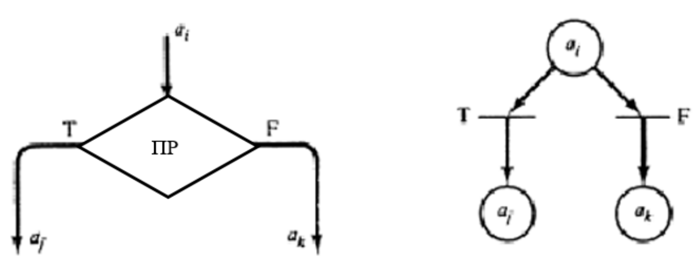
Кожна послідовна програма може бути представлена як блок-схеми.

Виявляється блок-схема багато в чому подібна до мережі Петрі: блок-схема представима у вигляді вузлів двох типів (прийняття рішення, показані ромбами, і обчислення, показані прямокутниками) і дуг між ними. Зручний спосіб виконання блок-схеми – введення фішки, що представляє поточну інструкцію. У міру виконання інструкцій фішка пересувається блок-схемою. Зі подібності між графічними уявленнями програми та мережі Петрі може здатися, що для отримання еквівалентної мережі Петрі можна замінити вузли блок-схеми на позиції, а дуги – на переходи. Але це – не так.

У мережі Петрі дії моделюються переходами, тоді як у блок-схемі дії моделюються вузлами. Крім того, якщо рух нашої фішки поточної інструкції в блок-схемі потрібно призупинити, це робиться між вузлами на дугах, а не всередині блоку. Таким чином, правильне переведення блок-схеми в мережу Петрі замінює вузли блок-схеми на переходи мережі Петрі, а дуги блок-схеми - на позиції мережі Петрі. Кожна дуга блок-схеми відповідає точно одній позиції мережі Петрі. Вузли блок-схеми подаються по-різному залежно від типу вузла: обчислення чи прийняття рішення.

Переведення вузлів обчислення блок-схеми в переходи мережі Петрі:

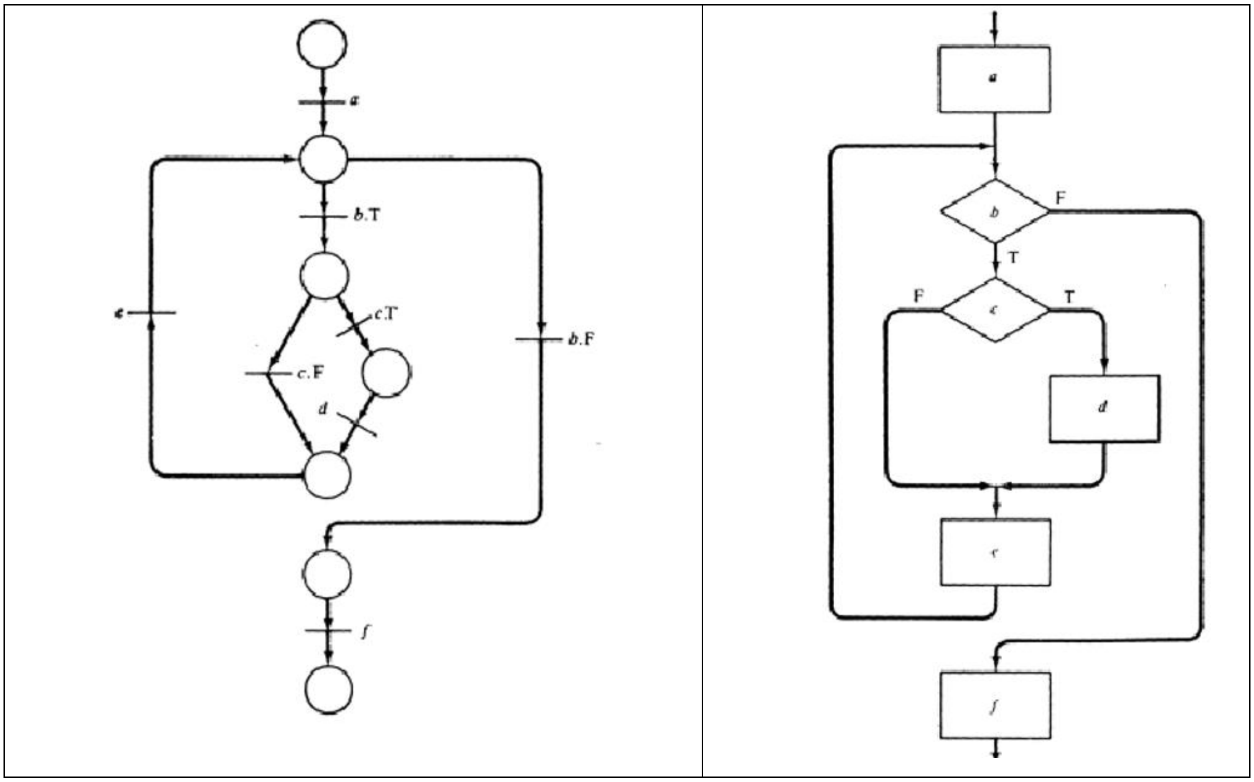
Переведення вузлів прийняття рішень (ПР) блок-схеми в переходи мережі Петрі:



Замислимося, що означають компоненти мережі Петрі - що таке позиція? Для відповіді розглянемо інтерпретацію фішок блок-схеми лічильником команд. Фішка, що знаходиться в позиції, означає, що лічильник команд встановлений на готовність до виконання наступної інструкції. Кожна позиція має єдиний вихідний перехід, за винятком позиції, яка передує ухваленню рішення; такі позиції мають по два вихідні переходи, що відповідають істинному та хибному значенню предикату.

Переходи, очевидно, пов'язуються з діями програми: обчислення та прийняття рішень. Для інтерпретації мережі Петрі необхідно тлумачити кожен перехід. Слід також зазначити, що переходи для обчислень мають один вхід та один вихід; перехід, що представляє обчислення, неспроможний перебувати у конфлікті, оскільки його вхідна позиція не буде вхідною будь-якого іншого переходу. А дія, пов'язана з прийняттям рішення, вводить у мережу конфлікт. Вибір способу вирішення конфлікту або недетермінований, або ним можна керувати ззовні (провісником, що обчислює істинність або хибність предикату і ініціює запуск необхідного переходу). Різниця між цими двома способами вирішення конфлікту – методологічне питання.

Повернемось до прикладу програми і блок-схеми. Представимо блок-схему еквівалентною мережею Петрі:



Завдання 5.1.1.

1) Описати роботу мережі Петрі: I{t1}={P1,P2,P3} I{t2}={P4} I{t3}={P3} O{t1}={P1} O{t2}={P2,P3} O{t3}={P4}, m0=(1,0,1,0).

2) Описати роботу мережі Петрі (позицій - 6, переходів - 5):

I {t1} = {P1}, I {t2} = {P3}, I {t3} = {P2, P3}, I {t4} = {P4, P5}, I {t5} = {P2};

O {t1} = {P2, P3}, O {t2} = {P3, P5}, O {t3} = {P2, P4}, O {t4} = {P4},

O {t5} = {P6}.

Побудувати таблицю перед- і постумов, намалювати схему.

Визначити приклади конфліктів та розпаралелювання.

Завдання 5.1.2.

1) За допомогою мережі Петрі описати процес складання іспиту. При складанні іспиту в аудиторії одночасно може перебувати 2 студенти. Одна людина

готується, а інша відповідає викладачеві. Коли студент закінчує відповідь і йде, другий сідає відповідати, а в аудиторію заходить ще одна людина для підготовки.
2) За допомогою мережі Петрі описати процес синхронізації стрижки в перукарні з двома перукарями.
3) **Задача про обідаючих китайських мудреців** була запропонована Дейкстрой і пов'язана з п'ятьма мудрецями, які поперемінно то думали, то їли. Мудреці сидять за круглим столом, на якому багато страв китайської кухні. Між сусідами лежить одна паличка для їжі. Однак для прийому китайської їжі необхідно дві палички, отже, кожен мудрець повинен взяти паличку зліва та паличку праворуч: проблема, звичайно, полягає в тому, що якщо всі мудреці візьмуть палички зліва і потім чекатимуть, коли звільняться палички з правого боку, то вони чекатимуть вічно і помруть з голоду (стан глухого кута).

Завдання 5.1.3.

1) Побудувати мережу Петрі для алгоритма пошуку мінімального значення з трьох чисел – a, b, c (попарне порівняння, результат зберігається в ‘a’).
2) Побудувати мережу Петрі для розрахунку суми елементів одномірного масиву.

Завдання 5.1.4.

Побудувати мережу Петрі для алгоритмів, що реалізовані наступними фрагментами програм (*Python*):

<table>
<tr><td>

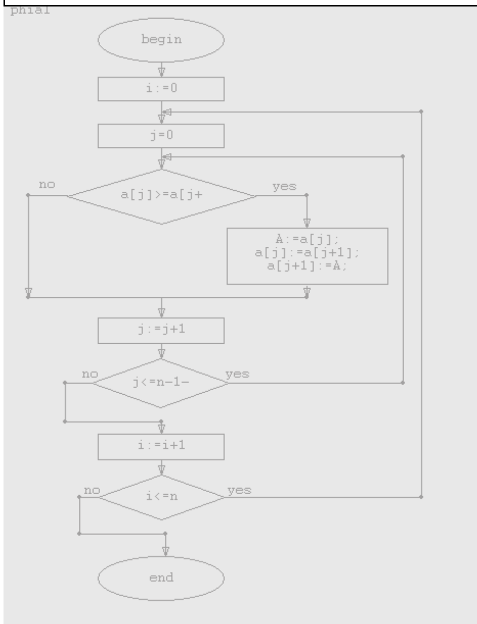
1)
```
do
{
scanf("%c",                    &k);
} while (k != 'y' && k != 'n');
```
</td><td>

2)
```
scanf("%d", &k);
while (k > 0)
{
   k -= 1;
   printf("%d", k);
}
```
</td></tr>
<tr><td>

3)
```
scanf("%d %d", &a, &b);
```
</td><td>

4)
</td></tr>
</table>

if (a > 0) *{int x = a * b; }* *printf("%d", x);*	*if (a > 0) x = a * b;* *else* *x = b;*



Завдання 5.1.5.

Для наведеної блок-схеми алгоритму сортування одномірного масиву методом «бульбочки» надати відповідну мережу Петрі.

5.2. Методи аналізу мереж Петрі

Методи аналізу мереж Петрі можна розбити на наступні три групи:

1) методи на основі дерева покриваємості (дерева досяжності);

2) методи алгебри на основі матричних рівнянь;

3) методи перетворення або декомпозиції.

