Міністерство освіти і науки України
Національний гірничий університет
Кафедра економічної кібернетики
та інформаційних технологій

Методичні вказівки
для самостійного вивчення
по дисципліні

«Теоретичні основи кібернетики»

напрям підготовки
«Економічна кібернетика»

Упорядники:
Професор, д.т.н, . Ігор Миколайович Пістунов
асистент Надія Володимирівна Лобова

Дніпропетровськ
2007

ВСТУП

Навчальна дисципліна «Теоретичні основи кібернетики» представляє знання з розділів математики, використовуваних при моделюванні і побудові складних систем управління.

Тут розглядаються т теми, матеріали яких найширше використовуються в теоретичних і експериментальних дослідженнях, і також в спеціальній літературі по кібернетиці.

Студентами повинні бути самостійно освоєні елементи теорії множин, поняття нечітких множин, основи математичної логіки, селективних функцій, а також мінімізацію булевих функцій.

Завдання по лабораторних роботах містять 30 варіантів. Номер вибираного варіанту повинен відповідати порядковому номеру конкретного студента в офіційному списку студентської групи.

Список літератури подано для кожного розділу окремо.

Для розуміння символіки, на початку подаємо таблиці спеціальних математичних символів.

Таблиця математичних символів

У математиці повсюдно використовуються символи для спрощення і скорочення тексту. Нижче приведений список найбільш математичних позначень, що часто зустрічаються? пояснення і приклади використання.

Окрім вказаних символів, іноді використовуються їх дзеркальні віддзеркалення, наприклад, $A \subset B$ позначає те ж, що і $B \supset A$.

Символ	Назва Вимова Розділ математики	Значення	Приклад
$\Rightarrow$	Імплікація, проходження «слідує» або «якщо., то» скрізь	$A \Rightarrow B$ означає «якщо *A* вірно, то *B* також вірно». Іноді замість нього використовують $\rightarrow$.	$x = 2 \Rightarrow x^2 = 4$ вірно, але $x^2 = 4 \Rightarrow x = 2$ невірно (оскільки *x* = - 2 також є рішенням).
$\Leftrightarrow$	Рівносильність «якщо і тільки якщо» або «рівносильне» скрізь	$A \Leftrightarrow B$ означає «*A* вірно тоді і тільки тоді, коли *B* вірно».	$x + 5 = y + 2 \Leftrightarrow x + 3 = y$
$\wedge$	Кон'юнкція «і» Математична логіка	$A \wedge B$ істинно тоді і тільки тоді, коли *A* і *B* обидва істинні.	$(n > 2) \wedge (n < 4) \Leftrightarrow (n = 3)$ якщо *n* натуральне число.
$\vee$	Диз'юнкція «або» Математична логіка	$A \vee B$ істинно, коли хоч би одна з умов *A* і *B* істинно.	$(n \leqslant 2) \vee (n \geqslant 4) \Leftrightarrow n \neq 3$ якщо *n* натуральне число.
$\neg$	Заперечення «не» Математична логіка	$\neg A$ істинно тоді і тільки тоді, коли *A* помилкове.	$\neg(A \wedge B) \Leftrightarrow (\neg A) \vee (\neg B)$

Символ	Назва / Вимова / Розділ математики	Значення	Приклад
$\forall$	Квантор загальності «Для будь-яких», «Для всіх» Математична логіка	$\forall x, P(x)$ позначає «*P(x)* вірно для всіх *x*».	$\forall n \in \mathbb{N},\ n^2 \geqslant n$
$\exists$	Квантор існування «існує» Математична логіка	$\exists x,\ P(x)$ означає «існує хоч би один *x* такий, що вірно *P(x)*»	$\exists n \in \mathbb{N},\ n + 5 = 2n$ (підходить число 5)
$=$	Рівність «рівно» скрізь	*x* = *y* позначає «*x* і *y* позначають один і той же об'єкт».	1 + 2 = 6 - 3
$:=$ $:\Leftrightarrow$	Визначення «рівно/ рівносильне за визначенням» скрізь	*x*: = *y* означає «*x* за визначенням рівний *y*». $P :\Leftrightarrow Q$ означає «*P* за визначенням рівносильне *Q*»	$\mathrm{ch}(x) := \frac{1}{2}\left(e^x + e^{-x}\right)$ (Гіперболічний косинус) $A \oplus B :\Leftrightarrow (A \vee B) \wedge \neg(A \wedge B)$ («або», що виключає)
$\{,\}$	Множина елементів «Множина.» Теорія множин	$\{a,\ b,\ c\}$ означає множину, елементами якої є *a*, *b* і *c*.	$\mathbb{N} = \{0,\ 1,\ 2,\ \ldots\}$ (множина натуральних чисел)

Символ	Назва / Вимова / Розділ математики	Значення	Приклад
{ | } {:}	Множина елементів, що задовольняють умові «Множина всіх. таких, що вірно.» Теорія множин	$\{x \mid P(x)\}$означає множину всіх *x* таких, що вірно *P(x)*.	$\{n \in \mathbb{N} \mid n^2 < 20\} = \{0,\ 1,\ 2,\ 3,\ 4\}$
∅ {}	Порожня множина «Порожня множина» Теорія множин	{ } означають множину, що не містить жодного елементу.	$\{n \in \mathbb{N} \mid 1 < n^2 < 4\} = \varnothing$
∈	Приналежність /не приналежність до множини «належить», «з» «не належить» Теорія множин	$a \in S$ означає «*a* є елементом множини *S*» $a \notin S$ означає «*a не* є елементом *S*»	$2 \in \mathbb{N}$
⊆	Підмножина «є підмножиною», «включено в» Теорія множин	$A \subseteq B$ означає «кожен елемент з *A* також є елементом з *B*». $A \subset B$ зазвичай означає те ж, що і $A \subseteq B$. Проте деякі автори використовують, щоб показати строге включення (тобто $\subsetneq$).	$(A \cap B) \subseteq A$

Символ	Назва Вимова Розділ математики	Значення	Приклад
$\subsetneq$	Власна підмножина «є власною підмножиною», «строго включається в» Теорія множин	$A \subsetneq B$ означає і $A \neq B$.	$\mathbb{N} \subsetneq \mathbb{Q}$
$\cup$	Об'єднання «Об'єднання . і .», «., об'єднане з .» Теорія множин	$A \cup B$означає множину елементів, належних *A* або *B* (або обом відразу).	$A \subseteq B \Leftrightarrow A \cup B = B$
$\cap$	Перетин «Перетин . і . », «., пересічене з .» Теорія множин	$A \cap B$ означає множину елементів, що належать і *A*, і *B*.	$\{x \in \mathbb{R} \mid x^2 = 1\} \cap \mathbb{N} = \{1\}$
$\setminus$	Різниця множин «різниця . і . », «мінус» «. без .» Теорія множин	$A \setminus B$ означає множину елементів, належних *A*, але не належних *B*.	$\{1, 2, 3, 4\} \setminus \{3, 4, 5, 6\} = \{1, 2\}$
$\to$	Функція «з . у»	$f\colon X \to Y$означає функцію *f* з областю визначення *X* і областю прибуття *Y*.	Функція, визначена як $f(x) = x^2$

Символ	Назва Вимова Розділ математики	Значення	Приклад
	скрізь		
$\mapsto$	Відображення «відображається в» скрізь	$x \mapsto f(x)$ означає, що образом *x* після застосування функції *f* буде *f(x)*.	Функцію, визначену як $f(x) = x^2$, можна записати так: $f : x \mapsto x^2$
$\mathbb{N}$	Натуральні числа «Ен» Числа	$\mathbb{N}$ означає множину $\{1,\ 2,\ 3,\ \ldots\}$ або $\{0,\ 1,\ 2,\ 3,\ \ldots\}$ (залежно від ситуа-ції).	$\{\lvert a\rvert \mid a \in \mathbb{Z}\} = \mathbb{N}$
$\mathbb{Z}$	Цілі числа «Зед» Числа	$\mathbb{Z}$ означає множину $\{\ldots,\ -3,\ -2,\ -1,\ 0,\ 1,\ 2,\ 3,\ \ldots\}$	$\{a,\ -a \mid a \in \mathbb{N}\} = \mathbb{Z}$
$\mathbb{Q}$	Раціональні числа «Ку» Числа	$\mathbb{Q}$ означає $\left\{\frac{p}{q}\ p \in \mathbb{Z} \wedge q \in \mathbb{Z} \wedge q \neq 0\right\}$	$3{,}14 \in \mathbb{Q}$
$\mathbb{R}$	Дійсні числа, або дійсні числа «Ер» Числа	$\mathbb{R}$означає множину всіх меж послідов-ностей з $\mathbb{Q}$	$\pi \in \mathbb{R}$ $i \notin \mathbb{R}$ (*i* комплексне число: $i^2 = -1$)

Символ	Назва	Вимова	Розділ математики	Значення	Приклад
$\mathbb{C}$	Комплексні числа	«Це»	Числа	$\mathbb{C}$ означає множину $\{a+b\cdot i \mid a\in\mathbb{R}\wedge b\in\mathbb{R}\}$	$i\in\mathbb{C}$
$<$	Порівняння	«менше ніж», «більш ніж»	Відношення порядку	$x < y$ означає, що x строго менше y. $x > y$ означає, що x строго більше y.	$x<y \Leftrightarrow y>x$
$\leqslant$	Порівняння	«менше або рівно»; «більше або рівно»	Відношення порядку	$x \leqslant y$ означає, що x менше або рівний y. $x \geqslant y$ означає, що x більше або рівний y.	$x\geqslant 1 \Rightarrow x^2 \geqslant x$
$\approx$	Приблизна рівність	«приблизно рівний»	Числа	$e\approx 2{,}718$ з точністю до 10^{-3} означає, що 2,718 відрізняється від e не більш ніж на 10^{-3}.	$\pi\approx 3{,}1415926$ з точністю до 10^{-7}.
$\surd$	Арифметичний квадратний корінь	«Корінь квадратний з .»	Числа	$\sqrt{x}$ означає позитивне дійсне число, яке в квадраті дає x.	$\sqrt{4}=2$

Символ	Назва Вимова Розділ математики	Значення	Приклад
∞	Нескінченність «Плюс/мінус нескінченність» Числа	$+\infty$ і суть елементи розширеної множини дійсних чисел. Ці символи позначають числа, меньше/більше всіх дійсних чисел.	$\lim\limits_{x\to 0} \frac{1}{\|x\|} = \infty$
| |	Модуль числа (абсолютне значення), модуль комплексного числа або потужність множини «Модуль»; «Потужність» Числа і Теорія множин	$\|x\|$ позначає абсолютну величину *x*. | *A* | позначає потужність множини *A* і дорівнює, якщо *A* конечно, числу елементів *A*.	$\|a + b \cdot i\| = \sqrt{a^2 + b^2}$
$\sum$	Сума, сума ряду «Сума . по . від . до .» Арифметика, Математичний аналіз	$\sum\limits_{k=1}^{n} a_k$ означає «сума a_k, де *k* приймає значення від 1 до *n*», тобто $a_1 + a_2 + \ldots + a_n$. $\sum\limits_{k=1}^{\infty} a_k$ означає суму ряду, що складається з a_k.	$\sum\limits_{k=1}^{4} k^2 =$ 12 + 22 + 32 + 42= 30
$\prod$	Добуток «Добуток . по . від . до .»	$\prod\limits_{k=1}^{n} a_k$ означає «добуток a_k для всіх *до*	$\prod\limits_{k=1}^{4} (k+2) =$

Символ	Назва Вимова Розділ математики	Значення	Приклад
	Арифметика	від 1 до *n*», тобто $a_1 \cdot a_2 \cdot \ldots \cdot a_n$	$= 3 \cdot 4 \cdot 5 \cdot 6 = 360$
$\int dx$	Інтеграл «Інтеграл (від . до .) функції . по (або d).» Математичний аналіз	$\int_a^b f(x)\,dx$ означає «інтеграл від *a* до *b* функції *f* від *x* по змінній *x*».	$\int_0^b x^2\,dx = b^3/3$
$\frac{df}{dx}$ *f'(x)*	Похідна «Похідна . по .» Математичний аналіз	$\frac{df}{dx}$ або *f'(x)* означає «(перша) похідна функції *f* від *x* по змінній *x*».	$\frac{d\cos x}{dx} = -\sin x$
$\frac{d^n f}{dx^n}$ *f(n)*	Похідна *n-го* порядку «*n-я* похідна . по .» Математичний аналіз	$\frac{d^n f}{dx^n}$ або *f(n)(x)* (у другому випадку якщо n фіксоване число, то воно пишеться римськими цифрами) означає «*n-я* похідна функції *f* від *x* по змінній *x*».	$\frac{d^4 \cos x}{dx^4} = \cos x$

Таблиця позначень абстрактної алгебри

У абстрактній алгебрі повсюдно використовуються символи для спрощення і скорочення тексту, а також стандартні позначення для деяких груп. Нижче приведений список найбільш позначень алгебри, що часто зустрічаються, пояснення і приклади використання.

Окрім вказаних символів, іноді використовуються їх дзеркальні віддзеркалення, наприклад, позначає те ж, що і $G \triangleright H$.

Символ)	**Назва** / **Вимова**	**Значення**
Символи абстрактної алгебри		
$\triangleleft$	Нормальна підгрупа, ідеал кільця «нормальна в» «. є ідеалом .»	$H \triangleleft G$ означає «*H* є нормальною підгрупою групи G», якщо G – група, і «H є (двосторонні́м) ідеалом кільця G», якщо G – кільце.
$[:]$	Індекс підгрупи, розмірність поля «індекс . у .», «розмірність . над .»	$[G : H]$означає «індекс підгрупи *H* в групі G», якщо G – група, і «розмірність поля H над полем G», якщо G і H – поля.
$\times$	Прямий добуток груп «прямий добуток . і .»	$G \times H$ означає «прямий добуток груп *G* і H».
$\oplus$	Пряма сума підпросторів «пряма сума . і .»	$V = V_1 \oplus V_2$означає «простір *V* розкладається в пряму суму підпросторів V_1 і V_2».
$\otimes$	Тензорний твір «тензорний твір . і .»	$T_1 \otimes T_2$означає «тензорний твір тензорів T_1 і T_2».
$[,]$	Комутатор елементів групи «комутатор . і .»	$[g, h]$означає «комутатор елементів *g* і h групи G», тобто елемент $ghg^{-1}h^{-1}$.
G'	Комутант «комутант»	G' означає «коммутантні групи *G*».
$<>_n$	Циклічна група «Циклічна група порядку *n*, породжена a»	$<a>_n$ означає «циклічна група порядку *n*, породжена елементом a».

$\perp$	Ортогональний підпростір «ортогональний підпростір до .»	$V^{\perp}$ означає «ортогональний підпростір до підпростору *V*».
A^T	Транспонована матриця «транспонована матриця .»	A^Tозначає «транспонована матриця *A*».
$E_{i,j}$	Матрична одиниця «матрична одиниця .»	$E_{i,j}$ означає «матрична $i,\ j$-одиниця», тобто матриця, у якої на місці $(i,\ j)$ стоїть одиниця, а на решті місць – нулі.
$*$	Зв'язаний оператор Зв'язаний простір Мультиплікативна група поля «оператор, зв'язаний до .»; «простір, зв'язаний до .»; «мультиплікативна група .»	$\mathcal{A}^*$ означає «лінійного оператора, зв'язаного до $\mathcal{A}$», якщо – лінійний оператор. V^* означає «лінійний простір, зв'язаний до *V* (дуальне до *V*)», якщо *V* – лінійний простір. F^* означає «мультиплікативна група поля *F*», якщо *F* – поле.

Стандартні позначення деяких груп

S_n	Симетрична група *n*-*ой* ступеня «эс .»	S_nозначає «симетрична група (або група перестановок) ступеня *n*».
A_n	Знакозмінна група *n*-*ого* ступеня «а .»	A_nозначає «знакозмінна група (тобто група парних підстановок) ступеня *n*».
$GL_n(F)$	Група не вироджених лінійних операторів «же ель . над .»	$GL_n(F)$ означає «група не вироджених лінійних операторів розмірності *n* над полем *F*» (від *general linear*).
$SL_n(F)$	Група лінійних операторів з визначником 1 «эс ель . над .»	$SL_n(F)$означає «група лінійних операторів розмірності *n* над полем *F* з визначником 1» (від *special linear*).
$UT_n(F)$	Група верхніх трикутних матриць «група верхніх трикутних матриць порядку	$UT_n(F)$означає «група верхніх трикутних матриць порядку *n* над полем F» (від *upper triangular*).

	. над .»	
$SUT_n(F)$	Група верхніх унітрикутних матриць «група верхніх унітрикугних матриць порядку . над .»	$SUT_n(F)$ означає «група верхніх унітрикутних матриць порядку *n* над полем *F*» (від *special upper triangular*), тобто верхніх трикутних матриць з одиницями на головній діагоналі.
$\mathbb{Z}_p$	Кільце вирахувань по модулю «зед .»	$\mathbb{Z}_p$ означає «кільце вирахувань по модулю *p*» (якщо p — просте, то це поле).
$\mathbb{Q}_p$	p-адичні числа «ку .»	$\mathbb{Q}_p$ означає «поле *p-адичних* чисел».
D_n	Група диедра *n*-ого ступеня «дэ .»	D_n означає «група диедра *n*-ого ступеня» (тобто група симетрій правильного *n*-кутника).
V_4	Четверна група Клейна «вэ чотири»	V_4означає «четверна група Клейна» (тобто група симетрій правильного тетраедра).

1. ТЕОРІЯ МНОЖИН

Теорія множин – розділ математики, в якому вивчаються загальні властивості множин. Теорія множин лежить в основі більшості математичних дисциплін; вона зробила глибокий вплив на розуміння предмету самої математики.

До другої половини XIX століття поняття «множини» не розглядалося як математичне («множина книг на полиці», «множина людських чеснот» і т. д. – все це чисто побутові обороти мови). Положення змінилося, коли німецький математик Георг Кантор розробив свою програму стандартизації математики, в рамках якої будь-який математичний об'єкт повинен був виявлятися тим або іншим «множиною». Наприклад, натуральне число, по Канторові, слід було розглядати як множину, що складається з єдиного елементу іншої множини, званої «натуральним рядом», – який, у свою чергу, сам є множиною, що задовольняє так званим аксіомам Пеано. При цьому загальному поняттю «множини», що розглядалося їм як центральна для математики, Кантор давав мало що визначальні визначення ніби «множина є багато такого, мислиме як єдине», і т.д. Це цілком відповідало умонастрою самого Кантора, що підкреслено називав свою програму не «теорією множин» (цей термін з'явився багато пізніше), а *вченням* про множини (*Mengenlehre*).

Програма Кантора викликала різкі протести з боку багатьох сучасних йому крупних математиків. Особливо виділявся своїм непримиренним до неї відношенням Леопольд Кронекер, що вважав, що математичними об'єктами можуть вважатися лише натуральні числа і те, що до них безпосередньо зводиться (відома його фраза про те, що «бог створив натуральні числа, а все інше – справа рук людських»). Проте, деякі інші математики – зокрема, Готлоб Фреге і Давид Гільберт – підтримали Кантора в його намірі перекласти всю математику теоретико-множинною мовою.

Проте незабаром з'ясувалося, що установка Кантора на необмежене свавілля при операції з множинами (виражений їм самим в принципі «суть математики полягає в її свободі») є спочатку порочною. А саме, були виявлені ряд теоретико-множинних антиномій: виявилось, що при використанні теоретико-множинних уявлень деякі твердження можуть бути доведені разом зі своїми запереченнями (а тоді, згідно правилам класичної логіки висловів, може бути «доведено» абсолютно будь-яке твердження!). Антиномії ознаменували собою повний провал програми Кантора.

На початку XX століття Бертран Рассел, вивчаючи теорію множин, прийшов до парадоксу (з тих пір відомому як парадокс Рассела). Він продемонстрував неспроможність наївної теорії множин і пов'язаною з нею канторовской програми стандартизації математики.

Після виявлення антиномії Рассела частина математиків (наприклад, Л. Е. Я. Брауер і його школа) вирішила повністю відмовитися від використання теоретико-множинних уявлень. Інша ж частина математиків, очолена Д. Гільбертом, зробила ряд спроб обгрунтувати ту частину теоретико-множинних уявлень, яка здавалася їм найменш відповідальною за виникнення антиномій, на основі явної надійної фінітної математики. З цією метою були розроблені різні аксіоматизації теорії множин.

В рамках аксіоматичних теорій множини «існують» виключно формальним чином, і їх «властивості» можуть істотно залежати від вибору аксіоматики.

Сучасна теорія множин будується на системі аксіом тверджень, що приймаються без доказу, з яких виводяться всі теореми і затвердження теорії множин.

Система аксіом Цермела Френкеля (ZF) є стандартною системою аксіом для теорії множин. Ця і подібні нею системи аксіом цікаві тому, що будь-яка математична теорія може бути «перекладена» мовою теорії множин таким чином, що теореми цієї теорії стануть теоремами про множини, доказовими з аксіом ZF.

До цієї системи аксіом часто додають аксіому вибору, і називають **системою Цермела- Френкеля з аксіомою вибору** (ZFC).

Ця система аксіом записана на мові логіки першого порядку, і містить нескінченну кількість аксіом. Існують та інші, кінцеві системи. Наприклад, система NBG (von Neumann Bernays Godel) разом з множинами розглядає так звані класи об'єктів. NBG рівносильна ZF в тому сенсі, що будь-яка теорема про множини (тобто що не згадує про класи), доказова в одній системі, також доказова і в іншій.

Ці аксіоми були розроблені Торальфом Сколемом (Thoralf Skolem) в 1922 році, і є розвитком системи аксіом Адольфа Френкеля (Adolf Fraenkel), яка, у свою чергу, була розвитком системи аксіом Эрнста Цермело (Ernst Zermelo).

Подама декілька понять, які дозволять краще засвоїти матеріал.

Конті нуум (від латів. *continuum* — безперервне) має декілька значень в математиці і філософії.

У теорії множин континуум може позначати два схожі поняття:

- кардинал або клас множин, рівнопотужних множинаі дійсних чисел.
- множина рівнопотужна множинаі дійсних чисел. Наприклад, сукупність всіх точок відрізання на прямій або множина всіх ірраціональних чисел. Говорять: «континуум», «множина потужності континуум» або «континуальное множина».

Підмножина — поняття частини в теорії множин. Множина *B* є підмножиною множини А (позначається $B \subseteq A$) у випадку, якщо кожен елемент множини *B* є також і елементом множини *A*. Наприклад, множина всіх парних чисел є підмножиною множини всіх цілих чисел.

Якщо *A* є підмножиною *B*, то В називається надмножиною *A*.

Порожня множина, поняття теорії множин; порожня множина не містить жодного елементу; позначається _ (∅). Порожня множина є підмножиною будь-якої множини. Потужність порожньої множини рівна нулю.

Поняття порожньої множини (подібно до поняття «нуль») виникає з потреби, щоб результат всякої операції над множинами був також множиною.

Рахункова множина – множина, елементи якої можливо занумерувати натуральними числами. Формальніше: множина *X* є рахунковою, якщо існує $X \to \mathbb{N}$ бієція $X \to \mathbb{N}$, де $\mathbb{N}$ позначає множину всіх натуральних чисел. Іншими словами, рахункова множина — це множина, равнопотужна множині натуральних чисел.

Рахункова множина є «найменшою» нескінченною множиною, т. е. у будь-якій нескінченній множині знайдеться рахункова підмножина.

Властивості

1. Будь-яка підмножина рахункової множини кінцева або рахункова;
2. Об'єднання кінцевого або рахункового числа рахункової множини є рахунковим;
3. Прямий добуток кінцевого числа рахункової множини є рахунковим;
4. Множина всіх кінцевих підмножин рахункової иножигни є рахунковою;
5. Множина всіх підмножин рахункової множини континуальна і, зокрема, не є рахунковою.

Незліченна множина — така, що не є рахунковою. Таким чином, будь-яка множина є або кінцевою, або рахунковою, або незліченною.

Множина раціональних чисел і множина алгебраічних чисел рахункові, проте множина натуральних чисел континуальна, а отже, незліченна.

1.1. Операції над множинами

Над множинами, як і над багатьма іншими математичними об'єктами, можна здійснювати різні операції. В результаті операцій з початкових множин виходять нові. Для кращого розуміння сенсу цих операцій використовуються діаграми Ейлера – Венна, на яких представлені результати операцій над геометричними фігурами як множиною точок.

Порівняння множин

Множина *A* **міститься** в множині В (множина В включає множину А), якщо кожен елемент А є елемент В:

$$A \subset B :\Leftrightarrow x \in A \Rightarrow x \in B.$$

В цьому випадку *A* називається **підмножиною** *B*, *B* — **надмножиною** А. Якщо $A \subset B$ і $A \neq B$, то *A* називається **власною підмножиною** *B*. Відмітимо, що $\forall M \quad M \subset M$. За визначенням $\forall M \quad \varnothing \subset M$.

Дві множини називаються **рівними**, якщо вони є підмножинами один одного: $A = B :\Leftrightarrow A \subset B \wedge B \subset A$

Основні операції над множинами:

- **Об'єднання множин —**, побудова множини, що містить в собі всі елементи декількох початкових множин.

Застарілі терміни для цього поняття: **сума** і **з'єднання множин**.
Об'єднання двох множин A і B позначають

$$A \cup B := \{x | x \in A \vee x \in B\}$$

(також застаріле $A + B$)
Наприклад якщо {1,2,3}U{2,3,4}={1,2,3,4}.

Об'єднання сукупності позначають $\bigcup_{x \in X} M_x$

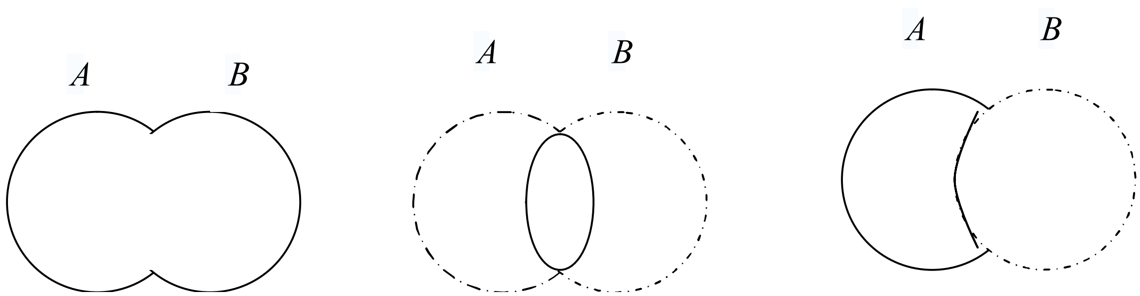
Строго кажучи, об'єднанням двох множин називається множина, що складається зі всіх елементів, які належать хоч би одному з них, тобто

$$\alpha \in A \cup B \iff \alpha \in A \vee \alpha \in B$$

Об'єднанням сукупності множин називається множина, що складається зі всіх елементів, які належать хоч би одному з множин даної сукупності, тобто

$$\alpha \in \bigcup_{x \in X} M_x \iff \exists x \in X : \alpha \in X$$

- **Перетин:**
$A \cap B := \{x | x \in A \wedge x \in B\}$ Перетин(множення) $A \cap B$ є множина всіх елементів, що належать одночасно як A, так і B. Наприклад якщо {1,2,3}$\cap${2,3,4}={2,3}. Множини, що не мають загальних елементів, називаються такими, що не перетинаються.
- **Різниця:**
$A \setminus B := \{x | x \in A \wedge x \notin B\}$ Різниця $A \backslash B$ (або A - B) є множина, що складаються зі всіх елементів A, що не входять у B. Наприклад якщо {1,2,3}2,3,4}={1}.



