Н. Р. Кондратенко, А. В. Остапенко-Боженова

РОЗДІЛИ ДИСКРЕТНОЇ МАТЕМАТИКИ ДЛЯ ЗАДАЧ ЗАХИСТУ ІНФОРМАЦІЇ

Міністерство освіти і науки України
Вінницький національний технічний університет

Н. Р. Кондратенко, А. В. Остапенко-Боженова

РОЗДІЛИ ДИСКРЕТНОЇ МАТЕМАТИКИ ДЛЯ ЗАДАЧ ЗАХИСТУ ІНФОРМАЦІЇ

Електронний навчальний посібник

Вінниця
ВНТУ
2022

УДК 004.056
К64

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 3 від 27.10.2022 р.)

Рецензенти:
О. М. Васілевський, доктор технічних наук, професор
О. В. Войцеховська, кандидат технічних наук, доцент
Л. В. Загоруйко, кандидат технічних наук, доцент

Кондратенко, Н. Р.

К64 Розділи дискретної математики для задач захисту інформації : електронний навчальний посібник [Електронний ресурс] / Н. Р. Кондратенко, А. В. Остапенко-Боженова. – Вінниця : ВНТУ, 2022. – (PDF, 88 с.)

Містить теоретичні відомості, тестові завдання та рекомендації до їх виконання при вивченні матеріалів з дисципліни «Теоретичні основи процесів у кібербезпеці» для студентів денної та заочної форм навчання спеціальності 125 «Кібербезпека». Подано основні поняття та методи теорії множин і відношень, алгебри логіки, закони логіки. Наведено основи теорії інформації та кодування. Наведено також основи логіко-лінгвістичних технологій, призначених для проєктування нечітких моделей оцінювання рівня безпеки інформації та розробки експертних систем захисту інформації. Розроблено та подано індивідуальні та тестові завдання з теорії множин і відношень, алгебри логіки та основ кодування.

УДК 004.056

ЗМІСТ

ПЕРЕДМОВА

Дискретна математика, до якої відносять теорію множин та відношень, математичну логіку та теорію кодування, як математична дисципліна почала формуватися в XVIII–XIX ст. завдяки роботам багатьох видатних математиків, зокрема Леонарда Ейлера, англійського вченого Дж. Буля та інших. Найбільшого поширення ця математична теорія досягла у теперішній час в зв'язку з розвитком обчислювальної техніки, інформатики, комп'ютерних засобів захисту інформації саме як наука для опису моделей захисту інформації в системах передавання інформації та в системах штучного інтелекту.

У першому розділі посібника подаються основні поняття та методи теорії множин і відношень. Можна вважати, що цей розділ має призначення розгорнутого словника для успішного опанування матеріалів інших розділів. Разом з тим, поряд з теорією множин, вводиться проблематика нечітких множин та відношень. Таке подання матеріалу буде допомагати студентам успішно розв'язувати задачі з поданням знань експертів з захисту інформації, що надаються в якісній формі, у кількісній формі, яка пристосована для подальшої обробки.

У другому розділі подаються поняття та означення алгебри логіки, основні закони логіки, їх доведення за допомогою таблиць істинності. Також розглядаються форми подання функцій алгебри логіки, способи переходу до нормальних форм, побудова досконалої кон'юнктивної та диз'юнктивної нормальних форм. В поширеному вигляді наводяться відомості про основні методи мінімізації логічних функцій.

Третій розділ присвячено основам теорії інформації та кодування, що відповідає вимогам різкого зростання потоків інформації в сучасних інформаційних системах та зумовлює необхідність приділяти увагу процесам передавання, перетворення та зберігання інформації.

В четвертому розділі подаються основи логіко-лінгвістичних технологій, призначених для проєктування нечітких моделей оцінювання рівня безпеки інформації та розробки експертних систем захисту інформації.

В останньому розділі наведено індивідуальні та тестові завдання. В тестові завдання внесені задачі з теорії множин та відношень, алгебри логіки та основ кодування.

В основу посібника покладено матеріал, який пропонувався студентам на лекційних, лабораторних та практичних заняттях з дисципліни «Теоретичні основи процесів у кібербезпеці» для студентів зі спеціальності 125 «Кібербезпека» Вінницького національного технічного університету.

1 ЕЛЕМЕНТИ ТЕОРІЇ МНОЖИН ТА ВІДНОШЕНЬ

Одним з основних понять математики є поняття множини та її елементів. Під множиною *S* будемо розуміти будь-яку сукупність певних об'єктів, які можна розрізнити, але розглядаються вони як єдине ціле. Таке інтуїтивне означення множини дано засновником теорії множин Г. Кантором. Це поняття в математиці є первісним і не має точного означення, яке б задовольняло сучасну математику.

1.1 Основні поняття та операції теорії множин

Множиною називають сукупність деяких різних об'єктів, які можна розглядати як єдине ціле. Множини позначають великими буквами *A, B, C.*

Об'єкти, які складають множину, називають її елементами, позначають їх малими буквами *a, b, c.* Якщо елемент належить множині то записують $a \in A$, у противному випадку – $a \notin A$.

Множину, яка не має жодного елемента, називають *порожньою* і позначають символом *Ø*.

Множину, яка складається зі скінченного числа елементів, називають *скінченною*. Число елементів множини *A* називають *потужністю множини* і позначають $|A|$.

Належність елементів $a_1, a_2, ..., a_n$ до множини *A* позначають так:

$$A = \{a_1, a_2, ..., a_n\}, \text{ або } A = \{a_n\}, \text{ де } n = 1,2, ..., n.$$

Низка множин має загальноприйняті позначення:

$N = \{1, 2, 3, \ldots, n\}$ – множина натуральних чисел;

$Z = \{\ldots - 3, - 2, - 1, 0, 1, 2, \ldots\}$ – множина цілих чисел;

$Z_0 = \{0; 1; 2; \ldots\}$ – множина цілих невід'ємних чисел;

$Q = \left\{\frac{p}{q} : (p \in Z; q \in N)\right\}$ – множина раціональних чисел;

$R = \{x \mid x = \pm a_0, a_1, a_2, \ldots, a_n\}$ – множина дійсних чисел.

Множину *B* називають *підмножиною A*, якщо кожний елемент множини *B* належить множині *A*, тобто,

$$B \subset A \leftrightarrow (b \in B \rightarrow b \in A),$$

де $\subset$ – знак належності.

Якщо *A* є підмножиною *B*, а *B*, в свою чергу, є підмножиною *A*, то множини *A* і *B* містять ті самі елементи. Ці множини називають *рівними A = B.*

Якщо $A \subseteq B$ і $A \neq B$, то *A* має множини, елементами яких є тільки підмножини *A*. Кожна множина *A* має множини, елементами яких є тільки підмножини множини *A*. Сім'ю всіх підмножин даної множини *A* або булеан

цієї множини позначають через $B(A)$, а множину A називають *універсальною множиною* або *простором* і позначають через U.

Приклад. Розглянемо булеан $B(v)$, якщо $U = \{x, y\}$. Першою множиною буде порожня $\varnothing$, далі множини $A = \{x\}$ та B = $\{y\}$, які складаються з одного елемента, множина $C = \{x, y\}$, що складається з усіх елементів множини, тобто,

$$B = \{\varnothing; \{x\}; \{y\}; \{x, y\}\}.$$

Множину часто задають за допомогою характеристик її елементів. Наприклад, якщо характеристику властивості множини А, елементами якої є х, позначити через *P(x)*, то множину задають у вигляді

$$A = \{x \mid P(x)\}, \text{ або } A = \{x; P(x)\}.$$

Нехай $A = \{x \in R \mid x^2 - 5x + 6 = 0\}$ тобто A є множина всіх дійсних коренів рівняння $x^2 - 5x + 6 = 0$, або A=$\{2; 3\}$.

Множини часто задаються графічно, за допомогою діаграм Ейлера – Венна. Наприклад, на рис. 1.1 зображено множини {1, 2}; {1, 4, 5} в універсальному просторі U = {1, 2, 4, 5}.

Замкнену лінію, що обмежує елементи множини, називають кругом Ейлера. Прямокутник, у верхньому правому кутку якого зображено *U,* обмежує елементи універсальної множини.

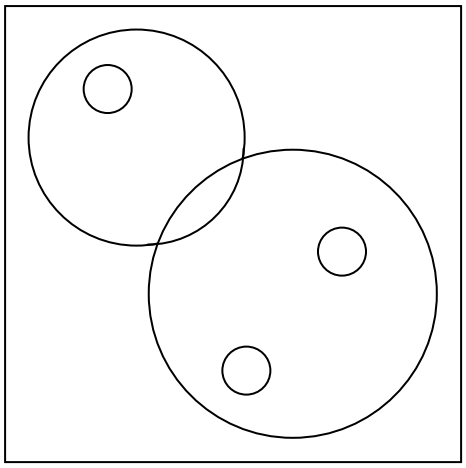


Рисунок 1.1 – Діаграм Ейлера-Венна

Розглянемо універсальну множину U і дамо означення чотирьох операцій над множинами.

Об'єднанням множин A і B називають множину C, яка містить елементи множини A та елементи множини B і позначають $A \cup B$

$$A \cup B = C = \{x \mid x \in Aa \text{ або } x \in B\}. \qquad (1.1)$$

Аналогічно позначають об’єднання довільного числа систем множин $\bigcup_{i=1}^{n} = A_i$. Наприклад, $A=\{1, 2, 3\}$, $B=\{0, 1\}$; $A \cup B = \{0,1,2,3\}$.

Властивості об’єднання множин

1. $A \cup A = A$.
2. $A \cup \varnothing = A$.
3. Якщо $A \supset B$,то $A \cup B = B$.
4. $A \cup B = B \cup A$.
5. $A \cup (B \cup C) = (A \cup B) \cup C$.

За допомогою діаграм Ейлера – Венна операцію об’єднання можна зобразити графічно (рис. 1.2).

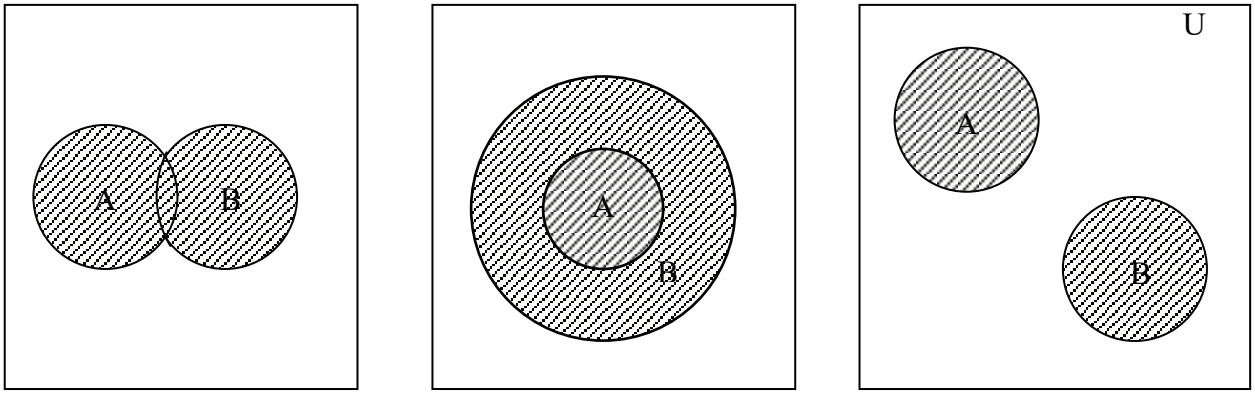


Рисунок 1.2 – Операція об’єднання

Перетином множин A і B називають множину C, яка містить ті і тільки ті елементи, які належать одночасно множинам $A \cap B$,

$$C = A \cap B = \{x : x \in A \text{ і } x \in B\} \qquad (1.2)$$

За допомогою діаграм Ейлера – Венна перетин можна зобразити таким чином (рис. 1.3).

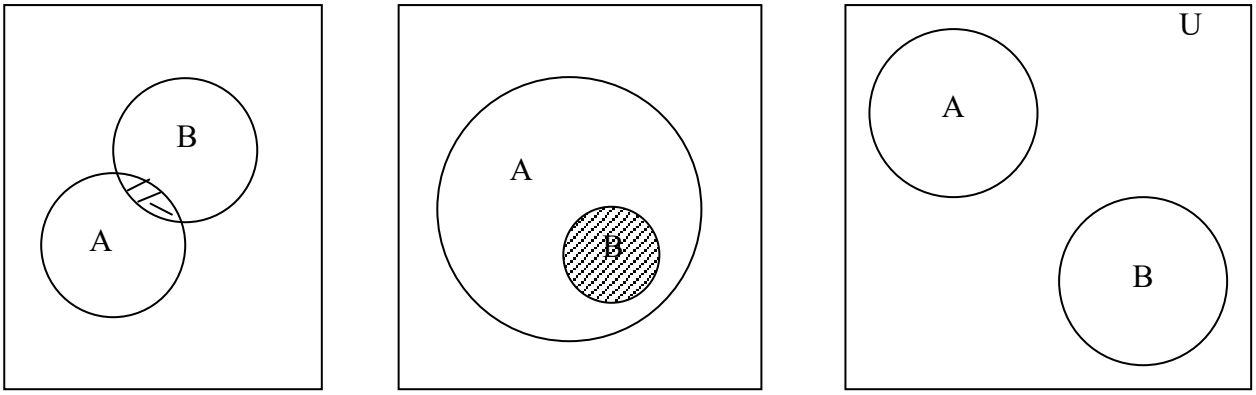


Рисунок 1.3 – Перетин множин A і B

Властивості перетину множин

1. $A \cap A = A.$
2. $A \cap \varnothing = \varnothing.$
3. $A \cap B = B \cap A.$
4. $A \cap (B \cap C) = (A \cap B) \cap C.$
5. Якщо $A \subset B$, то $A \subset B = A.$
6. $(A \cup B) \cap C = (A \cap C) \cup (B \cap C).$
7. $(A \cap B) \cup C = (A \cup C) \cap (B \cup C).$

Наприклад, $A = \{1,2,3\}, B = \{3,4,5\}, A \cap B = \{3\}.$

Аналогічно можна означити перетин довільної (зокрема нескінченної) системи множин $\bigcap_{i=1}^{n} A_i.$

Різницею множин A і B називають множину C, яка містить тільки ті елементи множини A, які не належать множині B (рис. 1.4), і позначають $A \setminus B$

$$C = A \setminus B = \{x \mid x \in A \text{ і } x \notin B\}. \qquad (1.3)$$

На відміну від перших двох операцій різниця множин означається лише для двох множин.

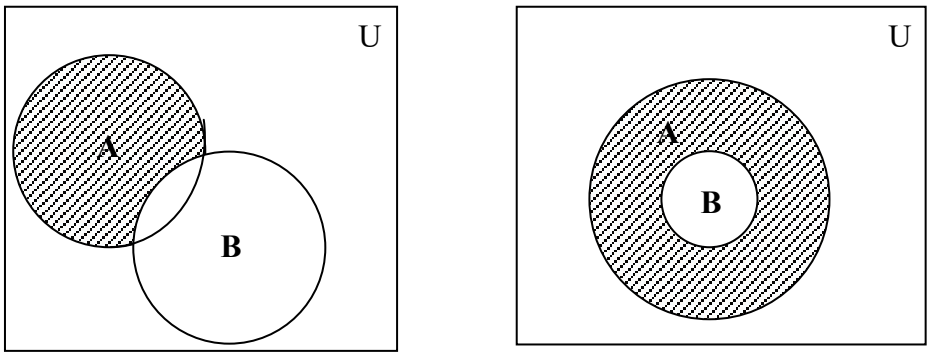


Рисунок1.4 – Різниця множин A і B

Властивості різниці множин

1. $A \setminus B \neq B \setminus A.$
2. Якщо $A \setminus B = \varnothing$, то $A \subseteq B.$

Наприклад, $U \setminus A$ називають доповненням до множини A і позначають $-A$ або $\overline{A}$.

Наприклад, $A = \{1,2,3\}, B = \{3,4,5\}, \quad A \setminus B. = \{1,\ 2\}.$

Різницею множин A і B називають множину

$$C = A - B = \{(A \setminus B) \cup (B \setminus A)\}. \qquad (1.4)$$

Теореми Моргана

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

Вважають, що теореми Моргана подвійні, тобто доповнення до перетину двох множин дорівнює об'єднанню їх доповнень; доповнення до об'єднань двох множин дорівнює перетину їх доповнень. Ці теореми переходять одна в одну при зміні слова *об'єднання* на *перетин* і, навпаки

$$\overline{(\overline{A})} = A \text{ (інволюція).}$$

1.2 Впорядковані множини. Поняття відношення

Одним з основних понять теорії множин є поняття декартового добутку множин.

Декартовим добутком $A \times B$ множин A і B називають множину

$$C = A \times B = \{(a,b) \mid a \in A, b \in B\}. \qquad (1.5)$$

Приклад 1. $A = \{1,2,3,4\}$.

$$A \times A = \{(1,1),(1,2),(1,3),(1,4),(2,1),(2,2),(2,3),(2,4),(3,1),(3,2),(3,3),(3,4),(4,1),(4,2),(4,3),(4,4)\}.$$

Приклад 2. Множина $R \times R = R^2$ є множиною точок на площині, тобто пар дійсних чисел (a, b), де $a \in R$ і $b \in R$, вони є координатами точок на площині.

Аналогічно можна ввести поняття декартового добутку n множин.

Декартовим добутком $A_1 \times A_2 \times,...,\times A_n = \prod_{i=1}^{n} A_i$ множин $A_1, A_2,..., A_n$ називають множину

$$C = \{(a_1, a_2,..., a_n) \mid a_1 \in A_1; a_2 \in A_2;...; a_n \in A_n\}. \qquad (1.6)$$

Якщо $A_1 = A_2 = ...= A_n$, то множину $C = A_1 \times A_2 \times, \ldots, \times A_n$ називають *прямим степенем* множини A і позначають A^n.

Підмножину $R \subseteq A^n$ називають *n-місним відношенням* на множині A. Інакше, a_1, a_2, …, a_n знаходяться у відношенні R, якщо $(a_1, a_2, \ldots, a_n) \in R$.

Одномісні, або унарні відношення R є підмножинами множини A. Властивості одномісного відношення R – це властивості підмножини A, тому термін «відношення» при $n = 1$ вживається дуже рідко.

Якщо $n = 2$, то відношення називають бінарним.

Бінарним відношенням між елементами множин $a \in A$ і $b \in B$ називають підмножину R декартового добутку $A \times B$. Якщо $A = B$, то R називають бінарним відношенням на A. Бінарні відношення позначають $(a, b) \in R$ або aRb.

Приклад 3. Відношення «≥» виконується для пар (10,6); (8,4) і не виконується для пар (4,8); (6,10).

Приклад 4. Для декартового добутку з прикладу 1 нижчевказані вирази є відношеннями: *R*1 = {(1,1), (2,1), (3,3), (4,4)} та *R*3 = {(1,2), (2,1), (1,3),(2,3)}.

Для задання бінарних відношень можна користуватися різними способами подання множин. Якщо множини скінченні, то, як правило, використовують матричний спосіб подання.

Нехай задано множину 2,4,6,8,10,12 і відношення $R: x \le y$. Розглянемо таблицю 1.1, де показано матричний спосіб подання відношення.

Таблиця 1.1

	2	4	6	8	10	12
2	1	1	1	1	1	1
4	0	1	1	1	1	1
6	0	0	1	1	1	1
8	0	0	0	1	1	1
10	0	0	0	0	1	1
12	0	0	0	0	0	1

Для будь-якої множини *A* відношення *E*, задане матрицею, в якій вздовж головної діагоналі стоять одиниці, а в інших місцях – нулі, називають *відношенням рівності* на *A*.

Областю визначення бінарного відношення називають множину

$$A_R=\{a \mid \text{ існує } b, \text{ таке, що } aRb\}. \tag{1.7}$$

Областю значень бінарного відношення називають множину

$$B_R=\{a \mid \text{ існує } b, \text{ таке, що } bRa\}. \tag{1.8}$$

Оскільки бінарні відношення є множинами, для них виконуються теоретико-множинні операції об'єднання, перетину та різниці:

$$R_1 \cup R_2;\ R_1 \cap R_2;\ R_1 \setminus R_2.$$

Приклад: нехай є бінарні відношення: *Q* = {(1,1), (1,2), (2,3), (3,4)} та *H* = {(0,0), (1,1), (1,2)} тоді результати операцій об'єднання перетину та різниці будуть такими:

$$Q \cup H = \{(0,0), (1,1), (1,2),(2,3),(3,4)\};$$
$$Q \cap H = \{(1,1)\};$$
$$Q \setminus H = \{(2,3), (3,4)\}.$$

Тільки для бінарного відношення *R* можна ввести поняття оберненого відношення *R*-1, для якого *aR*-1*b* справджується тоді і тільки тоді, коли *bRa*. Також для бінарного відношення визначають ліву область відношення R_- (ліві компоненти кортежів відношення), праву область R_+ (праві компоненти кортежів відношення) та поле відношення $F(R) = R_- + R_+$.

Приклад 5. Нехай є бінарне відношення: R={(1,2),(2,3),(3,4)}, тоді обернене відношення R-1= {(2,1), (3,2), (4,3)}. Для відношення R ліва область відношення буде дорівнювати $R_{\_}$ = {1,2,3}, права компонента відношення є R_{+} = { 2, 3, 4}. Поле відношення є $F(R) = R_{\_} + R_{+}$= {1, 2, 3, 4}.

1.3 Властивості бінарних відношень. Фактор-множина

Бінарне відношення R на множині A називають *рефлексивним відношенням*, якщо для будь-якого $a \in A$ виконується aRa ($(a,a) \in R$).

Головна діагональ рефлексивного відношення складається тільки з одиниць.

Бінарне відношення R називають *антирефлексивним* (іррефлексивним), якщо для всіх $a \in R$ не виконується відношення aRa ($(a,a) \in R$).

Головна діагональ матриці антирефлексивного відношення складається лише з нулів. Приклад рефлексивного та антирефлексивного відношень: є множина Y={1,2,3}, на цій множині відношення

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

є рефлексивним, антирефлексивним буде відношення

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

Бінарне відношення R називається *симетричним,* якщо для $(a,b) \in A$, aRb, то випливає, що bRa ($(a, b) \in R$, $(b, a) \in R$).

Матриця симетричного відношення є симетричною щодо головної діагоналі $b_{ij} = b_{ji}$ для будь-яких i та j.

Бінарне відношення R називають антисиметричним, якщо з aRb і bRa випливає $a = b$ (якщо $(a,b) \in R$, $(b,a) \in R$, то $a=b$).

Приклад симетричного відношення

$$S=\{(1,3), (2,1), (3,1), (1,2)\}.$$

Приклади антисиметричних відношень:

$$M = \{(1,2), (2,3), (3,4)\};$$
$$R = \{(1,1), (1,2), (2,2), (2,3),(3,3)\}.$$

Бінарне відношення є *транзитивним*, якщо для будь-яких $a, b, c \in A$ з aRb; bRc випливає, що aRc (якщо $(a,b) \in R$, $(b,c) \in R$, то $(a,c) \in R$).

Приклад транзитивного відношення

$$A=\{(1,1), (1,2), (1,4), (2,2), (2,4), (4,4)\}.$$

Переріз відношення. Фактор-множина

Нехай R – відношення від A до B, тобто,

$$R \subset A \times B$$

та елемент a належить A, тоді множина Ra – це всі елементи b, що належать B, для яких існує кортеж (a,b), що належить відношенню R. Множина Ra називається *перерізом відношення* R за елементом a. Множину всіх перерізів відношення називають *фактор-множиною*.

Приклад побудови перерізів відношення. На множині $A=B=\{1,2,3,4\}$ маємо відношення

$$R=\{(1,2),(2,1),(1,3),(2,3),(3,3),(3,4)\}.$$

Тоді:

Переріз за елементом $a = 1$ $R1=\{2,3\}$;

Переріз за елементом $a = 2$ $R2=\{1,3\}$;

Переріз за елементом $a = 3$ $R3=\{4\}$.

Властивість еквівалентності відношення. Класи еквівалентності

Бінарне відношення на множині A називають *еквівалентним*, якщо воно рефлексивне, симетричне та транзитивне.

Приклад відношення еквівалентності. На множині

$$B = \{1,2,3,4,5,6,7,8,9,10\}$$

задане відношення: «елемент a перебуває у відношенні з елементом b, якщо a і b мають однакову кількість натуральних дільників». Будуємо відношення, яке буде еквівалентним.