Методи першої групи зводяться до перерахування всіх досяжних маркувань або відповідних їм маркувань, що покриваються, і, взагалі кажучи, можуть бути застосовані до будь-яких класів мереж, однак на практиці їх використання обмежене «невеликими» мережами через різке зростання складності і потужності простору станів при збільшенні розмірів мережі. У той же час матричні рівняння і методи перетворення мереж, будучи могутнім засобом аналізу, виявляються придатними тільки для окремих підкласів мереж Петрі або лише в окремих випадках.

5.2.1. Дерево покриваемості

<u>Теоретичні відомості</u>

Дерево досяжності представляє собою множину досяжності мережі Петрі. Розглянемо, наприклад, марковану мережу Петрі на рис. 5.4. Початкове маркування – (1, 0, 0). З цього початкового маркування дозволено два переходи: t_1 і t_2. Оскільки ми хочемо розглянути всі множини досяжності, визначимо нові вершини в дереві досяжності *(початковою, корневою є вершина, що відповідає початковому маркуванню)* для досяжних маркувань, що виходять в результаті запуску кожного з цих двох переходів. Дуга, позначена переходом, що запускається, приводить з початкового маркування до кожного з нових маркувань (рис.5.4).

Це (часткове) дерево показує всі маркування, що безпосередньо досяжні із початкового маркування.

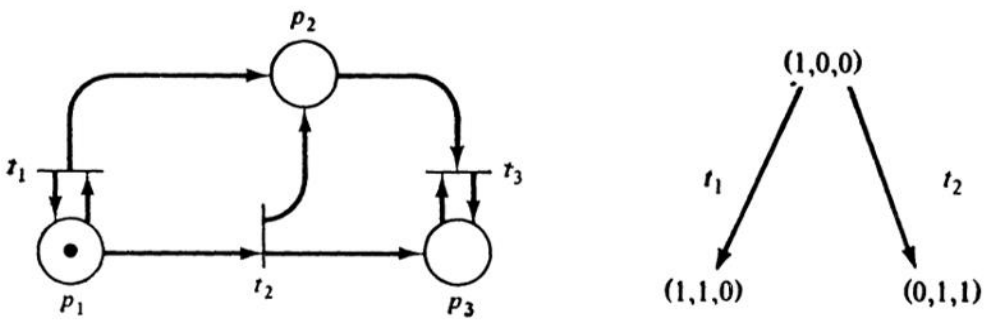


Рис. 5.4. Маркована мережа Петрі, для якої будується дерево досяжності, та перший крок побудови дерева досяжності.

Тепер необхідно розглянути всі маркування, які можна досягти з нових маркувань. З маркування (1, 1,0) можна знову запустити t_1 (отримуючи 1, 2, 0) і t_2 (отримуючи 0, 2, 1); з (0, 1, 1) можна запустити t_3 (отримуючи 0, 0, 1). Це породжує дерево, зображене на рис.5.5.

Рис. 5.5. Другий крок побудови дерева досяжності.

Починаючи з трьох нових маркувань необхідно повторити цей процес, породжуючи нові маркування, які потрібно ввести в дерево, як показано на рис. 5.6. Зауважимо, що маркування (0, 0, 1) пасивне; ніякий перехід у ньому не є дозволеним, тому жодні нові маркування із цього пасивного маркування у дереві не народжуватимуться. Крім того, необхідно зазначити, що маркування (0, 1, 1), що породжується запуском t_3 у (0, 2, 1), було вже породжене безпосередньо з початкового маркування запуском t_2.

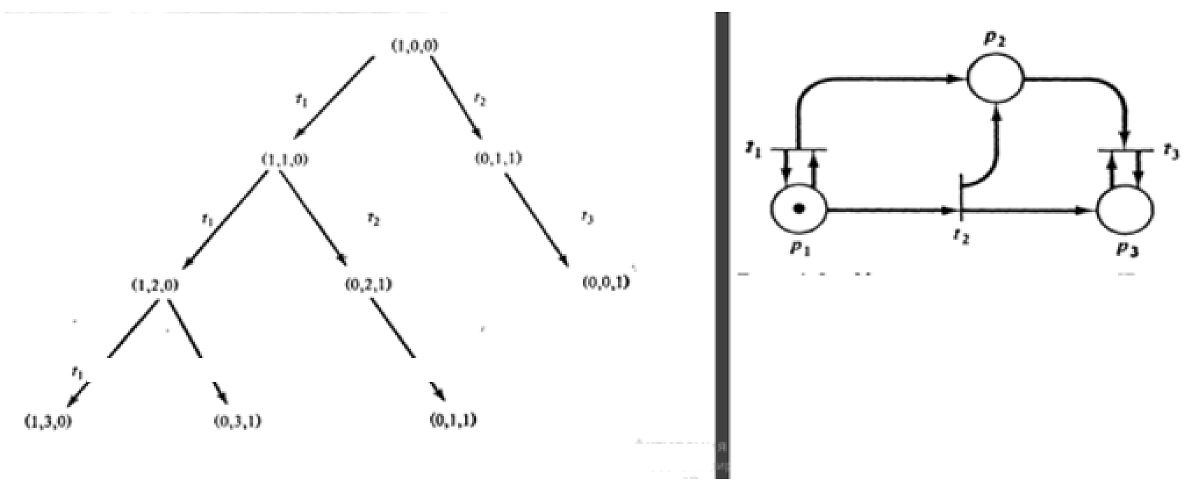


Рис. 5.6. Третій крок побудови дерева досяжності.

Якщо цю процедуру повторювати знову і знову, кожне досяжне маркування виявиться породженим. Проте вийшло, що дерево досяжності може виявитися нескінченним. Буде породжене кожне маркування з множини досяжності, тому для будь-якої мережі Петрі з множиною досяжності відповідне дерево також має бути нескінченним. Дерево репрезентує всі можливі послідовності запусків переходів. Кожен шлях у дереві, що починається в корені, відповідає допустимій послідовності переходів. Для перетворення дерева на корисний інструмент аналізу необхідно знайти засоби обмеження його до кінцевого розміру. Зауважимо, що якщо якесь уявлення нескінченної множини звичайно, то нескінченна множина маркувань має відображатися в таке уявлення. У загальному випадку це призведе до втрати інформації і, можливо, до того, що деякі властивості мереж Петрі визначити буде не можна, але все залежить від того, як уявлення було отримано.

Приведення до кінцевого уявлення здійснюється декількома способами. Нам необхідно знайти засоби, які обмежують введення нових маркувань (називаються *граничними вершинами*) на кожному кроці. Тут можуть допомогти пасивні маркування – маркування, які не мають дозволених переходів. Ці *пасивні маркування називають термінальними вершинами.* Інший клас маркувань – це маркування, що раніше зустрічалися в дереві. Такі *дублюючі маркування називаються дублюючими вершинами*; ніякі наступні маркування розглядати немає потреби – всі вони будуть породжені з місця першої появи дублюючого маркування в дереві.

Таким чином, у дереві на рис. 5.6 маркування (0, 1, 1), що вийшло в результаті виконання послідовності $t_1t_2t_3$, не породжуватиме жодних нових вершин у дереві, оскільки воно раніше зустрічалося в дереві в результаті виконання послідовності t_2 з початкового маркування.

Для зведення дерева досяжності до кінцевого втілення використовується ще один засіб. Розглянемо послідовність запусків переходів σ, що починається в початковому маркуванні μ і закінчується в маркуванні μ', $\mu' > \mu$. Маркування μ'

збігається із маркуванням μ за винятком того, що має деякі «додаткові» фішки в деяких позиціях, тобто $\mu' = \mu + (\mu' - \mu)$ і $(\mu' - \varphi) > 0$.

Тепер, оскільки на запуски переходів зайві фішки не впливають, послідовність σ можна запустити знову, починаючи в μ' і пересуваючись до маркування μ''. Отже, для тих позицій, які збільшують число фішок послідовністю σ, можна створити довільно велику кількість фішок, просто повторюючи послідовність стільки, скільки це потрібно. У мережі Петрі на рис. 1, наприклад, можна запустити перехід t_1 стільки разів, скільки необхідно для того, щоб отримати довільне число фішок в p_2.

Представимо нескінченну кількість маркувань, що виходять із циклів такого типу, за допомогою спеціального символу ω, який позначає «нескінченність». Для будь-якого постійного α $(\alpha < \omega)$ визначимо:

Зауважемо, що **$\omega+\alpha=\omega$, $\omega-\alpha=\omega$.** Для побудови дерева досяжності необхідні тільки ці операції над ω.

Тепер можна точно сформулювати дійсний алгоритм побудови дерева досяжності. Кожна вершина ***i*** дерева пов'язується з розширеним маркуванням $\mu[i]$; у розширеному маркування число фішок у позиції може бути або натуральним числом, або $\boldsymbol{\omega}$. **Кожна вершина класифікується або як гранична вершина, термінальна вершина, дублююча вершина, або як внутрішня вершина. Граничними є вершини, які ще оброблюються алгоритмом; алгоритм перетворить їх на термінальні, дублюючі або внутрішні вершини.**

Алгоритм починає з визначення початкового маркування коренем дерева, тобто граничною вершиною. Доки є граничні вершини, вони обробляються алгоритмом.

Нехай ***x*** гранична вершина, яку потрібно обробити.

1. Якщо в дереві є інша вершина, що не є граничною, і з нею пов'язане та ж маркування, то вершина дублююча.

2. Якщо для маркування жоден із переходів не дозволено, то *x* –термінальна вершина.

3. Для будь-якого переходу, дозволеного в $\mu[x]$, є можливість створити нову вершину *y* дерева досяжності. Тоді вершина *x* перевизначається як внутрішня, вершина *y* стає граничною.

Коли всі вершини дерева – термінальні, дублюючі чи внутрішні, алгоритм зупиняється.

На рис. 5.7 представлене дерево досяжності мережі Петрі з рис. 5.4.