1 2 3

Рис. 1.1. Приклад понять (позначено суцільними лініями): об’єднання (1), перетин (2) та різниця (3) множин A та B

- **Симетрична різниця:**
$A \triangle B \equiv A \dot{-} B := (A \cup B) \setminus (A \cap B) = \{x | (x \in A \wedge x \notin B) \vee (x \notin A \wedge x \in B)\}$
Диз'юнктивна сума (симетрична різниця) $A+B$ є множина всіх елементів, що належать або A, або B, але не обом разом. Наприклад, якщо {1,2,3}+{2,3,4}={1,4}.
- доповнення: $\overline{A} := \{x | x \notin A\}$

Операція доповнення має на увазі деякий універсум (множина U, яка містить A): $\overline{A} = U \setminus A$

- Декартове або пряме множення: $A \times B = \{(a,b) | a \in A, b \in B\}$

1.2. Аксіоми ZFC

1. *Аксіома об'ємності*. Дві множини a і b рівні тоді і тільки тоді, коли вони мають одні і ті ж елементи.

$$\forall a \forall b (a = b \leftrightarrow \forall c (c \in a \leftrightarrow c \in b))$$

2. *Аксіома порожньої множини*. Існує множина e без єдиного елементу. Ця множина зазвичай позначається {} або $\emptyset$.

$$\exists e \forall a (a \notin e)$$

3. *Аксіома пари*. Для будь-яких множин a *і* b існує множина c така, що a і b є його єдиними елементами. Множина c позначається $\{a,b\}$ і називається невпорядкованою парою a і b. Якщо $a = b$, то c складається з одного елементу.

$$\forall a \forall b \exists c \forall d (d \in c \leftrightarrow (d = a \lor d = b))$$

4. *Аксіома об'єднання*. Для будь-якого сімейства a множин існує множина, звана об'єднанням множини $b = \cup a$, що складається з тих і лише тих елементів, які містяться в елементах множини a.

$$\forall a \exists b \forall c (c \in b \leftrightarrow \exists d (d \in a \land c \in d))$$

5. *Аксіома нескінченності*. Аксіоми з 1 по 4 надають обмежені можливості для формування нових множин. Так, по теоремі Кантора в множині є елемент, що не належить a, тому, наприклад, не існує «множини всіх множин» (парадокс Рассела). Далі введемо визначення: множина називається індуктивною, якщо воно а) містить порожню множину і б) містить послідовник (тобто елемент)$a \cup \{a\} a \cup \{a\}$ кожного свого елементу. Аксіома нескінченності стверджує, що індуктивні множини існують.

$$\exists \omega (\emptyset \in \omega \land \forall x (x \in \omega \to x \cup \{x\} \in \omega))$$

6. *Схема виділення*. Будь-якій множині a і властивості φ відповідає множина b, елементами якого є ті і лише ті елементи a, які володіють властивістю φ. Схема виділення містить рахункову кількість аксіом, оскільки кожна формула логіки першого порядку породжує аксіому.

$$\forall a \exists b \forall c (c \in b \leftrightarrow (c \in a \land \varphi(c)))$$

7. *Аксіома множини підмножин*. Для будь-якої множини a існує множина b, що складається з тих і лише тих елементів, які є підмножинами множини a. Множина підмножин множини a позначається $\mathcal{P}(a)$.

$$\forall a \exists b \forall c (c \in b \leftrightarrow \forall d (d \in c \to d \in a))$$

Якщо ввести відношення підмножини, то цю формулу можна спростити.

$$\forall a \exists b \forall c (c \in b \leftrightarrow c \subseteq a)$$

8. *Схема підстановки*. Хай $\varphi(x,y)$ – така формула, що при будь-якому x_0 з множини X існує, і притому єдиний, об'єкт y_0 такий, що вираз істинний. Тоді об'єкти c, для кожного з яких існує d з X такий, що істинно, утворюють множи-

ну. Схема підстановки містить рахункову кількість аксіом, оскільки кожна відповідна формула породжує аксіому.

$$\forall x \exists !\ y(\varphi(x,y)) \to \forall a \exists b \forall c(c \in b \leftrightarrow (\exists d(d \in a \wedge \varphi(d,c)))$$

9. *Аксіома підстановки*. Кожна непорожня множина *s* містить елемент *a* такий, що $s \cap a = \emptyset$.

$$\forall s(s \neq \emptyset \to \exists a(a \in s \wedge a \cap s = \emptyset))$$

10. *Аксіома вибору*. Для будь-якого сімейства попарно неперетинних непорожніх множин існує множина *c* така, що, яка б не була множина *x* даного сімейства, множина складається з одного елементу.

Несуперечність приведеної аксіоматики досі не встановлена.

1.3. Потужність множини

Потужність множини – це узагальнення поняття кількості (числа елементів множини), яка має сенс для всіх множин, включаючи нескінченні.

Існують великі, є менші нескінченні множини, серед них рахункова множина є найменшою.

Визначення

Дві множини називаються **рівнопотужними**, якщо між ними існує бієкція. Існування бієкції між множинами є відношення еквівалентності, а **потужність** множини — це відповідний йому клас еквівалентності.

Приклад

Множина парних цілих чисел має таку ж потужність, що і множина цілих чисел $\mathbb{Z}$. Визначимо так: $f(x) = \frac{x}{2}$. *f* — бієкція, тому $|\mathbb{E}| = |\mathbb{Z}|$

Властивості

- Дві **кінцеві множини** множина рівно потужні тоді і тільки тоді, коли вони складаються з однакового числа елементів. Тобто для кінцевої множини поняття потужності співпадає зі звичним поняттям кількості.
- Для нескінченних множин потужність множини може співпадати з потужністю його власної підмножини, наприклад $|\mathbb{N}| = |\mathbb{Z}|$.
- Теорема Кантора гарантує існування потужнішої множини для будь-якого даного: *Множина всіх підмножин множини A потужніша A, або* $|2A| > |A|$.
- За допомогою канторова квадрата можна також довести наступне корисне твердження: *Декартове множення нескінченної множини* A *з самим собою рівнопотужне* A.

Слідуючи Канторові, потужність множини називається **кардинальним числом** і позначається потужність такої множини *A* через $|A|$ (сам Кантор використовував позначення $\overline{\overline{A}}$). Іноді зустрічається позначення $\#A$.

Потужність множини натуральних чисел позначається символом $\aleph_0$ («алеф-нуль»). Множина називається **нескінченною**, якщо її потужність $\geq \aleph_0$, таким чином, рахункові множини — це «найменші» з нескінченних множин. Наступні кардинальні числа в порядку зростання позначаються $\aleph_1, \aleph_2, \ldots$.

Про множини, рівнопотужні множині всіх дійсних чисел, говорять, що вони мають потужність континууму, і потужність таких множин позначається символом *c*. Континуум-гіпотеза стверджує, що $c = \aleph_1$.

Для потужностей, як і у разі кінцевих множин, є поняття: рівність, більше, менше. Тобто для будь-якої бмножини *A* і *B* можливе тільки одне з трьох:

1. $|A| = |B|$ або *A* і *B* рівнопотужні;
2. $|A| > |B|$ або *A* **потужніше** *B*, тоюто. *A* містить підмножину, рівнопотужну *B*, але *A* і *B* не рівнопотужні;
3. $|A| < |B|$ або *B* потуєніше *A*, в цьому випадку *B* містить підмножину, рівнопотужну *A*, але *A* і *B* не рівнопотужні.

Ситуація, в якій *A* і *B* не рівнопотужні і ні в одному з них немає частини, рівнопотужної іншому, неможлива. Інакше це означало б існування незрівняних між собою потужностей (що в принципі можливо, якщо не приймати аксіому вибору).

1.4. Булеан

Хай *A* – множина. Множина всіх підмножин множини *A* називається **булеаном** (або **ступенем множини** *A*, або множиною **частин** *A*) і позначається $\mathcal{P}(A)$ або 2^A. Ясно, що $\varnothing \in \mathcal{P}(A)$ и $A \in \mathcal{P}(A)$.

Справедливе наступне твердження: Число підмножин кінцевої множини, що складається з *n* елементів рівне 2^n.

Доказ проведемо методом математичної індукції.

База. Якщо $n = 0$, тобто множина порожня, то у нього тільки одна підмножина – воно само, і число, що цікавить нас, рівне $2^0 = 1$.

Індукційний крок. Хай твердження справедливе для деякого *n* і хай *M* – множина з кардинальным числом $n + 1$. Зафіксувавши деякий елемент $a_0 \in M$, розділимо підмножини множини *M* на два типи:

1. що містять a_0
2. що не містять a_0, тобто множини, що є підмножинами $M - \{a_0\}$.

Підмножин типу 2 по припущенню індукції 2^n. Але підмножин типу 1 рівно стільки ж, оскільки підмножина типу 1 виходить з деякої і притому єдиної підмножини типу 2 додаванням елементу a_0 і, отже, з кожної підмножини типу 2 виходить цим способом одне і лише одна підмножина типу 1.

Тому число всіх підмножин множини *M* дорівнює $2^n + 2^n = 2^{n+1}$.

1.5. Бієкція

Функцвя $f : X \to Y$ називається **бієкцією** (і позначається $f : X \leftrightarrow Y$), якщо вона:

1. Переводить різні елементи множини *X* в різні елементи множини *Y* (ін'єктивність).
2. Будь-який елемент з має свій прообраз (сюр'єктивність). Іншими словами $\forall y \in Y \exists x \in X : \ f(x) = y$.

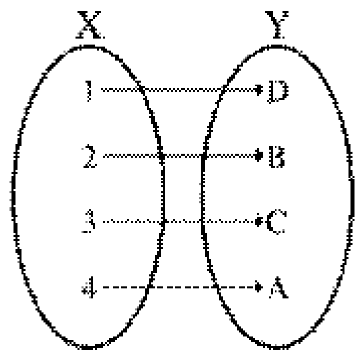


Рис. 1.2.
Бієктивна функція

Бієкцію також називають **взаємно однозначним відображенням**. Множини, для яких існує бієкція, називаються **рівнопотужними**.

Приклади

- $id : X \to X$ – функція, що зберігає всі елементи множини *X*, биективна на цій множині.
- $f(x) = x, \ f(x) = x^3$ – бієктивні функції з $\mathbb{R}$ в себе. Взагалі, будь-який моном однієї степені змінної степені є бієкцією.
- $f(x) = e^x$ – бієктивна функція в $\mathbb{R}_+ = (0, \infty)$. Але якщо її розглядати як функцію в $\mathbb{R}$, то вона вже не буде бієктивною (у негативних чисел не буде прообразів).
- $f(x) = \sin x$ не є бієктивною функцією, якщо вважати її визначеною на всьому $\mathbb{R}$.

Властивості

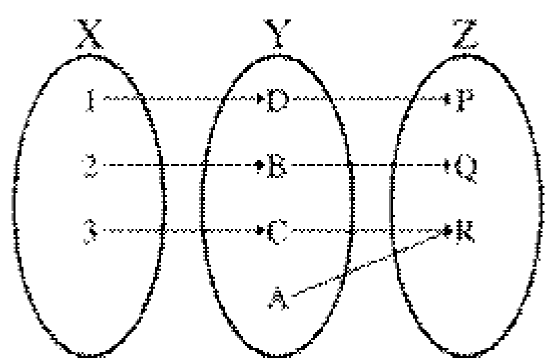


Рис. 1.3. Композиція ін'єкції і сюр'єкції, що дає бієкцію.

- Функція є бієктивною тоді і тільки тоді, коли існує $f^{-1} : Y \to X$ зворотня функція $f^{-1} : Y \to X$, така, що і $\forall y \in Y \ f(f^{-1}(y)) = y$.
- Якщо функції *f* і *g* бієктивні, то і композиція функцій бієктивна, в цьому випадку $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$. Коротко: композиція бієкцій є бієкцією. Зворотне, взагалі кажучи, невірно: якщо бієктивна, то ми можемо затверджувати лише, що *f* ін'єктивна, а *g* сюр'єктивна.

1.6. Індивідуальне завдання № 1

Кожен студент обирає собі по п’ять задач згідно номера за списком групи. Номера цих задач подано у настукпних таблицях, а самі задачі вміщено нижче.

Варіанти завдань

Варіант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Задача1	1	2	1	4	2	3	1	9	2	3	4	1	2	3	4
Задача2	9	12	3	14	5	6	7	10	12	5	6	5	6	7	9
Задача3	10	15	13	17	15	16	8	11	16	13	14	15	16	8	10
Задача4	11	18	16	21	18	19	17	21	18	19	20	21	19	17	11
Задача5	14	20	24	22	23	25	20	22	23	24	25	22	23	25	18

Варіант	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Задача1	2	3	4	5	6	1	2	3	4	5	6	7	9	12	13
Задача2	5	6	7	9	12	7	9	10	13	14	7	8	10	18	15
Задача3	9	13	8	10	14	8	10	12	16	17	8	16	11	21	19
Задача4	12	20	21	11	17	13	11	15	20	21	15	19	17	22	23
Задача5	19	24	22	23	25	24	14	18	24	22	18	23	20	25	24

ЗАВДАННЯ

1. Які з приведених нижче співвідношень невірні і чому?

а) x ∈ {2, a, x};
б) 3 ∈ {1 {2, 3}, 4};
в) x; ∈ {l, sin(x)};
г) {x, y} ∈ { a, { x,y},b}.

2. Чи рівні між собою множини *A і B* (якщо немає, то чому)?

а) A = {2, 5, 4}, B = {5, 4, 2};
б) A = {1, 2, 4, 2}, B = {I, 2, 4};
в) A = {2, 4, 5}, У = {2, 4, 3};
г) A = {1 {2, 5}, 6}, У = {1 (5, 2}, 6);
д) A **=** {1 {2, 5}, 6}, У == {1, 2, 5, 6}.

3. Чи зв'язані множини *A і У* відношенням включення (якщо так, то вкажіть, яке з них є підмножиною іншого)?

а) .A = { a, b, d}, У = {a, b, c,d};
б) A = { a, з, d, e}, У = { a, e, з};
в) A = { з, d, e}, У = { з, a}.

4. У яких відносинах знаходяться між собою наступні три множини:
A = {1, 3}; У – множина непарних позитивних чисел; З – множина вирішень рівняння x2 - 4x + 3 = 0?

5. Утворіть множину святкових днів 2006 р. Чи перетинається ця множина з множиною недільних днів того ж року? Якщо так, то запишіть елементи перетину цих двох множин.

6. До яких видів відносяться наступні множини:
A – множина конденсаторів в радіоприймачі; B – множина квадратів цілих чисел; З – множина вирішень рівняння 2x - 3 = 0**;** D – множина дерев на Місяці?

7. Прийнявши множину перших 20 натуральних чисел як універсум, запишіть наступні його підмножини: A – парних чисел; B – непарних чисел; C – квадратів чисел; D – простих чисел. У яких відносинах знаходяться ці підмножини?

8. Запишіть множини, що отримуються в результаті наступних операцій над множинами із завдання 7: $A \cup B$, $A \cap B$, $A \cap C$, A D, $C \backslash A$, $C \backslash B$, $C + D$. Сформулюйте визначальні властивості кожної з отриманих множин.

9. Три прилади *x, y,* ***z*** порівнюють за двома показниками, причому виділяють той з приладів, у якого даний показник якнайкращий (випадки однакових показників виключаються).

а) Утворіть множину ***U*** всіляких результатів такого порівняння, позначивши елементи цієї множини впорядкованими парами букв для приладів з якнайкращими показниками (наприклад, результат ***yz*** означає, що по першому показнику кращим виявився прилад ***y***, а по другому – прилад ***x*)**.

б) Скільки елементів містить множина всіляких результатів порівняння *m* приладів за *n* показниками?

в) Перерахуєте елементи множини можливих результатів, при яких прилад ***x*** виявляється кращим по першому показнику (A), по другому показнику (B), хоч би по одному показнику (C), за обома показниками (D), не є кращим ні по одному показнику (E).

10. Для множин A, B, З, D, E із завдання 9 в дайте відповіді на наступні питання:

а) Які множини виражаються через об'єднання, доповнення, перетин інших множин?

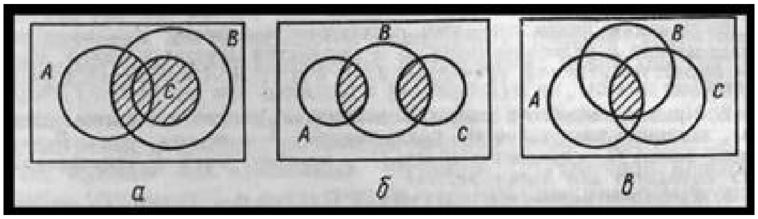
б) Якій множині відповідає різниця A \ Y і який його сенс?

в) Які множини зв'язані між собою відношенням включення?

г) Якій множині відповідає диз'юнктивна сума A + Y і який його сенс?

11. На прикладі множин A і *B* из задачі 9в покажіть справедливість співвідношення $A \backslash B = A \cap \overline{B}$ і проілюструйте його за допомогою кіл Ейлера.

12. Що можна сказати про відносини між множинами A, *B*, *C*, *представленими* колами Ейлера на рис. 1? Запишіть за допомогою операцій над множинами вирази для множин, які відповідають заштрихованим областям.



Рiс.1 Круги Ейлера до завдання 12.

13. Для написання цифр поштового індексу використовують множину з дев'яти елементів, які позначені буквами на рис. 5,а, а самі цифри зображені на рис. 5,б.

а) Скільки різних фігур можна зобразити за допомогою всіляких комбінацій з елементів початкової множини, вважаючи, що в кожній такій комбінації може брати участь від 0 до 9 елементів? Який відсоток цих комбінацій використовується для зображення цифр?

б) Запишіть множину A_k(k = 0, 1 ..., 9) елементів кожною з десяти цифр (наприклад, A_7 = { a, *з, f]).* Чи є серед них множини, що не пересікаються?

в) Запишіть для кожного з елементів s (s = *a,b ...,* i) множину B_s що складається з цифр, в написанні яких використовується елемент s (наприклад, *Bf* = {0, 6, 7, 8}). Які елементи використовуються найбільш рідко і найчастіше?

г) Вважаючи мірою близькості цифр, кількість загальних елементів, вкажіть цифри, найменш і найбільш близькі цифрі 3. Якій операції над множиначю A_k відповідає множина, що визначає міру близькості цифр?

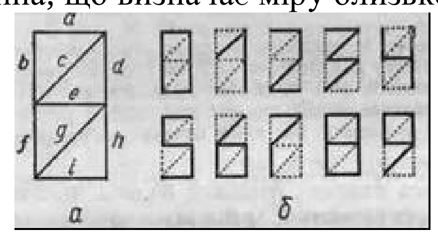


Рис. 2. Зображення цифр поштового індексу:
а — елементи початкової множини; б — цифри.

14. У хімічному продукті можуть опинитися домішки чотирьох видів, позначених через ***a, b, c, d***. Прийнявши як початкову множину A = { a, b, c, d}, утворіть множини всіх його підмножин P(A). Дайте змістовне тлумачення цієї множини і його елементів. Яким ситуаціям відповідають, зокрема, невласні підмножини?

15. Доведіть, що для кінцевої множини, що складається з ***n*** елементів, множина всіх його підмножин містить ***2n*** елементів.

16. Перевірте властивість транзитивності відношення включення на прикладі множин *X = {b, з}, Y = { a, b, з}, Z = {b}.*

17. Дайте словесний опис кожній з наступних множин:

а) {*x* | *x* — точка площини, що знаходиться на відстані **r** від початку координат};

б) {x | x2 - 4x+ 3 = 0};

в) *{ x* | *x—инженер* нашого відділу};

г) *{ x* | *x Î A* і *x Î B}; A* — множина транзисторів; *B* — множина деталей радіоприймача;

д) *{x* ∈ *R* | *x* = 3k, k ∈ *N}; N* — множина натуральних чисел;

е) $\{x^2 + 1$ | x – ціле число}.

18. Покажіть, що для будь-яких множин *A* і *B* справедливе відношення

Æ Ì A Ç B Ì A È B.

19. Покажіть, що для будь-якої множини *A* справедливі співвідношення:

A + *A* = 0; *A* + *Æ* = *A*.

20. Покажіть, що із співвідношення *A Ç B* = C слідує C *Ì A* і *C Ì B*.

21. Хай M_1 і M_2 — відповідно множина деталей першого і другого механізмів, а *P* — множина пластмасових деталей. Запишіть у вигляді співвідношень наступні умови.

а) Серед деталей першого механізму є всі пластмасові деталі.

б) Однакові деталі, що входять в обидва механізми, можуть бути тільки пластмасовими.

в) У другому механізмі немає пластмасових деталей.

22. Чи є сукупність отриманих в попередньому завданні співвідношень (*P Ì M1, M1 Ç M2 Ì P, M2 Ç P = Æ*) є несуперечливою? Якщо так, то чи можна її спростити? Для відповіді на поставлені питання проведіть спочатку логічні міркування, а потім скористайтеся колами Ейлера. Сформулюйте виводи, відповідні отриманому результату.

23. Запишіть множина впорядкованих пар *(x, y)*, що виражають відношення “х, — дільник у» на множині цілих чисел від 2 до 10 включно. Чи є це відношення функцією? Чи має воно властивість транзитивності?

24. Запишіть відношення між елементами множинаі цифр із завдання **13,** що виражається як « х має більше двох загальних елементів з у».

25. Хай *x* ∈ *X*, *y* ∈ *Y* і *A* — відношення між елементами множин *X* і *Y*, що виражається співвідношенням *x A y*. Вкажіть, в яких випадках *A* можна розглядати як функцію:

а) *X* — множина студентів, *Y* — множина учбових дисциплін, *x A y* — «*x* вивчає *y»;*

б*)* *X* — множина спортсменів, *Y* — зріст в одиницях довжини, *x A y*— «*x* має зріст *y»;*

в) *X* — множина компонентів електричного ланцюга, *Y* — множина вузлів ланцюга, *x A y* — «*x* пов'язаний з *y*».

Література до розділу 1

1. Болибрух А. А. , Проблемы Гильберта (100 лет спустя), Глава 2 Первая проблема Гильберта: континуум-гипотеза, Библиотека «Математическое просвещение», Выпуск 2
2. ВерещагинН. К. , ШеньА.. Лекции по математической логике и теории алгоритмов. Часть 1. Начала теории множеств.
3. Ершов Ю. Л., Палютин Е. А. Математическая логика: Учебное пособие. — 3-е, стереотип. изд. — СПб.: «Лань», 2004—336 с.
4. Колмогоров А. Н., Драгалин А. Г. Математическая логика. — М.: УРСС, 2005. – 240 с.
5. КурантР., РоббинсГ., Что такое математика? Глава II, § 4.
6. Френкель А. А., Бар-Хиллел И. Основания теории множеств. — М.: Мир, 1966. – 556 с.

02. НЕЧІТКІ МНОЖИНИ

Напевно, самою вражаючою людського інтелекту є здатність ухвалювати правильні рішення в умовах неповної і нечіткої інформації. Побудова моделей наближених роздумів людини і використання їх в комп'ютерних системах представляє сьогодні одну з найважливіших проблем науки.

Основи нечіткої логіки були закладені в кінці 60-х років в роботах відомого американського математика Лотфі Заде. Дослідження такого роду було викликано зростаючим незадоволенням експертними системами. Хвалений "штучний інтелект", який легко справлявся із завданнями управління складними технічними комплексами, був безпорадним при простих висловах повсякденному життю, типу "Якщо в машині перед тобою сидить недосвідчений водій – тримайся від неї подалі". Для створення дійсно інтелектуальних систем, здатних адекватно взаємодіяти з людиною, був необхідний новий математичний апарат, який переводить неоднозначні життєві твердження в мову чітких і формальних математичних формул. Першим серйозним кроком в цьому напрямі стала теорія нечітких множин, розроблена Заді. Його робота "Fuzzy Sets", опублікована в 1965 році в журналі "Information and Control", заклала основи моделювання інтелектуальної діяльності людини і стала початковим поштовхом до розвитку нової математичної теорії. Він же дав і назву для нової області науки - "fuzzy logic" (fuzzy – нечіткий, розмитий, м'який).

Щоб стати класиком, треба трохи випередити свій час. Існує легенда про те, яким чином була створена теорія "нечітких множин". Одного разу Заде мав довгу дискусію зі своїм другом щодо того, чия з дружин привабливіша. Термін "приваблива" є невизначеним і в результаті дискусії вони не змогли прийти до задовільного підсумку. Це змусило Заде сформулювати концепцію, яка виражає нечіткі поняття типу "приваблива" в числовій формі.

Подальші роботи професора Лотфі Заде і його послідовників заклали фундамент нової теорії і створили передумови для впровадження методів нечіткого управління в інженерну практику.

Апарат теорії нечітких множин, продемонструвавши ряд багатообіцяючих можливостей застосування - від систем управління літальними апаратами до прогнозування підсумків виборів, виявився разом з тим складним для втілення. Враховуючи наявний рівень технології, нечітка логіка зайняла своє місце серед інших спеціальних наукових дисциплін - десь посередині між експертними системами і нейронними мережами.

Тріумфальний хід нечіткої логіки по світу почався після доказу в кінці 80-х Бартоломеєм Косько знаменитої теореми FAT (Fuzzy Approximation Theorem). У бізнесі і фінансах нечітка логіка отримала визнання після того, як в 1988 році експертна система на основі нечітких правил для прогнозування фінансових індикаторів єдина передбачила біржовий крах. І кількість успішних фаззи-применений в даний час обчислюється тисячами.

Третій період почався з кінця 80-х років і до цих пір. Цей період характеризується бумом практичного застосування теорії нечіткої логіки в різних сферах науки і техніки. До 90-го року з'явилося близько 40 патентів, що відносяться до нечіткої логіки (30 - японських). Сорок вісім японських компаній створюють лабораторію LIFE (Laboratory for International Fuzzy Engineering), японський уряд фінансує 5-річну програму по нечіткій логіці, яка включає 19 різних проектів, - від систем оцінки глобального забруднення атмосфери і передбачення землетрусів до АСОВІ заводських цехів. Результатом виконання цієї програми була поява цілого ряду нових масових мікрочіпів, що базуються на нечіткій логіці. Сьогодні їх можна знайти в пральних машинах і відеокамерах, цехах заводів і моторних відсіках автомобілів, в системах управління складськими роботами і бойовими вертольотами.