$R=\{(1,1), (2,2), (2,3),(2,5),(2,7), (3,2), (3,3),(3,5),(3,7), (4,4), (4,9),(5,2), (5,3), (5,5), (5,7), (6,6), (6,8), (6,10), (7,2), (7,3), (7,5), (7,7), (8,6), (8,8), (8,10), (9,4), (9,9), (10,6), (10,8), (10,10)\}$.

Нехай на множині A задано відношення еквівалентності R. Виберемо елемент $a_1 \in A$ і утворимо клас A_1 (підмножину A), що складається з a_1 та всіх елементів, еквівалентних a_1; потім виберемо елемент $a_2 \notin A_1$ і утворимо клас A_2, що складається з a_2 та всіх елементів, еквівалентних a_2 і т. д.

Уявімо систему класів A_1, A_2, ... таку, що довільний елемент множини A належить хоча б одному класу, тобто $\cup A_i=A$. Ця система класів еквівалентності має такі властивості: вона створює класи, які попарно не перетинаються; будь-які елементи різних класів нееквівалентні, а з одного – еквівалентні.

Множини класів еквівалентності множини A за еквівалентністю R називають *фактор-множиною* A і позначають $A \mid R$.

Відношення порядку. Мажоранта та міноранта

Бінарне відношення на множині A називають *відношенням нестрогого порядку* (часткового порядку), якщо воно є рефлексивним, антисиметричним і транзитивним. Частковий порядок часто позначають символом $\leq$.

Бінарне відношення на множині A називають *відношенням строгого порядку*, якщо воно є антирефлексивним, антисиметричним та транзитивним. Строгий порядок позначають символом $<$.

Обидва типи відношень називають *відношенням порядку*. Елементи $a,b \in A$ називають *порівнянними за відношенням порядку*, якщо aRb або bRa.

Приклад відношення нестрогого порядку. Нехай на множині $B = \{1,2,3,4\}$ задане відношення $R =: \leq$. Будуємо це відношення.

$$R1 = \{ (1,1),(1,2),(2,2),(2,3),(2,4),(3,3), (3,4),(4,4),(1,3),(1,4) \}.$$

Приклад відношення строгого порядку. Нехай на множині $B = \{1,2,3,4\}$ задане відношення $R =: <$. Будуємо це відношення.

$$R2 = \{ (1,2), (2,3),(2,4), (3,4),(1,3),(1,4) \}.$$

Частковий порядок на множині А називають *лінійним*, якщо будь-які два елементи множин A можна порівняти за $\leqslant$, тобто $a \leqslant b$ або $b \leqslant a$ для $a, b \in A$. Множину A із заданим на ній частковим порядком називають частково впорядкованою. Розглянемо підмножину A' впорядкованої множини A.

Якщо існує елемент $b \in A$ такий, що всі $a \leqslant b$ належить підмножині A', говорять, що елемент b – *мажоранта* підмножини A'.

Аналогічно, якщо $a \leqslant b$, то кажуть, що b – *міноранта* підмножини A'.

Якщо множина мажорант (мінорант) має, в свою чергу, найбільший (найменший) елемент, то цей елемент називають *верхньою* (*нижньою*) *межею* множини A' і позначають $sup\ A'$ ($inf\ A'$).

Приклад знаходження мажоранти та міноранти. Нехай задана множина $A=\{1,2,3,4\}$ та отримано відношення порядку $R =: \leq$, яке має вигляд

$$R=\{(1,1),(1,2),(2,2),(2,3),(2,4),(3,3), (3,4),(4,4),(1,3),(1,4)\}.$$

Знайти мажоранту та міноранту для множини $B = \{1,2\}$.

Визначення мажоранти для елементів множини 1 та 2

$$1 \leq 2 \qquad 1 \leq 3 \qquad 1 \leq 4$$

$$2 \leq 2 \qquad 2 \leq 3 \qquad 2 \leq 4$$

Таким чином множина мажорант: $B+ = \{2,3,4\}$ supremum $B+ = \min B+ = 2$.

Визначення міноранти здійснюємо відповідно до кортежів відношення R для елементів 1 та 2

$$1 \leq 1 \qquad 1 \leq 2$$

Множина мінорант $B- = \{1\}$ infinum $B- = \max B- = 1$

Властивість функціональності відношення. Відображення

Відношення $F \subset A \times B$ називають *функціональним*, функцією, якщо для кожного $x \in A$ переріз F за x містить не більше одного елемента.

Всюди визначене в A функціональне відношення називається відображенням.

1.4 Основні поняття та операції теорії нечітких множин

Нечітка множина є сукупністю елементів довільної природи, щодо яких не можна з повною визначеністю сказати належить той чи інший елемент розглянутій сукупності даної множини чи ні.

Нехай $X = \{x\}$ – універсальна множина (універсум), тобто повна множина, що охоплює всю проблемну область.

Нечітка множина типу 1 $A \subseteq X$ є набором пар $\{(x, \mu_A(x))\}$, де $x \in X$ і $\mu_A : X \to [0,1]$ – функція належності, що є деякою суб'єктивною мірою відповідності елемента x нечіткій множині A.

Функція $\mu_A(x)$ може приймати значення від нуля, що позначає абсолютну неналежність, до одиниці, що, навпаки, говорить про абсолютну належність елемента x нечіткій множині A.

Якщо множину [0,1] замінити на {0,1}, то функція належності буде характеристичною функцію звичайної (не нечіткої) множини.

Якщо нечітка множина (типу 1) A визначена на кінцевій універсальній множині $X = \{x_1, x_2, ..., x_n\}$, то її зручно позначати так:

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + ... + \mu_A(x_n)/x_n = \sum_{i=1}^{n} \mu_A(x_i)/x_i \qquad (1.9)$$

де «$\mu_A(x_i)/x_i$» – пара «функція належності / елемент», яка називається синглтоном, а « + » позначає сукупність пар.

На практиці зручно використовувати кусково-лінійну апроксимацію функції належності нечіткої множини як це показано на рис. 1.5, тому що потрібно тільки два значення – $\underline{a}$ й $\overline{a}$.

У випадку неперервної множини X використовується таке позначення:

$$A = \int_X \mu_A(x)/x.$$

Знак $\int$ в цих формулах позначає сукупність пар $\mu_A(x)/x$).

Нечітка множина $A \subseteq X$ пуста, тобто $A = \varnothing$, якщо $\mu_A(x) = 0$, $\forall x \in X$.

Нечіткі множини A і $B \subseteq X$ еквівалентні, тобто $A = B$, якщо $\mu_A(x) = \mu_B(x), \forall x \in X$.

Нечітка множина $A \subseteq X$ є підмножиною нечіткої множини $B \subseteq X$, тобто $A \subseteq B$, якщо $\mu_A(x) \le \mu_B(x), \forall x \in X$.

Кардинальне число (потужність) нечіткої множини A знаходиться так:

$$cardA = |A| = \sum_{i=1}^{n} \mu_A(x_i).$$

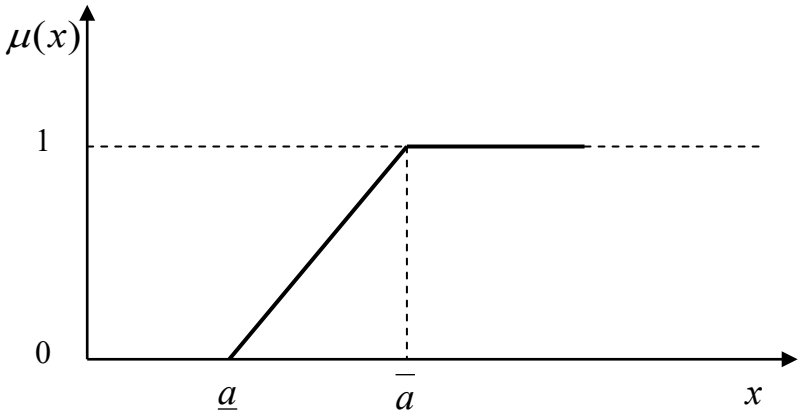


Рисунок 1.5 – Функція належності нечіткій множині

Носієм нечіткої множини A називається звичайна множина A_S, що містить тільки ті елементи універсума, для яких значення функції належності відповідної нечіткої множини відмінні від нуля. Математично носій нечіткої множини визначається умовою

$$A_S = \{x \in X \mid \mu_A(x) > 0\}.$$

Нечітка множина називається скінченною, якщо її носій є скінченною множиною. При цьому така нечітка множина має скінченну потужність, що чисельно дорівнює кількості елементів його носія як звичайної множини.

Аналогічним способом можна визначити й нескінченні нечіткі множини як такі нечіткі множини, носій яких не є скінченною множиною.

Нечіткі множини можуть бути задані двома основними способами:

1. У формі списку з явним переліком всіх елементів і відповідних їм значень функції належності, що утворюють розглянуту нечітку множину. При цьому найчастіше елементи з нульовими значеннями функції належності просто не вказуються в даному списку. Цей спосіб підходить для задання нечітких множин зі скінченним дискретним носієм і невеликим числом елементів. У цьому випадку нечітку множину зручно записувати у вигляді (1.1) або

$$A = \{< x_1, \mu_A(x_1) >, < x_2, \mu_A(x_2) >, ..., < x_n, \mu_A(x_n) >\}.$$

2. Аналітично у формі математичного виразу для відповідної функції належності. Цей спосіб може бути використаний для задання довільних нечітких множин як зі скінченним, так і з нескінченним носієм. У цьому випадку нечітку множину зручно записувати у вигляді:

$$A = \{< x, \mu_A(x) >\}$$
$$A = \{x, \mu_A(x)\}$$

де μ_A – деяка функція, задана аналітично у формі математичного виразу $f(x)$ або графічно у формі деякої кривої.

Зрізом (множиною рівня a) нечіткої множини $A \subseteq X$ називається (чітка) множина $A_a \subseteq X$ така, що

$$A_\alpha = \{x \in X : \mu_A(x) \geq \alpha\}, \forall \alpha \in [0,1].$$

У свою чергу, довільна функція належності $\mu(x)$ називається унімодальною на інтервалі $[a,b] \subset R$, якщо вона неперервна на $[a,b]$, а також існує деякий непустий $[c,d] \subset [a,b]$ такий, що $a \leq c \leq d \leq b$ і виконуються умови:

– функція $\mu(x)$ строго монотонно зростає на інтервалі $[a,c]$ при $a < c$;
– функція $\mu(x)$ строго монотонно спадає на інтервалі $[d,b]$ при $d < b$;
– функція $\mu(x)$ приймає своє максимальне значення на інтервалі $[c,d]$, тобто будь-яка точка $x_m \in [c,d]$ є точкою максимуму функції належності відносно інтервалу $[a,b]$

$$x_m = \arg\max_{x \in [a,b]} \{\mu(x)\}.$$

Функція належності $\mu_A(x)$ називається унімодальною (строго унімодальною), якщо вона унімодальна (строго унімодальна) на носії відповідної нечіткої множини A.

Ядром нечіткої множини A називається така звичайна множина A_{I}, елементи якої задовольняють умову $A_{\text{I}} = \{x \in X \mid \mu_A(x) = 1\}$.

Межами нечіткої множини A називаються такі елементи універсума, для яких значення функції належності відмінні від 0 і 1 ($0 < \mu_A(x) < 1$).

Елементи нечіткої множини $y \in A$, для яких виконується умова $\mu_A(y) = 0.5$, називаються точками переходу цієї нечіткої множини A.

Для характеристики нечітких множин використовують також поняття опуклості, що асоціюється з відповідним графічним зображенням функції належності.

Нечітка множина A з універсумом X називається опуклою, якщо її функція належності $\mu_A(x)$ задовольняє нерівність

$$\mu_A(x) \geq \min\{\mu_A(a), \mu_A(b)\}$$

для будь-яких значень $x, a, b \in X$, при яких $a<x<b$ і $a \neq b$.

Якщо A, B і C – нечіткі підмножини універсальної множини X, то виконуються такі властивості:

$$\left.\begin{aligned} A \cap B = B \cap A, \\ A \cup B = B \cup A, \end{aligned}\right\} \text{ комутативність} \qquad (1.10)\ (1.11)$$

$$\left.\begin{aligned} (A \cap B) \cap C = A \cap (B \cap C), \\ (A \cup B) \cup C = A \cup (B \cup C), \end{aligned}\right\} \text{ асоціативність} \qquad (1.12)\ (1.13)$$

$$\left.\begin{aligned} A \cap A = A, \\ A \cup A = A, \end{aligned}\right\} \text{ ідемпотентність} \quad (1.14)\ (1.15)$$

$$\left.\begin{aligned} A \cap (B \cup C) = (A \cap B) \cup (A \cap C), \\ A \cup (B \cap C) = (A \cup B) \cap (A \cup C), \end{aligned}\right\} \text{ дистрибутивність} \quad (1.16)\ (1.17)$$

$$A \cap \varnothing = \varnothing, \quad (1.18)$$

де $\varnothing$ – звичайна множина, така, що $\forall x_i \in X : \mu_{\varnothing}(x_i) = 0$.

$$A \cup \varnothing = A, \quad (1.19)$$

$$A \cap X = A, \quad (1.20)$$

де X – універсальна множина ($\forall x_i \in X : \mu_X(x_i) = 1$).

$$A \cup X = X, \quad (1.21)$$

$$\overline{\overline{A}} = A \text{ – інволюція}, \quad (1.22)$$

$$\left.\begin{aligned} \overline{A \cap B} = \overline{A} \cup \overline{B}, \\ \overline{A \cup B} = \overline{A} \cap \overline{B}. \end{aligned}\right\} \text{ теореми де Моргана} \quad (1.23)\ (1.24)$$

Розглянемо ***Основні операції над нечіткими множинами***.

Включення. Нехай X – універсальна множина, A і B – дві нечіткі підмножини X; будемо говорити, що A міститься в B, якщо

$$\forall x \in X : \mu_A(x) \le \mu_B(x), \quad (1.25)$$

і позначати $A \subset B$ або $A \subseteq B$.

Строге включення відповідає випадку, коли в (1.25) хоча б одна нерівність строга, позначається: $A \subset\subset B$.

Рівність. Нехай X – універсальна множина, A і B – дві нечіткі підмножини X; скажемо, що A і B рівні тоді і тільки тоді, коли

$$\forall x \in X : \mu_A(x) = \mu_B(x), \quad (1.26)$$

і будемо позначати $A = B$.

Якщо знайдеться принаймні один такий елемент x із X, що рівність $\mu_A(x) = \mu_B(x)$ не виконується, то будемо говорити, що $A \ne B$.

Нехай X – універсальна множина, A – нечітка підмножина X.

Доповненням нечіткої множини A називається нечітка множина $\neg A$ (або $\overline{A}$), функція належності якої дорівнює

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \forall x \in X. \quad (1.27)$$

Операція доповнення нечіткої множини A може бути проілюстрована графічно (рис. 1.6). Функції належності є лінійними Z-подібними функціями. При цьому результату операції доповнення $\neg A$ відповідає заштрихована ділянка на графіку. Неважко побачити, що графік функції належності доповнення нечіткої множини симетричний графіку функції належності вихідної нечіткої множини відносно лінії $\mu(x) = 0.5$.

Перетин. Перетином двох нечітких множин A і $B \subseteq X$ називається нечітка множина $A \cap B$, яку визначають як найбільшу нечітку множину, що одночасно міститься в A і B, і функція належності якої дорівнює

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), \forall x \in X . \qquad (1.28)$$

Або за виразом $\mu_{A \cap B}(x) = \{\mu_A(x) \cap \mu_B(y)\}$, де символ $\cap$ може замінюватись на символ, схожий на знак кон'юнкції $\wedge$, що означає мінімум.

Результат перетину двох нечітких множин, заданих на одному і тому ж універсумі X, також можна зобразити графічно в декартовій системі координат на площині (рис. 1.7).

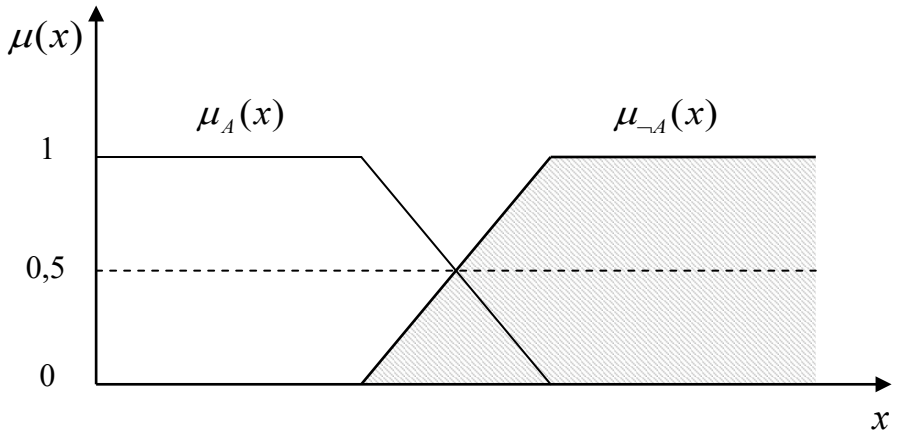


Рисунок 1.6 – Операція доповнення нечіткої множини A

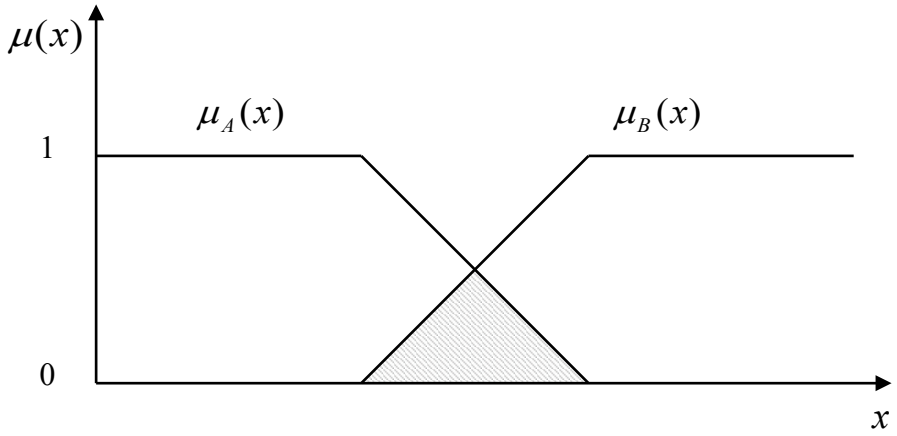


Рисунок 1.7 – Операція перетину двох нечітких множин A і B

Об'єднання. Об'єднанням двох нечітких множин A і $B \subseteq X$ називається нечітка множина $A \cup B$, яку визначають як найменшу нечітку множину, що містить як A, так і B, і функція належності якої дорівнює

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), \forall x \in X . \quad (1.29)$$

Або за виразом $\mu_{A \cup B}(x) = \{\mu_A(x) \bigcup \mu_B(y)\}$, де символ $\cup$ може замінюватись на символ, схожий на знак диз'юнкції $\vee$, що означає максимум.

Результат об'єднання двох нечітких множин, заданих на одному і тому ж універсумі X, зображено графічно на рис. 1.8.

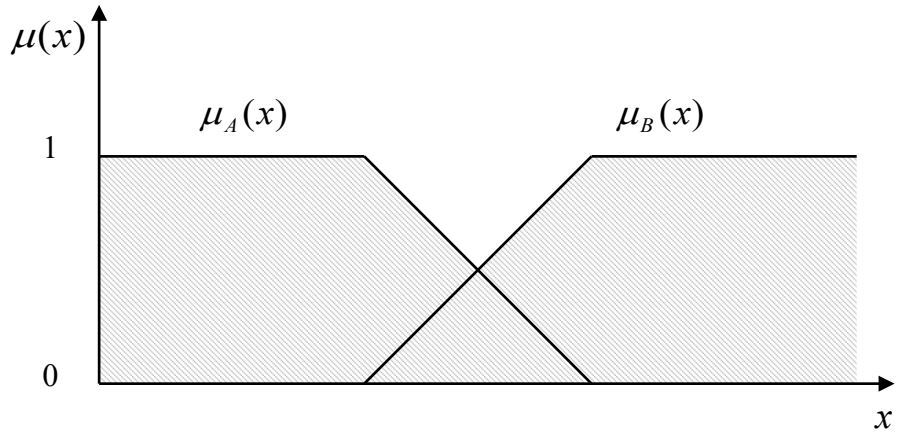


Рисунок 1.8 – Графічне подання операції об'єднання двох нечітких множин A і B, заданих лінійними Z-подібними функціями належності

1.5 Поняття нечіткого відношення та його властивості

Зазначимо, що, оскільки звичайне відношення подається як підмножина декартового добутку, то перехід до нечіткого відношення є таким самим як перехід від звичайної множини до нечіткої. Опис нечіткого відношення має містити не тільки перерахунок всіх пар елементів вихідної множини, які пов'язані цім відношенням, але і числа з інтервалу [0,1], що відображають ступінь виконання нечіткого відношення для цих пар. Також опис відношення містить множину, на якій це відношення побудовано.

Нечітким відношенням R на множині X називається нечітка підмножина декартового добутку $X \times X$, що містить функцію належності

$$\mu_R :. X \times X \rightarrow [0,1].$$

Значення функції належності розуміють як ступінь виконання відношення R. Звичайне відношення можна розглядати як окремий випадок нечіткого відношення, функція належності якого приймає значення 0 або 1.

Під носієм нечіткого відношення розуміють звичайне відношення на множині *X*, що пов'язує всі пари (*x*, *y*), для яких ступінь виконання даного нечіткого відношення не дорівнює нулю.

Поширеним є використання в задачах прийняття рішень множинами множин певного рівня нечіткого відношення. Оскільки нечітке відношення визначається як нечітка множина, то і його множини рівня подаються так:

$$R_\alpha = \{(x, y) \mid (x, y) \in X \times X, \mu_R(x, y) \geq \alpha\}.$$

Неважко побачити, що множина рівня α нечіткого відношення *R* на *X* є звичайним відношенням на *X*, яке пов'язує всі пари (*x*, *y*), для яких ступінь виконання відношення *R* не менший α (приклади наведено в табл.1.2).

Таблиця 1.2 – Матриця нечіткого відношення *R* на множині $X = \{x_1, x_2, x_3, x_4\}$.

	x_1	x_2	x_3	x_4
x_1	1	0.5	0	0.2
x_2	0.3	1	1	0.3
x_3	0	0.6	0.5	0.4
x_4	1	0.7	0.1	0

Тоді матриця звичайного відношення, яке є множиною рівня 0.5 цього нечіткого відношення, буде мати вигляд, наведений в табл. 1.3.

Таблиця 1.3 – Матриця звичайного відношення, нечіткого відношення *R*

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

1.6 Операції з нечіткими відношеннями

Нехай на множині *X* задано два нечітких відношення *A* та *B*, тобто в декартовому добутку *X*×*X* є дві нечітких множини *A* та *B*. Нечіткі множини $C=A \cup B$ та $D=A \cap B$ є, відповідно, об'єднанням та перетином нечітких відношень *A* та *B* на множині *X*.

Для функцій належності відношень *C* та *D* будемо мати вирази:

$$\mu_c(x, y) = \max\{\mu_A(x, y), \mu_B(x, y)\},$$
$$\mu_B(x, y) = \min\{\mu_A(x, y), \mu_B(x, y)\}.$$

Приклад операції об'єднання нечітких відношень. Нехай задано матрицю нечіткого відношення $R1$ на множині $X = \{x_1, x_2, x_3\}$. та матрицю нечіткого відношення $R2$ на множині $Y = \{y_1, y_2, y_3\}$ відповідно до табл. 1.4 та 1.5. Тоді таблиця 1.6 – це результат об'єднання відношень $R1$ та $R2$; таблиця 1.7 – це результат перетину відношень $R1$ та $R2$.

Таблиця 1.4 – Матриця нечіткого відношення $R1$

	y_1	y_2	y_3
x_1	1	0.5	0
x_2	0.3	1	1
x_3	0	0.6	0.5

Таблиця 1.5 – Матриця нечіткого відношення $R2$

	y_1	y_2	y_3
x_1	0.9	0.1	0
x_2	0.6	1	0
x_3	0	0.8	0.7

Таблиця 1.6 – Результат об'єднання $R1 \cup R2$

	y_1	y_2	y_3
x_1	1	0.5	0
x_2	0.6	1	1
x_3	0	0.8	0.7

Таблиця 1.7 – Результат перетину $R1 \cap R2$

	y_1	y_2	y_3
x_1	0.9	0.1	0
x_2	0.3	1	1
x_3	0	0.6	0.5

Операція композиції (згортки) для нечітких відношень. Розглянемо співвідношення для максмінної композиції, мінмаксної композиції та максмультиплікативної композиції. Наведемо приклади згортки для нечітких відношень A та B в табл.1.8, 1.9.