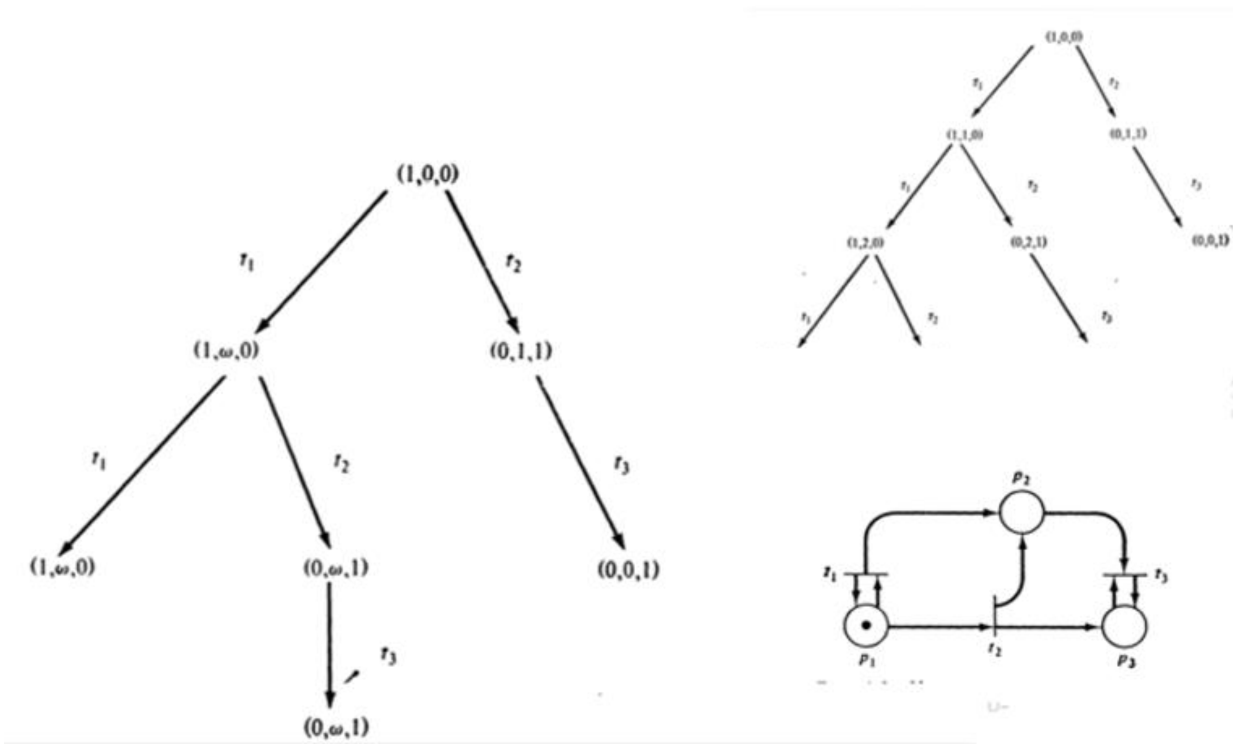
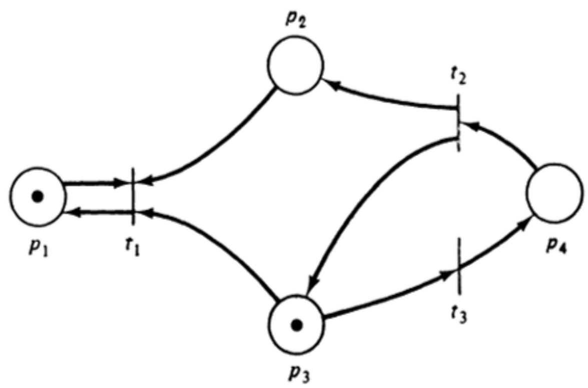
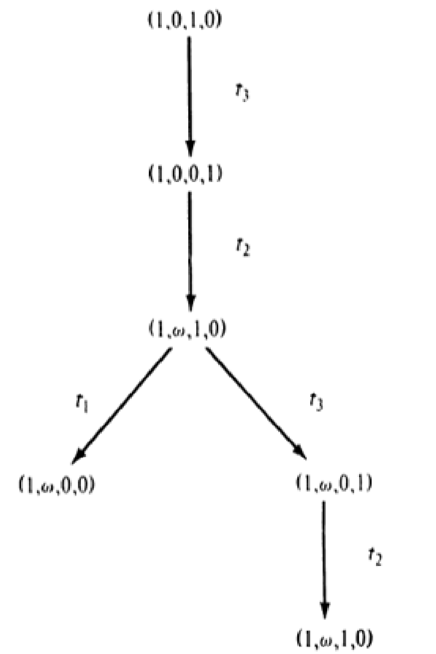


Рис. 5.7. Дерево досяжності мережі Петрі, наведеної на рис. 5.4.

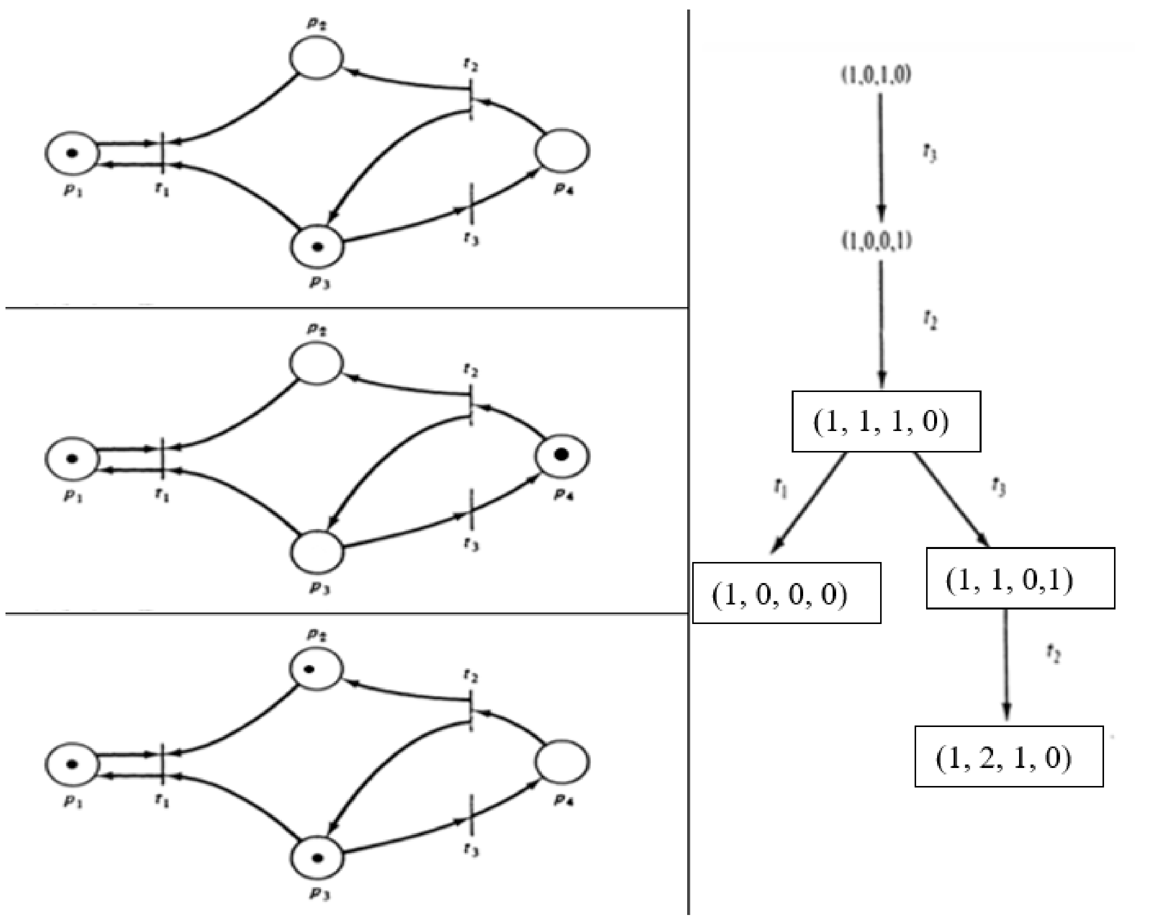
Розглянемо мережу Петрі і її дерево досяжності.

В дереві досяжності мережі Петрі, існує нескінченний шлях, що виходить із кореня.





Покроково:

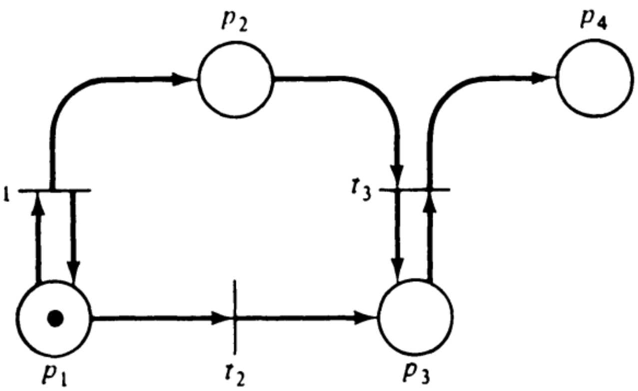


Дерево досяжності – дуже корисний інструмент аналізу мереж Петрі.

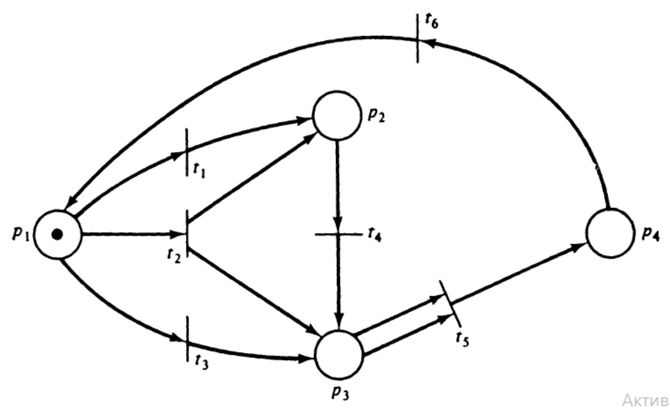
Завдання 5.2.1.

Побудувати дерево досяжності для зображеної мережі Петрі.

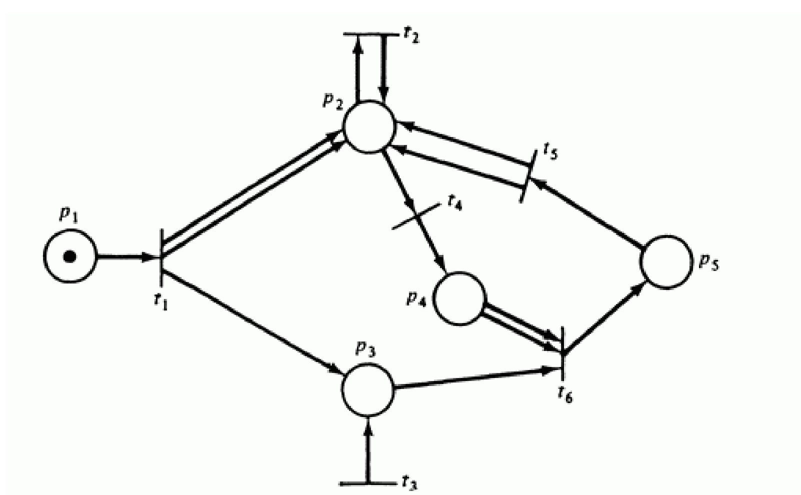
1)



2)



3)



5.2.2. Матриця інцидентності та рівняння стану

Теоретичні відомості

Другий підхід до аналізу мереж Петрі ґрунтується на матричному поданні мереж Петрі. Альтернативним по відношенню до визначення мережі Петрі у вигляді (P, T, I, O) є визначення двох матриць $\boldsymbol{D^-}$ і $\boldsymbol{D^+}$, що представляють вхідну та вихідну функції. Кожна матриця має $\boldsymbol{m}$ рядків (переходи) і $\boldsymbol{n}$ стовпців (позиції).