У США розвиток нечіткої логіки йде по шляху створення систем для великого бізнесу і військових. Нечітка логіка застосовується при аналізі нових ринків, біржовій грі, оцінки політичних рейтингів, виборі оптимальної цінової стратегії і тому подібне З'явилися і комерційні системи масового застосування.

Зсув центру досліджень нечітких систем у бік практичних застосувань привів до постановки цілого ряду проблем, зокрема:

нова архітектура комп'ютерів для нечітких обчислень;

елементна база нечітких комп'ютерів і контроллерів;

інструментальні засоби розробки;

інженерні методи розрахунку і розробки нечітких систем управління, і тому подібне.

Основні зусилля кібернетики зараз направлені на створення штучного інтелекту, що не поступається людському мозку. Оскільки комп'ютери розуміють тільки мову математики, то виникла необхідність представлення нечітких понять у вигляді математичних змінних, названих лінгвістичними.

Сукупність лінгвістичних змінних (мало, багато ...) складає нечітка множина. Наприклад, вік (молодий + не молодий + старий + немолодий + .).

Для опису нечітких множин вводяться поняття нечіткої і лінгвістичної змінних.

Нечітка змінна описується набором (N,X,A), де N – це назва змінної, X – універсальна множина (область міркувань), A – нечітка множина на X.

Значеннями лінгвістичної змінної можуть бути нечіткі змінні, тобто лінгвістична змінна знаходиться на більш високому рівні, чим нечітка змінна. Кожна лінгвістична змінна складається з:

- назви;
- множини своїх значень, яка також називається базовою терм–множиною T. Елементами базової терм-множини є назви нечітких змінних;
- універсальної множини X;
- синтаксичного правила G, за яким генеруються нові терми із застосуванням слів природної або формальної мови;
- семантичного правила P, яке кожному значенню лінгвістичної змінної ставить у відповідність нечітку підмножину множини X.

Розглянемо таке нечітке поняття як «Ціна акції». Це і є назва лінгвістичною змінною. Сформуємо для неї базове терм-множина, яке складатиметься з трьох нечітких змінних: «Низька», «Помірна», «Висока» і задамо область міркувань у вигляді X=[100;200] (одиниць). Останнє, що залишилося зробити – побудувати функції приналежності для кожного лінгвістичного терма з базового терм-множини T.

2.1. Поняття нечітких множин

Хай E - універсальна множина, x - елемент E, а R – певна властивість. Звичайна (чітка) підмножина A універсальної множини E, елементи якої задовольняють властивість R, визначається як множина впорядкованої пари $A = \{\mu A(x) /x\}$, де $\mu A(x)$ – характеристична функція, що приймає значення 1, коли x задовольняє властивості R, і 0 – в іншому випадку.

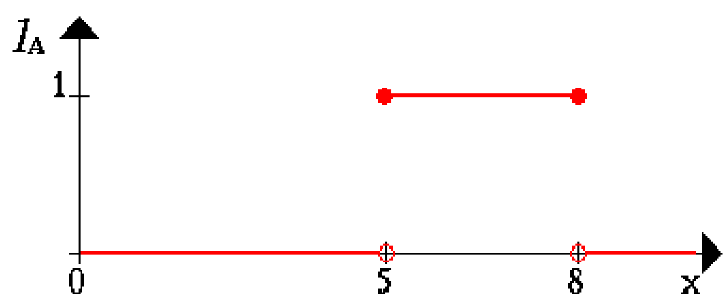
Нечітка підмножина відрізняється від звичайної тим, що для елементів x з E немає однозначної відповіді "ні" щодо властивості R. У зв'язку з цим, нечітка підмножина A універсальної множини E визначається як множина впорядкованою парі $A = \{\mu A(x) /x\}$, де $\mu A(x)$ - характеристична функція приналежності (або просто функція приналежності), що приймає значення в деякій впорядкованій множині M (наприклад, $M = [0,1]$).

Функція приналежності указує ступінь (або рівень) приналежності елементу x до підмножини A. Множину M називають множиною ознак. Якщо $M = \{0,1\}$, тоді нечітка підмножина A може розглядатися як звичайна або чітка множина.

Розглянемо множину X всіх чисел від 0 до 10. Визначимо підмножину A множини X всіх дійсних чисел від 5 до 8.

$$A = [5,8]$$

Покажемо функцію приналежності множині A, ця функція ставить у відповідність число 1 або 0 кожному елементу в X, залежно від того, належить даний елемент підмножині A чи ні. Результат представлений на наступному рисунку:



2.1. Зображення приналежності звичайної (чіткої) множини

Можна інтерпретувати елементи, відповідні 1, як елементи, що знаходяться в множині A, а елементи, відповідні 0, як елементи, що не знаходяться у множині A.

Ця концепція використовується в багатьох областях. Але існують ситуації, в яких даній концепції не вистачатиме гнучкості.

У даному прикладі опишемо множину молодих людей. Формально можна записати так

$$B = \{\text{ множина молодих людей}\}$$

Оскільки, взагалі, вік починається з 0, то нижня межа цієї множини повинна бути нулем. Верхню межу визначити складніше. Спочатку встановимо верхню межу, скажімо, рівну 20 рокам. Таким чином, маємо B як чітко обмежений інтервал, буквально: $B = [0,20]$. Виникає питання: чому хтось в свій двадцятирічний ювілей – молодий, а відразу наступного дня вже не молодий? Очевидно, це структурна проблема, і якщо пересунути верхню межу в іншу точку, то можна поставити таке ж питання.

Природніший шлях створення множини B полягає в ослабленні строгого ділення на молодих і не молодих. Зробимо це, виносячи не тільки чіткі думки "Так, він належить множині молодих людей" чи ні, вона не належить множині молодих людей", але і гнучкі формулювання "Так, він належить до досить молодих людей" чи ні, він не дуже молодий".

Розглянемо як за допомогою нечіткої множини визначити вираз "він ще молодий".

У першому прикладі ми кодували всі елементи множини за допомогою 0 чи 1. Простим способом узагальнити дану концепцію є введення значень між 0 і 1. Реально можна навіть допустити нескінченне число значень між 0 і 1, в одиничному інтервалі $I = [0, 1]$.

Інтерпретація чисел при співвідношенні всіх елементів множини тепер складніша. Звичайно, число 1 відповідає елементу, що належить множині B, а 0 означає, що елемент точно не належить множині B. Всі інші значення визначають ступінь приналежності до множини B.

Для наочності приведемо характеристичну функцію множини молодих людей, як і в першому прикладі.

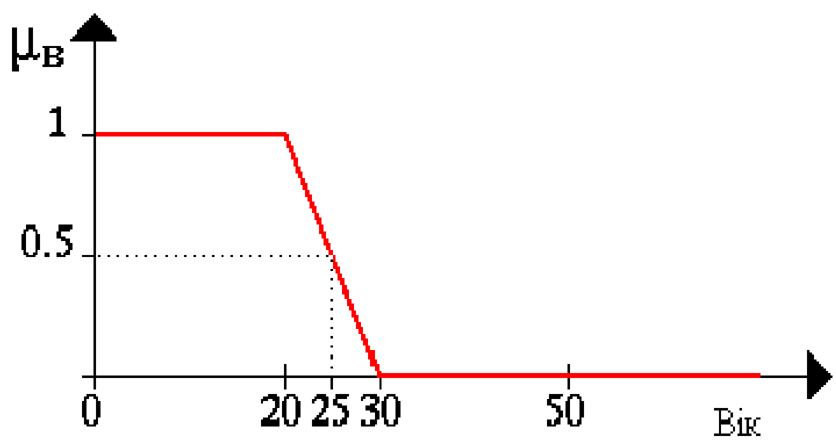


Рис. 2.2. Характеристична функція множини молодих людей

Хай $E = \{x_1, x_2, x_3, x_4, x_5\}$, $M = [0,1]$; A – нечітка множина, для якої

$$\mu A(x_1)=0{,}3;\ \mu A(x_2)=0;\ \mu A(x_3)=1;\ \mu A(x_4)=0{,}5;\ \mu A(x_5)=0{,}9$$

Тоді A можна представити у вигляді:

$$A = \{0{,}3/x_1;\ 0/x_2;\ 1/x_3;\ 0{,}5/x_4;\ 0{,}9/x_5\}\ \text{або}$$

$$A = 0{,}3/x_1 + 0/x_2 + 1/x_3 + 0{,}5/x_4 + 0{,}9/x_5$$

(знак "+" є операцією не складання, а об'єднання) або

	x_1	x_2	x_3	x_4	x_5
$A =$	0,3	0	1	0,5	0,9

Формалізуємо неточне визначення «гарячий чай». Як x (область існування) виступатиме шкала температури в градусах Цельсія. Очевидно, що вона буде змінюється від 0 до 100 градусів. Нечітка множина для поняття «Гарячий чай» може виглядати таким чином:
$C = \{0/0; 0/10; 0/20; 0{,}15/30; 0{,}30/40; 0{,}60/50; 0{,}80/60; 0{,}90/70; 1/80; 1/90; 1/100\}$.

Так, чай з температурою 60°С належить до множини «Гарячий» зі ступенем приналежності 0,80. Для однієї людини чай при температурі 60°С може опинитися гарячим, для іншого – не дуже гарячим. Саме у цьому і виявляється нечіткість завдання відповідної множини.

2.2. Основні характеристики нечітких множин

Хай $M = [0,1]$ і A – нечітка множина з елементами з універсальної множина E і з множиною визначення M

- Величина $\mu A(x)$ $\sup\limits_{x\in E}$ називається висотою нечіткої множини A. Нечітка множина A є нормальною, якщо її висота дорівнює 1, тобто верхня межа її функції приналежності дорівнює 1 ($\sup\limits_{x\in E}$ $\mu A(x)=1$). При $\mu A(x) <1$ нечітка множина називається субнормальною.
- Нечітка множина є порожньою, якщо $\forall x\in E$ μ $A(x)=0$. Не порожню субнормальну множину можна нормалізувати за формулою $\mu A(x) := \dfrac{\mu_A(x)}{\sup\limits_{x\in E}\mu_A(x)}$
- Нечітка множина є унімодальною, якщо $\mu A(x)=1$ лише для одного x з E.
- Носієм нечіткої множини A є звичайна підмножина з властивістю $\mu A(x) >0$, тобто носій $A = \{x / \mu A(x) > 0\}\ \forall\ x\in E$

- Елементи $\boldsymbol{x} \in \boldsymbol{E}$, для яких $\mu A(x)=0,5$ називаються точками переходу множини A.

Наведемо приклади нечітких множин з їх характеристиками:

1. Хай E = {0,1,2..,10}, M =[0,1]. Нечітку множину "декілька" можна визначити таким чином:

"декілька" = 0,5/3+0,8/4+1/5+1/6+0,8/7+0,5/8;

її характеристики: – висота = 1,

– носій={3,4,5,6,7,8},

– точки переходу – {3,8}.

2. Хай E = {0,1,2,3...,n,...}. Нечітку множину "малий" можна визначити:

$$\mu\text{"малий"}(x)= \left\{ \mu_{\text{"малий"}}(n) = \frac{1}{1+\left(\frac{n}{10}\right)^2} / n \right\}, \qquad (2.1)$$

3. Хай E = {1,2,3...,100} і відповідає поняттю "вік", тоді нечітка множина "молодий", можна визначити з допомогою

$$\mu\text{"молодий"}(x)= \begin{cases} 1, & x \in [1,25] \\ \dfrac{1}{1+\left(\dfrac{x-25}{5}\right)^2}, & x \geq 25 \end{cases}, \qquad (2.2)$$

$$m(x) = \frac{1}{1+\frac{(x-a)^2}{b}} / x$$

$$m(x) = a\sqrt{\frac{x}{x_{max}}\left(1-\frac{x}{x_{max}}\right)}$$

Нечітка множина "молодий" на універсальній множині E' ={ Іванов, Петров, Сидорів...} задається за допомогою функції приналежності μ"молодий"(x) на E = {1,2,3..100} (вік), що називається відносно E' функцією сумісності, при цьому:

μ"молодий"(Сидорів)= μ"молодий"(x), де x – вік Сидорова.

4. Хай E = { Запорожець, Жигулі, Мерседес....} – множина марок автомобілів, а E' = [0,μ] – універсальна множина "вартість", тоді на E' ми можемо визначити нечітку множина типу: "для небагатих ", "для середнього класу", "престижні", з функціями приналежності типу:

Рис. 2.3. Характеристична функція множини вартість автомобіля для людей різного достатку

Маючи ці функції і знаючи ціни автомобілів з E в даний момент часу, визначимо на E' нечіткі множини з цими ж назвами.

Так, наприклад, нечітка множина "для небагатих", задана на універсальній множині E = { Запорожець, Жигулі, Мерседес....} виглядає таким чином:

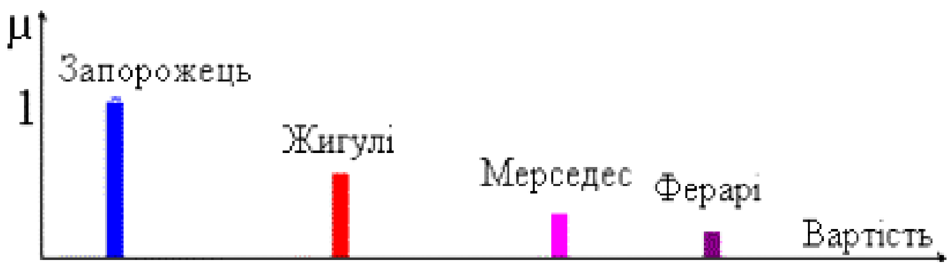


Рис. 2.4. Характеристична функція множини "для небагатих"

Аналогічно можна визначити нечітку множину "швидкісні", "середні", "тихохідні" і так далі

2.3. Функція приналежності

У приведених вище прикладах використані прямі методи, коли експерт або просто задає для будь-якого $\boldsymbol{x} \in \boldsymbol{E}$ значення $\mu A(x)$, або визначає функцію приналежності. Як правило, прямі методи завдання функції приналежності використовуються для вимірних понять, таких як швидкість, година, відстань, тиск, температура і так далі, тобто коли виділяються полярні значення.

У багатьох завданнях при характеристиці об'єкту можна виділити набір ознак і для будь-якого з них визначити полярні значення, що відповідають значенням функції приналежності, 0 або 1.

Наприклад, в завданні розпізнавання особи можна виділити наступні пункти:

Позначення	Опис характеристики	0	1
x_1	висота лоба	низький	широкий
x_2	профіль носа	кирпоносий	горбатий
x_3	довжина носа	короткий	довгий

x_4	розріз очей	вузький	широкий
x_5	колір очей	світлий	темний
x_6	форма підборіддя	гострий	квадратний
x_7	товщина губ	тонкі	товсті
x_8	колір особи	темний	світлий
x_9	овал особи	овальне	квадратне

Для конкретної особи *A* експерт, виходячи з приведеної шкали, задає $mA(\mathrm{x}) \in [0,1]$,], формуючи векторну функцію приналежності { $\mu A(x_1)$), $\mu A(x_2)$... $\mu A(x_9)$}.

Непрямі методи визначення значень функції приналежності використовуються у випадках, коли немає елементарних вимірних властивостей для визначення нечіткої множини. Як правило, це методи попарних порівнянь. Якби значення функцій приналежності були відомі, наприклад, $mA(x_i) = w_i$, $i=1,2,...,n$,, тоді попарні порівняння можна представити матрицею відносин А = $\{a_{ij}\}$, де $a_{ij}=w_i/w_j$ (операція ділення).

Існує понад десяток типових форм кривих для завдання функцій приналежності. Найбільшого поширення набули: трикутна, трапецеїдальна функції та функція приналежності Гауса.

Трикутна функція приналежності визначається трійкою чисел (a,b,c), і її значення в точці x обчислюється згідно виразу:

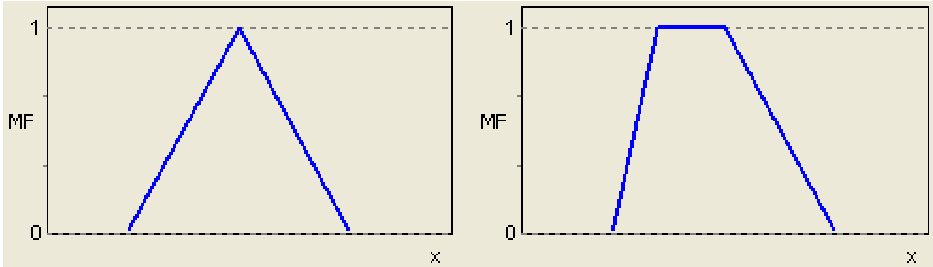
$$MF(x) = \begin{cases} 1 - \dfrac{b-x}{b-a}, & a \le x \le b \\ 1 - \dfrac{x-c}{c-b}, & b \le x \le c \\ 0, & \text{в інших випадках} \end{cases} , \tag{2.3}$$

При $(b\text{-}a)=(c\text{-}b)$ маємо випадок симетричної трикутної функції приналежності, яка може бути однозначно задана двома параметрами з трійки (a,b,c).

Аналогічно для завдання трапецеїдальній функції приналежності необхідна четвірка чисел (a,b,c,d):

$$MF(x) = \begin{cases} 1 - \dfrac{b-x}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ 1 - \dfrac{x-c}{d-c}, & c \le x \le d \\ 0, & \text{в інших випадках} \end{cases} , \tag{2.4}$$

При $(b\text{-}a)=(d\text{-}c)$ трапецеїдальна функція приналежності приймає симетричний вигляд.



a b

Рис. 2.5.. Типові кусочно-лінійні функції приналежності;
трикутна (a) та трапецеїдальна (b).

Функція приналежності гауссова типу описується формулою

$$MF(x) = \exp\left[-\left(\frac{x-c}{\sigma}\right)^2\right], \qquad (2.5)$$

і оперує двома параметрами. Параметр c позначає центр нечіткої множини, а параметр σ відповідає за крутизну функції.

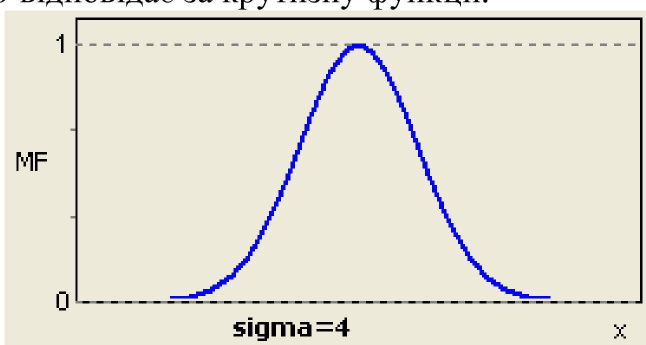


Рис. 2.6. Гауссова функція приналежності.

Сукупність функцій приналежності для кожного терма з базової терм-множини T зазвичай зображаються разом на одному графіку. На рис. 2.7 приведений приклад описаної вище лінгвістичної змінної «Ціна акції», на рис. 2.8 – формалізація неточного поняття «Вік людини». Так, для людини 48 років ступінь приналежності до множини «Молодий» рівна 0, «Середній» – 0,47, «Вище середнього» – 0,20.

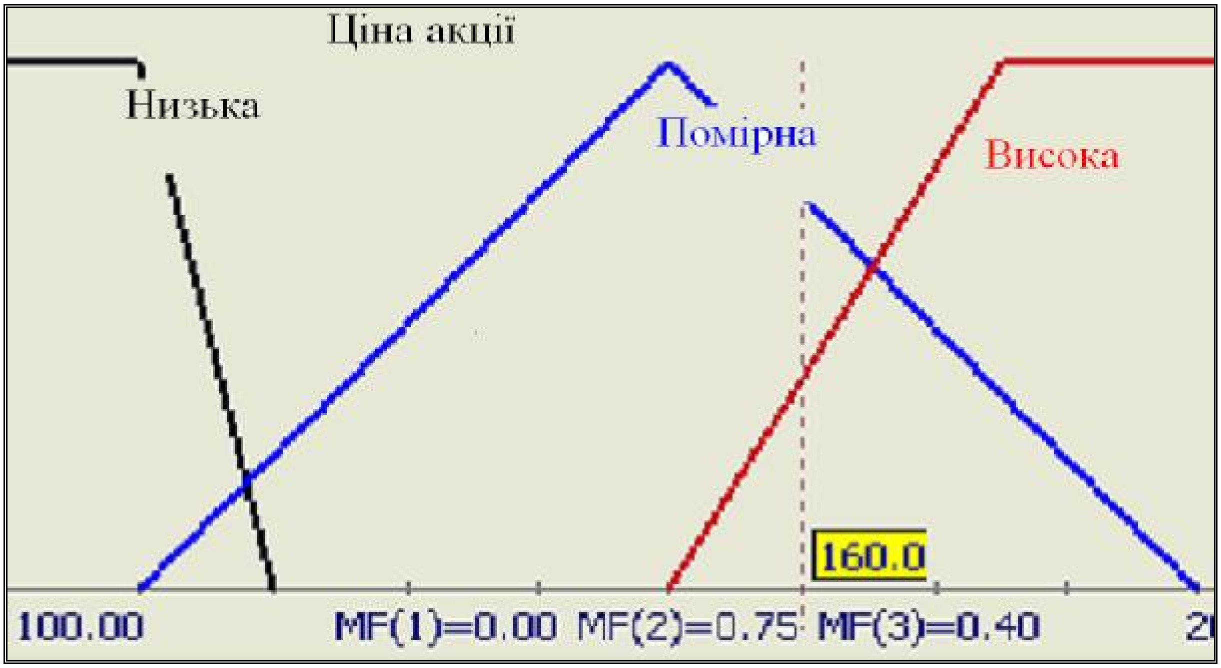


Рис. 2.7. Опис лінгвістичної змінної «Ціна акції».

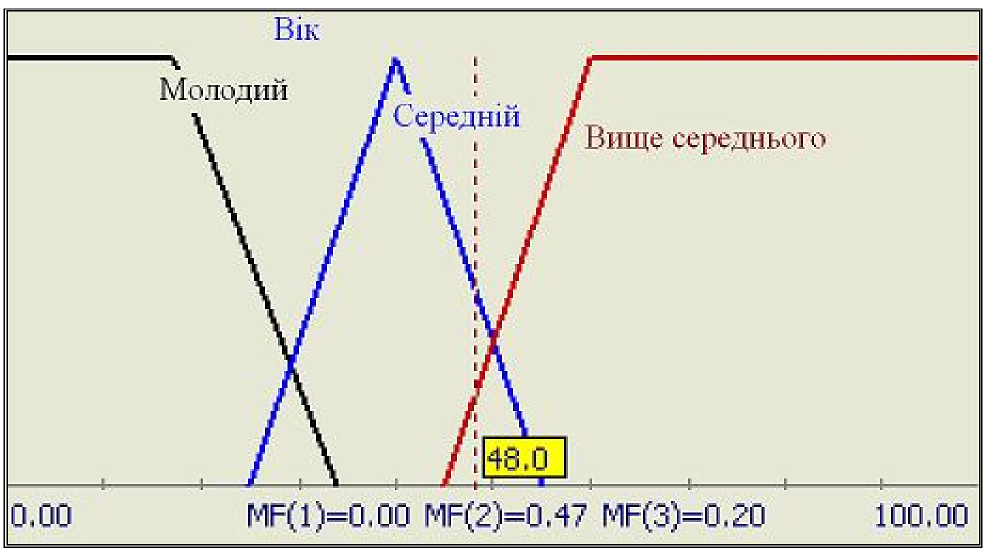


Рис. 2.8. Опис лінгвістичної змінної «Вік».

Кількість термів в лінгвістичній змінній рідко перевищує 7.

2.3. Операції над нечіткими числами

Цілий розділ теорії нечітких множин – м'які обчислення (нечітка арифметика) – вводить набір операцій над нечіткими числами. Ці операції вводяться через операції над функціями приналежності на основі так званого **сегментного принципу**.

Визначимо *рівень приналежності* a як ординату функції приналежності нечіткого числа. Тоді перетин функції приналежності з нечітким числом дає пару значень, які прийнято називати *межами інтервалу достовірності*.

Задамося фіксованим рівнем приналежності a і визначимо відповідні йому інтервали достовірності по двох нечітких числах $\underline{A}$ і $\underline{B}$: $[a_1, a_2]$ і $[b_1, b_2]$, відповідно. Тоді основні операції з нечіткими числами зводяться до операцій з їх інтервалами достовірності. А операції з інтервалами, у свою чергу, виражаються через операції з дійсними числами - межами інтервалів:

- операція "складання":

$$[a_1, a_2]\ (+)\ [b_1, b_2] = [a_1 + b_1, a_2 + b_2] \tag{2.6}$$

- операція "віднімання":

$$[a_1, a_2]\ (-)\ [b_1, b_2] = [a_1 - b_2, a_2 - b_1] \tag{2.7}$$

- операція "множення":

$$[a_1, a_2]\ (\times)\ [b_1, b_2] = [a_1 \times b_1, a_2 \times b_2], \tag{2.8}$$

- операція "ділення":

$$[a_1, a_2]\ (/)\ [b_1, b_2] = [a_1 / b_2, a_2 / b_1], \tag{2.9}$$

- операція "піднесення до ступеня":

$$[a_1, a_2]\ (\wedge)\ i = [a_1^{\,i}, a_2^{\,i}]. \tag{2.10}$$

З суті операцій з трапезоидными числами можна зробити ряд важливих тверджень :

- дійсне число є окремим випадком трикутного нечіткого числа;
- сума трикутних чисел є трикутне число;
- трикутне (трапезоїдне) число, помножене на дійсне число, є трикутне (трапезоїдне) число;
- сума трапезоїдних чисел є трапезоїдне число;

Аналізуючи властивості нелінійних операцій з нечіткими числами (наприклад, ділення), дослідники приходять до висновку, що форма функцій приналежності результуючих нечітких чисел часто близька до трикутної. Це дозволяє апроксимувати результат, приводячи його до трикутного вигляду. І, якщо приводимість в наявності, тоді *операції з трикутними числами зводяться до операцій з абсцисами вершин їх функцій приналежності*.

Тобто, якщо ми вводимо опис трикутного числа набором абсцис вершин (a, b, c), то можна записати:

$$(a_1, b_1, c_1) + (a_2, b_2, c_2) \equiv (a_1 + a_2, b_1 + b_2, c_1 + c_2) \tag{2.11}$$

Це – найпоширеніше правило м'яких обчислень.

2.4. Операції над нечіткими множинами:

1. Домінування (Вміщення)

Хай A і B - нечіткі множини на універсальній множині E.

Говорять, що A міститься в B, якщо $\forall x \in E\ \mu A(x) < \mu B(x)$. (2.12)

Позначення: $A \subset B$.