Таблиця 1.8 – Матриця нечіткого відношення A

	z_1	z_2
x_1	0.2	0.6
x_2	0.5	0.8

Таблиця 1.9 – Матриця нечіткого відношення *B*

	y_1	y_2
z_1	0.5	0.7
z_2	0.3	1

Максмінна композиція $A°B$ нечітких відношень A (див. табл.1.8) і B (табл. 1.9) на множині X має функцію належності $\mu_{A\circ B}(x,y) = \sup\min\{\mu_A(x,z), \mu_B(z,y)\}$.

Результати наведені в таблиці 1.10.

Таблиця 1.10 – Матриця максмінної композиції $A°B$:

	y_1	y_2
x_1	0.3	0.6
x_2	0.3	0.5

Мінмаксна композиція $A°B$ нечітких відношень A і B на множині X має функцію належності

$$\mu_{A\circ B}(x,y) = \inf\max\{\mu_A(x,z), \mu_B(z,y)\}.$$

Результати наведено в таблиці 1.11.

Таблиця 1.11 – Матриця мінімаксної композиції $A°B$

	y_1	y_2
x_1	0.5	0.7
x_2	0.5	0.7

Мультиплікативна композиція $A°B$ нечітких відношень A і B на множині X має функцію належності

$$\mu_{A\circ B}(x,y) = \sup\{\mu_A(x,z)\times\mu_B(z,y)\}.$$

Приклад мультиплікативної композиції $A°B$ нечітких відношень A і B на множині X показано у табл. 1.12.

Таблиця 1.12 – Матриця мультиплікативної композиції $A°B$

	y_1	y_2
x_1	0.18	0.6
x_2	0.25	0.8

1.7 Завдання для самостійної роботи

Тема. Множини та відношення

Варіант № 1

1. Зобразити множини на діаграмах Венна і довести рівність

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

2. Матричним та графічним способами задайте бінарне відношення на множинах:

   $A=\{1, 2, 3, 4, 5, 6, 7, 8\}$;
   $B=\{2, 9, 11, 17, 18, 19, 20\}$;
   $R=\{(a, b)|a\in A, b\in B, a > b\}$.

3. Для заданого відношення

   P:{<1,1>, <1,2>, <1,3>, <1,4>, <1,5>, <1,6>, <2,2>, <2,4>, <2,5>, <2,6>, <3,3>, <3,4>, <3,5>, <3,6>, <4,4>, <4,6>, <5,5>, <5,6>, <6,6>};
   B ={4,5};

визначте:

   a) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
   b) міноранту та мажоранту множини B;
   c) точні верхню та нижню межі множини B.

Варіант № 2

4. Зобразити множини на діаграмах Венна і довести рівність

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

5. Матричним та графічним способами задайте бінарне відношення на множинах:

   $A=\{1, 2, 3, -4, 5, 6, 7, -8\}$;
   $B=\{2, -9, 11, -17, 18, -19, 20\}$;
   $R=\{(a, b)|a\in A, b\in B, a + b < 0\}$.

6. Для заданого відношення P

   P : {<1,1>, <2,2>, <2,3>, <2,4>, <2,5>, <3,3>, <3,4>, <3,6>, <3,7>, <3,8>, <4,4>, <4,6>, <4,8>, <6,6>, <6,8>, <7,7>};
   B = { 3,4 };

визначте:

   a) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
   b) міноранту та мажоранту множини B;
   c) точні верхню та нижню межі множини B.

Варіант № 3

1. Зобразити множини на діаграмах Венна і довести рівність

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

2. Матричним та графічним способами задайте бінарне відношення на множинах:

   A={1, 2, 3, 4, 5, 6, 7, 8};
   B={2, 9, 11, 17, 18, 19, 20};
   $R=\{(a, b)|a\in A, b\in B, a = b^2\}$.

3. Для заданого відношення P

   P:{<1,1>, <1,3>, <1,4>, <1,6>, <1,7>, <1,8>, <3,3>, <3,6>, <3,7>, <3,8>, <4,4>, <4,6>, <4,8>, <6,6>, <6,8>, <7,7>, <7,8>, <8,8>};
   B = {6};

визначте:

   a) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
   b) міноранту та мажоранту множини B;
   c) точні верхню та нижню межі множини B.

Варіант № 4

1. Зобразити множини на діаграмах Венна і довести рівність

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

2. Матричним та графічним способами задайте бінарне відношення на множинах:

   A={1, 2, 3, 4, 5, 6, 7, 8};
   B={2, 9, 11, 17, 18, 19, 20};
   $R=\{(a, b)|a\in A, b\in B, a$ – взаємо прості b$\}$.

3. Для заданого відношення P

   P:{<1,1>, <1,4>, <1,5>, <1,7>, <1,8>, <2,2>, <2,4>, <2,5>, <2,7>, <2,8>, <4,4>, <4,5>, <4,7>, <5,5>, <5,7>, <5,8>, <7,7>, <7,8>};
   B = {4};

визначте:

   a) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
   b) міноранту та мажоранту множини B;
   c) точні верхню та нижню межі множини B.

Варіант № 5

1. Зобразити множини на діаграмах Венна і довести рівність

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

2. Матричним та графічним способами задайте бінарне відношення на множинах:

A={1, 2, 3, 4, 5, 6, 7, 8};
B={2, 9, 11, 17, 18, 19, 20};
$R=\{(a, b)|a\in A, b\in B, a \geq b\}$.

2. Для заданого відношення P

P:{<1,1>, <1,2>, <1,3>, <1,4>, <1,5>, <1,6>, <2,2>, <2,5>, <2,6>, <3,3>, <3,5>, <3,6>, <4,4>, <4,5>, <4,6>, <5,5>, <5,6>, <6,6>};
B = {2, 3, 4};

визначте:

a) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
b) міноранту та мажоранту множини B;
c) точні верхню та нижню межі множини B.

Варіант № 6

1. Зобразити множини на діаграмах Венна і довести рівність

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

2. Матричним та графічним способами задайте бінарне відношення на множинах:

A={1, 2, 3, 4, 5, 6, 7, 8};
B={2, 9, 11, 17, 18, 19, 20};
$R=\{(a, b)|a\in A, b\in B, a \geq b^2\}$.

3. Для заданого відношення P

P:{<1,1>, <2,2>, <3,3>, <3,4>, <3,5>, <3,6>, <3,7>, <3,8>, <4,4>, <4,5>, <4,6>, <4,7>, <4,8>, <5,5>, <5,6>, <5,8>, <6,6>, <6,7>, <6,8>, <7,7>, <8,8>};
B = {4, 5, 6, 7};

визначте:

a) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
b) міноранту та мажоранту множини B;
c) точні верхню та нижню межі множини B.

Варіант № 7

1. Зобразити множини на діаграмах Венна і довести рівність

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

2. Матричним та графічним способами задайте бінарне відношення на множинах:

   A={1, 2, 3, 4, 5, 6, 7, 8};
   B={2, 9, 11, 17, 18, 19, 20};
   R={(a, b)|$a \in A$, $b \in B$, $a + b$ - непарне}.

3. Для заданого відношення P

   P:{<1,1>, <1,2>, <1,3>, <1,4>, <2,2>, <2,3>, <2,5>, <2,7>, <2,8>, <3,3>, <3,4>, <3,5>, <3,7>, <4,4>, <5,5>, <5,6>, <5,7>, <5,8>, <6,6>, <7,7>, <7,8>};
   B = {3, 4, 5};

визначте :

a) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
b) міноранту та мажоранту множини B;
c) точні верхню та нижню межі множини B.

Варіант № 8

1. Зобразити множини на діаграмах Венна і довести рівність

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

2. Матричним та графічним способами задайте бінарне відношення на множинах:

   A={1, 2, 3, 4, 5, 6, 8, 10, 12};
   B={1, 2, 4, 6, 8, 10, 14};
   R={(a, b)|$a \in A$, $b \in B$, $a + b$ – ділиться на 4};

3. Для заданого відношення P

   P:{<1,1>, <1,2>, <1,3>, <1,4>, <1,5>, <1,6>, <1,7>, <2,2>, <3,3>, <3,1>, <3,2>, <3,4>, <3,5>, <3,8>, <4,4>, <5,5>, <5,6>, <5,7>, <5,8>, <6,6>};
   B = {3, 4};

визначте:

a) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
b) міноранту та мажоранту множини B;
c) точні верхню та нижню межі множини B.

Варіант № 9

1. Довести тотожність

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

2. Матричним та графічним способами задайте бінарне відношення на множинах:

   A={1, 2, 3, 4, 5, 6, 8, 10, 11}
   B={2, 4, 5, 6, 7, 10, 14}
   R={((a, b)|$a \in A$, $b \in B$, $a + b$ - парне}

3. Для заданого відношення P

   P:{<1,1>, <1,2>, <1,3>, <1,4>, <1,5>, <2,2>, <2,5>, <2,6>, <2,7>, <2,8>, <3,3>, <4,4>, <4,6>, <4,7>, <4,8>, <5,5>, <5,8>, <6,6>, <7,7>, <7,8>};
   B = {2, 3};

визначте:

   а) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
   b) міноранту та мажоранту множини B;
   c) точні верхню та нижню межі множини B.

×

Варіант № 10

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

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

2. Матричним та графічним способами задайте бінарне відношення:

   A={1, 2, 3, 4, 5, 6, 7, 10, 12};
   B={3, 4, 5, 6, 8};
   R={(a, b)|$a \in A$, $b \in B$, $a = b$}.

3. Для заданого відношення P

   P:{<1,1>, <1,2>, <1,3>, <1,4>, <1,5>, <1,6>, <2,2>, <2,3>, <2,4>, <2,7>, <2,8>, <3,3>, <3,5>, <3,6>, <4,4>, <4,7>, <4,8>, <5,5>, <5,6>, <6,6>, <6,7>, <6,8>, <7,7>, <7,8>};
   B = {3, 4, 5};

визначте:

   а) ліву $P\_$ та праву P_+ області та поле відношення $F(P)$;
   b) міноранту та мажоранту множини B;
   c) точні верхню та нижню межі множини B.

1.8 Контрольні питання

1. Поняття множини та підмножини. Які приклади множин ви знаєте?
2. Як прийнято позначати множину натуральних, цілих, невід'ємних чисел?
3. Які множини називають рівними?
4. Яку множину називають порожньою? Як її позначають?
5. Як позначається відношення включення множини, коли підмножина A є власною підмножиною B?
6. Що таке потужність множин?
7. Дайте означення основних операцій алгебри множин?
8. Яку множину називають декартовим (прямим) добутком двох і більше множин?
9. Що таке порожня множина?
10. Що таке бінарне відношення?
11. Що таке ліва, права області і поле бінарного відношення?
12. Яким чином визначені теоретико-множинні операції для бінарних відношень?
13. Дайте означення зворотного відношення для бінарного відношення.
14. Визначення спеціальних бінарних відношень. Коли бінарне відношення називається рефлексивним і антирефлексивним?
15. Дайте означення симетричного і антисиметричного бінарних відношень.
16. Як формулюється властивість транзитивності бінарних відношень?
17. Яке бінарне відношення називають еквівалентним?
18. Що таке класи еквівалентності?
19. Яке бінарне відношення називають відношенням нестрогого порядку?
20. Яке бінарне відношення називають відношенням строгого порядку?

РОЗДІЛ 2 ОСНОВНІ ЗАКОНИ ТА ТОТОЖНОСТІ АЛГЕБРИ ЛОГІКИ

Для формального опису логічних схем використовують математичний апарат алгебри логіки. Основним елементом, що його реалізують при проєктуванні технічних засобів захисту інформації, є логічні функції, які набувають, як і їх аргументи, тільки два значення – 0 та 1. Отже, вивчення властивостей логічних функцій є дуже важливим для фахівців, що займаються розв'язанням задач, пов'язаних з проєктуванням засобів захисту інформації.

2.1 Закони алгебри логіки та подання логічних функцій

Базовими поняттями алгебрі логіки є аксіоми алгебри логіки. Це – кон'юнкція, диз'юнкція та заперечення. Розглянемо їх змістовне значення. В логічних виразах порядок виконання дій такий: за відсутності дужок виконуються операції заперечення, потім кон'юнкції, останніми – диз'юнкції.

Логічний зв'язок «І» або кон'юнкція. Кон'юнкція описує складне висловлення, яке є істинним тоді і лише тоді, коли істинними є прості висловлення, і хибним, якщо хоч одне з простих висловлень хибне.

Подається логічний зв'язок «І» за кон'юнкцією таблицею (табл. 2.1) та позначається як «x і y» або xy або $x \wedge y$.

Таблиця 2.1

x	0	0	1	1
y	0	1	0	1
$x \cdot y = x \wedge y$	0	0	0	1

Логічний зв'язок «АБО» або диз'юнкція. Диз'юнкція описує складне висловлення, яке є істинним тоді, коли істинним буде хоча б одне з простих висловлень, які входять в це складне висловлення, і хибним, якщо всі прості висловлення хибні.

Подається диз'юнкція як логічний зв'язок «АБО» (табл. 2.2) і позначається $x \vee y$. Читається «x або y».

Таблиця 2.2

x	y	$x \vee y$ = «x або y»
0	0	0
0	1	1
1	0	1
1	1	1

Заперечення *інакше* ***логічний зв'язок «НЕ».*** Логічний зв'язок «НЕ» – це заперечення висловлення і читається «НЕ x», позначається $\bar{x}$ або $\neg x$ (табл. 2.3). Запереченням висловлення x є складне висловлення «НЕ x», яке є істинним, коли x хибне, і хибним, коли x істинно.

Таблиця 2.3

x	0	1
$\bar{x}$	1	0

Розглянемо набір незалежних властивостей, які вважають законами алгебри логіки, а саме :

– закон комутації

$$\left.\begin{array}{l} x \vee y = y \vee x \\ x \cdot y = y \cdot x \end{array}\right\}; \tag{2.1}$$

– закон асоціації

$$\left.\begin{array}{l} (x \vee y) \vee z = x \vee (y \vee z) \\ (xy)z = x(yz) \end{array}\right\}; \tag{2.2}$$

– закон дистрибутивності

$$\left.\begin{array}{l} x(y \vee z) = xy \vee xz \\ x \vee (y \cdot z) = (x \vee y) \cdot (x \vee z) \end{array}\right\}; \tag{2.3}$$

Також, співвідношення:

– логічне додавання до нуля

$$x \vee 0 = x; \tag{2.4}$$

– логічне додавання до одиниці

$$x \vee 1 = 1; \tag{2.5}$$

– логічне множення на 0

$$x \cdot 0 = 0; \tag{2.6}$$

– логічне множення на 1

$$x \cdot 1 = x; \tag{2.7}$$

– закон протиріччя

$$x \cdot \bar{x} = 0; \tag{2.8}$$

– закон виключеного третього

$$x \vee \bar{x} = 1. \tag{2.9}$$

– закон ідемпотентності

$$\left.\begin{array}{l} x \vee x \vee x = x \\ x \cdot x \cdot x = x \end{array}\right\}; \tag{2.10}$$

– закон подвійного заперечення

$$\bar{\bar{x}} = x; \tag{2.11}$$

– закон поглинання (x поглинає y)

$$\left.\begin{array}{l} x \vee xy = x \\ (x \vee y)x = x \end{array}\right\}; \tag{2.12}$$

– закон де Моргана

$$\overline{x \vee y} = \bar{x}\bar{y}; \tag{2.13}$$

$$\overline{xy} = \bar{x} \vee \bar{y}; \tag{2.14}$$

– наслідки законів де Моргана

$$x \vee y = \overline{\bar{x} \wedge \bar{y}}; \tag{2.15}$$

$$xy = \overline{\bar{x} \vee \bar{y}}. \tag{2.16}$$

Розглянемо доведення законів алгебри логіки. По-перше доведемо справедливість дистрибутивного закону для диз'юнкції відносно кон'юнкції

Звідки

Таблиця 2.4

x_1	x_2	x_3	x_2x_3	$x_1 \vee (x_2x_3)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Таблиця 2.5

x_1	x_2	x_3	$x_1 \vee x_2$	$x_1 \vee x_3$	$(x_1 \vee x_2)(x_1 \vee x_3)$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Твердження про справедливість закону доводить ідентичність крайніх стовпців в побудованих таблицях (див. табл. 2.4, 2.5).

Наведемо доведення законів *логічного додавання та множення з константою одиниці*:

– закони логічного додавання (табл. 2.6) та множення (табл. 2.7) з константою одиниці :

$$\left.\begin{array}{l} x \vee 1 = 1; \\ x1 = x. \end{array}\right\}$$

Таблиця 2.6

x	1	$x \vee 1$
0	1	1
1	1	1

Таблиця 2.7

x	1	$x1$
0	1	0
1	1	1

Наведемо *доведення законів логічного додавання та множення з константою нуля*:

– закони логічного додавання (табл. 2.8) та множення (табл. 2.9) з константою нуля:

$$\left.\begin{array}{l} x \vee 0 = x; \\ x0 = 0. \end{array}\right\}$$

Таблиця 2.8

x	0	$x \vee 0$
0	0	0
1	0	1

Таблиця 2.9

x	0	$x0$
0	0	0
1	0	0

Наведемо *доведення закона ідемпотентності*:

– закон ідемпотентності (табл. 2.10, 2.11) :

$$\left.\begin{array}{l} x \vee x = x; \\ xx = x. \end{array}\right\}$$

Таблиця 2.10

x	x	$x \vee x$
0	0	0
1	1	1

Таблиця 2.11

x	x	xx
0	0	0
1	1	1

Доведення закону виключеного третього та закону протиріччя:

– закон виключеного третього (табл. 2.12) та закон протиріччя (табл. 2.13):

$$\left.\begin{array}{l} x \vee \overline{x} = 1; \\ x\overline{x} = 0. \end{array}\right\}$$

Таблиця 2.12

x	$\overline{x}$	$x \vee \overline{x}$
0	1	1
1	0	1

Таблиця 2.13

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

Доведення закону подвійного заперечення:

– закон подвійного заперечення (табл. 2.14):

$$\overline{\overline{x}} = x\,.$$

Таблиця 2.14

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

Нарешті *доведення законів де Моргана*:

– закони де Моргана :

$$\overline{x \vee y} = \bar{x}\,\bar{y};$$

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

Доведення законів де Моргана наведені в табл. 2.15.

Таблиця 2.15

x	y	$x \vee y$	$\overline{x \vee y}$	$\bar{x}$	$\bar{y}$	$\bar{x}\,\bar{y}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Остання таблиця демонструє, що при різних значеннях x і y права і ліва частини однакові.

2.2 Способи переходу до нормальних форм логічних функцій

Логічні функції можна подати як в табличній, так і в аналітичній формі. Перший спосіб означає складання таблиць, де задаються логічні функції. В таблицях кожному з можливих наборів змінних ставиться у відповідність значення функції (0 або 1). Цей спосіб поширений і використовується для запису логічної функції від будь-якого числа аргументів. Але такий запис не є компактним. Кількість наборів, що визначають логічну функцію, дорівнює 2 в степені n, де n кількість змінних. Природно, при великих значеннях n таблиця стає громіздкою. Простіше мати аналітичну форму запису логічної функції у вигляді формул. Існує багато способів задання логічних функцій. Розглянемо функцію, яка подана у вигляді суперпозицій алгебри логіки:

$$f(x, y, z) = x(\overline{y \vee \bar{x}z}) \vee \bar{x}y.$$

Застосовуючи тотожності алгебри логіки, перетворимо цю функцію

$$f(x, y, z) = x(\overline{y \vee \bar{x}\,z}) \vee \bar{x}\,y = x(\bar{y}(\overline{\bar{x}\,z})) \vee \bar{x}\,y = x\,\bar{y}(x \vee \bar{z}) \vee \bar{x}\,y =$$

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

Отже, одна і та ж функція може бути подана різними формулами. В зв’язку з цим виникає задача знаходження форми запису функцій, при якій

кожній функції відповідає одна і лише одна формула, а формулі відповідає одна і лише одна функція.

Нормальні форми логічних функцій (НФ). Ці форми є лише диз'юнкціями елементарних кон'юнкцій або кон'юнкціями елементарних диз'юнкцій.

Розглянемо *побудову досконалої нормальної форми логічної функції (ДДНФ)*.

Елементарними кон'юнкціями в алгебрі логіки називають вирази у вигляді $x, x_1, \overline{x_3}, x_5, \overline{x}\,\overline{z}, xyz$, тобто, заборона ставиться тільки над кожною окремою змінною. Диз'юнкція кон'юнкцій називається диз'юнктивною нормальною формою (ДНФ).

Нехай є набір змінних $x_1, x_2, x_3, ..., x_n$. Кон'юнкцію всіх змінних, взятих з запереченнями або без них, називають конституентами одиниці або мінтермами. Будь-яка конституента одиниці (мінтерм) дорівнює одиниці лише на одному наборі змінних.

Запис конституенти одиниці *n* змінних, яка дорівнює одиниці на *m*-му наборі, здійснюється таким чином: потрібно число *m* подати у вигляді *n*-розрядного двійкового числа і в кон'юнкції взяти з запереченнями ті змінні, яким в двійковому числі відповідають нулі.

Конституента одиниці змінних x_1, x_2, x_3, x_4, x_5 яка дорівнює одиниці на 25-му наборі, має такий вигляд:

$$x_1 x_2 \overline{x_3}\,\overline{x_4} x_5 = 25_{10} = 11001.$$

Таким чином, диз'юнкція конституент одиниці це досконала диз'юнктивна нормальна форма (ДДНФ).

Розглянемо порядок визначення ДДНФ.

Є логічна функція п'яти аргументів $f(x_1, x_2, ..., x_5)$, яка дорівнює 1 на наборах з номерами 4, 10, 15, 20 і нулю – на решті наборів. Задача полягає в поданні цієї функції в ДДНФ.

Виконаємо такі операції:

1. Номери наборів, які мають значення 1, записуються в двійковому коді, потім подаються у вигляді кон'юнкції змінних, де над аргументами, які дорівнюють нулю, ставиться знак заборони :

4	00100	$\overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4}\,\overline{x_5}$
10	01010	$\overline{x_1}\,x_2\,\overline{x_3}\,x_4\,\overline{x_5}$
15	01111	$\overline{x_1}\,x_2 x_3 x_4 x_5$
20	10100	$x_1\,\overline{x_2}\,x_3\,\overline{x_4}\,\overline{x_5}$.

2. Набори кон’юнкцій об’єднуються знаком диз’юнкції.

Ще однією формою подання логічної функції є досконала кон’юктивна нормальна форма (ДКНФ). Конституентою нуля або макстермом називають логічну функцію *n* аргументів, яка набуває значення, що дорівнює нулю, лише на одному наборі.

Відомо, якщо наборів аргументів 2^n, то і конституент нуля може бути - 2^n.

Конституенти нуля подають у вигляді диз’юнкцій всіх аргументів, частина з яких береться з забороною.

Заперечення ставляться таким чином, щоб обернути в нуль диз’юнкцію в потрібному наборі.

Запис конституенти нуля на одинадцятому наборі, де число аргументів дорівнює шести:

$$\begin{array}{cc} 11 & 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1 \\ & x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \end{array}$$

Заперечення ставиться над аргументами, які дорівнюють одиниці

$$f_{11}(x_1, x_2, x_3, x_4, x_5, x_6) = x_1 \vee x_2 \vee \overline{x_3} \vee x_4 \vee \overline{x_5} \vee \overline{x_6}\ .$$

Кон’юнкція конституент нуля, які дорівнюють нулю на тих самих наборах, що і задана функція, називається досконалою кон’юктивною нормальною формою.

Будь-яка логічна функція має єдину ДКНФ. Розглянемо порядок отримання ДКНФ на прикладі.

Подати у ДКНФ функцію трьох аргументів, яка дорівнює нулю на наборах 1, 3, 6.