Визначимо $\boldsymbol{D^-[j, i] = \#(p_i, I(t_j))}$, а $\boldsymbol{D^+[j, i] = \#(p_i, O(t_j))}$. $\boldsymbol{D^-}$ визначає входи у переходи, $\boldsymbol{D^+}$ **–** виходи.

Матрична форма визначення мережі Петрі еквівалентна стандартній формі, але дозволяє дати визначення термінах векторів і матриць. Нехай e[j] –вектор-рядок, що містить нулі скрізь, крім j компоненти. Перехід t_j представляється m-вектором e[j].

Тепер перехід t_j у маркуванні μ дозволено, якщо $\mu \geq e[j]\, D^-$, результат запуску переходу t_j у маркуванні μ записується як:

$\delta(\mu, t_j) = \mu - e[j]\, D^- + e[j]\, D^+ = \mu + e[j]\,(-D^- + D^+) = \mu + e[j]\, D$,

де $D = D^+ - D^-$ – складова матриця змін.

ВХІДНА МАТРИЦЯ (входи в переходи з позицій)

$$\mathbf{D^{-}} = \begin{array}{c|cccc} & P_1 & P_2 & P_3 & P_4 \\ \hline t_1 & & & & \\ t_2 & & & & \\ t_3 & & & & \end{array}$$

фішки втрачаємо від P до t

ВИХІДНА МАТРИЦЯ (виходи з переходів в позиції)

$$\mathbf{D^{+}} = \begin{array}{c|cccc} & P_1 & P_2 & P_3 & P_4 \\ \hline t_1 & & & & \\ t_2 & & & & \\ t_3 & & & & \end{array}$$

фішки набуваємо від t до P

Тоді для послідовності запусків переходів $\sigma = t_{j1}t_{j2}\ldots t_{jk}$ маємо:

$\delta(\mu, \sigma) = \delta(\mu, t_{j1}, t_{j2}, \ldots, t_{jk}) =$

$= \mu + e[j_1]\,D + \ldots + e[j_k]\,D = \mu + (e[j_1] + \ldots + e[j_k])\,D =$

$= \mu + f(\sigma)\,D.$

Вектор $f(\sigma) = e[j_1] + \ldots + e[j_k]$ називається вектором запусків послідовності $t_{j1}t_{j2}\ldots t_{jk}$. ***і-ий*** елемент вектора $f(\sigma)$ – це число запусків переходу t_i в послідовності $t_{j1}t_{j2}\ldots t_{jk}$. Вектор запусків, отже, є вектором з натуральними компонентами.

Якщо маркування μ' досяжне з початкового маркування μ, то тоді існує послідовність запусків переходів σ, яка призводить μ до μ':

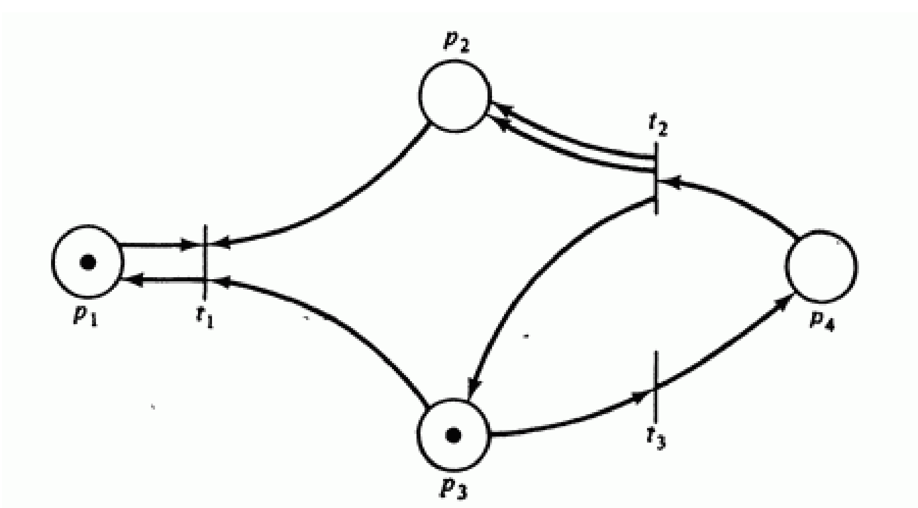
$\boldsymbol{\mu' = \delta(\mu, \sigma) = \mu + f(\sigma)\,D}$. Це і є наше головне матричне рівняння.

Якщо зліва х, то це пряма задача: пошук кінцевого маркування з початкового при заданому векторі запуску переходів: $\boldsymbol{x = \mu + f(\sigma)\,D}$.

Зворотна задача – визначити необхідний вектор запуску для досягнення кінцевого маркування (ліворуч) з початкового заданого: $\boldsymbol{\mu' = \mu + x\,D}$. Вектор-рядок $x = f(\sigma) = e[j_1] + \ldots + e[j_k]$, координати якого – кількість запусків відповідного переходу; порядок запусків переходів при цьому не визначається.

І пряма і зворотна задача має сенс коли рівняння має рішення в натуральних числах; якщо рівняння немає рішення, тоді $\boldsymbol{\mu'}$ недосяжна з $\boldsymbol{\mu}$, наприклад.

Розглянемо, наступну марковану мережу Петрі:



Матриці D^- і D^+ мають вигляд:

$$D^- = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$D^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

а матриця D:

$$D = \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix}.$$

У початковому маркуванні μ=(1, 0, 1, 0) перехід t_3 дозволено і призводить до маркування μ', де

$$\mu' = (1, 0, 1, 0) + (0, 0, 1) \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix} =$$
$$= (1, 0, 1, 0) + (0, 0, -1, +1) = (1, 0, 0, 1).$$

Послідовність $\sigma=t_3t_2t_3t_2t_1$ представляється вектором запусків $f(\sigma)$=(1, 2, 2) і отримує маркування μ':

$$= (1, 0, 1, 0) + (1, 2, 2) \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix} =$$
$$= (1, 0, 1, 0) + (0, 3, -1, 0) = (1, 3, 0, 0).$$

Для визначення того, чи є маркування (1, 8, 0, 1) досяжним з маркування (1, 0, 1, 0), маємо рівняння

$$(1, 8, 0, 1) = (1, 0, 1, 0) + x \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix},$$

$$(0, 8, -1, 1) = x \cdot \begin{bmatrix} 0 & -1 & -1 & 0 \\ 0 & +2 & +1 & -1 \\ 0 & 0 & -1 & +1 \end{bmatrix},$$

яке має рішення *x=(0 4, 5)*. Це відповідає послідовності $\sigma = t_3t_2t_3t_2t_3t_2t_3t_2t_3$.

Далі ми можемо показати, що маркування (1, 7, 0, 1) є недосяжним з маркування (1, 0, 1, 0), оскільки матричне рівняння

$$(1, 7, 0, 1)=(1, 0, 1, 0)+x\cdot\begin{bmatrix}0 & -1 & -1 & 0\\ 0 & +2 & +1 & -1\\ 0 & 0 & -1 & +1\end{bmatrix}$$

$$(0, 7, -1, 1)=x\cdot\begin{bmatrix}0 & -1 & -1 & 0\\ 0 & +2 & +1 & -1\\ 0 & 0 & -1 & +1\end{bmatrix}$$

не має рішення:

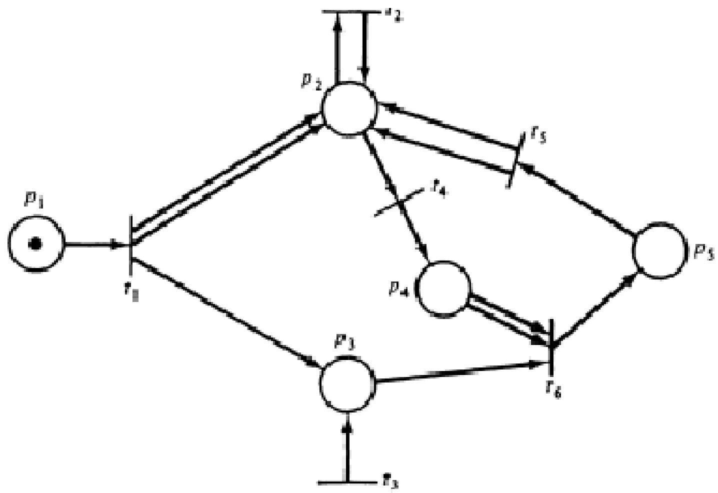
*1) 0= e1*0+e2*0+e3*0*

2) 7= e1(-1)+e2*2+e3*0*
3) -1= e1(-1)+e2*1+e3*(-1)*
*4) 1= e1*0+e2*(-1)+e3*1*

~~*1) 0= e1*0+e2*0+e3*0*~~
З 4) 1=–e2+e3.
Підставляємо в 3)
-1= –e1+ (-1), тобто, e1=0.
Тоді з 2) ***e2=3,5****.*

Матричний підхід до аналізу мереж Петрі дуже перспективний, але має деякі труднощі. Зауважимо насамперед, що матриця D не в повному обсязі відбиває структуру мережі Петрі. Переходи, що мають як входи, так і виходи з однієї позиції (петлі), представляються відповідними елементами матриць, D^+ і D^-, але потім взаємно знищуються в матриці D. Інша проблема – відсутність інформації о послідовності виконання переходів у векторі запуску.

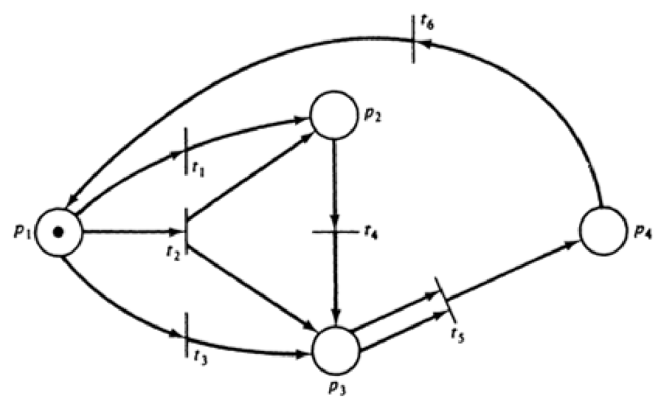
Завдання 5.2.2.



1) пряма задача: знайти маркування при початковому (0, 1, 1, 2, 0) та запуску переходів t_5 і t_6;

2) зворотна задача: знайти послідовність запуску перехлдів для досягнення маркування (0, 1, 1, 1, 0) з початкового (1, 0, 0, 0, 0).