Коли використовують термін "домінування", тобто у випадку якщо $A \subset B$, говорять, що B домінує A.

2. Рівність

A і B рівні, якщо $\forall x \in E\ \mu A(x) = \mu B(x)$. (2.13)

Позначення: $A = B$.

3. Доповнення

Хай $\mu = [0,1]$, A і B - нечіткі множини, задані на E. A і B доповнюють один одного, якщо $\forall x \in E\ \ \mu A(x) = 1 - \mu B(x)$. (2.14)

Позначення: $B = \overline{B}$ або $A = \overline{A}$

Очевидно, що $\overline{A} = A$. (Доповнення визначене для $\mu = [0,1]$, але очевидно, що його можна визначити для будь-якого впорядкованого M).

Доповнення нечіткої множини A позначається символом $\neg A$ і визначається

$$\neg A = \int_U (1 - m_A(u)) \Big/ u \ . \qquad (2.15)$$

Операція доповнення відповідає логічному запереченню.

4. Перетин

Перетин A і B позначається $A \cap B$ і визначається

$$A \cap B = \int_U (m_A(u) \wedge m_B(u)) \Big/ u \ . \qquad (2.16)$$

Перетин відповідає логічній зв'язці «і». $A \cap B$ – найменша нечітка підмножина, яка міститься одночасно в A і B:

$$\mu A \cap B(x) = \mu in(\ \mu A(x), \mu B(x)). \qquad (2.17)$$

5. Об'єднання

Об'єднання нечітких множин A і B ($A + B$)

$$A + B = \int_U (m_A(u) \vee m_B(u)) \Big/ u \ . \qquad (2.18)$$

Об'єднання відповідає логічній зв'язці «або».

$A \cup B$ – найбільша нечітка підмножина, яка включає як A, так і B, з функцією приналежності:

$$\mu A \cup B(x) = \mu ax(\mu A(x), \mu\ B(x)). \qquad (2.19)$$

6. Різниця

A - B = A$\cap$ $\bar{B}$ з функцією приналежності:

$$\mu A - B(x) = \mu A \cap \bar{B} (x) = \min(\mu A(x), 1 - \mu B(x)). \quad (2.20)$$

7. Диз'юнктивна сума

$A \oplus B = (A - B) \cup (B - A) = (A \cap \bar{B}) \cup (\bar{A} \cap B)$ з функцією приналежності:

$$\mu A - B(x) = \max\{[\min\{\mu A(x), 1 - \mu B(x)\}];$$

$$[\min\{1 - \mu A(x), \mu B(x)\}]\} \quad (2.21)$$

8. Добуток A і B позначається AB і визначається

$$AB = \int_U (m_A(u) m_B(u)) / u \, . \quad (2.22)$$

9. Піднесення до ступеня

$$a > 0 \quad A^a = \int_U (m_A(u))^a / u \, . \quad (2.23)$$

10. Концентрація, частковий випадок піднесення до ступеня*:*

$$CON(A) = A^2 . \quad (2.24)$$

11. Розтягування (розмивання) $DIL(A) = A^{0.5}$. (2.25)

Властивості операцій $\cup$ і $\cap$

Хай A, B, C - нечіткі множини, тоді виконуються наступні властивості:

- $\left.\begin{array}{l} A \cap B = B \cap A \\ A \cup B = B \cup A \end{array}\right\}$ – комутативність;
- $\left.\begin{array}{l} (A \cap B) \cap C = A \cap (B \cap C) \\ (A \cap B) \cap C = A \cap (B \cap C) \end{array}\right\}$ – асоціативність;
- $\left.\begin{array}{l} A \cap A = A \\ A \cup A = A \end{array}\right\}$ – ідемпотентність;
- $\left.\begin{array}{l} A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \end{array}\right\}$ – дистрибутивність;

- $A \cup \varnothing = A$, , де $\varnothing$ - порожня множина, тобто $\mu\varnothing(x) = 0$ $\forall x \in E$;
- $A \cap \varnothing = \varnothing$;
- $A \cap E = A$,, де E - універсальна множина;
- $A \cup E = E$;;
- $\left.\begin{array}{l}\overline{A \cap B} = \overline{A} \cup \overline{B} \\ \overline{A \cup B} = \overline{A} \cap \overline{B}\end{array}\right\}$ – теореми де Моргана.

На відміну від чітких множин, для нечітких множин в загальному випадку:

- $A \cap \overline{A} \neq \varnothing$,
- $A \cup \overline{A} \neq E$.

Приклади

Хай: $A = 0{,}4/x_1 + 0{,}2/x_2 + 0/x_3 + 1/x_4$;

$B = 0{,}7/x_1 + 0{,}9/x_2 + 0{,}1/x_3 + 1/x_4$;

$C = 0{,}1/x_1 + 1/x_2 + 0{,}2/x_3 + 0{,}9/x_4$.

Тоді, для визначених вище перетворень маємо:

1. $A \subset B$, тобто *A* міститься в *B* або *B* домінує *A*, *C* не зрівняно ні з *A*, ні з *B*, тобто пари {*A*, *C*} і {*B*, *C*} – пари не домінуємих нечітких множин.

2. $A \neq B \neq C$.

3. $\overline{A} = 0{,}6/x_1 + 0{,}8/x_2 + 1/x_3 + 0/x_4$;

$\overline{B} = 0{,}3/x_1 + 0{,}1/x_2 + 0{,}9/x_3 + 0/x_4$.

4. $A \cap B = 0{,}4/x_1 + 0{,}2/x_2 + 0/x_3 + 1/x_4$.

5. $A \cup C = 0{,}7/x_1 + 0{,}9/x_2 + 0{,}1/x_3 + 1/x4$.

6. $A - C = A \cap \overline{B} = 0{,}3/x_1 + 0{,}1/x_2 + 0/x_3 + 0/x_4$;

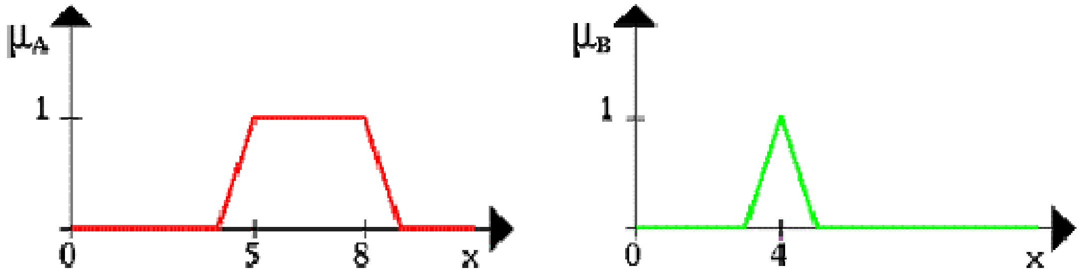
$B - A = \overline{A} \cap C = 0{,}6/x_1 + 0{,}8/x_2 + 0{,}1/x_3 + 0/x_4$.

7. $A \oplus B = 0{,}6/x_1 + 0{,}8/x_2 + 0{,}1/x_3 + 0/x_4$.

2.5. Наочне представлення операцій над нечіткими множинами

Для нечітких множин можна застосувати візуальне уявлення. Розглянемо прямокутну систему координат, на осі ординат якої відкладаються значення $\mu A(x)$, на осі абсцис в довільному порядку розташовані елементи *E*. Якщо *E* за своєю природою впорядковано, то цей порядок бажано зберегти в розташуванні елементів на осі абсцис. Таке уявлення робить наочними прості операції над нечіткими множинами.

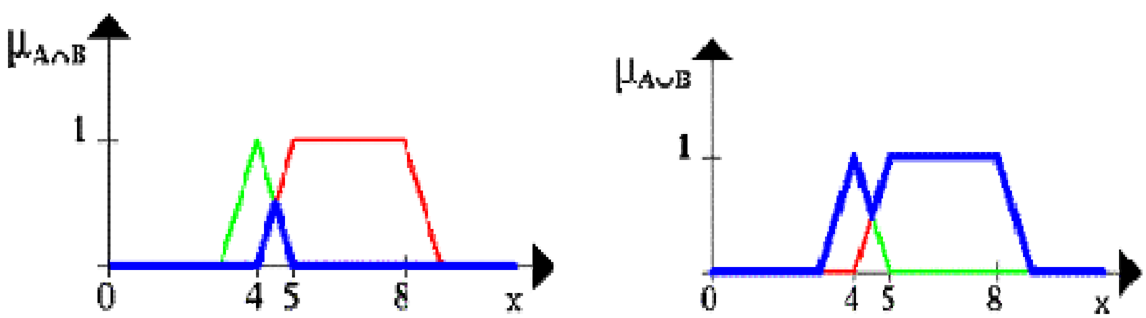
Хай *A* нечіткий інтервал між 5 до 8 і *B* нечітке число близько 4, як показано на рис. 2.9.



A *B*

Рис. 2.9. Наочне представлення нечіткого інтервалу *A* і нечіткого числа *B*

Проілюструємо нечітку множину між 5 і 8 «І» (AND) близько 4 (синя лінія на рис 2.10.а). Нечітка множина між 5 і 8 «АБО» (OR) близько 4 показано на наступному малюнку (знову синя лінія на рис. 2.10.б).



а б

Рис. 2.10. Наочне представлення операцій «І» (а) та «АБО» (б)

аступний малюнок ілюструє операцію доповнення. Синя лінія - це ДОПОВНЕННЯ нечіткої множинаі *A* (рис. 2.11).

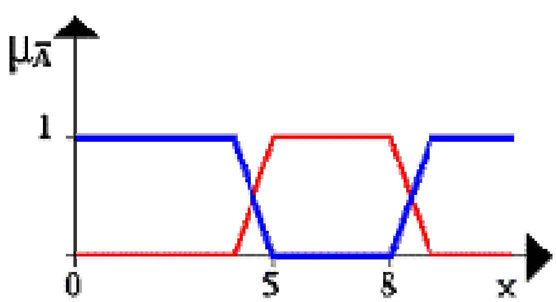


Рис. 2.11. Ілюстрація операції доповнення нечіткої множини А

На рис. 2.12 заштрихована частина відповідає нечіткій множинаі *A* і зображає область значень *A* і всіх нечітких множин, що містяться в *A*. Решта малюнків зображає відповідно, $\overline{A}$, $A\cap\overline{A}$, $A\cup\overline{A}$.

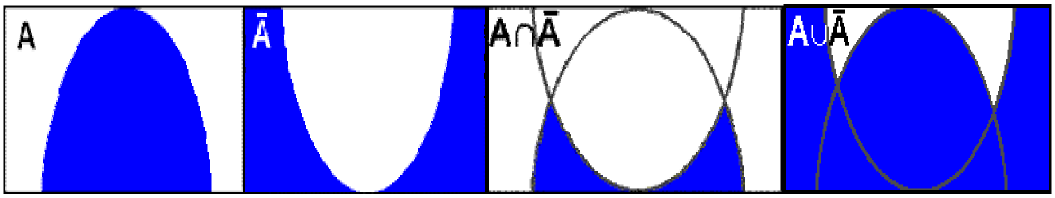


Рис. 2.12. Наочне представлення операцій $\bar{A}$, $A \cap \bar{A}$, $A \cup \bar{A}$ з нечіткою множиною A

На рис. 2.13 наведено наочний приклад використання операції концентрації та розмивання для нечіткої множини A.

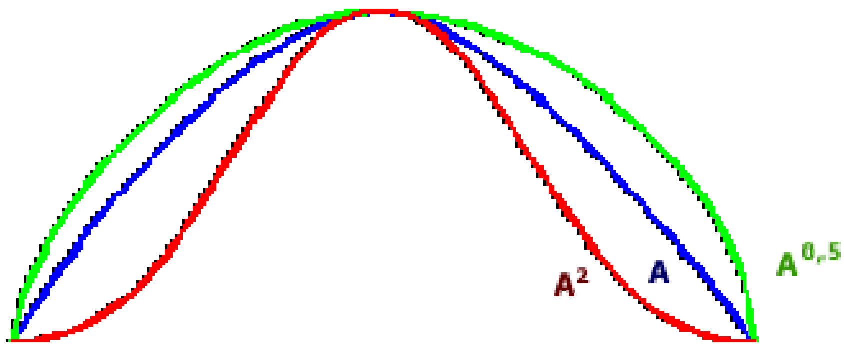


Рис. 2.13. Операції концентрації CON(A) та розмивання DIL(A)

2.6. Переваги нечітких та застосування нечітких систем

Коротко перерахуємо переваги fuzzy-систем в порівнянні з іншими:

- можливість оперувати нечіткими вхідними даними: наприклад, значення (динамічні завдання), що безперервно змінюються в часі, значення, які неможливо задати однозначно (результати статистичних опитів, рекламні компанії і так далі);
- можливість нечіткої формалізації критеріїв оцінки і порівняння: операція критеріями "більшість", "можливо", переважно" і т.д.;
- можливість проведення якісних оцінок як вхідних даних, так і вихідних результатів: ви оперуєте не тільки значеннями даних, але і їх мірою достовірності (не плутати з вірогідністю!) і її розподілом;
- можливість проведення швидкого моделювання складних динамічних систем і їх порівняльний аналіз із заданим ступенем точності: оперуючи принципами поведінки системи, описаними fuzzy-методами, ви по-перше, не витрачаєте багато часу на з'ясування точних значень змін-

них і складання рівнянь, що описують, по-друге, можете оцінити різні варіанти вихідних значень.

Що стосується вітчизняного ринку комерційних систем на основі нечіткої логіки, то його формування почалося в середині 1995 року. Популярними є наступні пакети:

- CubiCalc 2.0 RTC – одна з могутніх комерційних експертних систем на основі нечіткої логіки, що дозволяє створювати власні прикладні експертні системи ;
- CubiQuick – дешева "університетська" версія пакету CubiCalc ;
- RuleMaker – програма автоматичного витягання нечітких правил зі вхідних даних ;
- FuziCalc – електронна таблиця з нечіткими полями, що дозволяє робити швидкі оцінки при неточних даних без накопичення погрішності;
- OWL – пакет, що містить початкові тексти всіх відомих видів нейронних мереж, нечіткій асоціативній пам'яті і так далі

Основними споживачами нечіткої логіки на ринку СНД є банкіри і фінансисти, а також фахівці в області політичного і економічного аналізу. Вони використовують CubiCalc для створення моделей різних економічних, політичних, біржових ситуацій. Що ж до пакету FuziCalc, то він зайняв своє місце на комп'ютерах великих банкірів і фахівців з надзвичайних ситуацій – тобто тих, для кого важлива швидкість проведення розрахунків в умовах неповноти і неточності вхідної інформації. Проте можна з упевненістю сказати, що епоха розквіту прикладного використання нечіткої логіки на вітчизняному ринку ще попереду.

Сьогодні елементи нечіткої логіки можна знайти в десятках промислових виробів – від систем управління електропоїздами і бойовими вертольотами до пилососів і пральних машин. Без застосування нечіткої логіки немислимі сучасні ситуаційні центри керівників західних країн, де ухвалюються ключові політичні рішення і моделюються різні кризові ситуації. Одним з вражаючих прикладів масштабного застосування нечіткої логіки стало комплексне моделювання системи охорони здоров'я і соціального забезпечення Великобританії (National Health Service – NHS), яке вперше дозволило точно оцінити і оптимізувати витрати на соціальні потреби.

Не обійшли засоби нечіткої логіки і програмні системи, обслуговуючих великий бізнес. Першими, зрозуміло, були фінансисти, завдання яких вимагають щоденного ухвалення правильних рішень в складних умовах непередбаченого ринку. Перший рік використання системи Fuji Bank приніс банку в середньому $770000 на місяць (і це тільки офіційно оголошений прибуток!).

Услід за фінансистами, стурбовані успіхами японців і втратою стратегічної ініціативи, когнітивними нечіткими схемами зацікавилися промислові гіганти США. Motorola, General Electric, Otis Elevator, Pacific Gas & Electric, Ford та інші на початку 90-х почали інвестувати в розробку виробів, що використовують нечітку логіку. Маючи солідну фінансову "підтримку", фірми, що специалізуються на нечіткій логіці, дістали можливість адаптувати свої розробки для широкого круга застосувань. "Зброя еліти" вийшла на масовий ринок.

Серед лідерів нового ринку виділяється американська компанія Hyper Logic, заснована в 1987 році Фредом Уоткинсом (Fred Watkins). Спочатку компанія спеціалізувалася на нейронних мережах, проте незабаром цілком концентрувалася на нечіткій логіці. Недавно вийшла на ринок друга версія пакету CubiCalc фірми HyperLogic, яка є одній з щонайпотужніших експертних систем на основі нечіткої логіки. Пакет містить інтерактивну оболонку для розробки нечітких експертних систем і систем управління, а також run-time модуль, що дозволяє оформляти створені користувачем системи у вигляді окремих програм.

Окрім Hyper Logic серед "патріархів" нечіткої логіки можна назвати фірми IntelligenceWare, InfraLogic, Aptronix. Всього ж на світовому ринку представлено більше 100 пакетів, які так чи інакше використовують нечітку логіку. У трьох десятках СУБД реалізована функція нечіткого пошуку. Власні програми на основі нечіткої логіки анонсували такі гіганти як IBM, Oracle та інші.

На принципах нечіткої логіки створений і один з російських програмних продуктів – відомий пакет "Бізнес-прогноз". Призначення цього пакету – оцінка ризиків і потенційної прибутковості різних бізнес-планів, інвестиційних проектів і просто ідей щодо розвитку бізнесу. "Ведучи" користувача за сценарієм його задуму, програма задає ряд питань, які допускають як точні кількісні відповіді, так і наближені якісні оцінки, – типу "маловірогідно", "ступінь риски високий" і так далі Узагальнивши всю отриману інформацію у вигляді однієї схеми бизнес-проекта, програма не тільки виносить остаточний вердикт про ризикованy проекту і очікуваних прибутків, але і указує критичні крапки і слабкі місця в авторському задумі. Від аналогічних іноземних пакетів "Бізнес-прогноз" відрізняється простотою, дешевизною і, зрозуміло, російськомовним інтерфейсом. Втім, програма "Бізнес-прогноз" - лише перша ластівка, за якою неминуче з'являться нові розробки учених СНД.

2.7. Індивідуальне завдання №2.

Задача 1

Визначити спільності значень в трьох категоріях **вік**, **час доби, колір палітри.** Записати їх із застосуванням функції приналежності m_i, де $i \le 7$. Побудувати графіки функцій приналежності.

№ варіанту	**Вік**	**Час доби**	**Колір палітри**
1	Зрілість	ніч	фіолетовий
2	Юність	вечір	зелений
3	Студентський вік	ранок	жовтий

№ варіанту	Вік	Час доби	Колір палітри
4	Дитинство	полудень	червоний
5	Отроцтво	день	синій
6	Пенсійний вік	північ	білий
7	Підліток	«на зірці»	сірий
8	Старість	сутінки	рожевий
9	Репродуктивний вік	«ні день, ані ніч»	коричневий
10	Шкільний вік	полудень	фіолетовий
11	Зрілість	день	зелений
12	Юність	ніч	жовтий
13	Отроцтво	вечір	червоний
14	Пенсійний вік	ранок	синій
15	Підліток	полудень	білий
16	Репродуктивний вік	день	сірий
17	Зрілість	північ	рожевий
18	Юність	«на зірці»	коричневий
19	Студентський вік	сутінки	фіолетовий
20	Дитинство	«ні день, ані ніч»	зелений
21	Отроцтво	полудень	жовтий
22	Пенсійний вік	день	червоний
23	Підліток	північ	синій
24	Старість	«на зірці»	білий
25	Репродуктивний вік	сутінки	сірий
26	Шкільний вік	«ні день, ані ніч»	рожевий
27	Студентський вік	день	коричневий
28	Дитинство	північ	білий
29	Отроцтво	«на зірці»	червоний
30	Пенсійний вік	сутінки	синій

Задача 2

Виконати операції над нечіткими множинами: **доповнення** нечіткої множини *A* і *B*; **об'єднання** нечітких множин *A* і *B*; **перетин** *A* і *B*; **добуток** *A* і *B*; ***піднесення до ступеня; концентрація; розтягування***. Якщо *U* =1+2+3+…+10. Множини *A* і *B* вибираються з варіантів за номером студента в списку групи.

№ варіанту	*A*	*B*
1	0,8/4+1/6+0,6/8	0,3/2+0,8/4+1/6+0,6/8
2	0,7/2+0,9/4+1/6+0,8/7	0,7/2+1/4+0,3/8
3	0,6/3+0,8/5+1/7	0,4/1+0,6/3+0,7/5+1/7

№ варіан-ту	*A*	*B*
4	0,2/1+0,4/3+0,6/5+1/8	0,3/2+0,7/3+1/5
5	0,7/4+0,9/5+1/7	0,3/2+0,5/4+0,8/7+1/9
6	0,6/4+0,8/5+1/8	0,2/1+0,5/3+0,8/6+1/7
7	0,3/2+0,7/3+1/5	0,2/1+0,4/3+0,6/5+1/8
8	0,2/1+0,4/3+0,6/5+1/8	0,2/1+0,4/3+0,6/5+1/8
9	0,3/3+0,4/5+0,6/7+1/9	0,4/1+0,5/3+0,6/5+0,8/8
10	0,7/2+1/4+0,3/8	0,8/4+1/6+0,6/8+0,9/9
11	0,6/7+0,8/8+0,9/9+1/10	0,4/7+0,6/8+0,7/9
12	0,1/3+0,2/5+0,3/7+0,4/10	0,4/1+0,6/3+0,7/5+1/7
13	0,6/2+0,8/4+0,9/9+1/10	0,4/7+0,6/8+0,7/9
14	0,1/3+0,2/5+0,3/7+0,4/10	0,4/1+0,6/3+0,7/5+1/7
15	0,5/2+0,6/7+0,7/8+0,9/10	0,4/2+0,6/5+0,7/7+1/10
16	0,6/3+0,8/5+1/7	0,2/2+0,4/3+0,5/5+0,6/7
17	0,3/3+0,4/5+0,6/7+0,9/8	0,4/3+0,6/5+0,7/6+1/7
18	0,3/5+0,5/6+0,7/7+0,9/9	0,4/6+0,6/7+0,7/8+1/9
19	0,4/7+0,6/8+0,7/9	0,1/3+0,2/5+0,3/7+0,4/8
20	0,4/1+0,5/3+0,7/5+1/7	0,6/7+0,8/8+0,9/9+1/10
21	0,1/7+0,6/8+0,7/9	0,1/3+0,2/5+0,3/7+0,4/9
22	0,4/1+0,5/3+0,7/5+1/7	0,5/2+0,6/7+0,7/8+0,9/10
23	0,3/2+0,6/5+0,7/7+1/10	0,6/3+0,8/5+1/7
24	0,2/2+0,4/3+0,5/5+0,6/7	0,3/5+0,5/6+0,7/7+0,9/9
25	0,6/4+0,8/5+1/8	0,1/3+0,2/5+0,3/7+0,4/10
26	0,3/2+0,7/3+1/5	0,6/7+0,8/8+0,9/9+1/10
27	0,2/1+0,4/3+0,6/5+1/8	0,4/1+0,6/3+0,7/5+1/7
28	0,3/3+0,4/5+0,6/7+1/9	0,4/7+0,6/8+0,7/9
29	0,7/2+1/4+0,3/8	0,2/2+0,4/3+0,5/5+0,6/7
30	0,6/7+0,8/8+0,9/9+1/10	0,6/4+0,8/5+1/8

Література до розділу 2.

1. Cordon O., Herrera F., A General study on genetic fuzzy systems // Genetic Algorithms in engineering and computer science, 1995. – P. 33-57.
2. Kosko B. Fuzzy systems as universal approximators // IEEE Transactions on Computers, vol. 43, No. 11, November 1994. – P. 1329-1333.
3. Zadeh L.A. Fuzzy sets as a basis for a theory of possibility // Fuzzy Sets and Systems. - 1978. - Vol.1, №1.
4. Виленский П.Л., Лившиц В.Н., Орлова Е.Р., Смоляк С.А. Оценка эффективности инвестиционных проектов. М., Дело, 1998.

5. Заде Л. (1976) Понятие лингвистической переменной и ее применение к принятию приближенных решений, М., Мир.
6. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.: Мир, 1976.
7. Кофман А., Хил Алуха Х. Введение теории нечетких множеств в управлении предприятиями, Минск, Вышэйшая школа, 1992.
8. Кравец А.С. Природа вероятности, М., Мысль, 1976.
9. Круглов В.В., Дли М.И. Интеллектуальные информационные системы: компьютерная поддержка систем нечеткой логики и нечеткого вывода. – М.: Физматлит, 2002.
10. Леоленков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб., 2003.
11. Масалович А. Нечеткая логика в бизнесе и финансах. – www.tora-centre.ru/library/fuzzy/fuzzy-.htm
12. Недосекин А.О., Воронов К.И. Анализ риска инвестиций с применением нечетких множеств // Управление риском, 2000, №1.
13. Недосекин А.О., Максимов О.Б. Анализ риска банкротства предприятия с применением нечетких множеств // Вопросы анализа риска, 1999, № 2-3.
14. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М., 2004.
15. Смоляк С.А. Учет специфики инвестиционных проектов при оценке их эффективности // Аудит и финансовый анализ, 1999, №3.

3. АЛГЕБРА ЛОГІКИ

Своїм існуванням ця наука зобов'язана англійському математикові Джорджу Булю, який досліджував **логіку висловів** (початок XIX в).

Вислови будуються над множиною {B,,,, 0, 1}, де *B* – не порожня множина, над елементами якої визначено три операції:

$\neg$ *заперечення* – унарна операція, тобто операція з однією змінною.

$\wedge$ *кон'юнкція* – бінарна операція, тобто операція з двома змінними.

$\vee$ *диз'юнкція* – бінарна операція.

а також дві константи — логічний нуль **0** і логічна одиниця **1.**

Об'єкти з двома можливими станами характеризуються булевими змінними, які здатні приймати лише два різні значення. Для позначення цих двох значень зазвичай використовуються цифри 0 і 1 або букви Б (БРЕХНЯ) і І (ІСТИНА). Д. Буль позначив це як true і false. Тому частіше всього використовуються саме такі слова англійської мови.

Відносини між булевими змінними представляються булевими функціями, які подібно до числових функцій можуть . залежати від однієї, двох і, взагалі *n* змінних (аргументів). Запис $y = f(x_1, x_2, \ldots, x_n)$ означає, що y — функція аргументів $x_1, x_2, \ldots, x_n$. Найважливіша особливість булевих функцій полягає в тому, що вони, як і їх аргументи, приймають свої значення з двоелементної множини (0,1), або (І, Б}, тобто характеризуються одним з двох можливих станів.

Основними в двозначній логіці є наступні три функції.

Заперечення – функція $y = f(x)$ що приймає значення 1, коли x = 0, і значення 0, коли x = 1; вона позначається $y = \bar{x}$ (читається «не x»).