Виконуємо такі дії:

– запис диз’юнкції всіх аргументів для наборів, де функція перетворюється в нуль, і над аргументами, які дорівнюють одиниці, ставимо знак заперечення

$$\begin{array}{lll} 1 & 0\ 0\ 1 & x \vee y \vee \overline{z} \\ 3 & 0\ 1\ 1 & x \vee \overline{y} \vee \overline{z} \\ 6 & 1\ 1\ 0 & \overline{x} \vee \overline{y} \vee z \end{array}$$

– вигляд функції:

$$f(x, y, z) = (x \vee y \vee \overline{z})(x \vee \overline{y} \vee \overline{z})(\overline{x} \vee \overline{y} \vee z)\,.$$

Розглянемо способи переходу від нормальної до досконалої форми логічної функції. Є способи переходу від нормальної до досконалої форми логічної функції за таблицею, аналітично або графічно. Дуже часто для

переходу від нормальної до досконалої форм фахівці використовують таблиці.

Логічну функцію можна подати як диз'юнкцію мінтермів, відповідно до тих наборів, на яких функція дорівнює 1.

Таким чином, ця форма і є ДДНФ. Вносити в ДДНФ має сенс тільки конституенти одиниці або мінтерми, які відповідають одиничним наборам.

Таблиця 2.16

x_1	x_2	x_3	f_1	f_2
0	0	0	1	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

Для прикладу подамо ДДНФ для функцій, заданих таблицею істинності (табл. 2.16).

$$f_1 = \overline{x}_1\overline{x}_2\overline{x}_3 \vee \overline{x}_1\overline{x}_2x_3 \vee x_1\overline{x}_2\overline{x}_3 \vee x_1\overline{x}_2x_3;$$
$$f_2 = \overline{x}_1\overline{x}_2x_3 \vee \overline{x}_1x_2\overline{x}_3 \vee \overline{x}_1x_2x_3 \vee x_1x_2x_3.$$

Кон'юнктивна нормальна форма (ДКНФ) є іншою формою є досконалої кон'юнктивної нормальної форми (ДКНФ).

ДКНФ подається як кон'юнкція конституент нуля або макстермів, відповідних нульовим наборам функції.

Для прикладу наведемо ДКНФ для функцій, заданих таблицею істинності (табл. 2.16),

$$f_1 = (x_1 \vee \overline{x}_2 \vee x_3)(x_1 \vee \overline{x}_2 \vee \overline{x}_3)(\overline{x}_1 \vee \overline{x}_2 \vee x_3)(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3);$$
$$f_2 = (x_1 \vee x_2 \vee x_3)(\overline{x}_1 \vee x_2 \vee x_3)(\overline{x}_1 \vee x_2 \vee \overline{x}_3)(\overline{x}_1 \vee \overline{x}_2 \vee x_3).$$

Розглянемо аналітичний спосіб переходу від нормальної до досконалої форми логічної функції. Відомо, що досконала нормальна форма, на відміну від нормальної, завжди містить диз'юнкції (ДДНФ) або кон'юнкції (ДКНФ) лише максимального рангу *r*.

Тому для здійснення переходу від звичайної ДНФ до ДДНФ *r*-рангу потрібно кон'юнкції, які входять до ДНФ *k*-го ($k<r$) рангу, послідовно множити на логічний вираз $\left(y_i \vee \overline{y_i}\right)$, де $y_i = x_1, x_2, x_3, ..., x_n$ – одна зі змінних, якої нема в даній кон'юнкції.

Наведемо приклад перетворення в ДДНФ логічної функції, що задана в ДНФ,

$$f_{\text{ДНФ}}(x_1, x_2, x_3) = x_1x_2 \vee x_3.$$

За нижчевказаними законами

$$x_1 \vee x_2 = x_2 \vee x_1,\ x_1x_2 = x_2x_1,\ (x_1 \vee x_2)x_3 = x_1x_3 \vee x_2x_3,$$

та $x_1 \vee \overline{x_1} = 1;\ \ x_1 \cdot \overline{x_1} = 0$ алгебри логіки. Перетворимо кон'юнкції заданої функції в мінтерми 3-го рангу:

$$x_1x_2(x_3 \vee \overline{x_3}) = x_1x_2x_3 \vee x_1x_2\overline{x_3};$$

$$x_3 = x_3(x_1 \vee \overline{x_1}) = (x_1x_3 \vee \overline{x_1}x_3)(x_2 \vee \overline{x_2}) = x_1x_2x_3 \vee \overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3 \vee \overline{x_1}\,\overline{x_2}x_3.$$

Як результат перетворень мінтерми, що їх отримано, з'єднаємо диз'юнкцією і, на основі тотожності $x_1 \vee x_1 = x_1;\ x_1 \cdot x_1 = x_1,$ маємо:

$$f_{\text{ДДНФ}}(x_1, x_2, x_3) = x_1x_2x_3 \vee x_1x_2\overline{x_3} \vee x_1\overline{x_2}x_3 \vee \overline{x_1}x_2x_3 \vee \overline{x_1}\,\overline{x_2}x_3.$$

Для переходу від КНФ до ДКНФ *r*-го рангу потрібно диз'юнкції, що входять до КНФ *k*-го рангу, додавати з виразом $y_i\overline{y_i}$, де $y_i = x_1, x_2, x_3, ..., x_n$ – одна зі змінних, що не входить в дану диз'юнкцію. Наведемо приклад перетворення КНФ до ДКНФ

$$f_{\text{КНФ}}(x_1, x_2, x_3) = x_1(x_2 \vee \overline{x_3}).$$

На основі нижченаведених законів:

$$(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3),$$

$$(x_1x_2) \vee x_3 = (x_1 \vee x_3)(x_2 \vee x_3)$$

та тотожності $x_1 \vee 0 = x_1,\ \ x_1 \cdot 1 = x_1$ логіки, переробимо диз'юнкції заданої функції в макстерми 3-го рангу:

$$x_1 = x_1 \vee x_2\overline{x_2} = (x_1 \vee x_2)(x_1 \vee \overline{x_2}) = (x_1 \vee x_2 \vee x_3\overline{x_3})(x_1 \vee \overline{x_2} \vee x_3 \cdot \overline{x_3}) =$$

$$= (x_1 \vee x_2 \vee x_3)(x_1 \vee x_2 \vee \overline{x_3})(x_1 \vee \overline{x_2} \vee x_3)(x_1 \vee \overline{x_2} \vee \overline{x_3});$$

$$x_2 \vee \overline{x_3} = x_2 \vee \overline{x_3} \vee x_1\overline{x_1} = (x_1 \vee x_2 \vee \overline{x_3})(\overline{x_1} \vee x_2 \vee \overline{x_3}).$$

Результат перетворень дає макстерми, які з'єднаємо кон'юнкцією і, на основі тотожності

$$x_1 \vee x_1 = x_1;\ x_1 \cdot x_1 = x_1,$$

будемо мати

$$f_{ДКНФ}(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3)(x_1 \vee x_2 \vee \overline{x_3})(x_1 \vee \overline{x_2} \vee x_3)(x_1 \vee \overline{x_2} \vee \overline{x_3})(\overline{x_1} \vee x_2 \vee \overline{x_3}).$$

2.3 Методи отримання мінімальних форм логічних функцій

Мінімальною формою подання логічних функцій називають таку форму, яка не припускає більше ніяких спрощень. Процес спрощення логічної функції з метою отримання мінімальної нормальної форми називають *мінімізацією*. При мінімізації виходять з вимоги мінімальної витрати обладнання, оскільки кожній логічній функції відповідає певний фізичний елемент. Для мінімізації логічних функцій використовують різні методи послідовного вилучення змінних за допомогою тотожностей алгебри логіки, методу Квайна, карт Карно, діаграм Вейча.

Розглянемо методи мінімізації логічних функцій в класі диз'юнктивних нормальних форм. При цьому під мінімальними будемо розуміти диз'юнктивні нормальні форми (ДНФ).

Наведемо деякі поняття.

Елементарною кон'юнкцією будемо називати кон'юнкцію декількох різних змінних, що взяті з запереченнями або без них. Наприклад, $x, xy, \bar{x}y, \bar{x}yz$.

Диз'юнктивна нормальна форма (ДНФ) – це диз'юнкція елементарних кон'юнкцій.

Мінімальною диз'юнктивною нормальною формою логічної функції називають ДНФ, яка містить найменшу кількість символів щодо всіх інших ДНФ, які подають задану функцію.

Функція $\varphi(x_n)$ є імплікантою функції $f(x_n)$, якщо на будь-якому наборі значень змінних $x_1, x_2, ..., x_n$ справедлива умова

$$\varphi(x_n) \le f(x_n).$$

Наведемо два твердження, які є корисними при отриманні мінімальних ДНФ:

1) диз'юнкція будь-якого числа імплікант логічної функції f також є імплікантою цієї функції;

2) будь-яка логічна функція f еквівалентна диз'юнкції всіх своїх імплікант.

Прості імпліканти логічної функції f – це такі елементарні кон'юнкції, які самі входять до даної функції, є імплікантами функції f, але ніяка їх власна частина не є імплікантою функції f.

Звідси висновок, що логічна функція f дорівнює диз'юнкції всіх простих імплікант.

Метод послідовного вилучення змінних за допомогою законів та тотожностей алгебри логіки є найбільш простим методом мінімізації. Будь-яке спрощення логічної функції відбувається винесенням за дужки загальних множників з таких мінтермів, додавання яких приводить до вилучення окремих змінних. Очевидно, що вилучення певної змінної з даного мінтерма

відбувається при додаванні до нього мінтерма, який відрізняється лише значенням цієї змінної. Такий процес підбору мінтермів, що супроводжується зниженням рангу змінної, називається «*склеюванням мінтермів*».

Розглянемо метод Квайна. Метод Квайна полягає в отриманні скороченої ДНФ. Цей метод здійснює перетворення досконалої диз'юнктивної нормальної форми за допомогою операції неповного склеювання та поглинання. Метод Квайна використовується для логічних функцій невисокого рангу за умови, що вихідні функції подано в ДДНФ. Продемонструємо дію операцій неповного склеювання та поглинання.

Повне склеювання визначається співвідношенням

$$xy \vee x\bar{y} = x \tag{2.17}$$

Доведення повного склеювання:

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

Поглинання визначається співвідношенням

$$x \vee xy = x\,. \tag{2.18}$$

Доведення операції поглинання

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

Уявімо операцію неповного склеювання

$$xy \vee x\bar{y} = x \vee xy \vee x\bar{y}\,,$$

Її можна отримати з формул так:

$$x = x \vee x = x \vee xy \vee xy \vee x\bar{y} = x \vee xy \vee x\bar{y}\,.$$

Приклад знаходження скороченої ДНФ функції. Є функція

$$f = \overline{\overline{x(y \vee z)} \wedge (x \vee yz)} = \overline{\overline{x(y \vee z)} \vee \overline{(x \vee yz)}} = x(y \vee z) \vee \bar{x}\overline{(yz)} =$$
$$= x(y \vee z) \vee x(y \vee z) = xy \vee xz \vee xy \vee xz.$$

Знаходимо ДДНФ

$$xy(\bar{z} \vee z) \vee xz(\bar{y} \vee y) \vee \bar{x}\bar{y}(\bar{z} \vee z) \vee \bar{x}z(\bar{y} \vee y) =$$
$$= xyz \vee xy\bar{z} \vee xy\bar{z} \vee x\bar{y}z \vee x\bar{y}z \vee xy\bar{z} \vee \bar{x}yz \vee xyz =$$
$$= \frac{xyz}{1} \vee \frac{xyz}{2} \vee \frac{xyz}{3} \vee \frac{xyz}{4} \vee \frac{xyz}{5} \vee \frac{xyz}{6}.$$

Реалізація склеювання

1 – 2 xy

$$xyz \vee xy\bar{z} = xy(z + \bar{z}) = xy$$

$2-6\ \ y\bar{z}$

$$xy\bar{z} \vee \bar{x}y\bar{z} = y\bar{z}(x+\bar{x}) = y\bar{z}$$

$1-3\ \ xz$

$$xyz \vee x\bar{y}z = xz(y+\bar{y}) = xz$$

$3-4\ \ \bar{y}z$

$$x\bar{y}z \vee \bar{x}\bar{y}z = \bar{y}z(x+\bar{x}) = \bar{y}z$$

$4-5\ \ \overline{xy}$

$$\bar{x}\bar{y}z \vee \bar{x}\bar{y}\bar{z} = \bar{x}\bar{y}(z+\bar{z}) = \bar{x}\bar{y}$$

$5-6\ \ \overline{xz}$

$$\bar{x}\bar{y}\bar{z} \vee \bar{x}y\bar{z} = \bar{x}\bar{z}(y+\bar{y}) = \bar{x}\bar{z}$$

Побудова імплікантної таблиці, (табл. 2.17), де у вертикальні та горизонтальні входи записано конституенти одиниці та прості імпліканти заданої функції. Така таблиця використовується для пошуку мінімальних форм.

Таблиця 2.17

	xyz	$xy\bar{z}$	$x\bar{y}z$	$\bar{x}\bar{y}z$	$\bar{x}\bar{y}\bar{z}$	$\bar{x}y\bar{z}$
xy	$\vee$	$\vee$				
$y\bar{z}$		$\vee$	$\vee$			$\vee$
xz	$\vee$		$\vee$			
$\bar{y}z$			$\vee$	$\vee$		
$\overline{xy}$				$\vee$	$\vee$	
$\overline{xz}$					$\vee$	$\vee$

Виписуємо дві еквівалентні мінімальні диз'юнктивні нормальні форми:

$$f(xyz) = xy + \bar{x}\bar{z} + \bar{y}z = xz + y\bar{z} + \bar{x}\bar{y};$$

Розглянемо порядок виконання алгоритму Квайна по кроках на прикладі мінімізації логічної функції, що задана у ДДНФ. Є логічна функція від 4-х змінних, яка задана конституентами одиниці на наборах: 3,4,5,7,9,11,12.13.

Нижче наведена її ДДНФ

$$f(x_1, x_2, x_3, x_4) = V_1(3,4,5,7,9,11,12,13) =$$

$$= \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 + \bar{x}_1 \cdot x_2 \cdot x_3 \cdot x_4 +$$

$$+ x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 + x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \bar{x}_3 \cdot \bar{x}_4 + x_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 .$$

Мінімізацію функції проводимо в декілька етапів.

Крок 1. Визначення первинних імплікант. Процес знаходження первинних імплікант подано в таблиці 2.18, де знаходимо імпліканти четвертого і третього рангу, тобто знижуємо ранг термів, які входять до ДДНФ.

Таблиця 2.18

Вихідні терми	$\bar{x}_1\bar{x}_2x_3x_4$	$\bar{x}_1x_2\bar{x}_3\bar{x}_4$	$\bar{x}_1x_2\bar{x}_3x_4$	$\bar{x}_1x_2x_3x_4$	$x_1\bar{x}_2\bar{x}_3x_4$	$x_1\bar{x}_2x_3x_4$	$x_1x_2\bar{x}_3\bar{x}_4$	$x_1x_2\bar{x}_3x_4$
$\bar{x}_1\bar{x}_2x_3x_4$	1			$\bar{x}_1x_3x_4$		$\bar{x}_2x_3x_4$		
$\bar{x}_1x_2\bar{x}_3\bar{x}_4$		1	$\bar{x}_1x_2\bar{x}_3$				$x_2\bar{x}_3\bar{x}_4$	
$\bar{x}_1x_2\bar{x}_3x_4$		$\bar{x}_1x_2\bar{x}_3$	1	$\bar{x}_1x_2x_4$				$x_2\bar{x}_3x_4$
$\bar{x}_1x_2x_3x_4$	$\bar{x}_1x_3x_4$		$\bar{x}_1x_2x_4$	1				
$x_1\bar{x}_2\bar{x}_3x_4$					1	$x_1\bar{x}_2x_4$		$x_1\bar{x}_3x_4$
$x_1\bar{x}_2x_3x_4$	$\bar{x}_2x_3x_4$				$x_1\bar{x}_2x_4$	1		
$x_1x_2\bar{x}_3\bar{x}_4$		$x_2\bar{x}_3\bar{x}_4$					1	$x_1x_2\bar{x}_3$
$x_1x_2\bar{x}_3x_4$			$x_2\bar{x}_3x_4$		$x_1\bar{x}_3x_4$		$x_1x_2\bar{x}_3$	1

Далі складаємо другу таблицю (табл. 2.19), яка містить всі терми, що не піддалися поглинанню, а також первинні імпліканти третього рангу.

Таблиця 2.19

Терм рангу 3	$\bar{x}_1x_3x_4$	$\bar{x}_2x_3x_4$	$\bar{x}_1x_2\bar{x}_3$	$x_2\bar{x}_3\bar{x}_4$	$\bar{x}_1x_2x_4$	$x_2\bar{x}_3x_4$	$x_1\bar{x}_2x_4$	$x_1\bar{x}_3x_4$	$x_1x_2\bar{x}_3$
$\bar{x}_1x_3x_4$	1								
$\bar{x}_2x_3x_4$		1							
$\bar{x}_1x_2\bar{x}_3$			1						$x_2\bar{x}_3$
$x_2\bar{x}_3\bar{x}_4$				1		$x_2\bar{x}_3$			
$\bar{x}_1x_2x_4$					1				
$x_2\bar{x}_3x_4$				$x_2\bar{x}_3$		1			
$x_1\bar{x}_2x_4$							1		
$x_1\bar{x}_3x_4$								1	
$x_1x_2\bar{x}_3$			$x_2\bar{x}_3$						1

Процес складання таблиць продовжується до тих пір, поки буде неможливо застосувати правило поглинання. Для даної функції можна дійти до первинної імпліканти другого рангу (табл. 2.19):

Отже первинні імпліканти найменшого рангу – $x_2\overline{x_3}$.

Крок 2. Визначення та встановлення позначок. Будуємо таблицю, кількість рядків якої дорівнює кількості отриманих первинних імплікант, а кількість стовпців збігається з кількістю мінтермів ДДНФ. Помічаємо, якщо в деякий мінтерм ДДНФ входить будь-яка з первинних імплікант, то на перетині відповідного стовпця і рядка ставиться позначка (табл. 2.20).

Крок 3. Пошук суттєвих імплікант

Фіксуємо, якщо в будь-якому зі стовпців таблиці 2.20 є тільки одна позначка, то первинна імпліканта у відповідному рядку є суттєвою, оскільки без неї не буде отримана вся множина заданих мінтермів.

Маємо в таблиці 2.20 суттєвою імплікантою є терм $x_2\overline{x_3}$. Таким чином, стовпці, які відповідають суттєвим імплікантам, з таблиці викреслюються.

Крок 4. Видалення зайвих стовпців

Виконання третього кроку означає, що в результаті видалення стовпців 2, 3, 7 і 8 одержуємо таблицю 2.21. В ситуації, коли в таблиці є два стовпці, в яких є позначки в однакових рядках, то один з них викреслюється.

Таблиця 2.20

Вихідні терми / первинні імплікант и	$\overline{x_1}\,\overline{x_2}\,x_3x_4$	$\overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4}$	$\overline{x_1}\,x_2\,\overline{x_3}\,x_4$	$\overline{x_1}\,x_2x_3x_4$	$x_1\,\overline{x_2}\,\overline{x_3}\,x_4$	$x_1\,\overline{x_2}\,x_3x_4$	$x_1x_2\,\overline{x_3}\,\overline{x_4}$	$x_1x_2\,\overline{x_3}\,x_4$
$\overline{x_1}\,x_3x_4$	∨			∨				
$\overline{x_2}\,x_3x_4$	∨					∨		
$\overline{x_1}\,x_2x_4$			∨	∨				
$x_1\,\overline{x_2}\,x_4$					∨	∨		
$x_1\,\overline{x_3}\,x_4$					∨			∨
$x_2\,\overline{x_3}$		∨	∨				∨	∨

Крок 5. Видалення зайвих первинних імплікант

Розглядаємо таку ситуацію, якщо після видалення декількох стовпців на кроці 4 в табл. 2.21 з’являються рядки, в яких немає жодної позначки, то первинні імпліканти, які відповідають цим рядкам, в подальшому не розглядаються, оскільки вони не покривають мінтерми, що залишилися.

Таблиця 2.21

Первинні імпліканти	Вихідні терми			
	$\overline{x_1}\,\overline{x_2}\,x_3x_4$	$\overline{x_1}\,x_2x_3x_4$	$x_1\overline{x_2}\,\overline{x_3}\,x_4$	$x_1\overline{x_2}\,x_3x_4$
$\overline{x_1}\,x_3x_4$	∨	∨		
$\overline{x_2}\,x_3x_4$	∨			∨
$\overline{x_1}\,x_2x_4$		∨		
$x_1\overline{x_2}\,x_4$			∨	∨
$x_1\overline{x_3}\,x_4$			∨	

Крок 6. Отримання мінімального покриття

В таблиці 2.21 відзначається така сукупність первинних імплікант, які містять позначки в усіх стовпцях. Декілька можливих варіантів такого означає, що при виборі надається перевага варіанту покриття з мінімальним сумарним числом букв в імплікантах, що створюють покриття. Це – первинні імпліканти $\overline{x_1}x_3x_4$ і $x_1\overline{x_2}x_4$.

Отже, мінімальна форма заданої функції буде складатися з суми суттєвих імплікант і первинних імплікант, які покривають мінтерми, що залишились,

$$f(x_1, x_2, x_3, x_4) = x_2 \cdot \overline{x_3} + \overline{x_1}\,x_3x_4 + x_1\overline{x_2}\,x_4.$$

Для подання логічних функцій від невеликої кількості змінних використовують діаграми Вейча.

Мінімальні ДНФ логічної функції *f* невеликої кількості змінних дуже просто отримати за допомогою діаграм Вейча. Діаграма Вейча для логічної функції двох змінних має вигляд (рис. 2.1). Кожна комірка діаграми відповідає набору змінних булевої функції в її таблиці істинності. Комірка діаграми Вейча має одиницю, якщо логічна функція набуває одиничного значення на відповідному наборі. Якщо значення логічної функції нульові, в діаграмі Вейча вони не проставляються. Діаграма Вейча для логічної функції трьох змінних має вигляд (рис. 2.2), відповідно, діаграма Вейча для функції чотирьох змінних має вигляд (рис. 2.3).

	X_1	$\overline{X_1}$
X_2	1 1	0 1
$\overline{X_2}$	1 0	0 0

Рисунок 2.1 – Діаграма Вейча для логічної функції двох змінних

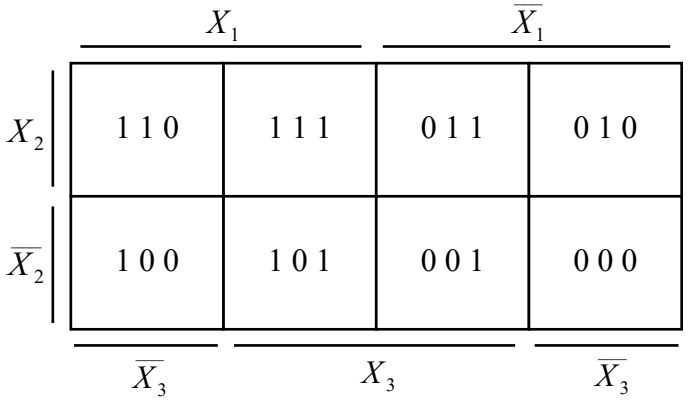


Рисунок 2.2 – Діаграма Вейча для логічної функції трьох змінних

Мінімізація відбувається за нижчевказаними правилами.

1. Сусідні дві комірки (0-куби) створюють один 1-куб. На межах карти аналогічно.

2. Можуть об’єднуватися чотири вершини, створюючи один 2-куб, що містить дві незалежні координати;

3. Можуть об’єднуватися вісім вершин, утворюючи один 3-куб;

4. Можуть об’єднуватися шістнадцять вершин, утворюють один 4-куб і т. д.

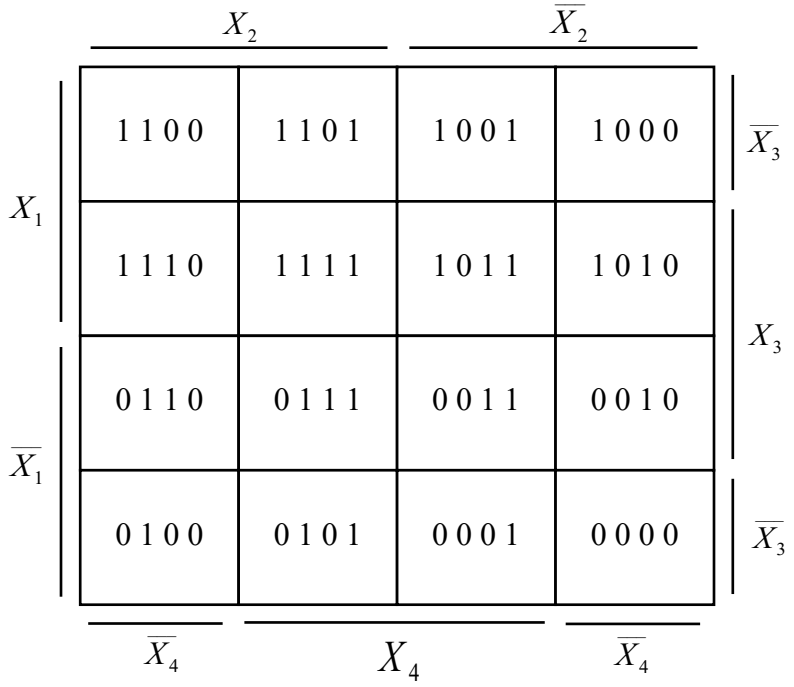


Рисунок 2.3 – Діаграма Вейча для логічної функції чотирьох змінних

Множина прямокутників, які покривають усі одиниці, називається покриттям, при цьому одна і та ж комірка може покриватися два або декілька разів.