Завдання 5.2.3.



1) Знайти досяжне маркування ***при*** $\boldsymbol{M_0 = (1, 0, 0, 0)}$ та при запуску:

а f = (0, 1, 0, 1, 0, 0)

б f= (0, 1, 0, 1, 1, 1)

в f = (0, 1, 1, 0, 0, 0)

2) Розглянути можливість досяжності, тобто знайти вектор запуску,

при $\boldsymbol{M_0 = (1, 0, 0, 0)}$ для:

а M = (1, 0, 0, 0)

б M = (0, 0, 1, 0)

в M = (0, 1, 2, 0)

Відповіді до розділу 5

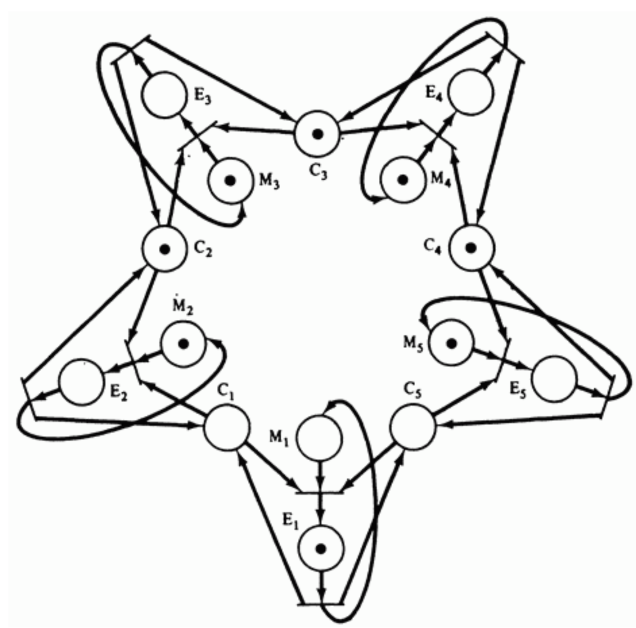
5.1.1

2) Конфлікт породжується через позиції P_2 і P_3, а розпаралелювання – в переходах t1, t2, t3.

ПЕРЕДУМОВА	ПЕРЕХІД	ПОСТУМОВА
P1	t1	P2, P3
P3	t2	P3, P5
P2, P3	t3	P2, P4
P4, P5	t4	P4
P2	t5	P6

5.1.2.

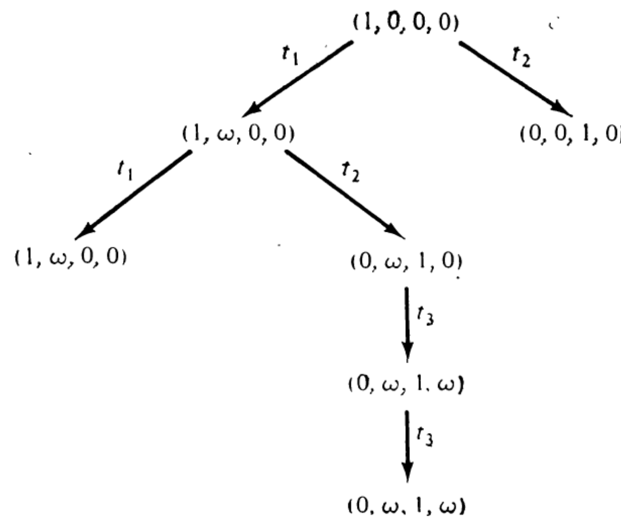
3) Позиції представляють палички для їжі C_1, …, C_5, і оскільки кожна з них спочатку вільна, то початковому маркуванні у кожній з цих позицій є фішка. Кожен мудрець представлений двома позиціями: роздум M_i і прийняття їжі E_i.



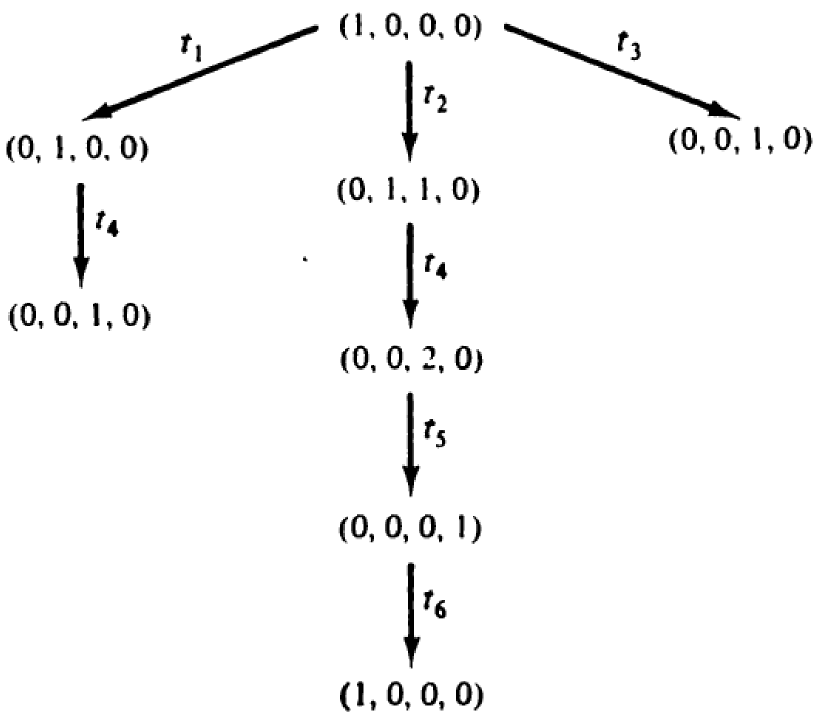
Для того щоб мудрець перейшов зі стану роздумів у стан їжі, обидві палички (ліворуч і праворуч) повинні бути вільні. Це легко моделюється мережею Петрі.

5.2.1.

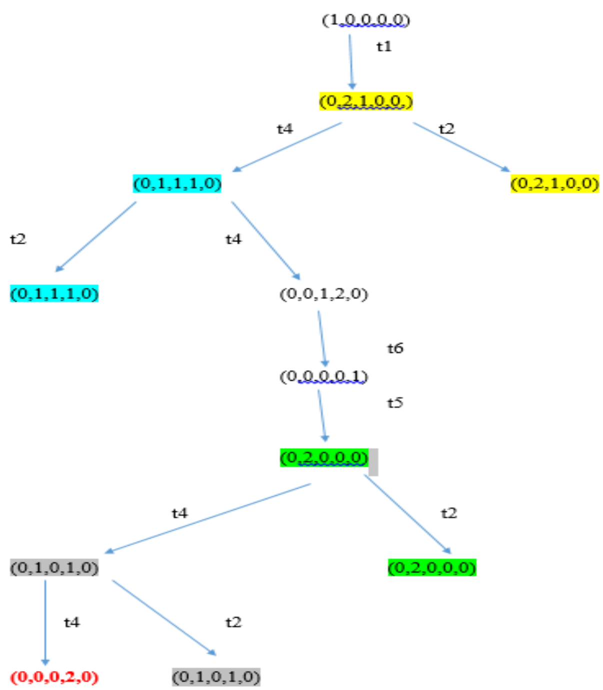
1)



2)



3)



5.2.2.

D^+	**P1**	**P2**	**P3**	**P4**	**P5**
t1	0	2	1	0	0
t2	0	1	0	0	0
t3	0	0	1	0	0
t4	0	0	0	1	0
t5	0	2	0	0	0
t6	0	0	0	0	1

D^-	**P1**	**P2**	**P3**	**P4**	**P5**
t1	1	0	0	0	0
t2	0	1	0	0	0
t3	0	0	0	0	0
t4	0	1	0	0	0
t5	0	0	0	0	1
t6	0	0	1	2	0

Рівняння: $\mu' = \mu + f(\sigma) * D$

Матриця змін D:

$$\begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & -1 & -2 & 1 \end{bmatrix}.$$

1) пряма задача: знайти маркування при початковому (0, 1, 1, 2, 0) та запуску переходів t_5 і t_6.

$$(0\ 1\ 1\ 2\ 0)\ +(0\ 0\ 0\ 0\ 1\ 1)\begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & -1 & -2 & 1 \end{bmatrix}. = (0\ 3\ 0\ 0\ 0)$$

2) зворотна задача: знайти послідовність запуску перехлдів для досягнення маркування (0, 1, 1, 1, 0) з початкового (1, 0, 0, 0, 0).

(0 1 1 1 0) = (1 0 0 0 0) + (t1 t2 t3 t4 t5 t6) D ;

$$(-1\ 1\ 1\ 1\ 0) = (t1\ t2\ t3\ t4\ t5\ t6)\ \begin{bmatrix} -1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & -1 & -2 & 1 \end{bmatrix}.$$

-t1=-1	**t1=1**	
2t1-t4+2t5=1	2t5-t4=-1 *(а)*	
t1+t3-t6=1 ➔	t3=t6	*(а, б)* ➔ **t4=1+2C**
t4-2t6=1	t4-2t6=1 *(б)*	
-t5+t6=0	**t5=t6 *(=t3=C)***	

Підбираємо. наприклад, C=0, тоді відповідь (1 0 0 1 0 0) ,
тобто запускаємо t1 і t4 *(але ще може і t2 – про нього нічого не відомо).*

При розв'язанні матричних рівнянь, якщо ми отримуємо дробові або від'ємні значення, то виходить ця ситуація до даної мережі не може бути застосована. Додатково - при вирішенні зворотньої задачі ми можемо не отримати однозначний розв'язання системи рівнянь (отриманої при множенні вектора-рядка на матрицю) - теж не може бути застосовано.

Крім цього, отриманий вектор запуску, або заданий при підрахунку маркування в прямій задачі, можуть не існувати в природі даної мережі. А порядок проходження запусків ніколи не буває визначено. Це і є недоліки аналізу мереж Петрі за допомогою матриць, необхідно додавати інші методи аналізу.

5.2.3.