Диз'юнкція – функція, що приймає значення 0 тоді і тільки тоді, коли обидва аргументи мають значення 0; вона позначається $y = x_1 \vee x_2$ (читається «x_1 *або* x_2»). В інших випадках вона дорівнює 1.

Кон'юнкція – функція, що приймає значення 1 тоді і тільки тоді, коли обидва аргументи мають значення 1; вона позначається $y = x_1 \wedge x_2$ (читається «x_1 *і* x_2»). В інших випадках вона дорівнює 0.

3.1. Логічні операції

Легко показати, що на даній множині *B* можна задати чотири унарні і шістнадцять бінарних відносин, представлених в таблицях 3.1. та 3.2., проте всі вони можуть бути отримані через суперпозицію (можливі варіанти комбінацій) трьох вибраних операцій.

Спираючись на цей математичний інструментарій, логіка висловів вивчає вислови і предикати. Також вводяться додаткові операції, такі як еквівалентність «$\leftrightarrow$» («тоді і тільки тоді, коли»), імплікація «$\rightarrow$» («отже»), складання по модулю два «$\oplus$» («що виключає або»), штрих Шеффера, стрілка Пірса та інші.

Логіка висловів послужила основним математичним інструментом при створенні комп'ютерів. Вона легко перетвориться в бітову логіку: істинність вислову позначається одним бітом (0 – БРЕХНЯ, 1 – ІСТИНА); тоді операція $\neg$ набуває сенсу віднімання з одиниці; $\vee$ – складання не по модулю; & – множення; $\leftrightarrow$ – рівність; $\oplus$ – в буквальному розумінні складання по модулю 2 (АБО, що виключає – XOR); $|$ – не перевищення суми над 1 (тобто A $|$ B = (A + B) <= 1).

Згодом булева алгебра була узагальнена від логіки висловів шляхом введення характерних для логіки висловів аксіом. Це дозволило розглядати, наприклад, логіку, потрійну логіку (коли є три варіанти істинності вислову: «істина», «брехня» і «не визначено») і ін.

Таблиця 3.1

Унарні логічні операції

x	g_1 ($\neg$)	g_2 (=)	g_3 (1)	g_4 (0)
0	1	0	1	0
1	0	1	1	0

g_2(x) — тотожна функція (**$g_2(x)$=** x)

g_1(x) — заперечення / негація (**$g_1(x)$=** $\neg$x = x')

Таблиця 3.2

Бінарні логічні операції

x	y	f_1 ($\wedge$)	f_2 ($\vee$)	f_3 ($\equiv$) $x_1 \approx x_2$	f_4 ($\oplus$)	f_5 ($\subset$)	f_6 ($\supset$) $x_1 \to x_2$	f_7 ($\downarrow$)	f_8 ($\mid$)
0	0	0	0	1	0	1	1	1	1
0	1	0	1	0	1	0	1	0	1
1	0	0	1	0	1	1	0	0	1
1	1	1	1	1	0	1	1	0	0
x	y	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	0	0	0	1	1	0	0	1	0
0	1	0	1	1	0	0	1	1	0
1	0	1	0	0	1	1	0	1	0
1	1	0	0	0	0	1	1	1	0

Тут **0** і **1** – тотожні нулю і одиниці відповідно

$f_1(x, y)$ – кон'юнкція ($f_1(x, y)$= x&y = xy = min(x, y))

$f_2(x, y)$ – диз'юнкція ($f_2(x, y)$= xy = max(x, y))

$f_3(x, y)$ – еквівалентність ($f_3(x, y)$= xy)

$f_4(x, y)$ – сума по модулю два ($f_4(x, y)$= xy)

$f_5(x, y)$ – імплікація від у до х ($f_5(x, y)$= xy)

$f_6(x, y)$ – імплікація від х до у ($f_6(x, y)$= xy)

$f_7(x, y)$ – стрілка Пірса = функція Таггера = функція Вебба («антидиз'юнкція») ($f_7(x, y)$= xy).

$f_8(x, y)$ – штрих Шеффера («антикон'юнкція») ($f_8(x, y)$= xy)

$f_9(x, y)$, $f_{10}(x, y)$ – інверсії імплікацій f_5 і f_6
f_{11} - f_{14} – функції тільки одного аргументу
f_{15}, f_{16} – тотожність.

Всі перелічені вище функції називаються **логічними зв'язками**.

3.2. Властивості логічних операцій

1. Комутативність: $xy = yx$ $\circ \in \{\&, \vee, \oplus, \sim, |, \downarrow\}$.
2. Ідемпотентність: $xx = x$ $\circ \in \{\&, \vee\}$.
3. Асоціативність: $(xy)\circ z = x\circ(yz)$ $\circ \in \{\&, \vee, \oplus, \sim\}$.
4. Дистрибутивність кон'юнкцій і диз'юнкції щодо диз'юнкції, кон'юнкції і суми по модулю два відповідно:
   - $x\wedge(y\vee z) = (x\wedge y)\vee(x\wedge z)$,
   - $x\vee(y\wedge z) = (x\vee y)\wedge(x\vee z)$,
   - $x\wedge(y\oplus z) = (x\wedge y)\oplus(x\wedge z)$.
5. Закони де Моргана:
   - $\neg(x\wedge y) = (\neg x)\vee(\neg y)$,
   - $\neg(x\vee y) = (\neg x)\wedge(\neg y)$.
6. Закони поглинання:
   - $x\wedge(xy)= x$
   - $x\vee(xy)= x$.
7. Інші (1):
   - $x\wedge(\neg x) = x\wedge 0 = x\oplus x = 0$.
   - $x\vee(\neg x) = x\vee 1 = x\sim x = x\to x = 1$.
   - $x\vee x = x\wedge x = x\wedge 1 = x\vee 0 = x\oplus 0 = x$.
   - $x\oplus 1 = x\to 0 = x\sim 0 = x|x = x\downarrow x = \neg x$.
   - $\bar{\bar{x}} = x$.
8. Інші (2):
   - $x\oplus y = (x\wedge\bar{y})\vee(\bar{x}\wedge y) = (x\vee y)\wedge(\bar{x}\vee\bar{y})$.
   - $x\sim y = \overline{x\oplus y} = (x\wedge y)\vee(\bar{x}\wedge\bar{y}) = (x\vee\bar{y})\wedge(\bar{x}\vee y)$
   - $x\to y = \bar{x}\vee y = ((x\wedge y)\oplus x)\oplus 1$.
9. Інші (3) (Доповнення законів де Моргана):
   - $x\mid y = \neg(x\wedge y) = \neg x\vee\neg y$.
   - $x\downarrow y = \neg(x\vee y) = \neg x\wedge\neg y$.

10.Закони склеювання: $(a\vee\bar{b})\wedge(a\vee b) = a$

$$(a\wedge\bar{b})\vee(a\wedge b) = a$$

11.Ще два очевидних закони

$$x \vee \overline{x} = 1; \qquad x \wedge \overline{x} = 0.$$

3.3. Правила перетворення логічних формул, які мають більше 2-х змінних

З простих формул можна конструювати складніші. Позначатимемо довільні формули як функції від деякого числа аргументів, наприклад $f(A)$ або $g(x_1, x_2 \dots x_n)$. У дужках перераховуються змінні, що входять у формулу або цілі фрагменти формули, що є єдине ціле, Якщо список змінних неважливий, то писатимемо багатокрапку, наприклад $f(\dots)$. Ці дужки важливі для того, щоб відрізнити два випадки:

1. Окрема вільна змінна f, що має певну область значень (наприклад, true і false, якщо змінна булева).
2. Ціла формула $f(\dots)$, в яку може входити багато змінних (або одна, або жодній), кожна з яких, в принципі, може мати свою область значень.

Для багатьох розділів математики прийняті правила побудови формул по певних правилах. Для булевої алгебри ці правила такі:

1. true – булева формула (проста).
2. false – булева формула (проста).
3. Одна окрема булева змінна – формула (проста). Якщо булева формула $f(\dots)$, то з неї можна побудувати формули:
$$(f(\dots))$$
$$\sim(f(\dots)).$$
4. Якщо є дві булеві формули (можливо однакові) $f(\dots)$ і $g(\dots)$, то з них можна побудувати формули:

$$(f(\dots) \vee g(\dots))$$
$$(f(\dots) \oplus g(\dots))$$
$$(f(\dots) \& g(\dots))$$
$$(f(\dots) \Leftrightarrow g(\dots))$$
$$(f(\dots) \Rightarrow g(\dots))$$
$$(f(\dots) \downarrow g(\dots))$$
$$(f(\dots) | g(\dots))$$
$$(f(\dots) \Leftarrow g(\dots))$$
$$(\mathrm{f}(\dots) \prec g(\dots))$$
$$(f(\dots) \succ g(\dots))$$

Таким чином, формули будуються послідовно, крок за кроком від простих формул ("атомів") до все більш складних формул ("молекул"), які усередині себе містять простіші. Кожну формулу, яка використовується як складова частина для побудови складнішої формули, називатимемо підформулою. Вище за підформулу позначені як $f(\dots)$ і $g(\dots)$.

Зовнішні дужки в правилах покликані дотримати той порядок операцій, який відповідає порядку побудови формули з атомів. Зайві дужки можна прибрати. Процес прибирання і відновлення дужок виглядає точно так, як і в шкільній алгебрі: залежно від порядку операцій. Коли дужки відсутні, то в першу чергу виконується операція "~" (перший, вищий пріоритет), потім "&" (другий пріоритет), потім "∨" і "⊕" (третій пріоритет), потім всі останні (четвертий, нижчий пріоритет). Операції одного пріоритету виконуються зліва направо.

Цей порядок пріоритетів поширений в найбільш сучасній математичній і комп'ютерній літературі. Щоб його запам'ятати, звернете увагу на схожість з арифметичними операціями зміни знаку ("-" аналог "~"), множення (аналогічна операція "&" іноді називається логічним множенням), складання (аналогічні операції ∨ і ⊕) і операції порівняння (значки операцій ⇔і ⇒) зовні схожі на значки операцій порівняння =, <, ≥ і ін.).

Всі булеві формули мають одну дуже важливу загальну властивість:

Істинність булевої формули залежить тільки від істинності значень змінних.

Можете проглянути ще раз визначення кожної з булевих операцій і переконатися: всюди обчислення істинності подібних формул зводиться до розгляду булевих величин, які відображають істинність яких-небудь висловів. Величини просто підставляються в таблицю, з таблиці виходить відповідь, і більше ця відповідь ні від чого не залежить.

Не має значення, які конкретно вислови аналізуються, якщо ми вже визначили їх істинність. Як наслідок, ми можемо складати таблиці істинності і для складніших булевих формул.

Наприклад, розглянемо формулу

$$(\sim X \Leftrightarrow Y)\ \&\ (X \oplus Y) \vee \text{false}$$

Складаємо таблицю, виписавши в ній всі можливі комбінації для істинності змінних:

X	Y	$(\sim X \Leftrightarrow Y)\ \&\ (X \oplus Y) \vee \text{false}$
false	false	
false	true	
true	false	
true	true	

Визначимо порядок дій, згідно дужкам і пріоритетам:

$$\begin{array}{ccccccccc} 1 & & 2 & & 4 & & 3 & & 5 \\ (\sim X & \Leftrightarrow & Y) & \& & (X & \oplus & Y) & \vee & \text{false} \end{array}$$

Для кожної операції визначаємо, звідки узяти операнди. Операнд може бути узятий

- безпосередньо з формули, якщо у формулі на місці операнда стоїть константа true або false;
- з таблиці істинності, якщо у формулі на місці операнда стоїть змінна;
- зі змінної, в якій був збережений проміжний результат, якщо у формулі на місці операнда стоїть складна формула.

Для кожної операції виконуємо обчислення істинності, згідно таблиці істинності цієї операції. Результат позначаємо проміжною змінною. Результат останньої операції буде результатом обчислення всієї формули. В даному випадку порядок дій такий:

1. Обчислити ~X, позначити як $R1$.
2. Обчислити $R1 \Leftrightarrow Y$, позначити як $R2$.
3. Обчислити $X \oplus Y$, позначити як $R3$.
4. Обчислити $R2$ & $R3$, позначити як $R4$.
5. Обчислити $R4 \vee$false, позначити як $R5$.
6. Прийняти $R5$ як результат обчислень.

Виконаємо ці дії для конкретного рядка таблиці істинності, де X = true, Y = false:

1. Обчислити ~X, позначити як $R1$ = false.
2. Обчислити $R1 \Leftrightarrow Y$, позначити як $R2$ = true.
3. Обчислити $X \oplus Y$, позначити як $R3$ = true.
4. Обчислити $R2$ & $R3$, позначити як $R4$ = true.
5. Обчислити $R4 \vee$false, позначити як $R5$ = true.
6. Прийняти $R5$ = true як результат обчислень.

Після цього можна заповнити один елемент таблиці істинності:

Аналогічно заповнюємо осередки, що залишилися:

X	Y	(~X ⇔Y) & (X Y) ∨false
false	false	false
false	true	true
true	false	true
true	true	false

В процесі обчислення можлива деяка оптимізація. Замість повторення одних і тих же дій можна обмежитися єдиним обчисленням, після чого використовувати значення, позначене в проміжній змінній. Для прикладу розглянете формулу ($A \vee$~B) & ~($A \vee$~B) & ~~($A \vee$~B). Складіть для неї інструкцію по обчисленню і оптимізуйте її так, щоб не обчислювати ($A \vee$~B) тричі.

3.4. Вислови
3.4.1. Вислови і операції над ними

Наведені вище функції, які можуть бути використані як елемент логічного вислову. На їх підставі можна розраховувати результат деяких логічних посилок.

Під *висловом* розуміють граматично правильну оповідну пропозицію, про яку можна сказати, що воно або *істинно*, або *помилково*, наприклад: «Київ – столиця України», «Париж – столиця Росії». Перший вислів *істинний*, другий – *помилковий*.

Візьмемо два *прості* вислови:

A = «На вулиці йде дощ», B = «Над моєю головою розкрита парасолька».

За допомогою п'яти логічних зв'язок можна утворити наступні *складні* вислови:

1) *заперечення*: $\underline{A}$ = «На вулиці не йде дощ»;

2) *диз'юнкція*: $\underline{A} \vee B$ = «На вулиці *не* йде дощ *або* над моєю головою розкрита парасолька»;

3) *кон'юнкція*: $A \wedge \underline{B}$ = «На вулиці йде дощ *і* над моєю головою *не* розкрита парасолька»;

4) *імплікація*: $A \to B$ = «*Якщо* на вулиці йде дощ, *то* над моєю головою розкрита парасолька»

5) *еквівалентність*: $B \sim A$ = «Над моєю головою розкрита парасолька *тоді і тільки тоді, коли* на вулиці йде дощ».

Інші логічні зв'язки, відомі нам за логікою Буля, в логіці вислови не використовуються. Тепер зробимо з приводу кожної з п'яти вказаних зв'язків невеликі зауваження.

Заперечення. Вислів A по-іншому можна прочитати так: «*Істинне те, що на вулиці йде дощ*». *Тому, якщо* $A = 0$, *то це означає, що на вулиці не йде дощ*. Вислів, що доповнює $\underline{A}$ також орієнтується на дійсний вислів, тобто його слід розуміти як «*Істинне те, що на вулиці не йде дощ*». Тоді $\underline{A} = 1$ позначатиме ту ж саму ситуацію, що і у попередньому випадку, тобто відсутність дощу.

Диз'юнкція. У нашому конкретному прикладі диз'юнкція двох висловів A і B, в принципі, може мати на увазі і кон'юнкцію цих же висловів. Проте часто граматичний союз *або* не включає союз *і*. Хай дані два інших вислови:

P = «Петро знаходиться в кінотеатрі»

Q = «Петро знаходиться в басейні».

Якщо для нас не так важливо, де знаходиться Петро, то ми, звичайно, можемо використовувати союз *або* з включеним в нього союзом *і*, формально записавши:

$P \vee Q$= «Петро знаходиться в кінотеатрі *або/і* в басейні».

Але якщо потрібно точно встановити, де знаходиться Петро, то ми зобов'язані виключити випадок одночасної присутності Петра в кінотеатрі і басейні, тобто формально записати:

$(P \vee Q) \wedge (\underline{P \wedge Q})$.

Подібні вислови називаються *строгою диз'юнкцією*, яка означає «*або* P, *або* Q, *але не* P *і* Q одночасно». І хоча, з погляду логіки Буля, ця логічна операція рівносильна операції *симетричної різниці:*

$P + Q = (P \vee Q) \wedge (\underline{P} \vee \underline{Q})$,

історично склалося так, що символ « + » у логіці висловів не використовується.

Кон'юнкція. Логічний союз і необов'язково повинен представлятися через граматичний союз *і*. Зокрема, вище приведений вираз можна прочитати трохи інакше:

$A \wedge \underline{B}$ = «На вулиці йде дощ, *а* над моєю головою не розкрита парасолька»;

Союзи *а* і *але* по змісту часто співпадають з союзом *і*, тому вони використовуються в складних кон’юнктивних пропозиціях. Проте мовна ситуація може стати такій, що союз *і* перестає грати роль *кон'юнкції*; приведемо дві пропозиції:

$P \wedge Q$= «Йому стало страшно і він убив людину».

$Q \wedge P$ =«Он убив людину і йому стало страшно».

Тут не комутативність двох простих пропозицій очевидна, оскільки ми маємо справу з прихованою *імплікацією,* коли одна проста пропозиція обумовлює інше.

$Q \to P$ = «Коли він убив людину, йому стало страшно».

Імплікація. Вислів типу «*якщо А, то В*» носить *пояснюючий* характер. Він як би роз'яснює нам, чому має місце подія *В* – тому що мала місце подія *А*. Ця властивість імплікації особлива цінно для логіки висловів, про що ми детально зупинимося в наступному підрозділі.

Пояснюючий характер імплікації тісно пов'язаний з *причинно-наслідковим відношенням*, при якому *А* виступає в ролі *причини*, а *В* – *слідства*. Причинно-наслідковий зв'язок між *А* і *В* граматично може бути оформлена пропозиціями: «*А* є достатньою підставою для *В*», «*В*, тому що *А*», «*В* за умови виконання *А*» і так далі Якщо під *А* і *В* розуміти колишні вислови, то результат причинно-наслідкового відношення можна оформити наступною таблицею істинності. Другий рядок таблиці говорить про відсутність причинно-наслідкового відношення між подіями *А* і *В*.

Таблиця

А	*В*	*А* → *В*	Результати
0	0	1	Залишуся сухим
1	0	0	Вимокну
0	1	1	Залишуся сухим
1	1	1	Залишуся сухим

Еквівалентність. Вислів «*А* еквівалентно *В*» може бути з успіхом замінено на «*А* рівно *В*», «*А* тотожно *В*», «*А* рівносильне *В*», «*А* тоді і тільки тоді, коли *В*» і так далі Оскільки еквівалентність виражається через кон'юнкцію двох імплікацій:

$A \sim B = (A \to B) \wedge (B \to A)$,

то це відношення часто виникає при одночасному виконанні двох умов: «з *А* слідує *В*» і «з *В* слідує *А*». Таким чином, при еквівалентності двох подій неможливо одному з них приписати роль тільки *причини, а іншому – тільки следствия. Наприклад, дві події:*

R = «Наростання анархії в суспільстві»

S = «Падіння авторитету влади»

є цілком подіями рівного порядку, оскільки причиною наростання анархії в суспільстві є падіння авторитету влади; і навпаки, падіння авторитету влади відбувається із-за наростання анархії в суспільстві. У даній ситуації безглуздо звинувачувати тільки владу в слабкості і некомпетентності або звинувачувати народ в несвідомості і недисциплінованості.

Події *R* і *S* утворюють логічне коло; їх називатимемо сильно зв'язаними подіями і виражати наступними тотожними формами:

R ~ S = (R Ù S) ~ (R Ú S) = (R Ú S) → (R Ù S).

Поняття «Сильної зв'язаності» співпадає з поняттям «еквівалентності», якщо мова йде про двох подіях. Але візьмемо, наприклад, добре відоме пояснення, на чому тримається Земля:

Земля (X) тримається на трьох китах (Y), кити (Y) тримаються на водах океану (Z), океан (Z) тримається на Землі (X).

Послідовність, куди входять три названі об'єкти *X, Y* і *Z*, теж утворюють логічне коло:

(X → Y) Ù (Y → Z) Ù (Z → X).

Проте відношення еквівалентності (бути взаємною опорою один для одного) між всіма трьома об'єктами, тобто

$$X \sim Y \sim Z$$

тут не виникає, та і не може виникнути, оскільки ми не стверджуємо, що Земля є безпосередньою опорою для китів (*X ~ Y*), або що кити є безпосередньою опорою для вод океану (*Y ~ Z*). Тому еквівалентність в даному випадку виявляється у вельми своєрідній формі:

(X Ù Y Ù Z) ~ (X Ú Y Ú Z) або

(X Ú Y Ú Z) → (X Ù Y Ù Z),

що можна тлумачити у разі операції еквівалентності як: одночасна поява всіх трьох опор відбудеться *тоді і тільки тоді, коли виникне хоч би одна з опор, і навпаки; для операції імплікації: якщо* виникне яка-небудь одна з опор, *то* це приведе до появи всіх трьох опор. Таким чином, *сильна зв'язаність*, або *логічне коло, є не логічна операція, а щось проміжне між причинно-наслідковим відношенням і відношенням еквівалентності. Подібні аномальні* відносини виникають нерідко, наприклад, між членами злочинної організації, де всі зв'язано круговою порукою, чиновники часто пускають людину по колу і він не може знайти крайнього з них і так далі. Всі ці соціальні явища випадають з розгляду математичної логіки.

Хай x_1 і x_2 — деякі вислови, які можуть бути істинними (1) або помилковими (0), наприклад: «Я піду в театр» (x_1) і «Я зустріну друга» (x_2). Диз'юнкцією $y = x_1 \vee x_2$ є складний вислів «Я піду в театр *або* зустріну друга», а кон'юнкцією $y = x_1 \wedge x_2$ – вислів «Я піду в театр *і* зустріну друга».

Отже, вислови можна розглядати як двійкові змінні, а зв'язки «не», «або», «і», за допомогою яких утворюються складні вислови, – як операції над цими змінними. У алгебрі висловів використовуються ще дві операції: *імплікація* $x_1 \to x_2$ відповідна зв'язці «*якщо, то*» і е*квіваленція* $x_1 \approx x_2$ відповідна зв'язці «*якщо і тільки якщо*».

У нашому прикладі імплікацією буде вислів: «Якщо я піду в театр, то зустріну друга», а еквіваленцією – «Я піду в театр, якщо і тільки якщо зустріну друга». Як видно з таблиць, імплікація висловів помилкова тільки у разі, коли перше з простих висловів істинно, а друге помилково. Еквіваленція є дійсним висловом, якщо обидва прості вислови істинні або помилкові одночасно.

Позначивши буквами прості вислови, можна представити складний вислів формулою за допомогою відповідних зв'язок. Наприклад, вислову «Якщо тиск масла на кульку клапана більше зусилля його пружини (x_1), то масло відкриває клапан (x_2) і частково перетікає з нагнітальної порожнини у впускну порожнину (x_3) відповідає формула $x_1 \to x_2 x_3$.

3.4.2. Побудова доказів в логіці висловів

Логіка — це наука про способи доказу. З'ясуємо, в чому, власне, полягає відмінність в побудові доказів в логіці висловів і логіці Буля.

У булевій логіці всі докази будувалися на *відношенні еквівалентності*. Навіть якщо в множинних виразах і фігурувало відношення включення, що є окремим випадком *відношення порядку,* то його ми переводили в тотожність. Дві логічні функції вважалися еквівалентними, якщо вони давали на відповідних наборах аргументів абсолютно однакові значення нулів і одиниць. При використанні формального запису логічних виразів окремі ланки ланцюга будь-якого доказу там були зв'язані через символ рівності « = ». Відношення еквівалентності задовольняє трьом законам:

рефлективності: $A = A$;
симетричності: якщо $A = B$, то $B = A$;
транзитивності: якщо $A = B$ *і* $B = C$, *то* $A = C$.

У логіці висловів докази будуються на *відношенні порядку,* тобто на відношенні, яке існує між причиною і наслідком. Тут вже окремі ланки ланцюга доказу зв'язані символом імплікації. Проте символ імплікації «→» при логічному висновку ми замінюватимемо на символ « ⇒ », подібно до того, як в логіці Буля використовуються два символи еквівалентності – « ~ » і « = ». Символ « ~ » є *об'єктним, а символ « = » – суб'єктним.*

Таким чином, слід розрізняти *мову логіки висловів* і *метамову дослідника*. Щоб уникнути плутанини, введемо ще два *метасимволи: замість об'єктної кон'юнкції « ∧ » використовуватимемо суб'єктний символ метаконъюнкции – «, », а замість об'єктної диз'юнкції «∨» – суб'єктну метадизъюнкцию « ; ». Тоді твердження, яке потрібно довести, оформляється у вигляді наступного причинно-наслідкового відношення:*

$$P_1, P_2, \dots , P_{n-1}, P_n \Rightarrow C \tag{3.1}$$

де Pi — *посилка* (*причина*), C — *висновок* (*слідство*). Читається: «Якщо посилки $P_1, P_2, \dots , P_{n-1}, P_n$ істинні, то й висновок C теж істинний» або, по-іншому: «Якщо причини $P_1, P_2, \dots , P_{n-1}, P_n$ мали місце, то матиме місце і слідство C».

Щоб не сплутати об'єктний вислів (пропозиція) з суб'єктним висловом, справедливість якого ми маємо намір встановити, умовимося пропозиції типу (3.1) називати *клаузой* (*clause — пропозиція*). Клауза — це метаречення, в якому використано відношення порядку, оформлене через символ метаімплікації «⇒». Як і відношення еквівалентності відношення порядку задовольняє трьом законам

рефлективності: A ⇒ A;

антисиметричності: якщо *A Þ B, то* $\underline{B}$ Þ $\underline{A}$*; ;*

транзитивності: якщо *A Þ B і B Þ C, то A Þ C.*

Відношення порядку припускає виконання закону антисиметричності, який записується ще і так:

якщо *A Þ B і B Þ A, то A = B;*

якщо *A = B, то A Þ B і B Þ A*

Клауза є саме формальний запис доводжуваної пропозиції. Замість букв в ній можна підставити об'єктні вислови, і тоді клауза наповнюється конкретним змістом, який вже іменується *семантикою* або *легендою*. Приклад клаузи:

A → B, A Þ B.

Якщо прийняти, що *A* = «виблискувала блискавка», *B* = «пролунав грім», то можна скласти наступну легенду: «Відомо, що коли виблискувала блискавка, то після цього пролунає грім. Блискавка виблискувала. Отже, повинен пролунати і грім».