Продемонструємо мінімізацію логічної функції за допомогою діаграм Вейча для функції, яка наведена нижче.

$$f(x_1, x_2, x_3, x_4) = V_1(0, 2, 3, 4, 6, 8, 10, 11, 14) =$$
$$= \overline{x}_1 \overline{x}_2 \overline{x}_3 \overline{x}_4 \vee \overline{x}_1 \overline{x}_2 x_3 \overline{x}_4 \vee \overline{x}_1 \overline{x}_2 x_3 x_4 \vee \overline{x}_1 x_2 \overline{x}_3 \overline{x}_4 \vee$$
$$\vee \overline{x}_1 x_2 x_3 \overline{x}_4 \vee x_1 \overline{x}_2 \overline{x}_3 \overline{x}_4 \vee x_1 \overline{x}_2 x_3 \overline{x}_4 \vee x_1 \overline{x}_2 x_3 x_4 \vee x_1 x_2 x_3 \overline{x}_4 .$$

Діаграма Вейча для даної функції наведена на рис. 2.4.

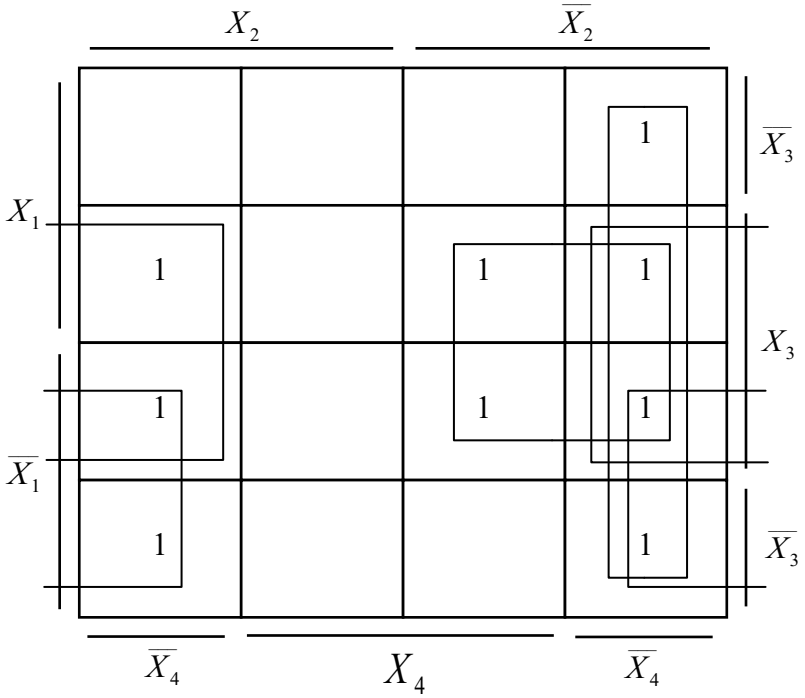


Рисунок 2.4

Отже, мінімальна форма заданої функції має такий вигляд:

$$f(x_1, x_2, x_3 x_4) = \overline{x}_1 \overline{x}_4 + x_3 \overline{x}_4 + \overline{x}_2 x_3 + \overline{x}_2 \overline{x}_4 .$$

2.4 Завдання для самостійної роботи

Тема. Мінімізація логічних функцій.

Виконайте :

- мінімізацію функцій за допомогою методу Квайна,
- діаграм Вейча,

відповідно до індивідуального варіанта (табл. 2.22).

Таблиця 2.22 – Варіанти індивідуальних завдань

$f(x_1,x_2,x_3,x_4)=$	
1. $V_1(0,2,3,5,7,8,10,13)$	23. $V_1(1,2,6,7,9,10,11,13)$
2. $V_1(0,1,2,3,4,9,10,11,12)$	24. $V_1(2,3,5,7,9,11,12,14)$
3. $V_1(0,1,2,4,6,9,8,11)$	25. $V_1(0,1,2,4,5,8,11,12,15)$
4. $V_1(0,1,3,5,7,9,10,14)$	26. $V_1(1,2,4,6,10,11,15)$
5. $V_1(1,2,5,6,8,9,11,13,14)$	27. $V_1(2,4,5,7,9,11,13,14)$
6. $V_1(1,3,4,7,9,12,13)$	28. $V_1(1,6,7,9,10,11,14,15)$
7. $V_1(0,1,2,5,7,8,11,14,15)$	29. $V_1(2,3,4,10,11,12,14)$
8. $V_1(1,2,3,5,8,9,10,11)$	30. $V_1(0,2,4,5,6,8,12,14)$
9. $V_1(0,1,3,4,6,8,10,12)$	31. $V_1(1,3,6,8,11,12,13,15)$
10. $V_1(1,4,5,9,11,12,13,14)$	32. $V_1(1,5,6,7,9,10,13,14)$
11. $V_1(1,3,5,9,11,13,14)$	33. $V_1(2,4,6,7,11,12,13,14)$
12. $V_1(0,1,2,8,9,10,11,14)$	34. $V_1(0,3,7, 9,10,11,12,14)$
13. $V_1(0,1,2,5,6,8,10,15)$	35. $V_1(0,2,6,7,10,12,13)$
14. $V_1(0,1,4,7,8,11,12,14)$	36.$V_1(1,4,6,7,9,11,13,14,15)$
15. $V_1(1,4,6,7,10,11,13,15)$	37. $V_1(0,3,5,7,9,10,11,14)$
16. $V_1(0,2,4,5,6,9,10,12)$	38. $V_1(0,1,4,5,7,12,13,15)$
17. $V_1(2,4,6,8,11,12,13,15)$	39. $V_1(0,2,4,5,7,11,12,13,15)$
18. $V_1(1,4,5,10,11,12,13,15)$	40. $V_1(0,1,4,5,6,8,10,11,13)$
19. $V_1(0,2,3,6,8,10,11,14)$	41. $V_1(1,2,3,4,5,7,8,9,13)$
20. $V_1(0,1,2,5,6,9,11,13)$	42. $V_1(0,1,2,4,5,7,8,9,11)$
21. $V_1(1,2,5,7,10,13,15)$	43. $V_1(0,3,4,7,9,11,12,14,15)$
22. $V_1(0,1,4,7,11,12,15)$	44. $V_1(3,4,7,9,10,12,14,15)$
45. $V_1(0,1,2,4,6,8,9,11,12)$	

2.5 Контрольні питання

1. Навести таблицю істинності логічної функції «кон'юнкція».
2. Навести таблицю істинності логічної функції «диз'юнкція».
3. Навести таблицю істинності логічної функції «операція Пірса».
4. Які основні властивості логічних функцій?
5. Як виконується закон протиріччя для кон'юнкції та закон виключеного третього для диз'юнкції?
6. Як виконується закон подвійного заперечення?
7. Як виконуються закони з константами для кон'юнкції (логічне множення на одиницю, логічне множення на нуль)?
8. Як виконуються закони з константами для диз'юнкції (логічне додавання до одиниці, логічне додавання з нулем)?
9. Як перевірити справедливість формул де Моргана?
10. Яку диз'юнкцію нормальних форм називають мінімальною формою логічної функції?
11. Що таке суттєва імпліканта?
12. З яких основних кроків складається процес мінімізації логічної функції за методом Квайна?
13. Що таке елементарна кон'юнкція, диз'юнкція?
14. Як використовуються для мінімізації логічної функції діаграми Вейча?

РОЗДІЛ 3 ОСНОВНІ ПОНЯТТЯ ТЕОРІЇ ІНФОРМАЦІЇ ТА КОДУВАННЯ

3.1 Основні задачі теорії інформації та кодування повідомлень

Основна задача теорії інформації – пошук способів передачі інформації при мінімальних часових і матеріальних затратах (передача більшої кількості інформації за найменший час). Кількість інформації визначається ступенем невизначеності тієї чи іншої ситуації, події, факту. Найпростішим є вибір з 2-х рівномірних повідомлень типу «Так – Ні». Результатом вирішення цієї проблеми є передача двох якісних ознак 0 і 1, чи «+» і «-» імпульсів тощо. Кількість інформації, переданої в цьому найпростішому випадку, прийнято вважати за одиницю кількості інформації. Ця двійкова одиниця називається «**БІТ»**.

Число повідомлень *N*, які можна отримати, комбінуючи *m* символів алфавіту по *n* елементів в повідомленні, складає

$$N = m^n .$$

У 1828 р. Хартлі запропонував логарифмічні міри кількості інформації:

$$I = \log_a N = \log_a m^n = n \times \log_a m ,$$

a=2 при основі log опускаємо і розуміємо, що це двійковий логарифм.

Поняття інформації пов’язано зі зняттям невизначеності, яка існувала до отримання повідомлення. Чим більша невизначеність була до передачі повідомлення, тим більша кількість інформації міститься в прийнятому повідомленні. З теорії інформації мірою невизначеності є ентропія – кількість інформації, що припадає на один елемент повідомлення,

$$H = -\sum_{i=1}^{m} p_i \log_2 p_i = \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} . \qquad (3.1)$$

Вираз (3.1) називається середньою ентропією повідомлення. Вперше його отримав К. Шеннон у 1948 р.

$$p_i = \frac{1}{m}, H = \log_a m .$$

3.2 Оптимальні коди та їх характеристики

При передачі повідомлень, закодованих двійковим рівномірним кодом, зазвичай не враховують статистичну структуру передаваних повідомлень, які, незалежно від ймовірності їх появи, є кодовими комбінаціями однакової довжини, тобто, кількість двійкових символів, що припадає на одне повідомлення, постійна. Такі коди мають надлишковість. Найбільш

ефективним способом зменшення надмірності повідомлення є побудова оптимальних кодів, що мають мінімальну середню довжину кодових слів. При побудові оптимальних кодів найбільшого поширення набули методики Шеннона – Фано та Хаффмана.

Код Шеннона – Фано будується так, як і двійковий код, тільки елементи повідомлення вписується в таблицю формування кодових груп в порядку зменшення ймовірності їх появи на виході джерела інформації. Ділення виконується таким чином, щоб суми ймовірностей в кожній групі були приблизно однаковими. Якщо кількість елементів повідомлень відома, то ці елементи, без урахування ймовірностей їх появи, можна відобразити двійковим кодом. Але таке відображення не є оптимальним, оскільки не враховує ймовірностей появи вказаних елементів в потоці інформації.

Якщо ймовірності появи відомі, то відображення елементів кодовими комбінаціями виконується за принципом їх послідовного розбиття на дві рівноймовірні групи. Ймовірності записуються в порядку зменшення. При розбитті елементів повідомлення на групи верхнім (більшим) частинам груп присвоюється символ 1, а нижнім (меншим) – 0.

Розпізнавання кодових комбінацій виконується за рахунок того, що ні одна з кодових комбінацій не має бути початковою частиною інших, більш довгих, комбінацій. Це забезпечує можливість однозначного декодування. Декодування має починатися з початку повідомлення, бо інакше однозначність зникає, і повідомлення спотворюється. За рахунок використання цього коду забезпечується швидкість передачі інформації по каналу передачі даних. Це забезпечується за рахунок ущільнення інформації, ефективність якої визначається коефіцієнтом ущільнення.

3.3 Методики побудови коду Шеннона – Фано, коду Хаффмана

Методики побудови коду Шеннона – Фано

Крок 1. Розташувати повідомлення в порядку спадання ймовірностей.

Крок 2. Розділити повідомлення на дві групи з рівними сумарними ймовірностями. Якщо цього не можна досягти, то їх ділять так, щоб у верхній частині залишалися символи, сумарна ймовірність яких менша сумарної ймовірності символів у нижній підгрупі.

Крок 3. Присвоїти першій групі символ 0, а другій – символ 1.

Крок 4. Кожну з підгруп розділити на дві частини таким чином, щоб сумарні ймовірності новостворених підгруп були, по можливості, рівні. Якщо цього не можна досягти, то їх ділять так, щоб у верхній частині залишалися символи, сумарна ймовірність яких менша сумарної ймовірності символів у нижній підгрупі.

Крок 5. Присвоїти першим групам кожної з підгруп 0, а другим – 1.

Поділ на групи проводиться до тих пір, поки в кожній з підгруп не залишиться по одному символу. Кодові комбінації коду Шеннона – Фано для кожного повідомлення отримують зчитуванням символів 0 і 1 відповідних груп та підгруп.

Приклад побудови коду Шеннона – Фано. Побудуємо код Шеннона – Фано для передачі шести повідомлень, що мають ймовірності: A = 0,4; B = 0,2; C = 0,2; D = 0,1; E = 0,05; F = 0,05. Повідомлення в порядку убування їх ймовірностей маємо у другому стовпці табл. 3.1.

На першому етапі розподілу на групи відокремлюємо лише перше повідомлення (група 1), залишивши групі 2 всі інші. Друге повідомлення складе першу підгрупу 2-ї групи, а решта чотири повідомлення складуть другу підгрупу, яка на наступних кроках буде ділиться на частини так, що кожен раз перша частина скрадатиметься з одного повідомлення.

Таблиця 3.1 – Етапи побудови коду Шеннона – Фано

Номер повідомлення	**Символ / Ймовірність**	**Ділення на групи / підгрупи**	**Кодові слова**
1	A=0,4	0	0
2	B=0,2	1	10
3	C=0,2	1	110
4	D=0,1	1	1110
5	E=0,05	1	11110
6	F=0,05	1	11111

В результаті виконання коду отримуємо кодові послідовності для всіх шести символів. При цьому довжина кодової комбінації символа A, з найбільшою ймовірністю появи, є найкоротшою.

Визначимо середню довжину кодової комбінації L_{cp}

$$L_{cp} = \sum_{i=1}^{m} P_i \times E_i ,$$

де E_i – кількість розрядів отриманої кодової комбінації відповідного символу.

$$L_{cp} = \sum_{i=1}^{m} P_i \times E_i = 0,4 \times 1 + 0,2 \times 2 + 0,2 \times 3 + 0,1 \times 4 + 0,05 \times 5 + 0,05 \times 5 = 1,9$$

При передачі цих же шести повідомлень рівномірним двійковим кодом довжина кодової комбінації дорівнюватиме 3.

Код Хаффмана. Алгоритм побудови коду Хафамана починається з того, що символи інформаційного повідомлення записуються в стовпчик в порядку зменшення ймовірності їх появи в повідомленні. Два останні символи з

найменшими ймовірностями об’єднуються в одне повідомлення таким чином, що з’являється один додатковий символ, який характеризує об’єднане повідомлення з сумарною ймовірністю

$$P=p1+p2.$$

При отриманні нового символа з ймовірністю P він займає відповідну позицію у стовпчику, не порушуючи порядку зменшення ймовірності появи повідомлень.

На наступному кроці алгоритму знову об’єднуються символи, що мають найменші ймовірності. Цей процес продовжується до тих пір, доки не буде одержана сумарна ймовірність $P = 1$.

Процес додавання ймовірностей можна зобразити за допомогою кодового дерева, у якого гілкам з більшою ймовірністю надається значення «1», а гілкам з меншою ймовірністю – «0». Якщо ймовірності однакові, то значення «1» присвоюється верхній гілці, а значення «0» – нижній. В результаті побудови кодового дерева ми можемо, рухаючись кодовим деревом від вершини з ймовірністю $P = 1$ до символа, який потрібно закодувати, одержимо певну кодову комбінацію в коді Хаффмана.

Методика побудови коду Хаффмана

Етап1. Згортання ймовірностей

Крок 1. Розташувати повідомлення в порядку спадання ймовірностей у вигляді таблиці.

Крок 2. Два останніх повідомлення об’єднати в одне допоміжне, якому приписати сумарну ймовірність.

Крок 3. У додатковому стовпці знову розташувати ймовірності в порядку спадання.

Крок 4. Дві останні ймовірності об’єднати в сумарну. Процес продовжується до тих пір, поки не вийде єдине повідомлення з імовірністю $P = 1$.

Етап 2. Побудова кодового дерева

Щоб скласти кодові комбінації, що відповідають даним повідомленням, потрібно простежити шлях переходу повідомлення по рядках і стовпцях таблиці ймовірностей. Для наочності будують кодове дерево. З точки, що відповідає ймовірності 1, направляють дві гілки, причому гілкам з більшою ймовірністю присвоюється символ 1, а з меншою – 0. Таке послідовне розгалуження продовжується до тих пір, поки гілки не дійдуть до ймовірності кожного повідомлення.

Приклад побудови коду Хаффмана. Побудуємо методом Хаффмана оптимальний код для передачі восьми повідомлень з ймовірностями: 0,49; 0,14; 0,14; 0,07; 0,07; 0,06; 0,02; 0,01.

Розташуємо повідомлення в порядку спадання ймовірностей у вигляді таблиці 3.2. Та проведемо кроки етапу згортання ймовірностей.

Таблиця 3.2 – Етап згортання ймовірностей

Номер повідомлення	Ймовірність	Допоміжні стовпці						
1	0,49	0,49	0,49	0,49	0,49	0,49	0,51	1,0
2	0,14	0,14	0,14	0,14	0,23	0,28	0,49	
3	0,14	0,14	0,14	0,14	0,14	0,23		
4	0,07	0,07	0,09	0,14	0,14			
5	0,07	0,07	0,07	0,09				
6	0,06	0,06	0,07					
7	0,02	0,03						
8	0,01							

Після цього можна перейти до етапу побудови кодового дерева (рис. 3.1).

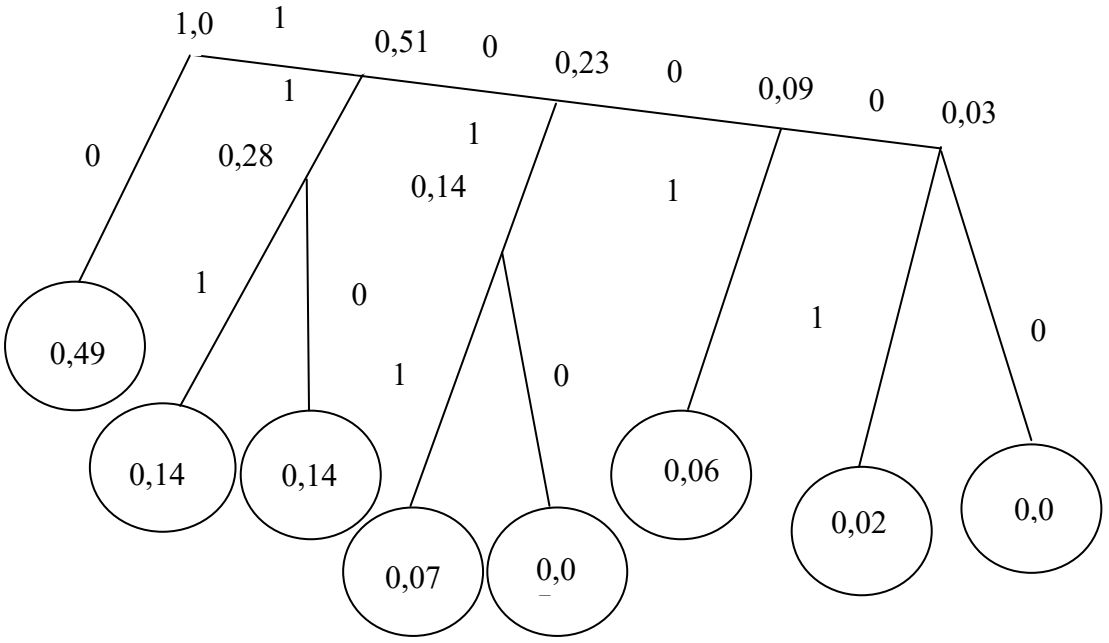


Рисунок 3.1 – Кодове дерево Хаффмана

Рухаючись по кодовому дереву від вершини (з ймовірністю 1) до ймовірності символів (зверху вниз), можна записати для кожного повідомлення відповідну кодову комбінацію:

1 (0,49) – 0
2 (0,14) – 111
3 (0,14) – 110
4 (0,07) – 1011
5(0,07) – 1010
6 (0,06) – 1001
7 (0,02) – 10001
8 (0,01) – 10000

Визначимо середню довжину кодової комбінації

$$L_{cp} = \sum_{i=1}^{m} P_i \times E_i ,$$

де E_i – кількість розрядів отриманої кодової комбінації відповідного символа.

$$L_{cp} = \sum_{i=1}^{m} P_i \times E_i =$$
$$= 0,49 \times 1 + 0,14 \times 3 + 0,14 \times 3 + 0,07 \times 4 + 0,07 \times 4 + 0,06 \times 4 + 0,02 \times 5 + 0,01 \times 5 = 2,28$$

При передачі цих же повідомлень рівномірним двійковим кодом довжина кодової комбінації буде дорівнювати $H =]\log_2 8[= 3$.

3.4 Основні надлишкові коди та їх характеристики

Одним з основних методів під час контролю, зберігання, передачі та обробки інформації є використання надлишкових кодів, призначених для виявлення та виправлення помилок. Найбільш значне використання на практиці знайшли двійкові коди, кожний розряд яких може приймати значення 0 або 1. Ці коди характеризуються такими показниками:

1. Кількість розрядів *n* в кодовій комбінації прийнято називати *довжиною* або *розрядністю коду*.

2. Кількість одиниць в кодовій комбінації називають *вагою кодової комбінації* і позначають ω .

Ступінь відмінності будь-яких двох кодових комбінацій двійкового коду називається *відстанню між кодами* α або *кодовою відстанню*. Вона виражається числом позицій або символів, в яких комбінації відрізняються одна від іншої. Кодову відстань можна також визначити як вагу кодової комбінації, отриману шляхом додавання за модулем двох кодових комбінацій.

Помилки, внаслідок дії завад або відмов, проявляються в тому, що в одному або кількох розрядах кодової комбінації нулі переходять в одиниці і навпаки – одиниці переходять в нулі. В результаті утворюється нова кодова комбінація, яка є хибною.

Для виявлення позицій в *n*-розрядній кодовій комбінації, де є спотворення символів, використовується вектор помилки. Цей вектор є *n*-розрядною комбінацією, в якій одиниці показують положення спотворених символів кодової комбінації.

Вага вектора помилки характеризує кратність помилки. Сума за правильну комбінацію.

Важливою характеристикою надлишкового коду є кодова відстань α, що показує кратність помилок, які можуть бути виявлені і виправлені цим кодом.

Для виявлення *t*-кратних помилок необхідно і достатньо, щоб

$$\alpha_{min} \geq t + 1.$$

Для виправлення *s*-кратних помилок –

$$\alpha_{min} \geq 2s + 1.$$

А для виявлення *t*-кратних і виправлення *s*-кратних помилок

$$\alpha_{min} \geq t + 2s + 1.$$

Коригувальні властивості коду залежать від його надлишковості. Для виявлення і тим паче виправлення помилок в *n*-розрядному коді має бути деяке число (наприклад *k*) контрольних розрядів, за допомогою яких формуються «заборонені» кодові комбінації. Виявлення помилок, в такому випадку, зводиться до виявлення заборонених кодових комбінацій, а виправлення – до визначення найбільш ймовірної дозволеної кодової комбінації, з якої отримана дана заборонена комбінація.

Кількість надлишкових позицій *k*. називають абсолютною надлишковістю коду, що виявляє чи виправляє помилки.