D^+	P1	P2	P3	P4
t1	0	1	0	0
t2	0	1	1	0
t3	0	0	1	0
t4	0	0	1	0
t5	0	0	0	1
t6	1	0	0	0

D^-	P1	P2	P3	P4
t1	1	0	0	0
t2	1	0	0	0
t3	1	0	0	0
t4	0	1	0	0
t5	0	0	2	0
t6	0	0	0	1

$D = D^+ - D^-$	P1	P2	P3	P4
t1	-1	1	0	0
t2	-1	1	1	0
t3	-1	0	1	0
t4	0	-1	1	0
t5	0	0	-2	1
t6	1	0	0	-1

$M = M_0 + f*D$

1)

а M = (0, 0, 2, 0)

б M = (1, 0, 0, 0) початкове

в M = (-1, 1, 2, 0) не існує

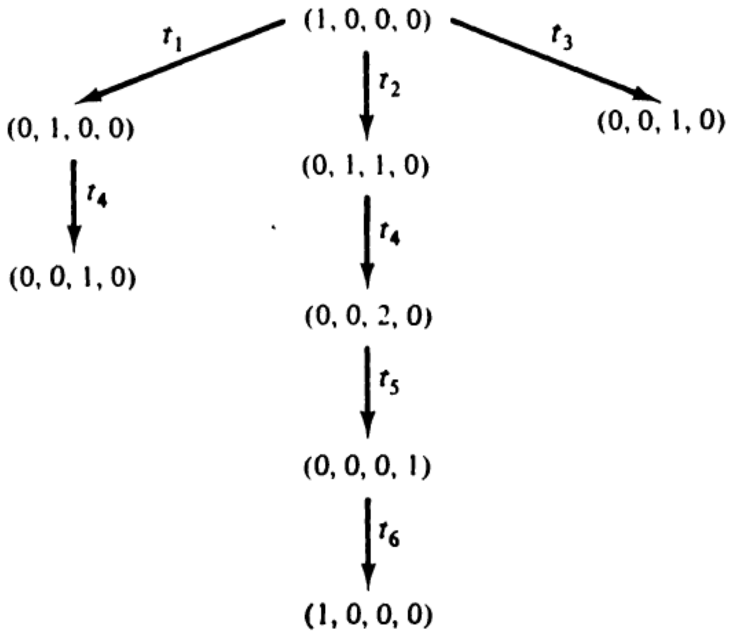
2)

а f = (0, 1, 0, 1, 1, 1) або (0, x, 0, x, x, x), де x = 0, 1, 2, 3,…. – центральна гілка дерева

б f = (0, 1, 1, 1, 1, 1) або (1, 0, 0, 1, 0, 0) – права або ліва гілки дерева – замість червоного можуть бути будь-які однакові значення x=0,1,2,… - центральна гілка, яка повертає до початкового маркування, а далі вже виконується права або ліва, що призводить до тупика (0, 0, 1,0)

в f = ( ) не існує

Для перевірки і повноцінного аналізу – побудуйте дерево досяжності.



ДОДАТОК 1.

ОГЛЯД ТЕОРІЇ КОМПЛЕКТІВ

Теорія множин використовується в математиці та обчислювальній техніці тривалий час. Теорія комплектів є природним розширенням теорії множин. Комплект, подібно до множини є набір елементів з певної області. Однак на відміну від множини комплекти допускають присутність *декількох* екземплярів одного й того самого елемента. В теорії множин елемент або є елементом множини, або ні є елементом множини. В теорії комплектів елемент може входити в комплект нуль разів (не входити в комплект) або один, два, три або будь-яке задане число разів.

Як приклад розглянемо наступні комплекти над областю {a, b, c, d}: B_1= {a, b, c), B_2 = {a}, B_3 = {a, b, c, c}, B_4 = {a , a, a}, B_5= { b, c, b, c}, B_6= {c, c, b, b }, B_7 = {a, a, c, a, a, b, b, c, d, d, d, d, d, d, d}. Деякі комплекти є множинами (наприклад, B_1 і B_2). Так само як і в множині, порядок елементів у комплекті не важливий. Тому B_5 і B_6 є тим самим комплектом (упорядковані комплекти називаються послідовностями).

В теорії множин основним поняттям є відношення приналежності. Це відношення пов'язує елементи та множини та визначає, які елементи є членами яких множин. Основним поняттям теорії комплектів є функція числа екземплярів. Ця функція визначає кількість екземплярів елемента у комплекті. Позначимо число екземплярів елемента x у комплекті B через #(x, B) (читається «число входжень, кількість x в B»).

Виходячи із цього поняття, можемо визначити основи теорії комплектів. Більшість понять та позначень запозичені з теорії множин. Якщо ми обмежимо число елементів у комплекті так, що $0 \leq \#(x, B) \leq 1$, то отримаємо теорію множин.

1. Членство

Функція #(x, B) визначає число екземплярів елемента x у комплекті B. Звідси випливає, що #(x, B) > 0 для всіх x та B. Ми розрізняємо нульовий та ненульовий випадки. Елемент x є членом комплекту, якщо # (x, B) > 0. Це ми позначатимемо через $x \in B$. Аналогічно, якщо #(x, B) = 0, то $x \notin B$.

Визначимо *пустий комплект* $\varnothing$, який не має членів (для всіх x: #(x, $\varnothing$) = 0).

2. Потужність

Потужність комплекту |B| є загальна кількість екземплярів елементів у комплекті $|B| = \Sigma\#(x, B)$.

3. Включення та рівність комплектів

Комплект A є *підкомплект* комплекту B (позначається $A \subseteq B$), якщо кожен елемент A є елементом щонайменше не більше разів: $A \subseteq B$ тоді і тільки тоді, коли $\#(x, A) \le \#(x, B)$ для всіх x. *Два комплекти рівні* (A=B), якщо #(x, A) = #(x, B) для всіх x.

З цих визначень ми можемо безпосередньо показати, що A=B тоді і тільки тоді, коли $A \subseteq B$ і $B \subseteq A$.

Комплект A *строго включений* у комплект B ($A \subset B$), якщо $A \subseteq B$ і $A \neq B$. Тобто не для всіх x виконується строга нерівність, але хоч одна повинна бути.

$\varnothing \subseteq B$ для будь - якого комплекту; з A=B слідує |A|=|B|; з $A \subseteq B$ випливає $|A| \le |B|$. Зазначимо, що з $|A| \le |B|$ не випливає $A \subseteq B$.

4. Операції

Над комплектами можна ввести чотири бінарних операції. Для двох комплектів *A* та *B* визначимо:

- ***об'єднання комплектів $A \cup B$: # (x, $A \cup B$) = max (#{x, A), #(x, B));***
- ***перетин комплектів $A \cap B$: # (x, $A \cap B$) = min (# (x, A), # (x, B));***
- ***суму комплектів A + B : # (x , A + B) = # (x, A) + # (x, B);***
- ***різницю комплектів A – B; # {x, A - B) = # (x, A) - # (x, $A \cap B$).***

Зауважимо, що + не означає симетричну різницю класичної теорії множин у даному випадку; співпадає тільки для диз'юнктних множин.

Ці операції мають більшість очікуваних властивостей. Об'єднання, перетин та сума комутативні та асоціативні, крім того, справедливі очікувані включення:

$$\boldsymbol{A \cap B \subseteq A \subseteq A \cup B;}$$

$$\boldsymbol{A - B \subseteq A \subseteq A + B.}$$

Відмінність між об'єднанням та сумою очевидна:

$$\boldsymbol{|A \cup B| \leq |A| + |B|;}$$

$$\boldsymbol{|A + B| = |A| + |B|^{*}.}$$

На жаль, відмінності між $A \cap B$ та A–B не можна так само легко проілюструвати, що пояснюється неможливістю для операції різниці видалення елементів із комплекту, які не входять до нього.

5. Простір комплектів

Визначимо область D як множину елементів, з яких складаються комплекти. Простір комплектів D^n є множина всіх таких комплектів, що їх елементи належать D, і жоден елемент не входить в комплект більше n разів. Інакше кажучи, для будь-якого $B \in D^n$:

1. З $x \in B$ випливає $x \in D$.

2. Для будь-кого x $\#(x, B) \leq n$.

Множина D^{∞} є множиною всіх комплектів над областю D без будь-якого обмеження на кількість екземплярів елемента в комплекті.

6. Відображення Паріха

Для кінцевої області D = {d1, d2, ..., dn} існує природна відповідність між кожним комплектом B над D і n-вектором f = (f1, f2, …, fn), що визначається співвідношенням fi = # (di, B)

Цей вектор відомий як відображення Паріха.

---

* Чи можна ввести поняття «диз'юнктних» комплектів по аналогії з теорією множин**,** *чи буде таке поняття доречним до цих нерівностей?*

ДОДАТОК 2

Методичні вказівки до виконання

КОМПЛЕКСНИХ КОНТРОЛЬНИХ РОБІТ

з кредитного модуля: Дискретна математика. Частина 2.

Булева алгебра

для студентів спеціальності 113 «Прикладна математика»,

освітньо-професійної програми «Наука про дані та математичне моделювання»

Метою викладання кредитного модуля «Дискретна математика. Частина 2. Булева алгебра» є оволодіння основними поняттями і методами, потрібними для вивчення наступних дисциплін спеціальності, формування світогляду на дискретну математику як на фундаментальну науку, що призначена для формалізації знань, у тому числі математичних наук.

Закріплення отриманих знань з кредитного модуля «Дискретна математика. Частина 2. Булева алгебра» перевіряється комплексною контрольною роботою (ККР). ККР містить 40 завдань, однакових за структурою, кожне із яких складається з трьох запитань, сформованих за різними темами. Всі завдання різняться, жодна з задач в запитаннях не повторюється. ККР перевіряє залишкові знання за дисципліною, але вирішення задач потребує уміння застосовувати інтегровані знання всього програмного матеріалу дисципліни, що дозволяє студентам продемонструвати саме вміння використовувати набуті знання для вирішення практично спрямованих завдань.

Під час ККР використання довідкової літератури, обладнання, приладів, матеріалів, комп’ютерних програм тощо не передбачено.

Максимальна кількість балів, яка нараховується за виконання окремого питання контрольного завдання (КЗ), визначена і враховує ступінь його важливості та рівень складності: перше запитання КЗ оцінюється в 50 балів, друге - 20 балів, третє -30 балів.

Перше запитання «Побудувати таблицю істинності для заданої булевої функції f(x,y,z)» перевіряє практичні знання, набуті за таким важливим для вивчення дисципліни «Дискретна математика» розділом, як булева алгебра.

Для побудови таблиці істинності для заданої булевої функції треба враховувати визначення формул, функцій, наборів тощо з даного розділу.

У зображенні булевих формул прийняті наступні допущення: зовнішні дужки опускають; установлюють пріоритети виконання операцій у наступному порядку:

$\neg$ – заперечення (найвищий пріоритет),

$\wedge$ - кон'юкція,

$\vee$ – диз'юнкція,

$\rightarrow$, $\equiv$ імплікація й еквівалентність (мають однаковий пріоритет). З урахуванням цих пріоритетів надлишкові дужки також опускаються.

Всі набори розташуємо в порядку зростання їхніх номерів, а поруч із кожним набором помістимо значення булевої функції на цьому наборі. Отриману в такий спосіб таблицю назвемо таблицею значень, або таблицею істинності булевої функції. Для кожної булевої формули можна побудувати її таблицю істинності, обчислюючи її значення на кожному наборі її змінних.