Над суб'єктом, який формулює метаречення, може стояти інший суб'єкт, для якої пропозиції першого суб'єкта виявляться об'єктними. Тоді клаузу (3.1) другий суб'єкт або метасуб’єкт запише для себе наступним логічним виразом:

P_1 Ù P_2 Ù ... Ù P_{n-1} Ù P_n → C.

Перетворимо цей вираз в диз’юнкт, отримаємо:

$$\underline{P}_1 \lor \underline{P}_2 \lor \ldots \lor \underline{P}_{n-1} \lor \underline{P}_n \lor C.$$

Звідси знаходимо:

$$(P_1 \land P_2 \land \ldots \land P_{n-1}) \to (\underline{P}_n \lor C).$$

Тому клауза (3.1) може бути представлена в іншій *еквівалентній* формі:

$$P_1, P_2, \ldots, P_{n-1} \Rightarrow \underline{P}_n ; C \qquad (3.2)$$

Через комутативність кон'юнкції на місці посилки P_n може опинитися будь-яка інша, причому не одна. Наприклад, клауза:

$$P_1, P_2, P_3, P_4 \Rightarrow C_1; C_2 ; C_3.$$

може бути перетворена в іншу еквівалентну форму:

$$P_4, \underline{C}_2, P_1, \underline{C}_1 \Rightarrow \underline{P}_1 ; C_3 ; \underline{P}_2. \qquad (3,3)$$

Проте клауза (3.1) в порівнянні з (3.2), (3.3) і іншими подібними формами має певні переваги і, зокрема, використовується в *мові логічного програмування ПРОЛОГ. Її називають хорновскою. Довільну клаузу завжди можна звести до хорновского вигляду шляхом еквівалентних перетворень.*

Якщо символ метаімплікації « ⇒ » клаузи (3.2) змістити в крайнє ліве положення, то вона перетвориться на *тавтологію*; якщо ж його змістити в крайнє праве положення, то – в *суперечність*:

$1 \Rightarrow \underline{P}_1 ; \underline{P}_2 ; \ldots ; \underline{P}_{n-1} ; \underline{P}_n ; C$ — *тавтологія*,

$P_1, P_2, \ldots, P_{n-1}, P_n, \underline{C} \Rightarrow 0$ — *суперечність*.

Як і в логіці Буля, в логіці висловів існують *аксіоматичний* і *конструктивний* підходи доказів логічних виразів. Аксіоматична побудова логіки висловів полягає в тому, щоб спробувати вичленувати з нескінченного числа дійсних клауз *незалежну систему аксіом*, за допомогою якої можна було б встановити справедливість будь-яких інших клауз.

Розглянемо ще приклади застосування алгебри логіки для визначення деяких висловлювань.

1. Затримані підозрювані в злочині Браун, Джон і Сміт. Один з їх говорить правду, інший - напівправду, третій - брехню. Приведемо їх свідчення.

Браун:"Я зробив це, Джон не винен."

Джон:"Браун не винен, злочинець - Сміт."

Сміт:"Я не винен, винен Браун."

Знайти злочинця, якщо відомо, що він один.

Рішення.

Введемо позначення:

B – винен Браун;

C – винен Сміт;

D – винен Джон.

Тоді умова завдання буде виражена двома рівняннями:

1)$BD'+B'C+B'C' = 1$ (свідчення підозрюваних, одне з них істинно);

2)$B'C'+B'D'+C'D' = 1$ (злочинець єдиний).

Складемо функцію, що об'єднує ці висловлювання

$$M = (BD'+B'C+B'C')(B'C'+B'D'+C'D') = B'CD'+BC'D' = 1.$$

BC'D' відпадає, оскільки інакше Браун і Сміт обидва говорять правду. Отже, істинно $B'CD'$, тобто злочинець – Сміт, він ще і брехун. Джон говорить правду, Браун – напівправду. До речі, відсіяти $BC'D'$ можна було на першому етапі, оскільки з умов завдання виходить $BD'+BC' = 0$, тому

$$M = B'C(B'C'+B'D'+C'D')= B'CD'.$$

2. Якщо в експедицію поїде Кавунів, то поїде або Бруквин або Вишневський. Якщо поїдуть Кавунів з Вишневським, то поїде і Бруквин. Хто відправиться в експедицію?

Рішення.

A - поїде Кавунів.

B - поїде Бруквин.

W - поїде Вишневський.

1)$A \rightarrow (B+W)$;

2)$AW \rightarrow B$.

$$M = (A' + B + W)(A'W' + B) = 1$$

$$\text{Отже } 1 + B[A'(W' + 1) + W] = 1$$

$$\text{Тоді } B[A'+W] = 0 \text{ або } BA'+BW = 0$$

тобто хибним є твердження, що поїде Бруквин і не поїде Кавунів або поїдуть Бруквин і Вишневський. Висновок: поїдуть Кавунів і Бруквин.

3.3. Індивідуальне завдання №3.

Варіант 1

1. Підстановкою у формулу $a \vee b$ змінних запишіть нові формули і спростіть їх, якщо це можливо:

a) $a = \bar{x}y, \quad b = z;$

b) $a = xy,. \quad b = x\bar{y};$

c) $a = x,. \quad b = xy;$

d) $a = x,. \quad b = \bar{x}y;$

e) $a = xy,. \quad b = c \vee d, \quad c = xz, \quad d = y\bar{z}.$

8. Перетворіть формули до такого вигляду, щоб операції заперечення застосовувалися тільки до логічних змінних:

a) $\overline{xy \vee \bar{z}};$

b) $x(xy \vee \overline{yz} \vee \overline{y \vee \bar{z}}v.$

16. Дані прості вислови: x_1—«йде дощ», x_2 – «дуже жарко».

a) Запишіть формулу складного вислову «Невірно, що йде дощ і дуже жарко».

b) Перетворіть формулу згідно із законом де Моргана і складіть відповідний вислів.

c) Переконаєтеся в тотожності початкового і перетвореного висловів.

Варіант ***2***

2. Запишіть таблиці відповідності для наступних формул:

a) $x\bar{x}$

b) $xy \vee \bar{x}$

c) $(p \vee q)\left(\overline{p \vee \bar{q}}\right)$

d) $\overline{x \vee y}.$

9. Переконаєтеся за допомогою таблиць відповідності в справедливості виразів для імплікації і еквіваленції:

a) $x_1 \to x_2 = \bar{x}_1 \vee x_2$

b) $x_1 \approx x_2 = x_1 x_2 \vee \bar{x}_1 \bar{x}_2 = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2);$

c) $x_1 \approx x_2 = (x_1 \to x_2)(x_2 \to x_1).$

17. Мандрівник зупинився у розвилки доріг, що ведуть в пункти *A* і *B*, і йому потрібно з'ясувати, в якій саме пункт веде кожна з доріг. Двоє людей, які знаходилися у розвилки заявили, що вони можуть відповісти тільки на одне запитання і що один з них завжди правдивий, а інший брехун. Яке питання повинне задати мандрівник, щоб у будь-якому випадку відповідь на нього містила необхідну інформацію?

a) Вирішите задачу шляхом безпосередніх міркувань без застосування алгебри логіки.
b) Представте ситуацію у вигляді складного вислову, складеного з простих.
c) Запишіть відповідну формулу і таблицю відповідності.
d) По таблиці відповідності сформулюйте шукане питання.

Варіант ***3***

3. Перевірте за допомогою таблиць відповідності наступну тотожність:

a) $\overline{x \vee y} = \overline{xy}$;
b) $x(x \vee y) = x$;
c) $x \vee \overline{x}y = x \vee y$.

14. Запишіть формули для наступних висловів, позначивши буквами вхідні в них прості вислови:

a) Тиск падає і система не працює.
b) Обчислення виконані точно або конструкція не досконала.
c) Проект розробив Андрій або Петро, а експеримент виконав Іван.
d) Якщо буде гарна погода, ми відправимося на стадіон або підемо за грибами.
e) Програма може бути виконана, якщо і тільки якщо матеріали поступають своєчасно.
f) Якщо я поїду на автобусі, то запізнюся на роботу, або я скористаюся таксі.
g) Андрій допомагає Петру або Петро допомагає Андрію, або вони допомагають один одному.

18. Вислів є логічно істинним, якщо відповідна йому формула тотожно рівна одиниці, і логічно помилковим, якщо формула рівна нулю. Визначите за допомогою таблиць відповідності, яким висловам відповідають приведені нижче формули (істинним, помилковим або ні тим не іншим):

a) $p \approx p$
b) $p \to \overline{p}$
c) $(p \vee q) \approx pq$
d) $(p \to \overline{q}) \to (q \to \overline{p})$
e) $(p \to q) \to p$
f) $((p \to q) \to p) \to p$;
g) $\overline{p \vee q} \approx \overline{p}\,\overline{q}$.

Варіант **4**

5. Спростіть наступні формули:

a) $\overline{x}yz \vee x\overline{y}\,\overline{z} \vee xy\overline{z}$;
b) $xy \vee z \vee \overline{xy \vee z}(zv \vee x)$;
c) $x\overline{y}\,\overline{z} \vee xy\overline{z} \vee \overline{x}yz \vee xyz$;

d) $(x \vee y)(\overline{xy} \vee z) \vee \bar{z} \vee (x \vee y)(u \vee v)$.

15. Запишіть формулу, відповідну вислову: «Програма буде виконана тоді і тільки тоді, коли закінчаться випробування і показники будуть задовільні; якщо програма не буде виконана, співробітники не отримають премію або будуть переглянуті технічні умови ».

19. При x_1=1; x_2=0; x_3=0 і x_4=1 знайдіть значення кожною з наступних функцій:

a) $\overline{x_1 \vee x_2} \vee x_3 x_4$;
b) $x_1 x_2 \vee x_3(x_1 \vee x_4) \vee x_4(x_2 \vee x_3)$;
c) $x_1 \to (x_2 \vee x_3)$;
d) $(x_1 \vee x_2) \approx x_2 \overline{x_3}$;
e) $x_1 \overline{x_2} \to (x_2 \approx x_3)$;
f) $x_1 x_2 \to (x_3 \to x_2 \overline{x_4})$.

Література до розділу 3

1. Клини С. К., Введение в метаматематику, пер. с англ., М., 1957, §§ 20, 23;
2. Генцен Г., Исследования логических выводов, пер. с. нем., в кн.: Математическая теория логического вывода, М., 1967;
3. Карри Х. Б., Основания математической логики, пер. с англ., М., 1969.
4. Акимов О.Е.Алгебра логики. – http://www.sceptic-ratio/narod.ru/ma.htm.

4. МІНІМІЗАЦІЯ БУЛЕВИХ ФУНКЦІЙ

Нагадаємо, що під булевою алгеброю мається на увазі розділ математичної логіки, в якому для побудови логічних формул використовуються тільки операції НЕ, І, АБО.

За допомогою булевих функцій описується робота кібернетичних систем типу перемикача – лічильників, дешифраторів, диспетчерів, і так далі Розробка системи починається з вербального опису завдань, які ставляться перед нею і умов, в яких вона функціонуватиме. Далі будується логічна модель, що відображає роботу системи.

Модель описується логічними функціями, які задаються у вигляді таблиць істинності. Самі таблиці формуються в процесі розробки по вербальному (змістовному) опису принципів роботи системи. Математичний опис роботи системи є формулами алгебри логіки. Кожна змінна формули відображає стан окремих елементів системи. Тому, чим більше за змінні містить формула, чим складніше вона, тим система, що складніше розробляється. Одна і та ж система може описуватися різними по складності формулами, а, отже, може бути реалізована різними по складності способами.

За всіх інших рівних умов, чим простіше система, тим краще. Тому прагнуть отримати найбільш прості логічні формули (функції), тобто оптимізувати створювані системи за критерієм складності.

Для розуміння цих положень розберемо наступну задачу.

Приймальна комісія у складі трьох членів комісії і одного голови вирішує долю абітурієнта більшістю голосів. У разі рівного розподілу голосів більшість визначається голосом голови Побудувати автомат для таємного голосування.

Виразимо умову завдання у вигляді таблиці істинності (табл. 4.1). Тут x_4 – голова, $x_3 \ldots x_1$ - члени комісії, y – вихідна функція: $y = 1$, якщо абітурієнт успішно пройшов співбесіду, $y = 0$, якщо – ні.

Таблиця 4.1.

x_4	x_3	x_2	x_1	y	x_4	x_3	x_2	x_1	y
0	0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	0	1	0	1
0	0	1	1	0	1	0	1	1	1
0	1	0	0	0	1	1	0	0	1
0	1	0	1	0	1	1	0	1	1
0	1	1	0	0	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1

Кон'юнкції вхідних змінних, на яких функція приймає значення 1, називатимемо одиничними, або робочими наборами (РН). Набори, на яких функція приймає значення 0, називатимемо нульовими або забороненими наборами(ЗН).

Для того, щоб по таблиці істинності знайти функцію, достатньо виписати всі робочі набори і з'єднати їх знаком логічного додавання . Якщо змінна входить в РН одиницею, то вона записується в прямому вигляді, інакше – в інверсному. З табл. 4.1 отримуємо

$$
\begin{aligned}
y = \overline{x}_4 x_3 x_2 x_1 + x_4 \overline{x}_3 \overline{x}_2 x_1 + x_4 \overline{x}_3 x_2 \overline{x}_1 + x_4 \overline{x}_3 x_2 x_1 + \\
+ x_4 x_3 \overline{x}_2 \overline{x}_1 + x_4 x_3 \overline{x}_2 x_1 + x_4 x_3 x_2 \overline{x}_1 + x_4 x_3 x_2 x_1
\end{aligned}
$$

Даний вираз є досконалою диз'юнктивною нормальною формою (ДДНФ) булевої функції. Очевидно, що така функція є незручною для використання, тому її потрібно зменшити (мінімізувати).

Відома безліч методів мінімізації булевих функцій. Поширення набули методи Квайна–мак–Класьки, Вейча–Карно, Гарвардський метод, метод невизначених коефіцієнтів.

4.1. Метод невизначених коефіцієнтів

Математичною чіткістю і простотою відрізняється метод невизначених коефіцієнтів. Суть його полягає в представленні функції як диз'юнктивної нормальної форми (ДНФ) наступного вигляду:

$$
f(x_1, x_2, \ldots, x_n) = k_1^1 x_1 \vee k_1^0 \overline{x_1} \vee k_2^1 x_2 \vee k_2^0 \overline{x_2} \vee \ldots \vee k_{12\ldots N}^{00\ldots 0} \overline{x_1}, \overline{x_2} \ldots \overline{x_N}. \quad (4.1)
$$

де k_i – коефіцієнти, які при входженні відповідної кон'юнкції в мінімальну ДНФ рівні одиниці, інакше рівні нулю, $i = 1, 2, \ldots N$.

У правій частині виразу (4.1) представлені всі можливі елементарні кон'юнкції N аргументів. Підставивши в ліву частину вирази (4.1) значення функції згідно заданій таблиці істинності, а в праву – відповідні набори значень аргументів, отримаємо систему логічних рівнянь, які легко вирішуються щодо невідомих коефіцієнтів. Для нульових значень лівої частини виразу (4.1) коефіцієнти при кон'юнкціях, що не обертаються в нуль на даному наборі, повинні дорівнювати нулю, оскільки диз'юнкція тільки тоді рівна нулю, коли всі доданки рівні нулю. Для одиничних значень функцію досить, щоб хоч би один доданок дорівнював одиниці.

З викладеного витікає простій алгоритм методу невизначених коефіцієнтів. Складається ДНФ у вигляді (4.1), підставляються набори значень аргументів, на яких функція рівна нулю (нульові набори) і всі коефіцієнти при ненульових кон'юнкціях прирівнюються нулю. Підставляються одиничні набори і для кожного з них коефіцієнти при ненульових кон'юнкціях мінімального рангу прирівнюються одиниці, решта коефіцієнтів прирівнюється нулю. З кон'юнкцій з одиничними значеннями коефіцієнтів складається мінімальна ДНФ.

Основний недолік методу обумовлений трудністю перебору всіх можливих елементарних кон'юнкцій. Очевидно, число кон'юнкцій рівне числу можливих різних комбінацій аргументів і їх заперечень. Неважко бачити, що можливих комбінацій без урахування заперечень визначається виразом:

$$S_1 = C_N^1 + C_N^2 + \ldots + C_N^i + \ldots + C_N^N, \tag{4.2}$$

де N – число аргументів; C_N^i – число сполучень з N по i.

Згідно формули розкладання бінома Ньютона

$$S_1 = (1+1)^N - C_N^0 = 2^N - 1. \tag{4.3}$$

З урахуванням заперечення кожна певна кон'юнкція з множини S_1 у свою чергу дасть 2^i різних комбінацій. Тому загальне число різних елементарних кон'юнкцій має вигляд

$$S_1 = C_N^1 2^1 + C_N^2 2^2 + \ldots + C_N^N 2^N = 3^N - 1. \tag{4.4}$$

Закодуємо змінні за позиційною ознакою. Хай x_1 відповідає 1, x_2 – 10, x_3 – 100, x_4 – 1000 і т.д.; не x_1 – 2, не x_2 – 20, не x_3– 200 і так далі Одиниці в старшому розряді коди відповідає певному аргументу, двійка – запереченню цього аргументу. Якщо кон'юнкції відобразити у вигляді арифметичної суми кодів, то числовий ряд від 1 до$3^N - 1$, в трійковій системі числення, дасть всі можливі елементарні кон'юнкції.

Представимо цей ряд у вигляді вектора

$$A = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 & 1 \\ 0 & 0 & \ldots & 0 & 0 & 2 \\ 0 & 0 & \ldots & 0 & 1 & 0 \\ 0 & 0 & \ldots & 0 & 1 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 2 & 2 & \ldots & 2 & 2 & 2 \end{bmatrix}. \tag{4.5}$$

Формування вектора дає просте мнемонічне правило перебору кон'юнкцій, яким зручно користуватися як при ручному рахунку, так і при вирішенні завдань на ЕОМ.

Весь процес мінімізації може бути зведений до виконання наступних етапів:

1. Формуємо трійковий числовий ряд від 1 до $3^N - 1$ (вектор A).
2. Кожному елементу вектора A приводимо у відповідність арифметичний добуток значень відповідних аргументів (вектор E), тобто

$$A \to E = \begin{bmatrix} x_1 \\ 1 - x_1 \\ x_2 \\ x_1 x_2 \\ \vdots \\ (1 - x_1)(1 - x_2)\ldots(1 - x_N) \end{bmatrix}. \tag{4.6}$$

Тут заперечення i -го аргументу відображається різницею $(1 - x_i)$.

3. У вектор E послідовно підставляємо ті набори значень аргументів, на яких функція рівна нулю, – нульові набори. При кожній такій підстановці частина елементів обернеться в одиницю. Відповідні ним елементи початкового вектора E виключаємо з подальшого розгляду, прирівнюючи їх нулю.

4. У вектор E по черзі підставляємо одиничні набори, і результуючі елементи умножаємо на відповідні елементи вектора A.

5. З вектора, отриманого в результаті такого множення, вибираємо ненульові елементи, що містять мінімальне число цифр, відмінних від нуля. Цими елементами є коди кон'юнкцій мінімального рангу.

Число кон'юнкцій, що «проглядаються», також можна істотно скоротити, опираючись на той факт, що число елементарних кон'юнкцій, що не обертаються в нуль при підстановці певного набору значень аргументів у виразі (4.1), дорівнюють $2^N - 1$.

Правила знаходження цих кон'юнкцій дуже прості.

1. Формується двійковий ряд чисел від 1 до $2^N - 1$ (вектор C).

Хай $N = 3$. Тоді

$$C = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \qquad (4.7)$$

2. Кожен елемент вектора умножається на 2 (вектор D).

$$D = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 2 & 0 \\ 0 & 2 & 2 \\ 2 & 0 & 0 \\ 2 & 0 & 2 \\ 2 & 2 & 0 \\ 2 & 2 & 2 \end{bmatrix}. \qquad (4.8)$$

3. З числа в певному розряді кожного елементу вектора D віднімається значення відповідного аргументу в наборі за умови, що це число відмінне від нуля. Якщо воно рівне нулю, то нуль переноситься в результат. Наприклад, хай значення аргументів x_3, x_2, x_1 рівні відповідно 1,0,1.

Тоді результат описаного перетворення представляється вектором F.

$$F=\begin{bmatrix}0&0&1\\0&2&0\\0&2&1\\1&0&0\\1&0&1\\1&2&0\\1&2&1\end{bmatrix} \qquad (4.9)$$

Це і є кон'юнкції, які на наборі 1,0,1 не обертаються в нуль.

Викладені правила дозволяють скоротити число кон'юнкцій 3^N-1, що проглядаються $3^N-1\,2^N-1$. Крім того, впорядкування елементарних кон'юнкцій за збільшенням їх рангів дозволяє добитися такої організації циклів, при якій можна вийти на кінець програми після отримання шуканого результату.

4.2. Метод алгебраїчних перетворень

Хай $x\in E_2$.

Позначимо $$x^{\sigma}=\begin{cases}x, \text{ якщо } \sigma=1\\ \underline{x}, \text{ якщо } \sigma=0\end{cases}$$

де для всіх $j=\overline{1,m}$

Формула, $x_{i_1}^{\sigma_1}\&x_{i_2}^{\sigma_2}\&\ldots\&x_{i_m}^{\sigma_m}$ називається *кон'юнкцією над множиною змінних* $\{x_1,x_2,\ldots,x_n\}$, $x_i\in E_2$. Кон'юнкція називається *елементарною* (скорочено е.к.), якщо $x_{i_j}\neq x_{i_k}\; j\neq k$.

Символи & у е.к. скорочено опускатимуться. Число букв в е.к. називається *рангом* е.к.

Формула вигляду $K_1\vee K_2\vee\ldots\vee K_s$ (короткий запис $\bigvee\limits_{i=1}^{s}K_i$), де K_i ($i-\overline{1,s}$) – диз'юнкція, називається *диз'юнктивною нормальною формою* (скорочено д.н.ф.).

Угрупування

Елементарні кон'юнкції K_i і K_j в д.н.ф. групуються в пари так, щоб після винесення за дужку загального множника *K* формула $K_i\vee K_j$ мала вигляд $K(x\vee\bar{x})$ або $K(1\vee x)$. Далі ці вирази замінюються еквівалентним ним виразам *K*.

Приклад

$$\bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 = \bar{x}_1 x_2 (\bar{x}_3 \vee x_3) \vee x_1 x_2 (\bar{x}_3 \vee x_3) =$$
$$= \bar{x}_1 x_2 \vee x_1 x_2 = x_2 (\bar{x}_1 \vee x_1) = x_2.$$

4.3. Метод Блейка

Метод полягає в послідовному застосуванні двох правил до д.н.ф.

Правило 1 називається *узагальненим склеюванням* і грунтується на рівності:

$$xK_1 \vee \bar{x}K_2 = xK_1 \vee \bar{x}K_2 \vee K_1 K_2.$$

Справедливість рівності можна показати за допомогою ланцюжка перетворень:

$$xK_1 \vee \bar{x}K_2 = xK_1 \& 1 \vee \bar{x}K_2 \& 1 = xK_1 (1 \vee K_2) \vee \bar{x}K_2 (1 \vee K_1) =$$
$$= xK_1 \vee xK_1 K_2 \vee \bar{x}K_2 \vee \bar{x}K_1 K_2 = xK_1 \vee \bar{x}K_2 \vee K_1 K_2 (x \vee \bar{x}) =$$
$$= xK_1 \vee \bar{x}K_2 \vee K_1 K_2.$$

На першому етапі проводяться операції узагальненого склеювання, поки це можливо. Для цього в д.н.ф. відшукуються формули вигляду $xK_1 \vee \bar{x}K_2$ і до д. н. ф. додаються е.к. $K_1 K_2$.

Зручне порівняння е.к.. проводити таким чином : 1-а е.к. порівнюється з 2-ою, 3-ою і так далі і у разі, коли диз'юнкція е.к. утворює формулу вигляду, $xK_1 \vee \bar{x}K_2$ до $K_1 K_2$ Потім 2-а е.к. у д.н.ф. порівнюється з 3-ою, 4-ою і т. д., зокрема <u>із знов отриманими при </u>всіх <u>попередніх </u>склеюваннях. І так далі для всіх е.к. у д.н.ф.. На цьому етапі формула ускладнюється.

Приклад

Хай $f(x_1, x_2, x_3) = x_1 x_2 \vee \bar{x}_1 x_3 \vee \bar{x}_2 x_3$.

Після першого етапу отримаємо

$f(x_1, x_2, x_3) = x_1 x_2 \vee \bar{x}_1 x_3 \vee \bar{x}_2 x_3 \vee x_2 x_3 \vee x_1 x_3 \vee x_3$.

Е.к. $x_2 x_3$ вийшла в результаті склеювання $x_1 x_2$ і $\bar{x}_1 x_3$ по x_1, $x_1 x_3$ – результат склеювання $\bar{x}_1 x_3$ із знов отриманою до. $x_1 x_3$ по x_1.

Правило 2 називається *поглинанням* і ґрунтується на рівності, що легко доводиться ланцюжком перетворень: $K_1 \vee K_1 K_2 = K_1 (1 \vee K_2) = K_1 \cdot 1 = K_1$. На другому етапі проводяться всілякі операції поглинання. Для цього в перетвореній після операції склеювання формулі знаходяться е.с. найменшого рангу, і всі е.к. у д.н.ф., що містять ці е.к. як співмножники, викреслюються.

Приклад

Для функції з попереднього прикладу застосування правила поглинання дає:

$$f(x_1, x_2, x_3) = x_1 x_2 \vee \bar{x}_1 x_3 \vee \bar{x}_2 x_3 \vee x_2 x_3 \vee x_1 x_3 \vee x_3 = x_1 x_2 \vee x_3.$$

Якщо задана функція своєї д. н. ф., то при її спрощенні доцільно спочатку застосувати метод угрупування, як більш простій, потім метод Блейка. Якщо відразу починати з методу Блейка, то перетворення бувають дуже громіздкими.