Тоді відносна надлишковість *R* дорівнює

$$R = \frac{k}{n}$$

Якщо при аналізі кодової комбінації надлишкового коду встановлена її належність до множини дозволених, то вважається, що помилки в ній немає. В іншому випадку робляться висновки, що комбінація спотворена. Для переважної більшості практичних задач очікувана кількість помилок є невідомою. Тому прогнозують, що можуть виникнути помилки будь-якої кратності (від одиниці до *n*), і тоді будь-яка дозволена комбінація може перетворитися в будь-яку дозволену комбінацію, тобто не всі спотворення можна виявити. При цьому коефіцієнт виявлення (частка помилкових комбінацій, що виявляються) складає

$$S_в = N_{заб} / N_{заг} = (N_{заг} - N_{дозв}) / N_{заг} = 1 - N_{дозв} / N_{заг} ,$$

де $N_{заг}$ – загальна кількість кодових комбінацій надлишкового коду;
$N_{дозв}$ – кількість дозволених кодових комбінацій;
$N_{заб}$ – кількість заборонених кодових комбінацій.

Код з перевіркою на парність (*c* = 2). Цей код дозволяє виявляти будь-яку помилку непарної кратності. Отримують його шляхом додавання до кодового слова одного контрольного розряду, в який записують 0 або 1 з тим, щоб число одиниць в кодовій комбінації, що її отримали, було парним. Ознакою спотвореної кодової комбінації є непарність кількості одиниць в комбінації.

Код з подвоєнням елементів. Код характеризується введенням додаткових розрядів для кожного символа вихідного коду, при цьому одиниця доповнюється нулем і перетворюється в 10, а нуль доповнюється одиницею і перетворюється 01. Ознакою спотворення коду буде поява в «парних» елементах сполучень типу 00 або 11. Код дозволяє виявляти всі помилки, за винятком випадків, коли мають місце двократні помилки в «парних» елементах.

Інверсний код. В основу побудови цього коду покладено метод повторення вихідної кодової комбінації. В тих випадках, коли вихідна комбінація містить парну кількість одиниць, вона просто повторюється; якщо вихідна комбінація містить непарну кількість одиниць, то повторення відбувається в інвертованому вигляді. Перевірка кодової комбінації інверсного коду відбувається шляхом підрахунку кількості одиниць в основній комбінації. Якщо вона парна, то елементи додаткової комбінації приймаються без змін. Якщо ж вона непарна, то елементи додаткової комбінації приймаються в інвертованому вигляді. Після цього основна і додаткова комбінації порівнюються за кожним елементом (перший елемент з першим, другий з другим і т. д.) і при виявленні хоча б одного елемента, що не збігається, формується висновок наявності помилки. Така побудова коду дозволяє виявляти всі помилки, що не призводять до одночасного спотворення парної кількості елементів в однакових позиціях основної і додаткової комбінацій.

Рівноважний код. Цей код характеризується тим, що всі дозволені кодові комбінації мають одну й ту саму вагу ω. Кодові комбінації цього коду не отримують з вихідного коду за певним правилом, а ставляться у певну відповідність вихідним повідомленням. Побудова кодових комбінацій здійснюється шляхом створення всіх можливих сполучень. При цьому вага кожної кодової комбінації дорівнює m, а кількість комбінацій з n символів по m одиниць є дорівнює

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

Для виявлення помилок в кодовій комбінації визначається її вага ω. Якщо вага дорівнює m, то ця комбінація правильна, в протилежному випадку кодову комбінацію спотворено. Цей код виявляє всі помилки, за винятком тих, які не змінюють ваги кодової комбінації.

Наведені вище надлишкові коди відносять до одновимірних кодів, на практиці для контроля зберігання та передавання інформації часто використовують двовимірні коди, які називають ітеративними або векторними кодами.

Ітеративний код. Методика побудови двовимірного ітеративного коду викладена нижче.

1. Задану сукупність інформаційних символів n подають у вигляді таблиці 3.3.

2. До кожного рядка та до кожного стовпця дописують контрольні символи таким чином, щоб рядки і стовпці стали словами певного надлишкового коду, де $L1$, $L2$, …, Ln – контрольні символи рядків, а $K1$, $K2$, …, Km – контрольні символи стовпців.

Таблиця 3.3 – Побудова двовимірного ітеративного коду

A_{11}	A_{12}	….	A_{1m}	$L1$
A_{21}	A_{22}		A_{2m}	$L2$
…	…	….	….	…
A_{n1}	A_{n2}	…	A_{nm}	Ln
$K1$	$K2$	…..	Km	-

Ітеративні коди дозволяють не тільки виявляти, але й виправляти помилки. Якщо не виконується контрольне співвідношення для будь-якого рядка або стовпця, то це означає наявність помилки. Символ, який знаходиться на перетині «помилкового» рядка і «помилкового» стовпця, є спотвореним і для виправлення помилки його потрібно інвертувати. Ітеративний код, який отримують за допомогою коду з перевіркою на парність, дозволяє виправити будь-яку поодиноку помилку та сукупність діагонально розташованих помилок, а також виявити будь-яку непарну помилку і більшість помилок парної кратності.

3.5 Методика побудови коду Геміңга та циклічного коду

Коди Геміңга відносять до найпоширеніших коригувальних (n, k) кодів, що містять k інформаційних і $r = n - k$ надлишкових розрядів. Надлишкові розряди будуються таким чином, щоб при декодуванні було можливим як виявлення помилки, так і місця її знаходження. Це досягається за рахунок перевірок на парність окремих груп розрядів. Кількість таких перевірок дорівнює кількості r надлишкових розрядів. Результатом перевірок є r-розрядний двійковий код, який і вказує номер помилкового розряду. Виправлення помилки виконується шляхом інверсії визначеного помилкового розряду.

Необхідна кількість надлишкових розрядів r і загальна довжина коду n пов’язані між собою співвідношенням

$$2^k \leqslant \frac{2^n}{1+n}.$$

Для спрощення процедури декодування перевірочні розряди розташовують на позиціях з номерами, що дорівнюють степені числа 2.

При цьому структура коду Гемінга буде мати вигляд:

№ позиції X_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	…
Розряди	k_1	k_2	$a1$	k_3	$a2$	$a3$	$a4$	k_4	$a5$	$a6$	$a7$	$a8$	$a9$	$a10$	$a11$	$к5$	$a12$	…

де $k_1,k_2,k_3,..$ – перевірочні розряди, *a1, a2, a3,..* – інформаційні розряди.

Значення контрольних розрядів визначаються таким чином. Значення першого перевірочного розряду k_1 отримують як результат суми за модулем 2 тих інформаційних розрядів, які розміщенні на позиціях з номерами, що мають одиницю в молодшому розряді (всі непарні номери позицій 1,3,5,7,9,..., тобто *a1, a3, a5, a6,...*).

Значення другого перевірочного розряду k_2 є результатом суми за модулем 2 тих інформаційних розрядів, що мають в номері одиницю в другому розряді (3, 6, 7,10, 13, ..., тобто *a1, a3, a4, a6, a7, a10,...).*

Аналогічним чином визначаються інші перевірочні розряди, тобто за правилом – в обчисленні суми за модулем 2 приймають ті інформаційні розряди, номери яких містять ту одиницю (з тією вагою), що міститься в номері перевірочного розряду.

Таким чином, перевірочні розряди обчислюються так:

$$k_1 = a\,1 \oplus a2 \oplus a4 \oplus a5 \oplus a7 \oplus a9 \oplus a11\ldots;$$
$$k_2 = a1 \oplus a3 \oplus a4 \oplus a6 \oplus a7 \oplus a10 \oplus a11\ldots;$$
$$k_3 = a2 \oplus a3 \oplus a4 \oplus a8 \oplus a9 \oplus a10 \oplus a11\ldots;$$
$$k_4 = a5 \oplus a6 \oplus a7 \oplus a8 \oplus a9 \oplus a10 \oplus a11\ldots;$$
$$k_5 = a12 \oplus \ldots .$$

При декодуванні визначають так званий синдром помилки *S1,S2,S3,S4 ... SK*, який вказує на помилковий розряд,

$$S1 = k_1 \oplus a1 \oplus a2 \oplus a4 \oplus a5 \oplus a7 \oplus a9 \oplus a11\ldots;$$
$$S2 = k_2 \oplus a1 \oplus a3 \oplus a4 \oplus a6 \oplus a7 \oplus a10 \oplus a11\ldots;$$
$$S3 = k_3 \oplus a2 \oplus a3 \oplus a4 \oplus a8 \oplus a9 \oplus a10 \oplus a11\ldots;$$
$$S4 = k_4 \oplus a5 \oplus a6 \oplus a7 \oplus a8 \oplus a9 \oplus a10 \oplus a11\ldots;$$
$$S5 = k_5 \oplus a12 \oplus \ldots .$$

При *S4 S3 S2 S1… SK* = 0000… кодова комбінація не містить помилок. В інших випадках синдром помилки буде вказувати на номер помилкового розряду кодової комбінації X_i.

Приклад побудови коду Геміңа (α = 3). Побудувати код Гемінга та показати процес виявлення та виправлення поодинокої помилки для α = 3. Інформаційна частина кодової комбінації: 01010100, перевірити помилку на позиції X_{10}.

Побудуємо кодову комбінацію для коду Гемінга (12, 8).

Кодова комбінація

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
k_1	k_2	0	k_3	1	0	1	k_4	0	1	0	0

Визначення контрольних розрядів:

1. $k_1 = X_3 \oplus X_5 \oplus X_7 \oplus X_9 \oplus X_{11} = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \Rightarrow k_1 = 0$
2. $k_2 = X_3 \oplus X_6 \oplus X_7 \oplus X_{10} \oplus X_{11} = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \Rightarrow k_2 = 0$
3. $k_3 = X_5 \oplus X_6 \oplus X_7 \oplus X_{12} = 1 \oplus 0 \oplus 1 \oplus 0 \Rightarrow k_3 = 0$
4. $k_4 = X_9 \oplus X_{10} \oplus X_{11} \oplus X_{12} = 0 \oplus 1 \oplus 0 \oplus 0 \Rightarrow k_4 = 1$

Отже, на кодер надійде таке повідомлення:

0000 1011 0100.

Після спотворення 10-го розряду (X_{10}) отримаємо

0000 1011 0000.

Перевірки на декодері:

1. $S1 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$
2. $S2 = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 1$
3. $S3 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$
4. $S4 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 1$

Отримавши $1010_2 = 10_{10}$, випливає, що спотворена позиція – це X_{10}.

Виправити кодову комбінацію можна, інвертувавши помилковий розряд:

$$0000\ 1011\ 0000 \Rightarrow 0000\ 1011\ 0100.$$

Метод побудови коду Гемінга ($\alpha = 4$)

Двійковий код Гемінга з кодовою відстанню $\alpha = 4$ формується шляхом додавання до коду Гемінга з $\alpha = 3$ ще одного розряду перевірки, що є результатом суми за модулем 2 всіх розрядів кодової комбінації X_i.

Операція кодування може виконуватися в два етапи. На першому етапі визначається кодова комбінація (X_i) з використанням основних співвідношень для $\alpha = 3$, а на другому – додається один розряд (r), в якому записується результат суми за модулем 2 всіх розрядів кодового слова, отриманого на першому етапі,

$$r = X_1 \oplus X_2 \oplus X_4 \oplus X_5 \oplus X_7 \ldots \oplus X_n.$$

Операція декодування також складається з двох етапів. На першому перевіряються основні контрольні співвідношення Si (для $\alpha = 3$), на другому – додаткове контрольне співвідношення r.

Результати виконання цих перевірок і відповідні їм висновки наведені в табл. 3.4

Таблиця 3.4 – Співвідношення результатів перевірки кодової комбінації

Основні контрольні співвідношення *Si*	Додаткові контрольні співвідношення *r*	Висновки
Не виконуються	Виконуються	Відбулася подвійна помилка
Не виконуються	Не виконуються	Відбулася поодинока помилка
Виконуються	Виконуються	Помилок немає
Виконуються	Не виконуються	Відбулася потрійна або більш високої кратності помилка але непарна

Приклад побудови коду Гемінга (α = 4). Побудувати код Гемінга та показати процес виявлення й виправлення поодинокої/подвійної помилки для α = 4. Інформаційна частина кодової комбінації: 01010100.

На кодер буде подане повідомлення 0000 1011 0100

Визначення додаткового розряду:

$$r = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$$

Кодова комбінація для 0000 1011 0100 **0**.

Перевіримо виявлення поодинокої помилки на прикладі.

Після спотворення 10-го розряду (X_{10}), отримаємо

0000 1011 0000 0.

Після перевірки на декодері основних контрольних співвідношень отримали

1010_2 – порушення на 10-й позиції.

Перевіримо додаткове контрольне співвідношення

$$r = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 1$$

порушення на контрольному співвідношенні.

Відповідно до табл. 3.4 робимо висновок: відбулась поодинока помилка, що відповідає дійсності.

Перевіримо виявлення подвійної помилки

1000 1011 0000 0

$S1 = X_1 \oplus X_3 \oplus X_5 \oplus X_7 \oplus X_9 \oplus X_{11} = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 1$

$S2 = X_2 \oplus X_3 \oplus X_6 \oplus X_7 \oplus X_{10} \oplus X_{11} = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$

$S3 = X_4 \oplus X_5 \oplus X_6 \oplus X_7 \oplus X_{12} = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$

$S4 = X_8 \oplus X_9 \oplus X_{10} \oplus X_{11} \oplus X_{12} = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus \quad = 1$

Після перевірки на декодері основних контрольних співвідношень отримали:

1 Етап: 1001_2 – порушення.
2 Етап: $r = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$ – немає порушення

Висновок: відбулась подвійна помилка.

Циклічний код

Для зручності використання циклічні коди розглядаються шляхом подання двійкової кодової комбінації не у вигляді послідовності нулів і одиниць, а у вигляді полінома певного степеня:

$$F(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_1x + a_0, \qquad (3.2)$$

де x – основа системи числення;

a_i – алфавіт системи числення (в двійковій системі 0 та 1).

Наприклад, двійкова послідовність 01001 може бути записана у вигляді полінома від змінної x так:

$$F(x) = 0x^4 + 1x^3 + 0x^2 + 0x^1 + 1x^0 = x^3 + 1 \quad . \qquad (3.3)$$

Подання кодових комбінацій у вигляді (3.3) дозволяє звести дії над комбінаціями до дій з многочленами. При цьому додавання двійкових многочленів зводиться до додавання за модулем 2 коефіцієнтів при однакових степенях x. Наприклад, операція додавання має вигляд:

$$(x^3 + x^2 + 0 + 1) + (x + 1) = x^3 + x^2 + x\,.$$

Множення здійснюється за звичайним правилом множення степеневих функцій, але отримані в цьому випадку коефіцієнти при певному степені додаються за модулем 2. Операція множення має вигляд:

$$(x^3 + x^2 + 0 + 1) * (x + 1) = x4 + 0 + x^2 + x + 1\,.$$

Ділення здійснюється за правилами ділення степеневих функцій, при цьому операції віднімання замінюються операціями додавання за модулем 2. Операція ділення

$$\frac{x^4 + 0 \times x^3 + 0 \times x^2 + x + 1}{x + 1}$$

буде складатися з послідовності таких кроків:

1. Перше ділення

$$x^4 + 0 + x^2 + x + 1$$
$$x^4 + x^3$$

2. Друге ділення

$$x^3 + x^2 + x$$
$$x^3 + x^2$$

3. Третє ділення

$$x + 1$$

де $x + 1$ – результат суми за модулем 2 є 0. Ділення завершено.

Основна властивість циклічних кодів, що визначає їх назву, полягає в нижчевикладеному. Якщо комбінація $a_0, a_1, a_2, ..., a_{n-1}$ є дозволеною, то комбінація, яку отримують з неї шляхом циклічної перестановки розрядів, тобто комбінація $a_{n-1}, a_0, a_1, a_2, ..., a_{n-1}, a_{n-2}$, також належить цьому коду.

Принцип побудови циклічних кодів. Основний підхід до побудови циклічних кодів базується на використанні многочленів. Незвідним називається многочлен, який не може бути поданий у вигляді добутку многочленів нижчого порядку, тобто такий многочлен ділиться тільки на самого себе або на одиницю і не ділиться ні на який інший.

Принцип побудови циклічного коду викладено нижче. Множимо комбінацію простого *k*-значного коду $Q(x)$ на одночлен x^r, потім ділимо на твірний поліном $P(x)$, степінь якого дорівнює *r*. При діленні добутку $x^r * Q(x)$ на твірний поліном отримуємо частку $C(x)$ такого ж степеня, що і $Q(x)$. Результат множення та ділення подамо так:

$$x^r * Q(x) / P(x) = C(x) + R(x) / P(x), \qquad (3.4)$$

де $R(x)$ – залишок від ділення $x^r * Q(x)$ на $P(x)$.

Частка $C(x)$ має такий же степінь, як і кодова комбінація $Q(x)$ простого коду, тому $C(x)$ є кодовою комбінацією цього простого *k*-значного коду. Крім того, степінь залишку не може бути більше степеня твірного полінома, тобто, його найстарший степінь має бути не більше або дорівнювати (*r*-1). Таким чином, найбільша кількість розрядів залишку $R(x)$ не перевищує числа *r*. Множимо обидві частини рівності (3.4) на $P(x)$, робимо алгебраїчні перестановки та отримуємо

$$\mathrm{F(x)} = C(x) * P(x) = x^r * Q(x) + R(x) \qquad (3.5)$$

В (3.5) знак «-» перед $R(x)$ замінимо на знак «+», оскільки віднімання за модулем 2 зводиться до додавання.

Таким чином, кодову комбінацію циклічного *n*-значного коду можна отримати двома способами.

1. *Перший спосіб*: множимо кодову комбінацію $Q(x)$ простого коду на x^r, додаємо до цього добутку залишок $R(x)$, який отримали шляхом ділення добутку $x^r * Q(x)$ на твірний поліном $P(x)$.
2. *Другий спосіб*: множення кодової комбінації $C(x)$ простого *k*-значного коду на твірний поліном $P(x)$.

При побудові циклічних кодів першим способом розташування інформаційних символів всіх кодових комбінаціях строго впорядковано, а саме: вони займають k старших розрядів, інші (n-k) відводяться під контрольні розряди.

При другому способі створення циклічних кодів інформаційні та контрольні символи в комбінаціях циклічного коду не відділяються, що ускладнює процес декодування. Тому, в основному, використовується перший спосіб.

Методика корекції помилок циклічними кодами. Для виявлення та виправлення помилкового розряду потрібно зробити такі операції:

1. Прийняту комбінацію ділимо на твірний поліном $P(x)$;
2. Підраховуємо кількість одиниць в залишку ω (вага залишку). Якщо $w \le t_v$, де t_v – припустима кількість помилок, що виправляються даним кодом, прийняту комбінацію додають за модулем 2 з отриманим залишком. Сума дає виправлену комбінацію. Якщо $\omega > t_v$ то перехід на п. 3;
3. Здійснюють циклічний зсув прийнятої комбінації вліво на один розряд. Комбінацію, що отримали після циклічного зсуву, ділимо на твірний поліном $P(x)$. Якщо в результаті повторного ділення $w \le t_v$, то ділене додають з залишком, а потім перехід до п. 4;
4. Здійснюють циклічний зсув вправо на один розряд комбінації, що отримали в результаті додавання останнього діленого з останнім залишком. Отримана комбінація вже не має помилок. Якщо після першого циклічного зсуву та наступного ділення залишок є таким, що його вага $w > t_v$, то перехід до п. 5;
5. Повторюють операцію п. 3 до тих пір, поки не буде досягнуто $w \le t_v$. В цьому випадку комбінацію, що отримано в результаті останнього циклічного зсуву, додають з залишком від ділення цієї комбінації на твірний поліном, перехід до п. 6;
6. Здійснюють циклічний зсув вправо рівно на стільки розрядів, на скільки зсунута комбінація, що її отримали додаванням з останнім залишком, відносно прийнятої комбінації. В результаті отримаємо виправлену комбінацію.

3.6 Практичні приклади розв'язання задач з теорії кодування

Задача 1. Визначити відстань між комбінаціями 11100101 та 10100110.

Розв'язання. Для визначення кодової відстані потрібно підсумувати за модулем 2 дві ці комбінації:

$$\begin{array}{ll} & 11100101 \\ \oplus & \underline{10100110} \\ & 01000011 \end{array}$$

Отримана в результаті додавання нова кодова комбінація характеризується вагою $w = 3$ (кількість одиниць). Звідси, відстань між вихідними кодовими комбінаціями $d = 3$.

*Відповідь***.** Відстань між вихідними кодовими комбінаціями $d = 3$.

Задача 2. Визначити здатність до виявлення та виправлення помилок для коду, який має такі дозволені комбінації:

1010, 1001, 0101, 1100, 0110, 0011.

Розв'язання. Спочатку потрібно визначити мінімальну кодову відстань між комбінаціями. Для цього рекомендується зробити таблицю відстаней (табл. 3.5).

Як видно з таблиці, $d_{min} = 2$. З урахуванням цього випливає, що код може виявляти лише одиничні помилки. Виправляти помилки він не може.

Відповідь. Досліджуваний код може виявляти одиничні помилки без можливості їх виправлення.

Таблиця 3.5 – Відстані кодових комбінацій

	1010	1001	0101	1100	0110	0011
1010	0	2	4	2	2	2
1001		0	2	2	4	2
0101			0	2	2	2
1100				0	2	4
0110					0	2
0011						0

Задача 3. Побудувати код з перевіркою на парність для кодових комбінацій 101001100 и 101010110.

Розв'язання. Код з перевіркою на парність має в собі один контрольний розряд, що є сумою за модулем 2 всіх розрядів вихідного коду.

Перша кодова комбінація має парне число одиниць, ось чому код з перевіркою на парність буде мати вигляд

1010011000.

Контрольний розряд, який додається до другого коду, буде мати в собі цифру 1 і, звідси, отримаємо

1010101101.

Відповідь. Код з перевіркою на парність 101001100→1010011000; 101010110 →1010101101.

Задача 4. Побудувати код з подвоєнням для вихідного коду 101101.

Розв'язання. Замінимо кожну одиницю комбінацією 10, а кожний нуль – 01, отримаємо 100110100110.

Відповідь. Код з подвоєнням для вихідного коду 101101 → 100110100110.

Задача 5. Виявити, чи є помилка в коді з подвоєнням 1001001011.

Розв'язання. Комбінацію потрібно розділити на групи по два розряди:

10.01.00.10.11.

В коді є група, яка має в собі два нулі, і група, яка має в своєму складі дві одиниці, що говорить про наявність помилок.

Відповідь. Помилки присутні.

Задача 6. Побудувати інверсний код для вихідної кодової комбінації 10101101.

Розв'язання. Спочатку потрібно підрахувати кількість одиниць в кодовій комбінації. Оскільки вона непарна, то до даної комбінації дописується її інверсія

10101101→ 01010010.

Відповідь. Інверсний код 01010010.

Задача 7. Перевірити правильність прийому кодової комбінації 0110001 за умови, що був переданий код Гемінга ($d = 3$).

Розв'язання. В результаті перевірки отримаємо:

$S1 = 0 + 1 + 0 + 1 = 0,$
$S2 = 1 + 1 + 0 + 1 = 1,$
$S3 = 0 + 0 + 0 + 1 = 1.$

Таким чином, в результаті перевірок контрольне двійкове число 110, що вказує на спотворення 6-го символа. Змінивши цей символ, отримаємо код 0110011, який був переданий.

Відповідь. Відбувся прийом кодової комбінації з похибкою у 6-у символі.

3.7 Завдання для самостійної роботи

Завдання 1. Побудувати коди: Шеннона-Фано та Хаффмана для передачі восьми повідомлень з імовірностями їх появи, наведеними в табл. 3.6. Визначити середню довжину кодових слів $L_{ср}$.

Таблиця 3.6 – Варіанти індивідуальних завдань для самостійної роботи

Варіант	Ймовірність							
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
0	0,3	0,2	0,05	0,05	0,1	0,1	0,1	0,1
1	0,35	0,15	0,15	0,1	0,1	0,05	0,05	0,05
2	0,4	0,1	0,2	0,17	0,08	0,01	0,02	0,02
3	0,45	0,05	0,3	0,11	0,02	0,02	0,02	0,03
4	0,5	0,1	0,05	0,15	0,06	0,04	0,02	0,08
5	0,55	0,15	0,03	0,06	0,06	0,07	0,01	0,03
6	0,6	0,21	0,09	0,04	0,03	0,01	0,01	0,01
7	0,5	0,31	0,04	0,05	0,04	0,02	0,02	0,02
8	0,6	0,2	0,07	0,05	0,03	0,02	0,02	0,01
9	0,4	0,25	0,09	0,08	0,07	0,06	0,03	0,02

Завдання 2. Дослідження властивостей завадостійкого коду Гемінга.