У таблиці наведені деякі функції від двох змінних.

x	y	f1	f6	f7	f9	f10	f12	f13
0	0	0	0	0	1	1	1	1
0	1	0	1	1	0	0	1	1
1	0	0	1	1	0	1	0	0
1	1	1	0	1	1	0	0	1

Це функції:

f1

– $x \wedge y$ (или x & y) – кон'юнкція

f6

– $x \oplus y$ – додавання по mod 2 (операція Жегалкіна)

f7

– $x \vee y$ – диз'юнкція

f9

– $x \equiv y$ (или $x \sim y$) – еквівалентність

f10

– заперечення $f(y) = \neg y$

f12

– заперечення $f(x) = \neg x$

f13

– $x \rightarrow y$ (или $x \supset y$) – імплікація

Друге запитання «Визначити, чи є формула тавтологією, за допомогою методу редукції» перевіряє здібності до логічного мислення, вікористання формальних законів алгебри висловлювань.

Визначити, чи є формула тавтологією, можна одним із трьох способів: побудувати таблицю істинності, перевірити це алгебраїчно або методом редукції (від супротивного).

Наприклад, для формули $E=((A\to B)\to(A\to C))\to(A\to(B\to C))$ вирішимо задачу методом редукції.

Припустимо, що формула не є тавтологією (тавтологія – формула, яка на всіх наборах приймає значення «істина»). Це має означати, що існує такий набір, на якому формула $|((A\to B)\to(A\to C))\to(A\to(B\to C))|=F$ (позначення наступні: істина – true – T, хибна – false – F).

Імплікація приймає значення хибна тільки в тому випадку, коли посилка – істина, а вивід – хибний. Отже, $|(A\to B)\to(A\to C)|=T$, а $|(A\to(B\to C)|=F$.

З останнього треба, що $|A|=T$, а $|B\to C|=F$, і, тоді $|B|=T$ и $|C|=F$. Підставимо отримані значення в першу частину виразу : $|(A\to B)\to(A\to C)|= |(T\to T)\to(T\to F)|= |T\to F|=F$, а не T, як припускалося. Отримане протиріччя доказує неможливість рівності виразу F, тобто, при будь-яких входженнях воно буде істинно. Це - тавтологія.

Але можна побудувати й таблицю істинності:

A	B	C	$A\to B$	$A\to C$	$(A\to B)\to(A\to C)$	$B\to C$	$A\to(B\to C)$	**E**
F	F	F	T	T	T	T	T	**T**
F	F	T	T	T	T	T	T	**T**
F	T	F	T	T	T	F	T	**T**
F	T	T	T	T	T	T	T	**T**
T	F	F	F	F	T	T	T	**T**
T	F	T	F	T	T	T	T	**T**
T	T	F	T	F	F	F	F	**T**
T	T	T	T	T	T	T	T	**T**

Значення формули при будь-яких входженнях A, B, C дорівнює T, отже, це тавтологія.

Третє запитання перевіряє закріплення знань з теми «Абстрактні автомати» і передбачає видати у якості відповіді на вхідне слово вихідне слово у заданому алфавіті при наявності опису дії автомата у вигляді таблиць виходів та переходів. Наприклад, автомат заданий множиною станів $Q = \{A, B, C\}$, множиною вхідних і вихідних сигналів $X = Y = \{0,1\}$ і таблицями переходів(1) і виходів(2). Початковий стан – A.

Табл.1

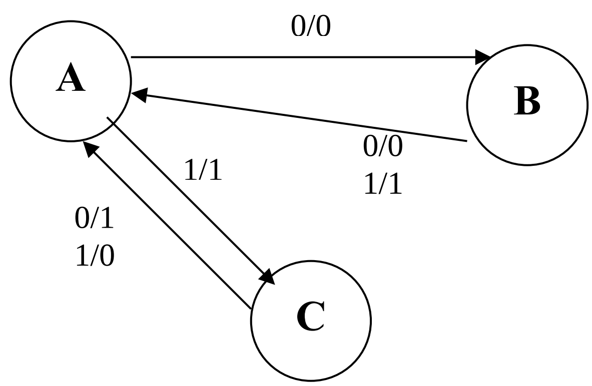
	A	**B**	**C**
0	B	A	A
1	C	A	A

Табл.2

	A	**B**	**C**
0	0	0	1
1	1	1	0

Сформувати вихідне слово, якщо на вхід подано 10110001.

Цей автомат реалізує булеву функцію «додавання за модулем 2» (x⊕y) двох двійкових чисел, які подаються на вхід за розрядами по черзі. У прикладі вхідне слово *1***0***1***1***0***0***0***1** сформовано з двох двійкових чисел *1100* та **0101**. Відповіддю має бути «додавання за модулем 2» за розрядами – число 1001.



При перетворюванні інформації за допомогою абстрактного автомату вихідний сигнал повинен формуватися у відповідь на кожний вхідний, тобто, кожен непарний символ вхідного слова (розряд першого числа) дублюється, а кожен парний (розряд другого числа) перетворюється на відповідь з урахуванням попереднього символу (відповідного розряду першого числа).

Відповідь (вихідне слово): **11100001.**

КРИТЕРІЇ ОЦІНЮВАННЯ ВИКОНАННЯ ККР

з кредитного модуля Дискретна математика. Частина 2. Булева алгебра.

Комплексна контрольна робота (ККР) оцінюється за шкалою 100 балів. Кожне контрольне завдання ККР містить три запитання. Відповідь на п**ерше запитання оцінюється в 50 балів, на друге - 20 балів, на третє -30 балів.**

Кожне питання оцінюється відповідно до системи оцінювання наступним чином.

По першому запитанню «Побудувати таблицю істинності для заданої булевої функції f(x,y,z)» - таблиця істинності для заданої функції містить 8 значень для кожного з можливих наборів, таким чином знімається по 5 балів за кожну неправильну відповідь на одному наборі, 5 балів можна зняти за вибудуваний не в лексикографічному порядку список наборів або пропущене значення, 5 балів знімається за помилки в розстановці пріоритетів операцій для заданої функції.

Рішення **другого запитання** «Визначити, чи є формула тавтологією, за допомогою методу редукції» передбачає відповідь у формі «так» чи «ні». Рішення оцінюється в 20 балів, якщо отримана повна правильна відповідь в результаті застосування методу редукції; 15 балів – якщо відповідь вірна, метод редукції застосовано не в повній мірі або рішення отримано за допомогою іншого методу (алгебраїчно, резолюції); 10 балів - відповідь вірна, але отримана за допомогою побудування таблиці істинності, або відповідь неправильна, але було застосовано метод редукції, у ході рішення якого було допущено декілька помилок; 5 балів – значні неточності, вжито декілька кроків рішення.

Третє запитання за темою «Абстрактні автомати як дискретні перетворювачі інформації» передбачає видати у якості відповіді на вхідне слово вихідне слово у заданому алфавіті при наявності опису дії автомата у вигляді таблиць виходів та переходів. Рішення оцінюється в 30 балів, якщо отримана повна правильна відповідь, є деякі пояснення; по 5 балів знімається за кожну помилку при формуванні рішення.

Сума балів переводиться до оцінки згідно з таблицею.

Кількість балів за виконання ККР	Оцінка за чотирьохбальною системою
100……….90	“відмінно”
89……….75	“добре”
74……….60	“задовільно”
59……….0	“незадовільно”

Використання довідкової літератури, обладнання, приладів, матеріалів, комп’ютерних програм тощо, для користування при виконанні контрольної роботи не передбачено.

ДОДАТОК 3.

КОНТРОЛЬНІ ЗАПИТАННЯ І ТЕСТОВІ ЗАВДАННЯ

Контрольні запитання й тестові завдання до розділу 4:

БУЛЕВА АЛГЕБРА

1) Дослідити булеві функції:

$\neg(x \to (\neg x \equiv z))\ (y \to \neg z)$;
$\neg(x \vee y\, \neg z)\ (x \vee z)$;
$xy \to (y \equiv z)$;
$(x|y) \downarrow (y|z)$;
$(x \downarrow y) \mid (y \downarrow z)$;
$(x \vee y)(y \to \neg z) \vee x$;
$(x \to yz) \vee \neg\, xy$;
$(x \vee (y \equiv z)) \to xz$.

2) Розв’язати рівняння булевих функцій:
$[(xy \to z)x] = [x \neg y \equiv yz]$;
$[(x \oplus y) \to z] = [(x \to y)\ \neg z]$.

3) Знайти мінімальні ДНФ і КНФ булевих функцій f(x, y, z, t), що дорівнюють 1 на вказанних наборах:

	¬y		y		
¬x	1	1		1	¬z
¬x	1	1		1	z
x	1		1	1	z
x		1	1		¬z
	¬t	t		¬t	

	¬y		y		
¬x		1	1		¬z
¬x	1	1		1	z
x	1		1	1	z
x		1	1		¬z
	¬t	t		¬t	

	¬y		y		
¬x	1	1			¬z
¬x	1	1		1	z
x	1			1	z
x		1	1		¬z
	¬t	t		¬t	

	¬y		y		
¬x			1	1	¬z
¬x		1		1	z
x	1			1	z
x		1	1		¬z
	¬t	t		¬t	

Контрольні запитання й тестові завдання до розділу 4:
ТЕОРІЯ ВИСЛОВЛЮВАНЬ

1) Контрольні запитання:

- В чому відмінність між нейтральною формулою та здійсненною?
- У чому полягає монотонність достовірних міркувань? Формальне вираження закону?
- Що означає несуперечність множини посилок? Як перевірити?
- Правило підстановки та правило еквівалентної заміни: спільне та відмінності.
- Що являє собою множина інтерпретацій формули?
- З чого складається порожній диз'юнкт? Які значення він приймає,
- В чому суть методу резолюції? З чого формується множина посилок для перевірки на суперечлівість?
- Що являє собою таблиця істинності?
- Яким фразам природної мови відповідають логічні зв'язки еквівалентність, імплікація, диз'юнкція, кон'юнкція? Як за їх (імплікація, диз'юнкція, кон'юнкція) допомогою виразити формально
- Визначення простого висловлювання. Як побудувати складне висловлення?
- Назвати чотири основні принципи логіки та записати їх формальне вираження.
- В чому полягає розв'язність (вирішуваність) теорії? Чи розв'язна теорія висловлювань, за рахунок чого? Перелічити способи доведення тавтологій.