Приклад 1

$$f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2x_3 \vee x_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 =$$
$$= \bar{x}_1\bar{x}_2x_3 \vee x_1x_2(x_3 \vee \bar{x}_3) \vee x_1\bar{x}_2(x_3 \vee \bar{x}_3) = \bar{x}_1\bar{x}_2x_3 \vee x_1x_2 \vee x_1\bar{x}_2 =$$
$$= \bar{x}_1\bar{x}_2x_3 \vee x_1(x_2 \vee \bar{x}_2) = \bar{x}_1\bar{x}_2x_3 \vee x_1 = \bar{x}_1\bar{x}_2x_3 \vee x_1 \vee \bar{x}_2x_3 = \bar{x}_2x_3 \vee x_1.$$

Приклад 2

$$f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2\bar{x}_3 =$$
$$= \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2(\bar{x}_3 \vee x_3) \vee x_1\bar{x}_3(\bar{x}_2 \vee x_2) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2 \vee x_1\bar{x}_3 =$$
$$- \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2 \vee x_1\bar{x}_3 \vee \bar{x}_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3 \vee \bar{x}_2x_3 \vee \bar{x}_3 - \bar{x}_1x_2 \vee \bar{x}_3$$

4.4. Мінімізація булевих функцій по Куайну

Хай будуть задані номери наборів з чотирьох змінних, на яких логічна функція приймає одиничне значення:

$$f\,(2, 5, 6, 7, 10, 12, 13, 14) = 1.$$

Виразимо цю логічну функцію в ДНФ (символ кон'юнкції писати не будемо):

F (0010, 0101, 0110, 0111, 1010, 1100, 1101, 1110) =

$$= \underline{x}_4\underline{x}_3x_2\underline{x}_1 \vee \underline{x}_4x_3\underline{x}_2x_1 \vee \underline{x}_4x_3x_2\underline{x}_1 \vee \underline{x}_4x_3x_2x_1 \vee$$
$$\vee\, x_4\underline{x}_3x_2\underline{x}_1 \vee x_4x_3\underline{x}_2\underline{x}_1 \vee x_4x_3\underline{x}_2x_1 \vee x_4x_3x_2\underline{x}_1$$

На першому етапі мінімізації початкову ДНФ можна спростити за рахунок використання *закону склеювання*:

$$f = x_2\underline{x}_1 \vee \underline{x}_4x_3x_1 \vee \underline{x}_1x_3x_2 \vee x_3\underline{x}_2x_1 \vee x_4x_3\underline{x}_2 \vee x_4x_3\underline{x}_1.$$

Звертаємо увагу на те, що одну і ту ж конституанту (імпліканту) можна склеювати з іншими конституантами (імплікантами) багато разів, оскільки в логіці Буля діє *закон идемпотентности*:

$$a = a \vee a = a \vee a \vee a = \dots, \quad a = a \wedge a = a \wedge a \wedge a = \dots,$$

тому будь-яку конституанту можна розмножити.

На другому етапі скористаємося *таблицею Куайна* (табл. 1), відповідно до якої метод мінімізації отримав найменування — *метод Куайна*.

Таблиця 4.1.

$x_4x_3x_2x_1$	0010	0101	0110	0111	1010	1100	1101	1110
– – 0 1	1	0	1	0	1	0	0	1
0 1 – 1	0	1	0	1	0	0	0	0
0 1 1 –	0	0	1	1	0	0	0	0
– 1 0 1	0	1	0	0	0	0	1	0
1 1 0 –	0	0	0	0	0	1	1	0
1 1 – 0	0	0	0	0	0	1	0	1

У таблиці по вертикалі перераховані всі отримані на першому етапі спрощення імпліканти, а по горизонталі – початкові конституанти. Одиниця в табл. 1. стоїть там, де імпліканта «накриває» конституенту. Річ у тому, що конституанта завжди може бути замінена імплікантою або навіть окремим термом за законом *поглинання*:

$$a = a \vee (a \wedge b) = a \vee (a \wedge abc) = \ldots ,$$

$$a = a \wedge (a \vee b) = a \wedge (a \vee abc) = \ldots$$

Після заповнення таблиці Куайна вийшло так, що майже в кожній графі опинилося по дві одиниці; тим часом досить мати одну одиницю на графі. Тому, по можливості, потрібно виключити надмірні одиниці. Вибір одиниць проводиться з міркувань мінімальності числа термів (вибрані одиниці заштриховані). У результаті виявилось, що можна обійтися тільки трьома імплікантами замість шести:

$$f = x_2\underline{x}_1 \vee \underline{x}_4x_3x_1 \vee x_4x_3\underline{x}_2.$$

За допомогою таблиць істинності перевіряємо, що отримана в МНФ функція відтворює всі значення початкової функції. Відзначимо, що в загальному випадку рішень за критерієм мінімуму термів може бути декілька.

4.5. Мінімізація за методом поєднання індексів

Не менш ефективним способом мінімізації логічних функцій є *метод поєднання індексів*. Для його викладу складемо табл.. 4.2., в стовпцях якої записані можливі поєднання індексів. У останній графі виписані значення функції *y*. Аналіз таблиці починається зліва по стовпцях. Принцип виключення *i, j - коду* наступний. На перетині *i-стовпця*, наприклад з поєднанням індексів 23, і *j-строки*, наприклад 3-го, знаходиться код 10, що відповідає імпліканті $x_2\underline{x}_3$.. Отже, в цьому стовпці скрізь, де зустрічається код 10, тобто в рядках 2, 3, 10 і 11, ці коди виключаються, оскільки значення функції в 3-му рядку рівне нулю. Тепер візьмемо стовпець з поєднанням індексів 124. Тут в 2-му і 6-му рядках залишені коди 010, а в 10-му і 14-му рядках – код 011. Зроблено це тому, що ці коди зустрічаються тільки на рядках зі значенням функції, рівним одиниці. Навпаки, код 110 цього ж стовпця зустрічається як при одиничних значеннях функції, так і при нульових.

Таблиця 4.2

n	1	2	3	4	12	13	14	23	24	34	123	124	134	234	1234	y
0	0	0	0	0	00	00	00	00	00	00	000	000	000	000	0000	0
1	1	0	0	0	10	10	10	00	00	00	100	100	100	000	1000	0
2	0	1	0	0	**01**	00	00	10	10	00	**010**	**010**	000	100	0100	1
3	1	1	0	0	11	10	10	10	10	00	110	110	100	100	1100	0
4	0	0	1	0	00	01	00	01	00	10	001	000	010	010	0010	0
5	1	0	1	0	10	11	10	01	00	10	**101**	100	**110**	010	1010	1
6	0	1	1	0	**01**	01	00	11	10	10	011	**010**	010	**110**	0110	1
7	1	1	1	0	11	11	10	11	10	10	111	110	**110**	**110**	1110	1
8	0	0	0	1	00	00	01	00	01	01	000	001	001	001	0001	0
9	1	0	0	1	10	10	11	00	01	01	100	101	101	001	1001	0
10	0	1	0	1	**01**	00	01	10	11	01	**010**	**011**	001	101	0101	1
11	1	1	0	1	11	10	11	10	11	01	110	111	101	101	1101	0
12	0	0	1	1	00	01	01	01	01	11	001	001	011	**011**	0011	1
13	1	0	1	1	10	11	11	01	01	11	**101**	101	111	**011**	1011	1
14	0	1	1	1	**01**	01	01	11	11	11	**011**	**011**	**011**	111	0111	1
15	1	1	1	1	11	11	11	11	11	11	111	111	111	111	1111	0

Отже, всі коди на рядках, що закінчуються нульовими значеннями функції, виключаються автоматично. Якщо ці коди потрапляють на рядки, що закінчуються одиничним значенням функції, то вони також не враховуються. Залишаються тільки ті коди, які розташовані на рядках з одиничним значенням функції (ці коди затемнені).

Далі керуються наступним правилом. Для того, щоб функція прийняла значення, рівне одиниці, достатньо того, щоб тільки яка-небудь одна імпліканта на рядку прийняла одиничне значення. Перш за все залишаємо мінімальну імпліканту $x_2\underline{x}_1$, яка перекриває одиниці в рядках 2, 6, 10 і 14. Потім, природно, звертаємося до 12-го рядка. Тут залишаємо єдиний на рядку код 011, що відповідає імпліканті $x_4x_3\underline{x}_2$. Ця ж імпліканта відповідальна за 13-й рядок, що закінчується теж одиницею. Залишилося розглянути 5-й і 7-й рядки. Загальною для них є імпліканта $\underline{x}_4x_3x_1$.

Таким чином, трьома імплікантами ми перекрили всі одиничні значення функції, що співпадає з результатом, отриманим на основі таблиці Куайна.

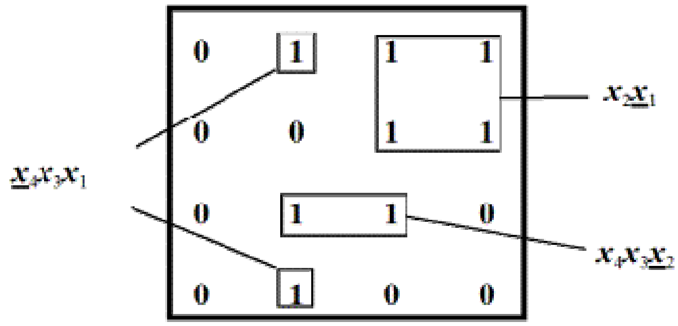
4.6. Карти Карно

Карта Карно для чотирьох змінних представлена табл. 4.3. Кожна клітка карти відповідає конституєнті. Заповнена карта представлена табл. 4 (функція узята та ж, що і по-перше двох методах). Згідно *закону склеювання*, дві суміжні конституєнти з одиничними значеннями завжди можна об'єднати для отримання відповідної імпліканти. Причому суміжними вважаються і ті, які лежать на межах карти. Які саме одиниці потрібно об'єднати для отримання відповідної імпліканти, легко визначити візуально. Слід також пам'ятати, що відповідно до закону идемпотентности одна і та ж одиниця табл. 4.4 може склеюватися з двома або трьома суміжними одиницями.

Таблиця 4.3

	x_1	x_1	$\underline{x}_1$	$\underline{x}_1$	
x_2	1100	1110	0110	0100	$\underline{x}_4$
x_2	1101	1111	0111	0101	x_4
$\underline{x}_2$	1001	1011	0011	0001	x_4
$\underline{x}_2$	1000	1010	0010	0000	$\underline{x}_4$
	$\underline{x}_3$	x_3	x_3	$\underline{x}_3$	

Таблиця 4.4



Пояснимо ідею карт Карно іншим способом.

Всю математичну логіку складають 3 операції: «заперечення» – «¬», «АБО» – «+», «І» – «&». За допомогою цих операцій можна реалізувати будь-яку функцію алгебри логіки. Значення, що повертаються цими логічними операціями видно з таблиці істинності.

A	B	$\neg A$	$\neg B$	$A + B$	$A \& B$
0	0	1	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	1	1

Ідея карти Карно полягає в тому, щоб усі можливі варіанти розташувати в одній таблиці. На наступному малюнку показана нумерація таблиць Карно для 2, 3 і 4 змінних.

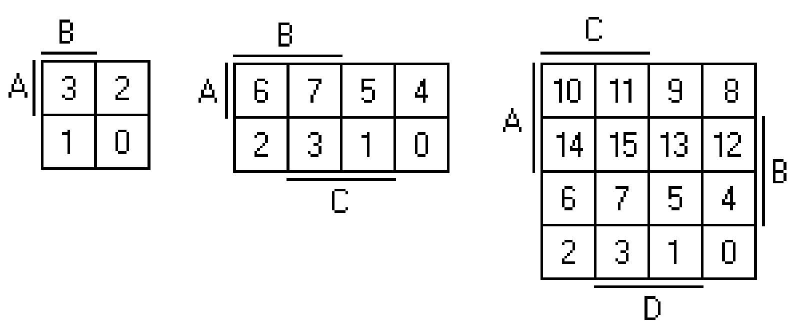


Рис. 4.1. Нумерація таблиць Карно для 2, 3 і 4 змінних

Нумерація таблиці така, що наприклад осередку з номером 2 відповідає варіант, в якому A = 1 і B = 0. Риски над елементами таблиці обмежують такі зони, в яких значення даної змінної завжди рівне одиниці.

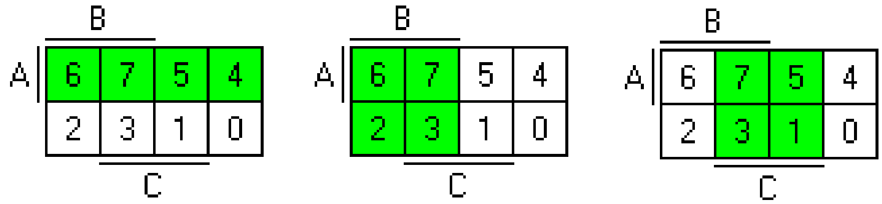


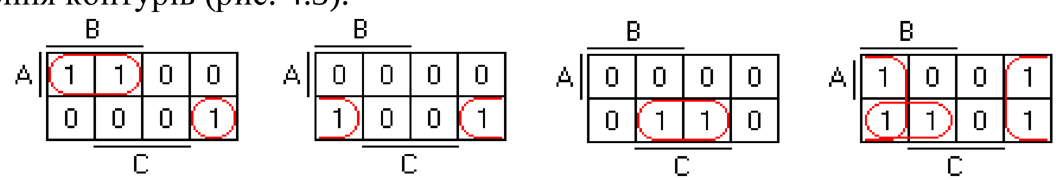
Рис. 4.2. Зони обмеження

Бачите яку зону охоплює змінна ***A*** (вона зелена)? У двійкових записах чисел 6=110, 7=111, 5=101 і 4=100 перша двійкова цифра (а це і є ***A***) рівна 1. Так от вся таблиця ділиться на зони. На малюнку перша таблиця для ***A***, друга для ***B***, а третя для ***C*.**

Далі Ви складаєте таблицю істинності і заносите нулі і одиниці в клітки з номерами, десяткові значення яких рівні двійковому значенню набору змінних.

Осередок	A B C
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Далі наступає ще складніша операція – це знаходження контурів. Контуром тут називається область в таблиці з одиниць. При цьому ніж більше одиниць потрапили в один контур, тим краще. Кількість одиниць в одному контурі може бути тільки кратна двом, тобто 1 2 4 8 ... Ось дивитеся приклад знаходження контурів (рис. 4.3).



$A \& B + \bar{A} \& \bar{B} \& \bar{C}$ $\quad$ $\bar{A} \& \bar{C}$ $\quad$ $\bar{A} \& C$ $\quad$ $\bar{C} + \bar{A} \& B$

1 2 3 4

Рис. 4.3. Принцип знаходження контурів

Існують цілі контури, як на 1 з рис. 4.3. А є також розірвані як на 2 і на 4 з рис. 4.3. Але розірвані вони тільки з вигляду, а якщо в думках склеїти протилежні краї таблиці, то контури будуть цілими. Вся ідея проста: Вам потрібно обвести якомога більше одиниць одночасно, і при цьому зробити стільки різних контурів, щоб всі одиниці були покриті. Одиниці можуть розташовуватися тільки в стовпець, рядок або декілька стовпців і рядків, тобто по діагоналі контури будувати не можна.

Це запис логічного виразу. Ви бачите контур вгорі і контур на один осередок внизу справа. Давайте запишемо формулу для верхнього контуру. Для цього дивимося в яку область контур повністю потрапляє. В даному випадку контур повністю потрапляє в область ***A*** і ***B***, тому для нього буде формула ***A&B***. Нижній контур не потрапляє ні в одну з відомих нам областей. Всі такі області називаються протилежними і виходять операцією НЕ.

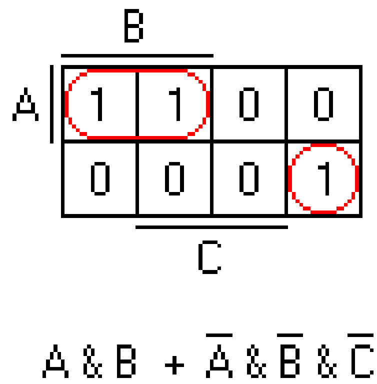


Рис. 4.4. Карта Карно, що відповідає певному логічному виразу

4.7. Індивідуальне завдання №4

Вивчити всі методи мінімізації булевих функцій.

1. Побудувати ДСНФ і КСНФ по відповідному варіанту завдання, представленому в таблиці завдань.
2. Знайти мінімальну форму булевої функції всіма методами, описаними вище.

.

Таблиця варіантів завдань.

№№ п/п	1	2	3	4	5	6	7	8
змінні								
X1	0	0	0	0	1	1	1	1
X2	0	0	1	1	0	0	1	1
X3	0	1	0	1	0	1	0	1

Варіанти	Варіанти наборів значень Y							
1	1	1	1	0	0	0	0	0
2	0	1	1	1	0	0	0	0
3	0	0	1	1	1	0	0	0
4	0	0	0	1	1	1	0	0
5	0	0	0	0	1	1	1	0
6	0	0	0	0	0	1	1	1

Варіан-ти	Варіанти наборів значень Y							
7	1	0	1	1	0	0	0	0
8	1	0	0	1	1	0	0	0
9	1	0	0	0	1	1	0	0
10	1	0	0	0	0	1	1	0
11	1	0	0	0	0	0	1	1
12	0	1	0	0	0	0	1	1
13	0	0	1	0	0	0	1	1
14	0	0	0	1	0	0	1	1
15	0	0	0	0	1	0	1	1
16	0	0	0	1	0	1	1	1
17	1	1	0	0	0	0	1	0
18	0	1	1	0	0	0	1	0
19	0	0	1	1	0	0	1	0
20	0	0	0	1	1	0	1	0
21	0	0	1	0	1	1	1	0
22	1	1	0	1	0	0	0	0
23	1	0	1	1	0	0	0	0
24	0	1	1	1	0	0	0	0
25	0	0	1	1	1	0	0	5
26	1	0	1	1	1	0	0	0
27	1	0	0	1	0	0	0	1
28	0	1	0	0	1	0	0	1
29	0	0	1	0	0	1	0	1
30	0	0	0	1	0	1	0	1
31	0	0	0	0	1	1	0	1
32	1	1	0	0	0	0	0	0
33	0	1	1	0	0	0	0	0
34	0	0	1	1	0	0	0	0
35	0	0	0	1	1	0	0	0
36	0	0	0	0	1	1	0	0
37	0	0	0	0	0	1	1	0
38	0	0	0	0	0	0	1	1
39	1	0	0	0	0	0	0	1
40	0	1	0	0	0	0	1	0
41	0	0	1	0	0	1	0	0
42	0	0	0	1	1	0	0	0
43	0	1	0	1	0	1	0	0
44	1	1	1	1	0	0	0	0
45	1	1	1	0	1	0	0	0
46	1	1	1	0	0	1	0	0
47	1	1	1	0	0	0	1	0

Варіан-ти	Варіанти наборів значень Y							
48	1	1	0	0	0	1	1	0
49	0	1	1	1	1	0	0	0
50	1	0	1	1	0	0	0	0
51	0	1	1	1	1	0	0	0
52	0	0	1	1	1	1	0	0
53	0	0	0	1	1	1	1	0
54	0	0	0	0	1	1	1	1
55	1	1	1	0	0	0	0	1
56	1	1	0	0	0	0	1	1
57	1	1	0	0	0	1	1	0
58	1	1	0	0	1	1	0	0
59	1	1	0	1	1	0	0	0
60	0	1	1	1	1	0	0	0

5. НЕЧІТКА ЛОГІКА

Нечітка логіка базується на поняттях нечітких множин (описаних у розділі 2), які представляють собою множину можливих значень нечіткої величини у формі функції приналежності

$$A = \{x / mA(x) > 0\}.$$

Фактично нечітка множина визначає логічну змінну з якою можна проводити деякі логічні операції, описані у розділі 2.

5.1. Нечіткий логічний висновок

Основою для проведення операції нечіткого логічного висновку є база правив, що містить нечіткі вислови у формі «Якщо-то» і функції приналежності для відповідних лінгвістичних термів (тобто, нечітких множин, які визначають лінгвістичну змінну). При цьому повинні дотримуватися наступні умови:

1. Існує хоч би одне правило для кожного лінгвістичного терма вихідний змінної.
2. Для будь-якого терма вхідної змінної є хоча б одне правило, в якому цей терм використовується як передумова (ліва частина правила).

Інакше має місце неповна база нечітких правил.

Хай в базі правил є m правил вигляду:

R_1: ЯКЩО x_1 це A_{11} . І . x_n це A_{1n}, ТО y це B_1.

R_i: ЯКЩО x_1 це A_{i1} . І . x_n це A_{in}, ТО y це B_i.

R_m: ЯКЩО x_1 це A_{m1} . І . x_n це A_{mn}, ТО y це B_m.

де x_k, k=1..n – вхідні змінні; y – вихідна змінна; A_{ik} – задані нечіткі множини з функціями приналежності.

Результатом нечіткого висновку є чітке значення змінної y^* на підставі заданих чітких значень x_k, k=1..n

У загальному випадку механізм логічного висновку включає чотири етапи: введення нечіткості (фазифікація), нечіткий висновок, композиція і приведення до чіткості, або дефазифікація (див. рис. 5.1).

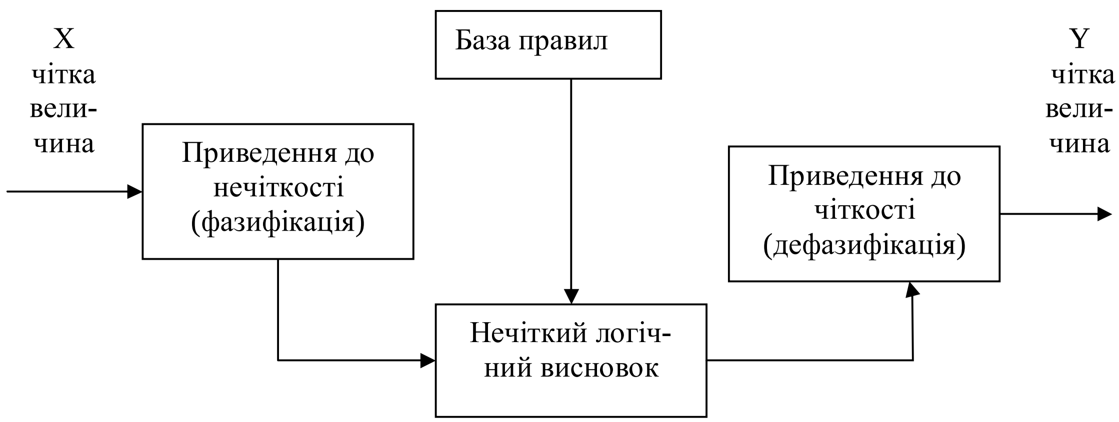


Рис. 5.1. Система нечіткого логічного висновку

Алгоритми нечіткого висновку розрізняються головним чином видом використовуваних правил, логічних операцій і різновидом методу дефазифікації. Розроблені моделі нечіткого виведення Мамдані, Сугено, Ларсена, Цукамото.

5.2. Модель нечіткого виведення Мамдані

Розглянемо докладніше нечіткий вивід на прикладі механізму Мамдані (Mamdani). Це найбільш поширений спосіб логічного висновку в нечітких системах. У нім використовується мінімаксна композиція нечітких множин. Даний механізм включає наступну послідовність дій.

1. Процедура фазифікації така: визначаються ступені істинності, тобто значення функцій приналежності для лівих частин кожного правила (передумов). Для бази правил з m правилами позначимо ступені істинності як $A_{ik}(x_k)$, i=1..m, k=1..n.

2. Нечіткий висновок. Спочатку визначаються рівні «відсікання» для лівої частини кожного з правил:

$$alfa_i = \min_i \left(A_{ik}(x_k)\right)$$

Далі знаходяться «усічені» функції приналежності:

$$B_i^*(y) = \min_i \left(alfa_i , B_i(y)\right)$$

3. Композиція, або об'єднання отриманих усічених функцій, для чого використовується максимальна композиція нечітких множин:

$$MF(y) – \max_i \left(B_i^*(y)\right)$$

де $MF(y)$ – функція приналежності підсумкової нечіткої множини.

4. Дефазифікація, або приведення до чіткості. Існує декілька методів дефазифікації. Наприклад, метод середнього центру, або центроїдний метод:

$$MF(y) = \max_i \left(B_i^*(y)\right).$$

Геометричний сенс такого значення – центр тяжіння для кривої $MF(y)$. На рис. 5.2. показано процес нечіткого висновку за Мамдані для двох вхідних змінних і двох нечітких правил R_1 і R_2.

Розглянемо випадок управління мобільним роботом, завданням якого є об'їзд перешкод. Введемо дві лінгвістичні змінні: ДИСТАНЦІЯ (відстань від робота до перешкоди) і НАПРЯМ (кут між подовжньою віссю робота і напрямом до перешкоди).

Розглянемо лінгвістичну змінну ДИСТАНЦІЯ. Її значення можна визначити термами ДАЛЕКО, СЕРЕДНЬО, БЛИЗЬКО і ДУЖЕ БЛИЗЬКО. Для фізичної реалізації лінгвістичної змінної необхідно визначити точні фізичні значення термів цієї змінної. Хай змінна ДИСТАНЦІЯ може приймати будь-які значення з діапазону від нуля до безкінечності. Згідно теорії нечітких множин, у

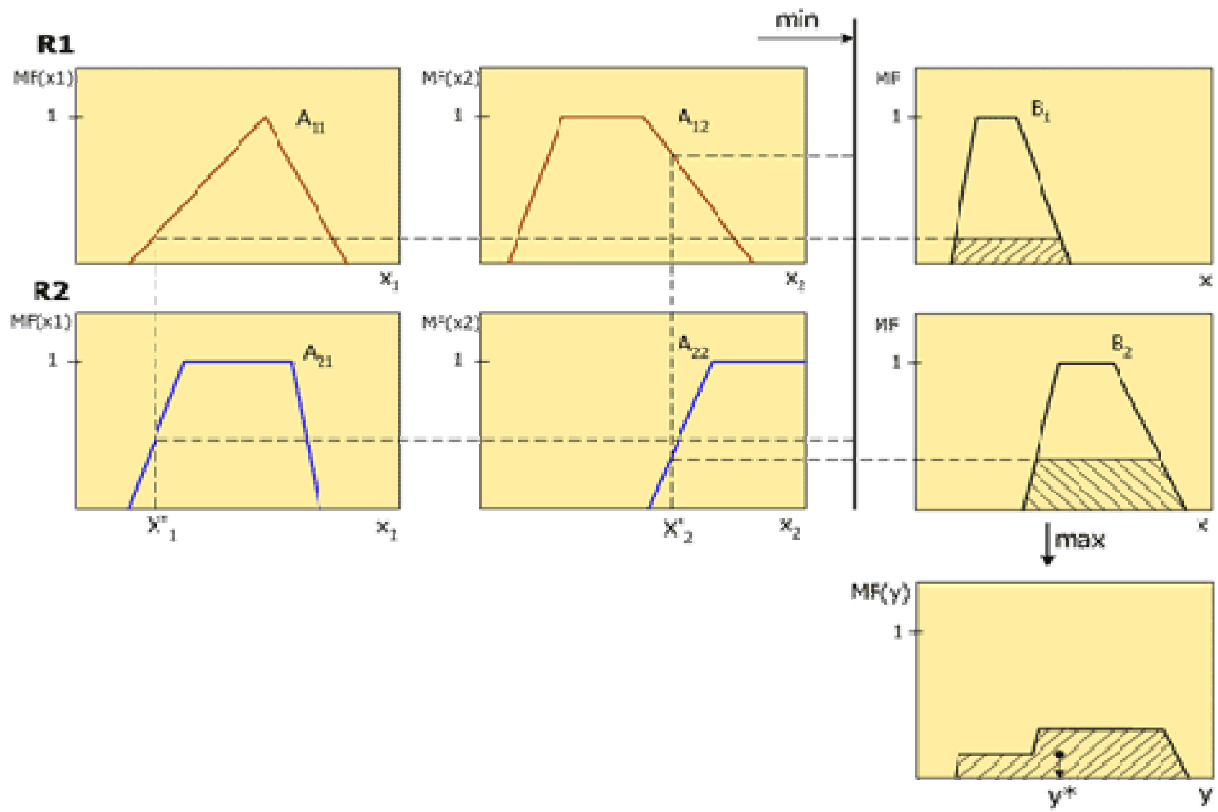


Рис. 5.2.. Схема нечіткого висновку за Мамдані

такому разі кожному значенню відстані з вказаного діапазону може бути поставлено у відповідність деяке число від нуля до одиниці, яка визначає ступінь приналежності даної фізичної відстані (допустимо 40 см) до того або іншого терма лінгвістичної змінної ДИСТАНЦІЯ. Ступінь приналежності визначаємо функцією приналежності $M(d)$, де d – відстань до перешкоди. У нашому випадку відстань 40 см. Можна задати ступінь приналежності до терма ДУЖЕ БЛИЗЬКО рівне 0,7, а до терма БЛИЗЬКО – 0,3 (рис. 5.3). Конкретне визначення ступеня приналежності проходить тільки при роботі з експертами.