1. Побудувати код Гемінга (12,8) і показати приклад для кодової комбінації, заданої в таблиці 3.7.

2. Дослідити можливості виявлення та виправлення одиничної помилки (в інформаційному та контрольному розрядах) в кодах з відстанню $d = 3$.

Таблиця 3.7 – Варіанти індивідуальних контрольних кодових комбінацій

Варіант	Контрольна кодова комбінація
1.	01001110
2.	11001100
3.	10101001
4.	00010001
5.	01010010
6.	00111000
7.	01110001
8.	10000111
9.	10100011
10.	11000101
11.	11100011
12.	00001000
13.	11000100
14.	10101000
15.	01001111

Завдання 3. Дослідження властивостей завадостійкого циклічного коду.

1. Побудувати циклічний код (7,4) і показати приклад для кодової комбінації, що задана в таблиці 3.8.

2. Дослідити можливості виявлення та виправлення поодинокої помилки.

Таблиця 3.8 – Варіанти індивідуальних контрольних кодових комбінацій циклічного коду

Варіант	Контрольна кодова комбінація циклічного коду
1.	1001
2.	1011
3.	1010
4.	1011
5.	1101
6.	1110
7.	1101
8.	1111
9.	0111
10.	0100
11.	0110
12.	0101
13.	0011
14.	1100
15.	1001

3.8 Контрольні питання

1. Вкажіть переваги та недоліки кодів Шеннона-Фано та Хаффмена.
2. Вкажіть основні кроки методики побудови коду Шеннона-Фано.
3. Вкажіть основні кроки методики побудови коду Хаффмена.
5. Назвіть основні характеристики надлишкових кодів.
6. Що розуміють під значністю та вагою кодової комбінації?
7. Що розуміють під відстанню між кодовими комбінаціями?
8. Що таке вектор помилки?
9. Що розуміють під кратністю помилки?
10. Який зв’язок між здатністю виявлення та виправлення помилок надлишковим кодом та кодовою відстанню?
11. Назвіть основні етапи побудови надлишкових кодів.
12. Як визначається кількість контрольних символів в коді Гемінга?
13. На яких позиціях розміщуються контрольні символи у коді Гемінга
14. Яким чином будуються контрольні перевірки в коді Гемінга?
15. Як виявляються та виправляються помилки в кодовій комбінації за допомогою коду Гемінга?
16. Як визначається відстань між кодовими комбінаціями?
17. Які властивості циклічного коду?
18. Як виявляються та виправляються помилки в кодовій комбінації за допомогою циклічного коду?
19. Які основні етапи побудови циклічного коду?
20. У чому відмінна особливість циклічних кодів?

РОЗДІЛ 4 ЛОГІКО-ЛІНГВІСТИЧНІ ТЕХНОЛОГІЇ В ЗАДАЧАХ ЗАХИСТУ ІНФРМАЦІЇ

В багатьох роботах, де розглядаються сучасні методи оцінювання рівня захисту комп'ютерної системи, пропонується при визначенні захищеної комп'ютерної системи притримуватись точки зору, що безпека є якісною характеристикою системи. В цьому випадку виникають труднощі відносно її вимірювання в будь-яких одиницях, і тоді рішення стосовно стану безпеки приймає експерт чи група експертів. Але потрібно враховувати той факт, що прийняття рішень стосовно безпеки залежить від суб'єктивних тверджень експерта, його знань та наявності методичного чи наукового забезпечення, тому використання нечітких множин стосовно прийняття рішень в галузі безпеки є доцільним.

4.1 Побудова нечітких моделей для оцінювання рівня захисту інформації в комп'ютерних системах

В різних експертних системах, зокрема й таких, що пристосовані до оцінювання рівня захисту в інформаційних системах, використовується нечіткий логічний висновок. В основі нечіткого висновку лежить база знань, що формується фахівцями з захисту інформації, вона має вигляд сукупності нечітких предикатних правил:

$$P_1: \text{якщо } x \in A_1, \text{ то } y \in B_1;$$
$$P_2: \text{якщо } x \in A_2, \text{ то } y \in B_2;$$
$$\dots$$
$$P_n: \text{якщо } x \in A_n, \text{ то } y \in B_n,$$

де x – вхідна змінна, їй надається значення з відомої множини даних;

y – змінна, що характеризує вихідне значення системи, воно буде обчислюватись;

A та B – функції належності, відповідно, для x та y.

В експертній системі знання експерта формуються у вигляді логічного виразу $A \to B$, який вказує на нечітке відношення посилання та висновку, позначається як R, а саме: $R = A \to B$, де позначення $\to$ називають нечіткою імплікацією.

Розглядаємо відношення R як нечітку підмножину прямого добутку $X \times Y$ універсальної множини посилань X та висновків Y. Отримання кінцевого результату (нечіткого) опрацювання висновку B^* відбувається з використанням певного спостереження A^* та знань $A \to B$, подається у вигляді композиційного правила нечіткого «modus ponens»

$$B^* = A^* \bullet R = A^* \bullet (A \to B),$$

де «•» – операція згортки або максмінної композиції. Ця операція може відбуватися за різними алгоритмами, але в будь-якому випадку процедура логічного висновку виконується в чотири етапи.

Перший етап полягає у введенні нечіткості або фазифікації. Функції належності, які визначаються для вхідних змінних, застосовуються до їх реальних значень для визначення ступеня істинності кожного посилання в кожному правилі.

Другий етап – це процедура логічного висновку. Значення істинності, що обчислюється для посилань кожного правила, застосовується до висновку в кожному правилі.

Третій етап полягає в реалізації операції композиції. Всі нечіткі підмножини, що їх отримано для кожної змінної висновку за всіма правилами, об'єднуються разом, щоб сформувати одну нечітку підмножину для всіх змінних висновку. Для такого об'єднання, як правило, використовується операція *max* (максимум, за операцією нечіткої логіки «АБО»). Реалізація максимуму відбувається, як максимум, по точках всіх нечітких підмножин.

Четвертий етап – це приведення до чіткості або дефазифікація. Процедура дефазифікації використовується, коли потрібно перетворити нечіткий набір значень висновку в чітке число. Існує багато підходів до реалізації цієї процедури.

Для опису вхідних параметрів та формалізації лінгвістичних даних, які характеризують безпеку інформації в системі за допомогою нечітких множин, потрібно використовувати нечітку та лінгвістичну змінні.

Нечітка змінна характеризується трійкою параметрів (α, X, A), де α – ім'я змінної; X – універсальна множина, A – нечітка множина на X, яка описує функцію належності $\mu_A(x)$, що описує значення нечіткої змінної α.

Лінгвістична змінна характеризується такими параметрами:

$$(\beta, T, X, W, M),$$

де β – ім'я лінгвістичної змінної; T – множина її значень (терм-множина). Ці терми є назвами значень нечіткої змінної, область визначення кожного значення множина X; W – синтаксична процедура, яка дозволяє оперувати значеннями терм-множини, наприклад, генерувати нові терми; M – семантична процедура, що дозволяє перетворити кожне нове значення лінгвістичної змінної після процедури W в нечітку змінну, тобто побудувати нову нечітку множину.

Залежно від універсальної множини X лінгвістичні змінні можуть мати числовий та нечисловий характер. Числовою лінгвістичною змінною буде така, у якої область визначення – інтервал на дійсній осі, а нечітку змінну, якій відповідають значення числової лінгвістичної змінної, називають нечітким числом.

Нечислові лінгвістичні змінні використовують, коли змінній неможливо дати кількісну характеристику, але є еталони, з якими її можна порівняти. Наприклад, нечисловою лінгвістичною змінною може бути така: «ефективність» з термами низька, середня, підвищена, висока.

4.2 Практичне використання логіко-лінгвістичного підходу в задачах апроксимації знань експертів в задачах захисту інформації

Розглянемо процедури впровадження нечітких множин щодо апроксимації експертних знань на прикладах.

Приклад 1. Нехай X – множина, що визначає довжину пароля для захисту комп'ютерної системи від несанкціонованого доступу. Треба побудувати нечітку множину A, яка описує терми $T \in$ { «мала» (M), «менша середньої» (MC), «середня» (C), «більша середньої» ($БС$), «велика» (B) } для лінгвістичної змінної «Довжина пароля».

Обчислення проводимо на основі опитування експертів. Число експертів n. Експерти дають відповіді на питання про належність елемента $x \in X$ або позитивно (n_1 – число позитивних відповідей), або негативно (n_2 – число негативних відповідей). В таблиці 4.1 зведено відповіді експертів щодо оцінювання довжини пароля за допомогою лінгвістичної оцінки для терма.

Таблиця 4.1 – Відповіді експертів для терма $T = MC$

N	X								
	6	7	8	9	10	11	12	13	14
n_1	1	2	4	5	5	4	3	2	1
n_2	4	3	1	0	0	1	2	3	4

Значення функції належності обчислюється за допомогою методу опитування за формулою

$$\mu_A(x) = n_1 / (n_1 + n_2).$$

Результати обчислень наведено в таблиці 4.2.

Таблиця 4.2 – Результати обчислень функції належності

N	X								
	6	7	8	9	10	11	12	13	14
$\mu_A(x)$	0.2	0.4	0.8	1	1	0.8	0.6	0.4	0.2

Таким чином, для терма $T = MC$ найбільш вагомим значенням довжини пароля є 9 або 10.

Приклад 2. Для оцінювання надійності захисту від несанкціонованого доступу певної комп'ютерної системи було залучено трьох експертів. В результаті оцінювання кожен з експертів озвучив свою оцінку у вигляді нечіткого значення. Треба обчислити середнє нечітке значення оцінок експертів за допомогою операції максмінної композиції.

Сума двох нечітких чисел за максмінною композицією дорівнює

$$X+Y=\bigcup_{i=1}^{n}\bigcup_{j=1}^{m}\left\{\left(\mu_x(x_i)\cap\mu_y(y_j)\right)/\left(x_i+y_j\right)\right\}.$$

Нечіткі множини для оцінок експертів за 10-бальною шкалою мають такий вигляд:

$$A_1 = \{\ 0{,}1/3;\ 0{,}6/4;\ 1/5;\ 0{,}7/6;\ 0{,}1/7\ \};$$
$$A_2 = \{\ 0{,}1/2;\ 0{,}6/3;\ 1/4;\ 0{,}5/5;\ 0{,}2/6\ \};$$
$$A_3 = \{\ 0{,}1/3;\ 0{,}4/4;\ 0.9/5;\ 1/6\ \ 0{,}7/7;\ 0.1/8\ \}.$$

Суму двох нечітких чисел обчислюємо за формулою максмінної композиції $A_1+A_2=$

0,1/5 0.1/6 0.1/7 0.1/8 0.1/9
0.1/6 0.6/7 0.6/8 0.5/9 0.2/10
0.1/7 0.6/8 1/9 0.5/10 0.2/11
0.1/8 0.6/9 0.7/10 0.5/11 0.2/12
0.1/9 0.1/10 0.1/11 0.1/12 0.1/13

$= \{0.1/5;\ 0.1/6;\ 0.6/7;\ 0.6/8;\ 1/9;\ 0.7/10;\ 0.5/11;\ 0.2/12;\ 0.1/13\ \}$

За наведеною формулою обчислюємо : $\left(A_1+A_2\right)+A_3=$

0.1/8 0.1/9 0.1/10 0.1/11 0.1/12 0.1/13
0.1/9 0.1/10 0.1/11 0.1/12 0.1/13 0.1/14
0.1/10 0.1/11 0.1/12 0.1/13 0.1/14 0.1/15
0.1/11 0.1/12 0.1/13 0.1/14 0.1/15 0.1/16
0.1/12 0.1/13 0.1/14 0.1/15 0.1/16 0.1/17
0.1/13 0.1/14 0.1/15 0.1/16 0.1/17 0.1/18
0.1/14 0.1/15 0.1/16 0.1/17 0.1/18 0.1/19
0.1/15 0.1/16 0.1/17 0.1/18 0.1/19 0.1/20
0.1/16 0.1/17 0.1/18 0.1/19 0.1/20 0.1/21

$= \{\ 0.1/8;\ 0.1/9;\ 0.1/10;\ 0.4/11;\ 0.6/12;\ 0.6/13;\ 0.9/14;\ 1/15;\ 0.7/16;\ 0.7/17;\ 0.5/18;$ $0.2/19;\ 0.1/20;\ 0.1\ /\ 21\}.$

Тоді середнє значення дорівнює : $\left(A_1+A_2+A_3\right)/3=$

$\{\ 0.1/2.67;\ 0.1/3;\ 0.1/3.33;\ 0.4/3.67;\ 0.6/4;\ 0.6/4.33;\ 0.9/4.67;\ 1/5;$ $0.7/5.33;\ 0.7/5.67;\ 0.5/6;\ 0.2/6.33;\ 0.1/6.67;\ 0.1\ /\ 7\}.$

Висновок: середнє нечітке значення оцінок експертів за допомогою операції максмінної композиції вказує на найбільше значення функції належності 0.9 для 4.67 балів, тобто в цілому оцінка близька до 5-ти балів.

Приклад 3. Розглянемо побудову нечіткої моделі для оцінювання рівня захищеності певної комп'ютерної системи. Модель буде мати бальну шкалу оцінювання та реалізована на основі операцій з нечіткими множинами. Вихідними даними для здійснення операцій з нечіткими множинами є дані від експертів та результати опитувань користувачів. Задача оцінювання рівня захищеності розв'язується за методикою, викладеною в роботі [1] за нижчевказаною послідовністю кроків.

По-перше складається експертний запит, компоненти якого попередньо ранжуються, та обчислюється коефіцієнт важливості для кожної компоненти запиту. В основі ранжування – матриця попарних порівнянь Сааті, або перетворена матриця. Далі користувачі дають відповіді на попередньо ранжовані компоненти експертного запиту за *N*-бальною шкалою. Шкала для оцінок теж складається експертом, діапазон шкали може змінюватись залежно від важливості загрози.

По-друге, крім процедури ранжування альтернатив, експерти будують нечіткі еталони, які відображають лінгвістичну змінну «рівень безпеки». Ці еталони будуть зразком для порівняння з нечіткими числами. Подамо поширену терм-множину лінгвістичної змінної «рівень безпеки», що має п'ять нечітких термів

$$T = \left\{ T_1,\ T_2, T_3, T_4, T_5 \right\}$$

з відповідними назвами «Низький» (Н), «Нижче середнього» (НС), «Середній» (С), «Вище середнього» (ВС) , «Високий» (В). Позначимо число еталонів

L = 5, шкала для всіх 0 – 4. Побудова еталонів для вказаних термів базується на таких нечітких множинах [1]:

Н = {1/0, 0,5/1, 0,2/2, 0,1/3, 0,06/4};
НС = {0,5/0, 1/1, 0,5/2, 0,2/3, 0,1/4};
С = {0,1/0, 0,2/1, 0,5/2, 1/3, 0,5/4};
ВС = {1/0, 0,5/1, 0,2/2, 0,1/3, 0,06/4};
В = {0,06/0, 0,1/1, 0,2/2, 0,5/3, 1/4}.

Для проведення розрахунків вводиться діапазон $\left[\underline{X}_j, \overline{X}_j\right]$, ($j$=1…$n$). Це зміни параметра X^*_j, нижня межа якого 0, верхня дорівнює N_j – максимально можлива кількість балів за компонентами запиту. Значення X^*_j відображається на множину еталонних нечітких чисел U = [0, L-1], де L – кількість еталонів, за формулою [1]

$$U_j^* = (L-1)\frac{X_j^* - \underline{X}_j}{\overline{X}_j - \underline{X}_j}$$

Далі відбувається розрахунок функції належності, де i = 1…L, за співвідношенням

$$\mu_i^j\left(U_j^*\right) = \left[\frac{1}{1+\left(U_j^* - i + 1\right)^2}\right]^{PN_j \times n}$$

Значення PN_j відповідає коефіцієнту важливості KV, де j змінюється від одиниці до n. Значення рівня захисту в системі буде визначатися за допомогою логічного виразу [ ]

$$\mu_S(X_j^*) = \bigvee_{i=1}^{L} \bigwedge_{j=1}^{n} \mu_i^j,$$

де $i = 1$,

L – номер терма з терм-множини T;

$j = 1, n$ – номер компоненти експертного запиту.

Розглянемо практичні розрахунки вказаних методик.

Є чотирикомпонентний запит, який складається з таких компонент.

Перша компонента: чи оновлюються програмні компоненти.

Друга компонента: чи проводяться курси навчання персоналу.

Третя компонента: чи ведеться спостереження важливих системних процесів.

Четверта компонента: чи проводиться аналіз системи на вразливості.

Межі діапазону шкали задає експерт для кожної компоненти окремо. В даній задачі для першого діапазону [0; 8], другого [0; 5], третього [0; 3], четвертого [0; 6].

Використовуємо матрицю парних порівнянь та перетворену матрицю для обчислення KV (коефіцієнт важливості) кожної компоненти.

Матриця парних порівнянь має такий вигляд:

$$A = (a_{ij}) = \begin{vmatrix} 1 & 7 & 3 & 5 \\ \frac{1}{7} & 1 & 9 & 3 \\ \frac{1}{3} & \frac{1}{9} & 1 & 7 \\ \frac{1}{5} & \frac{1}{3} & \frac{1}{7} & 1 \end{vmatrix}.$$

Елемент перетвореної матриці визначають як

$$a'_{vw} = \begin{cases} \dfrac{100}{(a_{ij}+1)\times a_{ij}}, & \forall i<j: v=i, w=j \\ 1, & \forall i=j: v=i, w=j \\ \dfrac{100}{(a_{ij}+1)}, & \forall i>j: v=i, w=j \end{cases},$$

де $i=j=\overline{1,n}$, а n – кількість компонент запиту.

Тоді елементи перетвореної матриці будуть мати такі значення:

$$A' = \left(a'_{ij} \right) = \begin{vmatrix} 1 & 87,5 & 75 & 83,33 \\ 12,5 & 1 & 90 & 75 \\ 25 & 10 & 1 & 87,5 \\ 16,67 & 25 & 12,5 & 1 \end{vmatrix}.$$

Занесемо в таблицю результати обчислення KV для всіх компонентів $\left(P_j,\ j=\overline{1,4} \right)$ та нормалізовані KV $\left(PN_j,\ j=\overline{1,4} \right)$ і відповіді користувачів. Нормалізований KV обчислюється за співвідношенням $PN_j = Pj / \sum_{i=1}^{4} P_j$.

Таблиця 4.3 – Результати опитування користувачів і експертів

Номер компоненти (j)	$P_j,\ j=\overline{1,4}$	$PN_j,\ j=\overline{1,4}$	Відповіді (в балах) $X_j^*,\ j=\overline{1,4}$
1	87,5+75+83,33=245,83	0,41	6
2	12,5+90+75=177,5	0,3	4,5
3	25+10+87,5=122,5	0,2	2,3
4	16,67+25+12,5 =54,17	0,09	1

Використовуючи вирази для U_j^* та $\mu_i^j\left(U_j^* \right)$ проводимо обчислення і визначаємо значення $U_j^*,\ j=\overline{1,4}$:

$$U_1^* = (5-1) \times \frac{6-0}{8-0} = 4 \times \frac{6}{8} = 3;$$

$$U_2^* = (5-1) \times \frac{4,5-0}{5-0} = 4 \times \frac{4,5}{5} = 3,6;$$

$$U_3^* = (5-1) \times \frac{2,3-0}{3-0} = 4 \times \frac{2,3}{3} = 3,06;$$

$$U_4^* = (5-1) \times \frac{1-0}{6-0} = 4 \times \frac{1}{6} = 0,66;$$

визначаємо значення $\mu_i^j\left(U_j^* \right),\ i=\overline{1,5}$:

$$\mu_1^1\left(U_1^* \right) = \left[\frac{1}{1+(3-1+1)^2} \right]^{0,41 \times 4} = \left[\frac{1}{1+9} \right]^{1,64} = 0,02;$$

$$\mu_3^1\left(U_1^* \right) = \left[\frac{1}{1+(3-3+1)^2} \right]^{0,41 \times 4} = \left[\frac{1}{1+1} \right]^{1,64} = 0,19;$$

$$\mu_1^2\left(U_2^* \right) = \left[\frac{1}{1+(3,6-1+1)^2} \right]^{0,2 \times 4} = \left[\frac{1}{1+12,96} \right]^{0,8} = 0,44;$$

$$\mu_4^2\left(U_2^* \right) = \left[\frac{1}{1+(3,6-3+1)^2} \right]^{0,2 \times 4} = \left[\frac{1}{1+2,56} \right]^{0,8} = 0,14;$$

$$\mu_4^3\left(U_2^* \right) = \left[\frac{1}{1+(3,06-7+1)^2} \right]^{0,18 \times 4} = \left[\frac{1}{1+5,76} \right]^{0,72} = 0,27;$$

$$\mu_4^4\left(U_2^* \right) = \left[\frac{1}{1+(0,66-1+1)^2} \right]^{0,21 \times 4} = \left[\frac{1}{1+0,43} \right]^{0,84} = 0,69;$$

а результати обчислень заносимо у таблицю 4.4.

Таблиця 4.4 – Результати обчислень

Номер компоненти (j)	$U_j^*,\ j=\overline{1,4}$	$\mu_1^j(U_j^*)$	$\mu_2^j(U_j^*)$	$\mu_3^j(U_j^*)$	$\mu_4^j(U_j^*)$	$\mu_5^j(U_j^*)$
1	4,5	0,02	0,05	0,19	0,84	0,54
2	2	0,14	0,44	1,00	0,44	0,14
3	2	0,27	0,57	1,00	0,57	0,27
4	2,67	0,47	0,62	0,88	0,96	0,69

Згідно з виразом $\mu_s(X_j^*) = \vee_{i=1}^{L} \wedge_{j=1}^{n} \mu_i^j$, знаходимо показник рівня захищеності

$$\mu_s(X_j^*) = \bigvee_{i=1}^{L} \bigwedge_{j=1}^{n} \mu_i^j = \left[\mu_1^1(3) \wedge \mu_1^2(3,6) \wedge \mu_1^3(3,06) \wedge \mu_1^4(0,66)\right]$$
$$\vee \left[\mu_2^1(3) \wedge \mu_2^2(3,6) \wedge \mu_2^3(3,06) \wedge \mu_2^4(0,66)\right]$$
$$\vee \left[\mu_3^1(3) \wedge \mu_3^2(3,6) \wedge \mu_3^3(3,06) \wedge \mu_3^4(0,66)\right]$$
$$\vee \left[\mu_4^1(3) \wedge \mu_4^2(3,6) \wedge \mu_4^3(3,06) \wedge \mu_4^4(0,66)\right]$$
$$\vee \left[\mu_5^1(3) \wedge \mu_5^2(3,6) \wedge \mu_5^3(3,06) \wedge \mu_5^4(0,66)\right]$$
$$= [0,02 \wedge 0,14 \wedge 0,27 \wedge 0,47] \vee [0,05 \wedge 0,44 \wedge 0,57 \wedge 0,62]$$
$$\vee\ [0,19 \wedge 1 \wedge 1 \wedge 0,88] \vee [0,84 \wedge 0,44 \wedge 0,57 \wedge 0,96]$$
$$\vee\ [0,54 \wedge 0,14 \wedge 0,27 \wedge 0,69] = 0,02 \vee 0,05 \vee 0,19 \vee 0,44 \vee 0,14 = 0,44$$

З табл. 4.4 видно, що обчислене значення $\mu_s = \mu_4$ (тобто $i = 4$), тому для прийняття рішення вибирається нечіткий терм T_4, назва якого «Вище середнього», що і визначає рівень захищеності у даному прикладі.

Приклад 4. Побудова нечіткої моделі розпізнавання стану складної системи за допомогою нечітких множин та лінгвістичних змінних

Задачу розпізнавання стану складної системи будемо розглядати як процес прийняття рішення в системі з одним вихідним параметром (станом) та n вхідними змінними. Робимо формалізацію причинно-наслідкових зв'язків між параметрами системи та її станом, ця процедура полягає в описі цих зв'язків природною мовою з використанням теорії нечітких множин та лінгвістичних змінних у вигляді матриці знань.

Будемо позначати: y – певний вихідний параметр, значення якого визначає стан системи в певній галузі; x_1, x_2, …, x_n – вхідні зміні, які характеризують стан системи і які впливають на її стан, тобто,

$$y = f_y(x_1, x_2, \ldots, x_n),$$

де f_y – певна апріорно невідома функція, яка пов'язує вхідні і вихідні змінні.