2) Перевірити формули, чи є вони тавтологіями:

$\vdash (A \to B \vee C) \to (\neg A \vee B \vee C)$;

$\vdash \neg(A \,\&\, B) \to (\neg A \vee \neg B)$;

$\vdash (\neg A \vee \neg B) \rightarrow \neg(A \ \& \ B)$;

$\vdash (B \rightarrow A \vee C) \rightarrow ((B \rightarrow C) \rightarrow ((D \rightarrow C) \rightarrow (B \vee D \rightarrow C)))$;

$\vdash (A \& B \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C)) \& (B \rightarrow (A \rightarrow C))$;

$\vdash (A \ \& \ B) \vee (A \ \& \ C) \rightarrow A \ \& \ (B \vee C)$;

$\vdash A \vee (B \vee C) \rightarrow (A \vee B) \vee C$;

$\vdash (A \vee B) \vee C \rightarrow A \vee (B \vee C)$;

$\vdash (\neg A \ \& \ \neg B) \rightarrow \neg(A \vee B)$;

$\vdash \neg (A \vee B) \rightarrow \neg A \ \& \ \neg B$;

$\vdash (A \vee B \rightarrow C) \rightarrow (B \rightarrow C) \& (A \rightarrow C)$.

3) Перевірити на несуперечливість системи посилок:

$\{\neg B \vee A \rightarrow C, \neg B \vee C, \neg D \vee C, B \vee D, \neg C \rightarrow A\}$;

$\{B \rightarrow A, B \rightarrow C, D \rightarrow C, B \vee D \rightarrow C\}$;

$\{C \rightarrow A \& B, B \& D \rightarrow A \vee C, B \& \neg A \rightarrow C \vee D, B \& \neg C \rightarrow \neg A\}$.

4) Перевірити логічне слідування:

- Профспілки штату будуть підтримувати губернатора (P) **тільки** в тому разі, якщо він підпише цей закон (Z). Фермери нададуть йому підтримку (F), **тільки** у разі накладеного на нього вето (V). Очевидно, що він або не підпише закон, або не накладе на нього вето. Отже, губернатор втратить голоси робітників, об'єднаних в профспілки, (¬P) або голоси фермерів (¬F).
- Намічена атака вдасться (A), **тільки тоді,** коли противник захоплений зненацька (Z) та його позиції погано захищені (P). Захопити противника зненацька можна **тільки,** якщо він безтурботний (B). Він не буде безтурботний, якщо його позиції погано захищені. Отже, намічена атака не вдасться.

Контрольні запитання й тестові завдання до розділу 4: ФОРМАЛЬНІ ТЕОРІЇ

Довести теореми теорії L:

$\vdash A \vee B \to B \vee A$;

$\vdash (A \to B \vee C) \to (\neg A \vee B \vee C)$;

$\vdash \neg(A \& B) \to (\neg A \vee \neg B)$;

$\vdash (\neg A \vee \neg B) \to \neg(A \& B)$;

$\vdash (B \to A \vee C) \to ((B \to C) \to ((D \to C) \to (B \vee D \to C)))$;

$\vdash (A \& B \to C) \to (A \to (B \to C)) \& (B \to (A \to C))$;

$\vdash (A \& B) \vee (A \& C) \to A \& (B \vee C)$; є розв'язок

$\vdash A \vee (B \vee C) \to (A \vee B) \vee C$;

$\vdash (A \vee B) \vee C \to A \vee (B \vee C)$;

$\vdash (\neg A \& \neg B) \to \neg(A \vee B)$;

$\vdash \neg(A \vee B) \to \neg A \& \neg B$;

$\vdash (A \vee B \to C) \to (B \to C) \& (A \to C)$;

$\vdash A \vee (B \& C) \to (A \vee B) \& (A \vee C)$;

Контрольні запитання й тестові завдання до розділу 4: АБСТРАКТНІ АВТОМАТИ

1) Побудувати автомат, який розпізнає в алфавіті {a, b, c, d}, чи є слово впорядкованим. Вихідні символи-літери дублюються. У відповідь на «пробіл» - λ видається сигнал Y або N. Стани в таблиці – великі літери {A, B, C, D, S, F}. Початковий стан S, порожнє слово, тобто просто пробіл, вважається впорядкованим.

2) Побудувати автомат, який переходить у стан *T*, якщо на його вхід було подано послідовність символів, що відповідають ідентифікатору деякої мови програмування, і переходить у стан *H* в іншому випадку.
Вхідними сигналами автомата є такі:

x_1 – літера, x_2 – цифра, x_3 – будь-який інший символ; β – пробіл.

Слова розділені пробілами. Ідентифікатор починається завжди з літери, може містить цифри, до нього не можуть входити будь-які інші символи.

3) Побудувати автомат, який переходить у стан T, якщо на його вхід було подано послідовність символів, що відповідає слову END, і переходить у стан H в іншому випадку.
Вхідними сигналами автомата є такі: E, N, D, x – пробіл, y – будь-який інший символ. Слова розділені пробілами.

4) Побудувати автомат для $X=Y=\{0,1\}$, що порівнює за величиною два додатні двійкові числа однакової розрядності. Пари розрядів порівнюваних чисел подаються на вхід автомата починаючи зі старших розрядів. Вихідним словом повинно бути більше з цих чисел. Роздільників немає.

Приклади:

Перше число **101**

Друге число 110 Вхідний потік сигналів **1**1**0**1**1**0

Вихідний потік 110110

Перше число **110**

Друге число 101 Вхідний потік сигналів **1**1**1**0**0**1

Вихідний потік 111100

5) Побудувати автомат, що за допомогою своїх станів запам'ятовує три останні символи (двійкові розряди), які було подано на його вхід. Вихідним сигналом є останній символ, що «забувається» (це - фактично питання про затримку сигналу - вихідний сигнал в момент часу t + 3 збігається з вхідним сигналом в момент часу t, X = Y = {0,1}, y (0) = y (1) = y (2) = 0).

Вхід	*0*	*1*	*1*	*0*	*1*	*0*	*1*	*0*	*1*
Вихід	***0***	***0***	***0***	*0*	*1*	*1*	*0*	*1*	*0*

Контрольні запитання й тестові завдання до розділу 4:

МЕРЕЖІ ПЕТРІ

1) За допомогою мережі Петрі описати процес видачі готівки в банкоматі. Передбачити такі дії, як введення коду та його розпізнавання або повтор вводу, доступність коштів або відмова тощо.

2) Дослідити мережу Петрі:

- побудувати таблицю перед-пост – умов, визначити конфлікти і розпаралелювання;
- побудувати дерево досяжності для початкового маркування (0,0,1);
- за допомогою матричного представлення визначити

I досяжне маркування

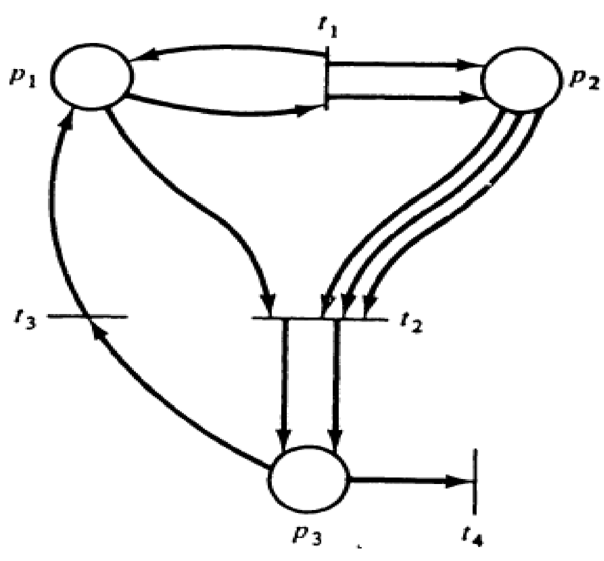
а) при початковому (0, 0, 1) і векторі запуску (3, 0, 1, 0, 0);

б) при початковому (1, 0, 0) і векторі запуску (2, 1, 0, 1);

II вектор запуску

в) при початковому (0, 0, 1) маркуванні для досяжного (0, 1, 2);

г) при початковому маркуванні (0, 1, 2) для досяжного (0, 0, 0).



Контрольні запитання й тестові завдання до ДОДАТКУ 1:
ОГЛЯД ТЕОРІЇ КОМПЛЕКТІВ

Для наступних комплектів з області D = {a, b, c} визначити потужністі комплектів та результати операцій:

A = {a, b}, B = {a, a, b, c}, C = {a, a, a, c, c}

$A \cup B$;	$A \cap B$;	$B - A$;
$A \cup C$;	$A + B$;	$C - B$;
$B \cup C$;	$A + C$;	$B - C$;
$A \cap C$;	$B + C$;	$A - C$;
$B \cap C$;	$A - B$;	$C - A$.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Бардачов Ю.М. Дискретна математика: підручник для студентів вищих технічних навчальних закладів / Ю. М. Бардачов, Н. А. Соколова, В. Є. Ходаков ; за ред. В.Є. Ходакова. - Київ : Вища школа, 2008. – 383 с.

2. Висоцька В.А. Дискретна математика: практикум (збірник задач з дискретної математики): навчальний посібник / В.А. Висоцька, В.В. Литвин, О.В. Лозинська; Міністерство освіти і науки України, Національний університет "Львівська політехніка". - Львів: "Новий світ-2000", 2020. – 575 с.

3. Журавчак Л.М. Дискретна математика для програмістів :навчальний посібник / Л.М. Журавчак, Н.І. Мельникова, П.В. Сердюк ; Міністерство освіти і науки України, Національний університет "Львівська політехніка". - Львів: Видавництво Львівської політехніки, 2019.- 417 с.

4. Нікольський Ю.В. Дискретна математика: підручник / Ю.В. Нікольський, В.В. Пасічник, Ю.М. Щербина; за науковою редакцією В.В. Пасічника ; Міністерство освіти і науки України. - Львів: Видавництво "Магнолія-2006", 2021. – 431 с.

5. Основи дискретної математики: підручник / Ю. В. Капітонова [та ін.]; НАН України, Інститут кібернетики ім. В. М. Глушкова, Міжнародний науково-навчальний центр інформаційних технологій та систем, Міністерство освіти і науки України, НТУУ "КПІ". - Київ: Наукова Думка, 2002. - 579 с.

6. Питерсон Дж. Теория сетей Петри и моделирование систем: Пер. с англ. / Дж.Петерсон – М.: Мир, 1984. – 264 с

7. Темнікова О.Л. Дискретна математика: Конспект лекцій (Частина 1) [Електронний ресурс]: навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані та математичне моделювання» / О.Л.Темнікова ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 2,97 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 154 с.

8. Темнікова О.Л. Дискретна математика: Конспект лекцій (Частина 2) [Електронний ресурс]: навч. посіб. для студ. спеціальності 113 «Прикладна математика», освітньої програми «Наука про дані та математичне моделювання» / О.Л.Темнікова ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,84 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 128 с.

9. Темнікова О.Л. Дискретна математика: практикум з дисципліни «Дискретна математика» для студентів спеціальності 113 «Прикладна математика» / О.Л.Темнікова [Електронне видання] – К. : КПІ ім. Ігоря Сікорського, 2018. – 88 с.