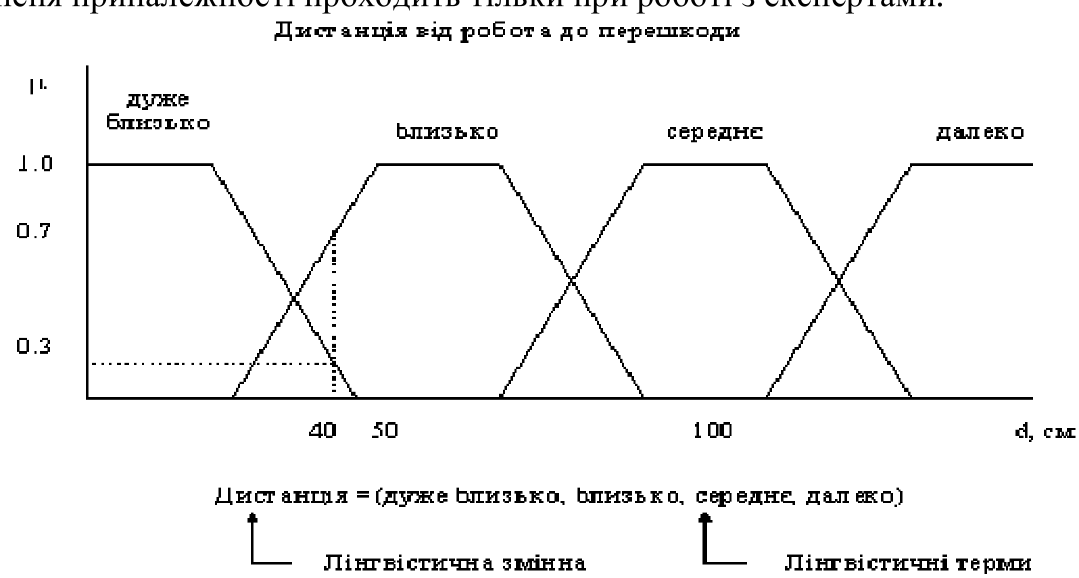


Рис. 5.3.. Лінгвістична змінна і функція приналежності

Змінній НАПРЯМ, яка приймає значення в діапазоні від 0 до 360 градусів, задамо терми ЛІВИЙ, ПРЯМИЙ і ПРАВИЙ.

Тепер необхідно задати початкові змінні. У даному прикладі достатньо однієї, яку назвемо РУЛЬОВИЙ КУТ. Вона може містити терми: РІЗКО ВЛІВО, ВЛІВО, ПРЯМО, УПРАВО, РІЗКО УПРАВО. Зв'язок між входом і виходом запам'ятовується в таблиці нечітких правил наступного вигляду.

		дістанція			
		дуже близько	близько	середнє	далеко
напрямок	правий	різко вліво	різко вліво	вліво	прямо
	прямий	різко вліво	вліво	вліво	прямо
	лівий	різко вправо	різко вправо	вправо	прямо

Кожен запис в даній таблиці відповідає своєму нечіткому правилу, наприклад "Якщо дистанція близько і напрям правий, тоді рульовий кут різко вліво".

Таким чином, мобільний робот з нечіткою логікою працюватиме за наступним принципом: дані від сенсорів про відстань до перешкоди і напрям до нього будуть фазифіковані, оброблені згідно табличним правилам, дефазифіковані, і отримані дані, у вигляді управляючих сигналів, які поступають на приводи робота.

5.3. Інтеграція з інтелектуальними парадигмами

Гібридизація методів інтелектуальної обробки інформації – девіз, під яким пройшли 90-і роки у західних і американських дослідників. В результаті об'єднання декількох технологій штучного інтелекту з'явився спеціальний термін – «м'які обчислення» (soft computing), який ввів Л. Заде в 1994 році. В даний час м'які обчислення об'єднують такі області як: нечітка логіка, штучні нейронні мережі, імовірнісні міркування і еволюційні алгоритми. Вони доповнюють один одного і використовуються в різних комбінаціях для створення гібридних інтелектуальних систем.

Вплив нечіткої логіки виявився обширним. Подібно до того, як нечіткі множини розширили рамки класичної математичної теорії множин, нечітка логіка «втрутилася» практично в більшість методів Data Mining, наділивши їх новою функціональністю. Нижче наводяться найцікавіші приклади таких об'єднань.

Нечіткі нейронні мережі

Нечіткі нейронні мережі (fuzzy-neural networks) здійснюють висновки на підставі апарату нечіткої логіки, проте параметри функцій приналежності настроюються з використанням алгоритмів навчання НС. Тому для підбору параметрів таких мереж застосуємо метод зворотного розповсюдження помилки, спочатку запропонований для навчання багатошарового персептрона. Для цього модуль нечіткого управління представляється у формі багатошарової мережі. Нечітка нейронна мережа як правило складається з чотирьох шарів: шаруючи фазифікації вхідних змінних, шаруючи агрегації значень активації умови, шаруючи агрегації нечітких правил і вихідного шаруючи.

Найбільшого поширення в даний час набула архітектура нечіткої НС виду ANFIS і TSK. Доведено, що такі мережі є універсальними апроксиматорами.

Швидкі алгоритми навчання і інтерпретується накопичених знань – ці чинники зробили сьогодні нечіткі нейронні мережі одним з найперспективніших і ефективніших інструментів м'яких обчислень.

Адаптивні нечіткі системи

Класичні нечіткі системи володіють тим недоліком, що для формулювання правив і функцій приналежності необхідно привертати експертів тієї або іншої наочної області, що не завжди вдається забезпечити. Адаптивні нечіткі системи (adaptive fuzzy systems) вирішують цю проблему. У таких системах підбір параметрів нечіткої системи проводиться в процесі навчання на експериментальних даних. Алгоритми навчання адаптивних нечітких систем відносно трудомісткі і складні в порівнянні з алгоритмами навчання нейронних мереж, і, як правило, складаються з двох стадій: 1. Генерація лінгвістичних правил; 2. Коректування функцій приналежності. Перше завдання відноситься до завдання переборного типу, друга – до оптимізації в безперервних просторах. При цьому виникає певна суперечність: для генерації нечітких правил необхідні функції приналежності, а для проведення нечіткого висновку – правила. Крім того, при автоматичній генерації нечітких правил необхідно забезпечити їх повноту і несуперечність.

Значна частина методів навчання нечітких систем використовує генетичні алгоритми. У англомовній літературі цьому відповідає спеціальний термін – Genetic Fuzzy Systems.

Значний внесок в розвиток теорії і практики нечітких систем з еволюційною адаптацією внесла група іспанських дослідників на чолі з Ф. Херрера (F. Herrera).

Нечіткі запити

Нечіткі запити до баз даних (fuzzy queries) – перспективний напрям в сучасних системах обробки інформації. Даний інструмент дає можливість формулювати запити на природній мові, наприклад: «Вивести список недорогих пропозицій про знімання житла близько до центру міста», що неможливе при використанні стандартного механізму запитів. Для цієї мети розроблена нечітка реляційна алгебра і спеціальні розширення мов SQL для нечітких запитів. Велика частина досліджень в цій області належить західноєвропейським ученим Д. Дюбуа и Г. Праде.

Нечіткі асоціативні правила

Нечіткі асоціативні правила (fuzzy associative rules) – інструмент для витягання з баз даних закономірностей, які формулюються у вигляді лінгвістичних висловів. Тут введені спеціальні поняття нечіткої трансакції, підтримки і достовірності нечіткого асоціативного правила.

Нечіткі когнітивні карти

Нечіткі когнітивні карти (fuzzy cognitive maps) були запропоновані Б. Косько в 1986 р. і використовуються для моделювання причинних взаємозв'язків, виявлених між концептами деякої області. На відміну від простих когнітивних карт, нечіткі когнітивні карти є нечіткий орієнтований граф, вузли якого є нечіткими множинами. Направлені ребра графа не тільки відображають причинно-наслідкові зв'язки між концептами, але і визначають ступінь впливу (вага) зв'язаних концептів. Активне використання нечітких когнітивних карт як засіб моделювання систем обумовлене можливістю наочного представлення аналізованої системи і легкістю інтерпретації причинно-наслідкових зв'язків між концептами. Основні проблеми пов'язані з процесом побудови когнітивної карти, який не піддається формалізації. Крім того, необхідно довести, що побудована когнітивна карта адекватна реальній модельованій системі. Для вирішення даних проблем розроблені алгоритми автоматичної побудови когнітивних карт на основі вибірки даних.

Нечітка кластеризація

Нечіткі методи кластеризації, на відміну від чітких методів (наприклад, нейронні мережі Кохонена), дозволяють одному і тому ж об'єкту належати одночасно декільком кластерам, але з різним ступенем. Нечітка кластеризація в багатьох ситуаціях «природніша», ніж чітка, наприклад, для об'єктів, розташованих на межі кластерів. Найбільш поширені: алгоритм нечіткої самоорганізації c-means і його узагальнення у вигляді алгоритму Густафсона-Кесселя.

Список можна продовжити і далі: нечіткі дерева рішень, нечіткі мережі Петрі, нечітка асоціативна пам'ять, нечіткі карти, що організуються самі, і інші гібридні методи.

5.4. Індивідуальне завдання №5

Задано наступну базу з трьох лінгвістичних правил:

R_1: ЯКЩО x_1 це A_{11} . І . x_2 це A_{12}, ТО y це B_1.

R_2: ЯКЩО x_2 це A_{22} . І . x_3 це A_{23}, ТО y це B_2.

R_3: ЯКЩО x_1 це A_{31} . І . x_3 це A_{32}, ТО y це B_3.

Нечіткі множини задані наступними функціями приналежності

X	$0,1N$	$0,9N+i$	$1,5N+i$	$2,1N+i$
A_{11}	0	$0,75+0,15i$		
A_{12}	0	$0,3-0,08i$	0,6	0,9
A_{22}	0,2	0,4	0	
A_{23}	0,8	$0,7+0,05i$	$0,3+0,22i$	0
A_{31}	0,2	0		
A_{32}	0,1	0,3	$0,7+0,1i$	0,2
B_1	0	$0,5-0,3i$	0,5	0
B_2	0	0,7	$0,7-0,2i$	0
B_3	0	$0,8-0,15i$	0,8	0

На вхід системи надійшло три чітких сигнали $x_1 = 0,7N + i$, $x_2 = 1,2N + 3i$, $x_3 = 1,7N + 0.2i$.

Знайти чіткий вихід системи y^* методом Мамдані.

Тут N – номер студента за списком групи, i – номер групи.

Рішення можна знаходити графічно.

Література до розділу 5.

1. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.: Мир, 1976.
2. Круглов В.В., Дли М.И. Интеллектуальные информационные системы: компьютерная поддержка систем нечеткой логики и нечеткого вывода. – М.: Физматлит, 2002.
3. Леоленков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб., 2003.
4. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М., 2004.
5. Масалович А. Нечеткая логика в бизнесе и финансах. www.tora-centre.ru/library/fuzzy/fuzzy-.htm
6. Kosko B. Fuzzy systems as universal approximators // IEEE Transactions on Computers, vol. 43, No. 11, November 1994. – P. 1329-1333.
7. Cordon O., Herrera F., A General study on genetic fuzzy systems // Genetic Algorithms in engineering and computer science, 1995. – P. 33-57.

6. СЕЛЕКТИВНІ ФУНКЦІЇ

При моделюванні кібернетичних систем нерідко виникає необхідність виділення ділянок складних об'єктів, що описуються порівняно простими математичними виразами.

Наприклад, деякі складні залежності можуть бути представлені лінійними відрізками. Заміна кривих залежностей деякими лінійними наближенням називається кусочно-лінійною апроксимацією. Для її здійснення необхідно навчиться виділяти відрізки за допомогою математичних виразів.

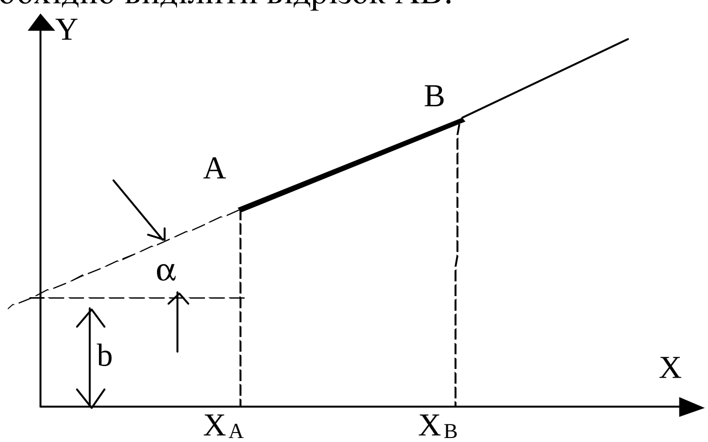
Це можна зробити за допомогою селективних функцій.

6.1. Прості селективні функції

Хай необхідно описати відрізок прямої:

$$y = a \cdot x + b \tag{6.1}$$

Рівняння (1) описує пряму лінію на площині *X*, *Y* від $x = -\infty$ до $x = +\infty$. Припустимо, що нам необхідно виділити відрізок АВ:



$a = tg(a)$

Рис. 6.1. Приклад опису відрізка АВ

Введемо спеціальну функцію:
Вона виділяє (селектує) відрізок на осі *X*. Тому вона називається селективною.

Тоді рівняння відрізку АВ можна представити у вигляді:

$$y = Si(x, x_A, x_B) \cdot (a \cdot x + b) \tag{6.2}$$

Відмітимо, що функція $Si(x, x_A, x_B)$ селектує замкнутий відрізок, тобто в нього включаються і кінці відрізку А і В.

За допомогою селективних функцій можна описувати складні фігури і конструкції. Наприклад, можна написати формулу гайки у вигляді суми відрізків прямих: Окрім лінійних функцій можна використовувати і нелінійні.

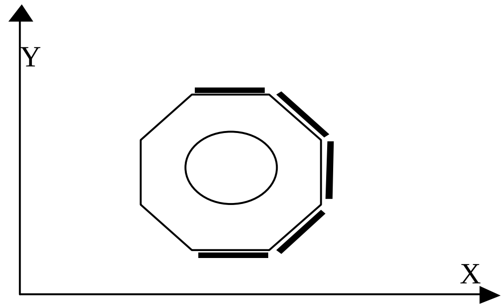


Рис. 6.2. Опис форми гайки відрізками прямих

Реалізувати селективні функції можна різними способами. Наприклад, за допомогою знакових функцій

$$sign(a)=\begin{cases}1, & при\ b>0;\\ 0, & при\ b=0;\\ -1, & при\ b<0.\end{cases} \text{ або } \operatorname{sgn}(a)=\begin{cases}1, & при\ b\ge 0;\\ -1, & при\ b<0.\end{cases} \qquad (6.3)$$

Ці функції можуть бути представлені графічно таким чином (рис. 6.3). Як видно з формул, вони відрізняються тільки тим, що sign(0) = 0, а sgn(0) = 1.

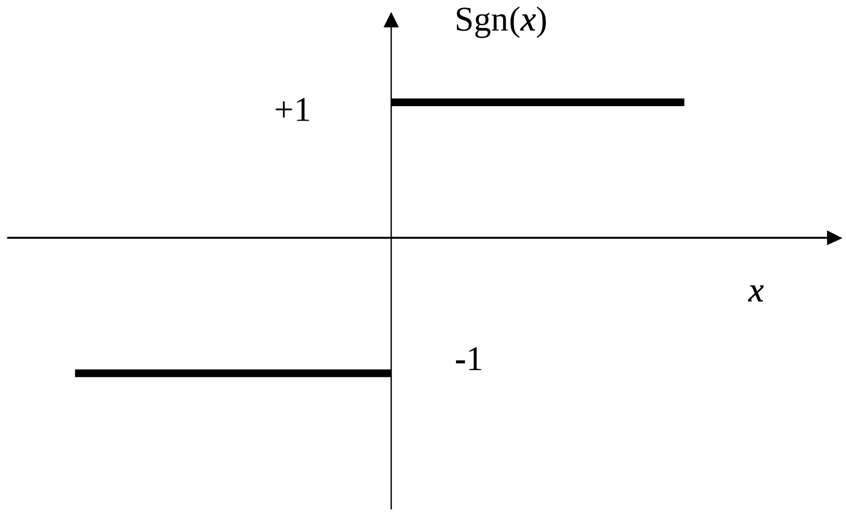


Рис. 6.3. Вигляд функції sgn(x)

Їх легко реалізувати в електронних таблицях Excel. Для SIGN э готова функція ЗНАК, а SGN можна реалізувати логічним оператором

ЕСЛИ(A1>=0;1;-1)

Інший тип селективних функцій, який є комбінацією вже відомих селективних функцій (6.3), такий

$$\left.\begin{aligned} Si(x,x_{\min},x_{\max}) &= \frac{1}{2}\cdot\{1+\operatorname{sgn}[(x-x_{\min})\cdot(x_{\max}-x)]\} \\ &\text{або} \\ Si(x,x_{\min},x_{\max}) &= sign\{1+sign[(x-x_{\min})\cdot(x_{\max}-x)]\}. \end{aligned}\right\} \qquad (6.4)$$

Ці функції вже пристосовані до можливості використання функцій sign та sgn до конкретних значень меж відрізка $x_{\min}$ *та* $x_{\max}$.Вони генерують значення 1 в діапазоні $x_{\min}$ *та* $x_{\max}$ і значення 01 – за межами цього діапазону. Причому, вони дають однакові результати . Їх можна реалізувати в електронних таблицях Excel за допомогою операторів

=(1+ЕСЛИ(B2>=0;1;-1)/2 або =ЗНАК(1+ЗНАК(B2)),

де B2 – адреса клітинки, що містить значення $(x - x_{\min}) \cdot (x_{\max} - x)$.
Для прикладу покажемо розрахунки в наступній таблиці для значень *Xmin* = 10, *Xmax* = 22.

X	(*X*-*Xmin*)(*X*-*Xmax*)	SGN	SIGN	Si_1	Si_2
4	-108	-1	-1	0	0
7	-45	-1	-1	0	0
10	0	1	0	1	1
13	27	1	1	1	1
16	36	1	1	1	1
19	27	1	1	1	1
22	0	1	0	1	1
25	-45	-1	-1	0	0
28	-108	-1	-1	0	0
31	-189	-1	-1	0	0

Коли ми хочемо виділити певну функцію за межами діапазону $x_{\min}$ *та* $x_{\max}$, потрібно використовувати перетворення вигляду

f(*x*)[1-Si(*x*;*Xmin*;*Xmax*)].

Для прикладу покажемо використання селективних функцій для того, щоб задати функцію вигляду

$$f(x) = \begin{cases} 5x - 12, x \quad < 3 \\ 0.8x^2 + 2 \ \ , 3 \le x \le 10 \\ 2 \ \lambda^{\,x}, \quad x \ > \ 10 \end{cases}$$

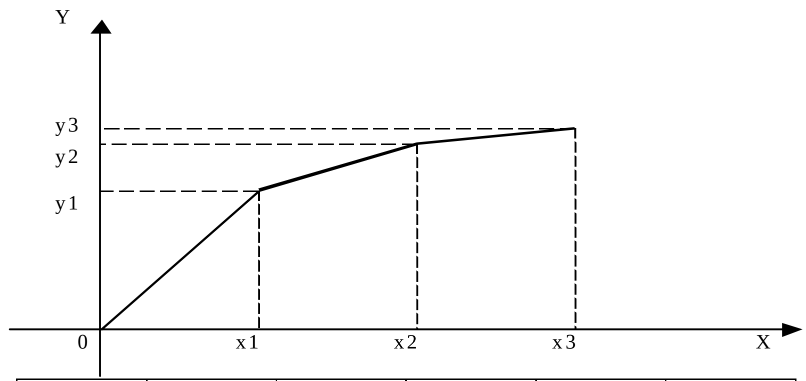
Така функція матиме вигляд

$$f(x) = (5x - 12)(1 - SGN(3) + (0.8x^2 + 2)Si(x;3;10) + 2\lambda^x \, SGN(10)$$

6.2. Індивідуальне завдання №6

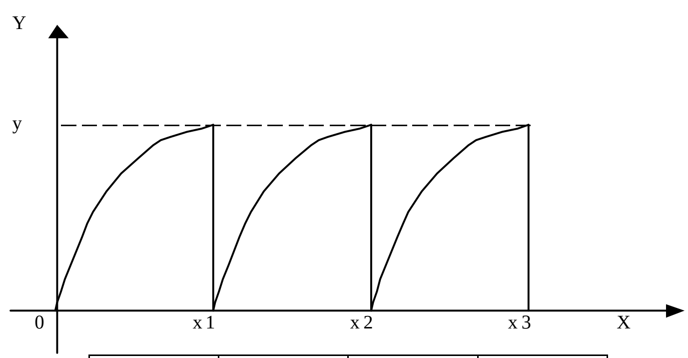
Необхідно побудувати рівняння для наступних графіків. Номер варіанта N – це номер студента за списком групи.

1.



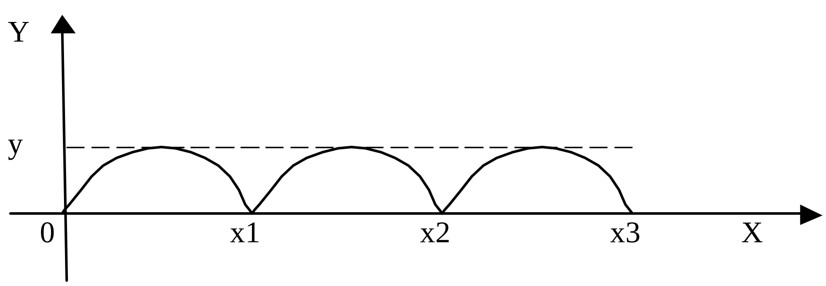
$x1$	$x2$	$x3$	$y1$	$y2$	$y3$
N	N+2	3N	N+1	3N+1	4N

2.



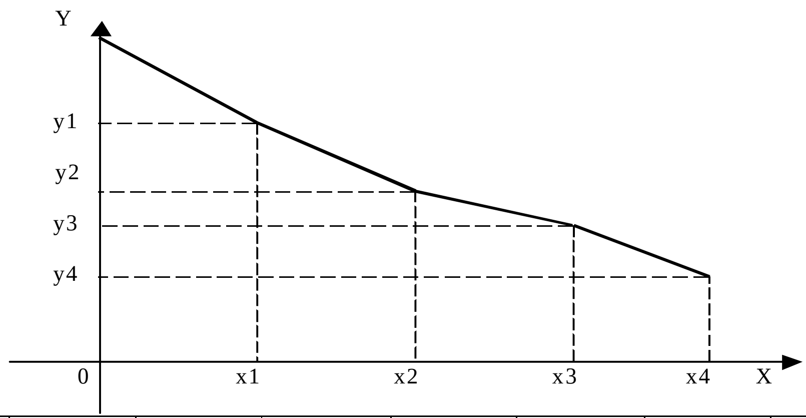
$x1$	$x2$	$x3$	y
N	2N	3N	N+1

3.



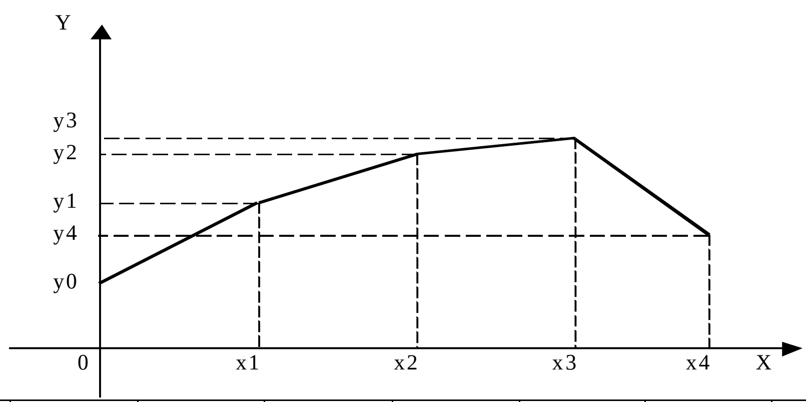
$x1$	$x2$	$x3$	Y
N	2N	3N	N/2

4.



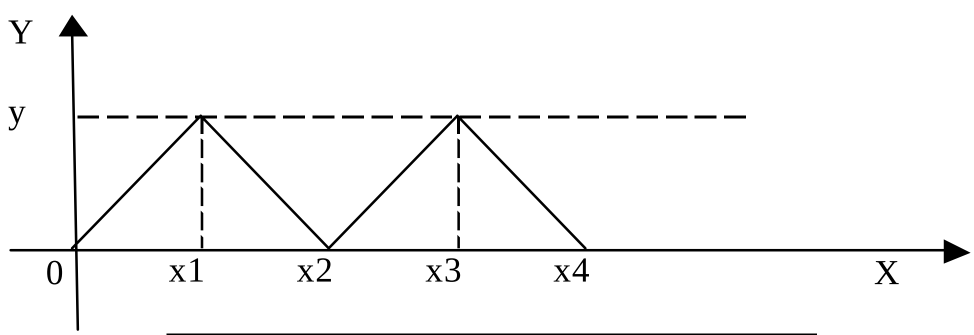
$x1$	$x2$	$x3$	$x4$	$y0$	$y1$	$y2$	$y3$	$y4$
N	N+2	3N	3N+3	5N	4N+1	3N	2N+4	N

5.



$x1$	$x2$	$x3$	$x4$	$y0$	$y1$	$y2$	$y3$	$y4$
N	N+2	3N	3N+3	5	7	18	15	6

6.



$x1$	$x2$	$x3$	$x4$	y
N	2N	3N	4N	N+1