Залежно від постановки задачі, вхідні і вихідні параметри можуть бути як кількісними, так і якісними. Кількісні параметри задаються діапазонами зміни, а якісні визначаються множинами можливих значень.

Всі параметри розглядаються як лінгвістичні змінні. Для оцінювання лінгвістичних змінних x_i, $i = 1, n$ будемо використовувати якісні терми з нижчевказаних терм-множин:

$$A_i = \left\{a_i^1, a_i^2, ..., a_i^{l_i}\right\}$$

– терм-множина змінної x_i. $i = 1, n$;

$$Y = \left\{y_1, y_2, ..., y_m\right\}$$

– терм-множина змінної y,

де a_i^p – p-ий лінгвістичний терм параметра x_i;

y_j – j-ий лінгвістичний терм параметра y, який визначає j-ий стан;

m – кількість різних станів системи.

Кожен терм має власну функцію належності.

Матрицю знань подамо таблицею 4.5, де k_i – кількість випадків зі станом y_j.

Таблиця 4.5 – Матриця знань

Номер правила	Параметри стану системи				Стан
	x_1	x_2	… x_i …	x_n	y
11 12 … 1_{k1}	a_1^{11} a_1^{12} … $a_1^{1k_1}$	a_2^{11} a_2^{12} … $a_2^{1k_1}$	… a_i^{11} … … a_i^{12} … … … $a_i^{1k_1}$ …	a_n^{11} a_n^{12} … $a_n^{1k_1}$	y_1
…	…	…	…	…	…
j_1 j_2 … j_{kj}	a_1^{j1} a_1^{j2} … $a_1^{jk_1}$	a_2^{j1} a_2^{j2} … $a_2^{jk_1}$	… a_i^{j1} … … a_i^{j2} … … … $a_i^{jk_1}$ …	a_n^{j1} a_n^{j2} … $a_n^{jk_1}$	y_j
…	…	…	…	…	…
m_1 m_2 … m_{km}	a_1^{m1} a_1^{m2} … $a_1^{mk_1}$	a_2^{m1} a_2^{m2} … $a_2^{mk_1}$	… a_i^{m1} … … a_i^{m2} … … … $a_i^{mk_1}$ …	a_n^{m1} a_n^{m2} … $a_n^{mk_1}$	y_m

Матриця знань – це система логічних висловлень типу «ЯКЩО – ТО, ІНАКШЕ», яка пов'язує значення параметрів $x_1, x_2, \ldots, x_n$ з одним зі станів.

ЯКЩО $(x_1 = a_1^{11})$ І $(x_2 = a_2^{11})$ І ... І $(x_n = a_n^{11})$ АБО
$(x_1 = a_1^{12})$ І $(x_2 = a_2^{12})$ І ... І $(x_n = a_n^{12})$ АБО ....
$(x_1 = a_1^{1k_1})$ І $(x_2 = a_2^{1k_1})$ І ... І $(x_n = a_n^{1k_1})$,

то $y = y_1$, ІНАКШЕ

ЯКЩО $(x_1 = a_1^{j1})$ І $(x_2 = a_2^{j1})$ І ... І $(x_n = a_n^{j1})$ АБО
$(x_1 = a_1^{j2})$ І $(x_2 = a_2^{j2})$ І ... І $(x_n = a_n^{j2})$ АБО ...
$(x_1 = a_1^{jk_1})$ І $(x_2 = a_2^{jk_1})$ І ... І $(x_n = a_n^{jk_1})$,

то y = y_2, ІНАКШЕ

ЯКЩО $(x_1 = a_1^{m1})$ І $(x_2 = a_2^{m1})$ І ... І $(x_n = a_n^{m1})$ АБО
$(x_1 = a_1^{m2})$ І $(x_2 = a_2^{m2})$ І ... І $(x_n = a_n^{m2})$ АБО
$(x_1 = a_1^{mk_1})$ І $(x_2 = a_2^{mk_1})$ І ... І $(x_n = a_n^{mk_1})$,

то $y = y_m$.

Або, з використанням символів $\cap$ та $\cup$, система може бути записана в більш компактному вигляді

$$\bigcup_{p=1}^{k_j}\left[\bigcap_{i=1}^{n}\left(x_i = a_i^{jp}\right)\right] \to d_j,\ j = 1, m$$

Базі знань відповідає система нечітких логічних рівнянь, яка дозволяє обраховувати функції належності різних станів системи за фіксованими значеннями вхідних параметрів стану системи. Система нечітких логічних рівнянь може бути подана таким чином:

$$\mu^{y_1}(x_1, x_2, ..., x_n) = \mu^{11}(x_1)\cdot\mu^{11}(x_2)\cdot\ldots\cdot\mu^{11}(x_n)\vee$$
$$\mu^{12}(x_1)\cdot\mu^{12}(x_2)\cdot\ldots\cdot\mu^{12}(x_n)\vee\ldots$$
$$\mu^{1k_1}(x_1)\cdot\mu^{1k_1}(x_2)\cdot\ldots\cdot\mu^{1k_1}(x_n),$$

$$\mu^{y_2}(x_1, x_2, ..., x_n) = \mu^{21}(x_1)\cdot\mu^{21}(x_2)\cdot\ldots\cdot\mu^{21}(x_n)\vee$$
$$\mu^{22}(x_1)\cdot\mu^{22}(x_2)\cdot\ldots\cdot\mu^{22}(x_n)\vee\ldots$$
$$\mu^{2k_1}(x_1)\cdot\mu^{2k_1}(x_2)\cdot\ldots\cdot\mu^{2k_1}(x_n),\ldots$$

$$\mu^{y_m}(x_1, x_2, ..., x_n) = \mu^{m1}(x_1)\cdot\mu^{m1}(x_2)\cdot\ldots\cdot\mu^{m1}(x_n)\vee$$
$$\mu^{m2}(x_1)\cdot\mu^{m2}(x_2)\cdot\ldots\cdot\mu^{m2}(x_n)\vee\ldots$$
$$\mu^{mk_1}(x_1)\cdot\mu^{mk_1}(x_2)\cdot\ldots\cdot\mu^{mk_1}(x_n).$$

де «$\vee$» – логічне АБО, «$\cdot$» – логічне І.

В загальному випадку система нечітких логічних рівнянь записується співвідношенням

$$\mu^{y_j}(x_1, x_2, \ldots, x_n) = \bigvee_{p=1}^{k_j} \left[\bigwedge_{i=1}^{n} \mu^{jp}(x_i) \right], \; j = 1, m.$$

Для побудови моделі для оцінювання рівня захисту в практичних задачах інформація, яка потрібна для формування моделі, може бути отримана на основі знань експертів в галузі захисту інформації.

4.3 Контрольні питання

1. Поняття нечіткої множини та підмножини.
2. Як прийнято позначати нечітку множину?
3. Які нечіткі множини називають рівними?
4. Як позначається функція належності нечіткої множини?
5. Що таке лінгвістична змінна?
6. Дайте означення основних операцій алгебри нечітких множин.
7. Які методи побудови функцій належності ви знаєте?
8. Що таке лінгвістичний терм?
9. Як побудувати множину термів для лінгвістичної змінної?
10. Що таке нечітке бінарне відношення?
11. Що таке нечітка змінна?
12. Що таке нечіткий еталон?
13. Дайте означення основних операцій нечіткої логіки.
14. Які операції нечіткої арифметики ви знаєте?
15. Як записати логічні висловлення?
16. Як будується матриця знань?
17. Що таке нечітке число?
18. Як відбувається апроксимація знань експертів за допомогою нечітких множин?
19. Як будується матриця попарних порівнянь?
20. Які операції нечіткої логіки використовуються при побудові нечітких моделей для визначення рівня захисту інформації?

РОЗДІЛ 5 ТЕСТОВІ ЗАВДАННЯ

5.1 Теорія множин та відношень

1. Нехай є множини A = {1, 2, 3, 4, 5}, B = {6, 7, 8, 9, 10}, множина, яка є їх об'єднанням, дорівнює:

1.{1, 2, 3, 4, 5,6, 7, 8, 9, 10};
2.{1, 2, 3, 4, 7, 8, 9, 10};
3.{1, 2, 3, 4, 7, 8, 9, 10, 5};
4.{2, 3, 4, 7, 8, 9, 10, 5}.

2. Нехай є множини A = { 2, 3, 4, 5}, B = {2, 3, 4, 7, 8, 9, 10}, множина, яка є їх перетином, дорівнює:

1.{2, 3, 4, 5, 7, 8, 9, 10};
2.{2, 3, 4 };
3.{7, 8, 9, 10, 5};
4.{2, 3, 4, 7, 8, 9, 10, 5}.

3. Нехай є множини A = {1, 2, 3, 4, 5 }, B = {5, 6, 7, 8, 9, 10}, множина, яка є їх різницею, дорівнює:

1. {1, 2, 3, 4, 5};
2. {1, 2, 3, 4, 7, 8, 9, 10};
3. {1, 2, 3, 4};
4. { 2, 3, 4, 7, 8, 9, 10, 5}.

4. Нехай є універсальна множина U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, для множини A={5, 6, 7, 8} множина-доповнення універсальної множини U дорівнює:

1.{1, 2, 3, 4, 5 };
2.{1, 2, 3, 4, 9, 10};
3.{1, 2, 3, 4};
4.{ 2, 3, 4, 7, 8, 9, 10, 5}.

5. Нехай є множини A ={1, 2, 3}, B ={9, 10}, множина, яка є їх декартовим добутком (A*B), дорівнює:

1.{ (1, 9), (1,10), (2,9), (2,10), (3,9), (3,10) };
2.{ (1,10), (2,9), (2,10), (3,9), (3,10) };
3.{ (2,10), (3,9), (3,10) ,1, 2, 3};
4.{2, 3, 4, 7, 8, 9, 10, 5}.

6. Нехай є впорядкована множина *C* = { (1, 9), (1,10), (2,9), (2,10), (3,9), (3,10), (3,11)}, множина, яка є перерізом множини *C* за елементом 3, дорівнює:

1. { (1, 9), (1,10), (2,9), (2,10) };
2. { (1,10), (2,9), (2,10), (3,9), 3, 10};
3. { (2,10), (3,9), (3,10) ,1, 2, 3};
4. { 9, 10, 11 }.

7. Відношенням строгого порядку є відношення, яке має властивості:

1. Рефлексивності, симетрії, транзитивності;
2. Рівності, симетрії, транзитивності;
3. Антирефлексивності, антисиметрії, транзитивності;
4. Симетрії, транзитивності.

8. Нехай є множина *A* = { а, б, с }, сукупність всіх підмножин цієї множини дорівнює:

1. {а} { б}, { с}, { а, б, с }, { а, с };
2. {а} { б}, { с}, { а, б, с }, { };
3. {б}, {с}, { а, б, с }, { а, с }, { б, с }, { а, б, с },{ };
4. {а}, {б}, {с}, { а, б }, { а, с }, { б, с }, { а, б, с },{ }.

9. Нехай є впорядкована множина *C* = {(1, 9),(1,10),(2,9),(2,10),(3,9),(3,10)}, для цієї множини полем відношення є:

1. { (1,2) };
2. { (1,1), (2,2), (2,3) };
3. { (2,1), (3,3) ,1, 2, 3};
4. {1, 2, 3, 9, 10}.

10. Нехай є впорядковані множини *C* = { (1,10), (2,9), (2,10) } та А = { (2,10), (2,11), (10,3) }, об’єднанням цих множин є:

1. { (1,10), (2,9), (2,10), (2,11), (10,3) };
2. { (2,11), (2,3), (3,11) };
3. { 10, 9, 3, 11 };
4. { 2, 3, 9 }.

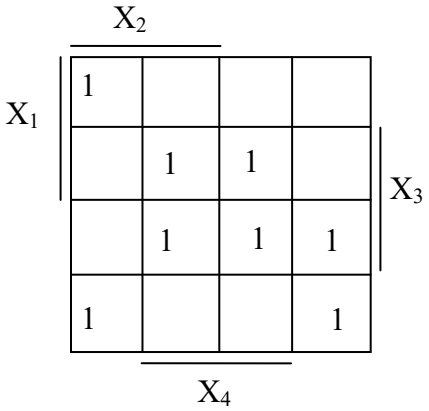
Таблиця 5.1 – Відповіді до тестових завдань 5.1

1.1	2.2	3.3	4.2	5.1	6.4	7.3	8.4	9.4	10.1

5.2 Закони та тотожності алгебри логіки

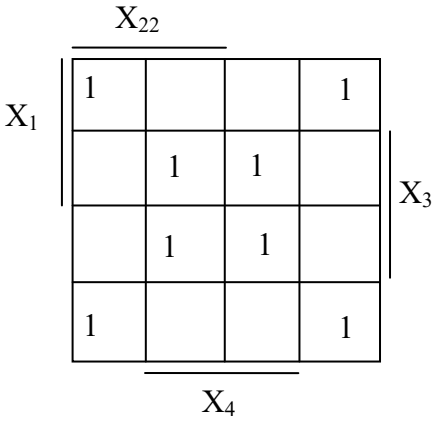
Знайти правильний запис однієї з можливих мінімальних форм логічної функції від чотирьох аргументів за діаграмами Вейча:

1.



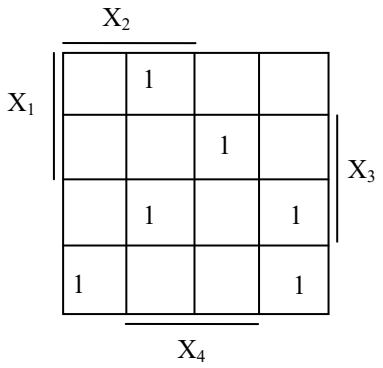
1. $f = x_2\overline{x_3}\,\overline{x_4} + x_2x_3x_4 + \overline{x_1}x_3x_4 + \overline{x_1}\,\overline{x_2}\,\overline{x_4}$
2. $f = x_1\overline{x_2}x_4 + \overline{x_3}x_4 + x_1\overline{x_3}\,\overline{x_4} + x_2x_3\overline{x_4} + x_1x_3x_4 + x_1\overline{x_2}x_3$;
3. $f = x_1\overline{x_3}x_4 + x_1\overline{x_4} + x_1\overline{x_3}\,\overline{x_4} + \overline{x_2}\,\overline{x_3}\,\overline{x_4} + x_1\overline{x_3}x_4 + x_1x_2\overline{x_4}$;
4. $f = x_2\overline{x_3}\,\overline{x_4} + x_3x_4 + \overline{x_1}\,\overline{x_2}\,\overline{x_4}$.

2.



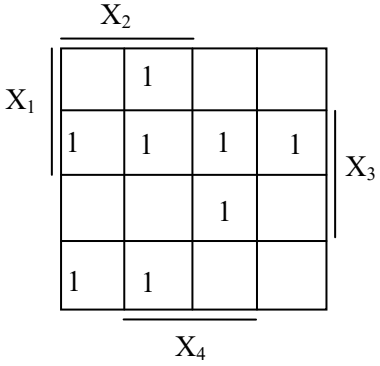
1. $f = x_1\overline{x_3}\,\overline{x_4} + x_2x_3\overline{x_4} + \overline{x_1}x_2x_3 + x_2\overline{x_3}x_4 + x_1\overline{x_3}x_4 + \overline{x_2}\,\overline{x_3}\,\overline{x_4}$;
2. $f = x_2x_3\overline{x_4} + \overline{x_1}x_2\overline{x_4} + \overline{x_1}x_2x_3 + \overline{x_1}x_3x_4 + \overline{x_2}x_3x_4 + \overline{x_1}\,\overline{x_2}\,\overline{x_4}$;
3. $f = \overline{x_3}\,\overline{x_4} + x_1x_3x_4 + \overline{x_2}x_3x_4$;
4. $f = \overline{x_3}\,\overline{x_4} + x_3x_4$.

3.



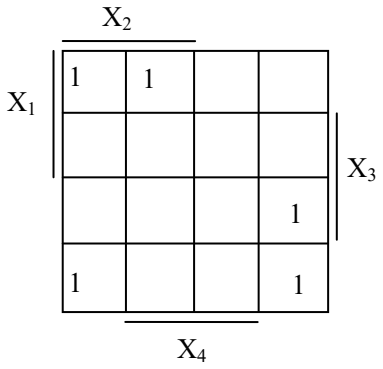
1. $f = x_2\overline{x_3}x_4 + \overline{x_1}x_2\overline{x_3} + \overline{x_2}x_3x_4 + x_1\overline{x_2}x_3 + x_1x_3\overline{x_4}$;
2. $f = x_1x_2\overline{x_3} + x_2\overline{x_3x_4} + \overline{x_1x_2x_4} + \overline{x_1}x_2x_3x_4 + x_1\overline{x_2}x_3x_4$;
3. $f = x_2\overline{x_3x_4} + x_1x_2\overline{x_3} + \overline{x_1x_3x_4} + \overline{x_1x_2x_4}$;
4. $f = x_1x_2\overline{x_3}x_4 + \overline{x_1x_3x_4} + \overline{x_1x_2x_4} + \overline{x_1}x_2x_3x_4 + x_1\overline{x_2}x_3x_4$.

4.



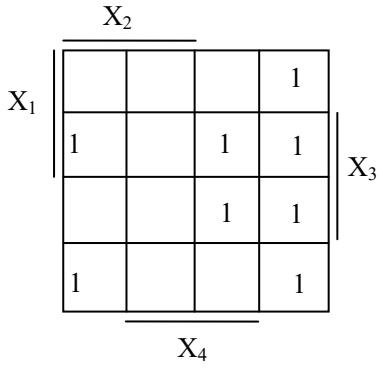
1. $f = x_1x_3\overline{x_4} + x_2\overline{x_3}x_4 + \overline{x_2}x_3x_4 + \overline{x_1}x_2\overline{x_3}$;
2. $f = x_2\overline{x_3x_4} + x_1x_2\overline{x_3} + x_1x_2x_4 + x_2x_3x_4 + \overline{x_1}x_3x_4 + \overline{x_1x_2}x_3$;
3. $f = x_2\overline{x_3x_4} + x_1x_2\overline{x_3} + \overline{x_1x_3x_4} + \overline{x_1x_2x_4}$;
4. $f = x_1x_2x_4 + x_1x_3 + \overline{x_2}x_3x_4 + \overline{x_1}x_2\overline{x_3}$.

5.



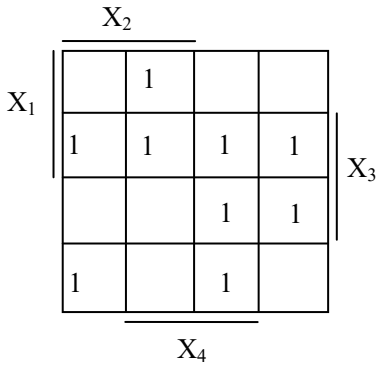
1. $f = x_2 x_3 \overline{x_4} + \overline{x_1} x_2 \overline{x_4} + \overline{x_1} x_2 x_3 + \overline{x_1} x_3 x_4 + \overline{x_2} x_3 x_4 + \overline{x_1 x_2} x_4$;
2. $f = x_1 x_2 \overline{x_3} + x_2 \overline{x_3 x_4} + \overline{x_1 x_2 x_4} + \overline{x_1 x_3 x_4}$;
3. $f = x_1 x_2 \overline{x_3} + \overline{x_1 x_3 x_4} + \overline{x_1 x_2 x_4}$;
4. $f = x_1 x_2 \overline{x_3} + \overline{x_1 x_3 x_4} + \overline{x_1 x_2 x_4} + \overline{x_1} x_2 x_3 x_4$.

6.



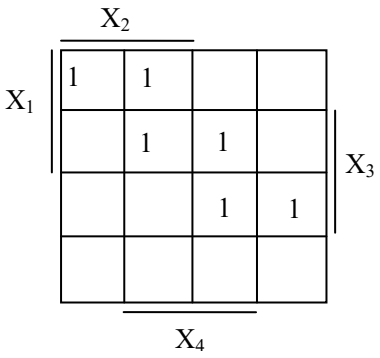
1. $f = x_1 x_3 \overline{x_4} + x_1 \overline{x_2 x_4} + \overline{x_2} x_3 x_4 + \overline{x_1 x_3 x_4}$;
2. $f = x_1 x_3 \overline{x_4} + \overline{x_1 x_3 x_4} + \overline{x_2} x_3 + \overline{x_2 x_4}$;
3. $f = x_1 x_2 \overline{x_3} + x_1 x_2 x_4 + x_1 x_3 x_4 + \overline{x_2} x_3 x_4 + \overline{x_1 x_2} x_3 + \overline{x_1 x_2 x_4}$;
4. $f = \overline{x_1 x_3 x_4} + x_1 x_3 \overline{x_4} + \overline{x_2 x_3 x_4} + x_1 \overline{x_2 x_4} + x_1 \overline{x_2} x_3 + \overline{x_2} x_3 x_4$.

7.



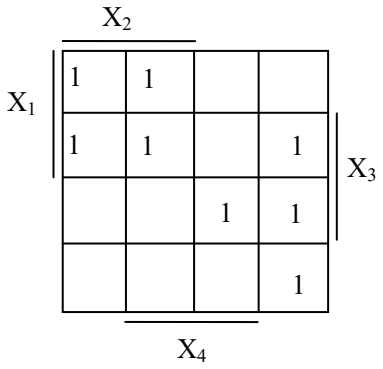
1. $f = x_1x_2x_3 + x_1x_2x_4 + \overline{x_1}\,x_2x_4 + \overline{x_1}\,x_2x_3 + \overline{x_1}\,x_2\,\overline{x_3}\,x_4$;
2. $f = x_2\,\overline{x_3}\,\overline{x_4} + x_1x_2\,\overline{x_3} + \overline{x_1}\,x_3x_4 + \overline{x_1}\,x_2\,x_4$;
3. $f = x_1x_2 + \overline{x_1}\,x_2x_3 + \overline{x_1}\,x_2\,x_4$;
4. $f = x_1x_3 + x_1x_2x_4 + \overline{x_2}\,x_3 + \overline{x_1}\,x_2\,x_4 + \overline{x_1}\,x_2\,\overline{x_3}\,x_4$.

8.



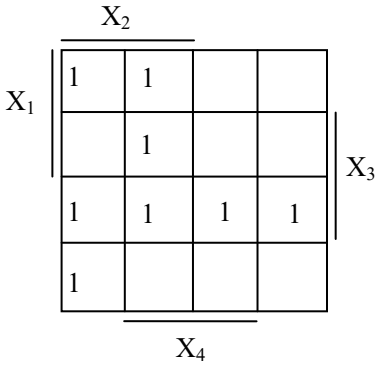
1. $f = x_1x_2\,\overline{x_4} + x_2x_3\,\overline{x_4} + x_1x_2x_3 + \overline{x_1}\,x_2x_3 + \overline{x_1}\,x_3\,\overline{x_4}$;
2. $f = x_1x_2x_3 + x_1x_2x_4 + \overline{x_1}\,x_2x_4 + \overline{x_1}\,x_2x_3$;
3. $f = x_1x_2\,\overline{x_3} + x_1x_2x_4 + \overline{x_2}\,x_3x_4 + \overline{x_1}\,x_2\,x_4$;
4. $f = x_1x_2\,\overline{x_3} + x_1x_3x_4 + \overline{x_2}\,x_3x_4$.

9.



1. $f = x_2\overline{x_3}\,\overline{x_4} + x_1\overline{x_3}\,\overline{x_4} + \overline{x_2}\,\overline{x_3}\,\overline{x_4} + \overline{x_1}\,\overline{x_3}\,\overline{x_4} + x_1x_3x_4 + \overline{x_2}x_3x_4$;
2. $f = x_1x_2 + \overline{x_1}\,\overline{x_2}x_3 + \overline{x_1}\,\overline{x_2}\,\overline{x_4}$;
3. $f = x_1x_2 + \overline{x_1}\,\overline{x_2}x_3 + \overline{x_1}\,\overline{x_2}\,\overline{x_4} + \overline{x_2}x_3\overline{x_4}$;
4. $f = x_1\overline{x_3}\,\overline{x_4} + \overline{x_2}\,\overline{x_3}\,\overline{x_4} + x_3x_4$.

10.



1. $f = x_1x_2\overline{x_3} + x_2\overline{x_3}\,\overline{x_4} + x_2x_3x_4 + \overline{x_1}\,\overline{x_2}x_3$;
2. $f = x_2\overline{x_3}\,\overline{x_4} + x_1x_2\overline{x_3} + x_1x_2x_4 + x_2x_3x_4 + \overline{x_1}x_3x_4 + \overline{x_1}\,\overline{x_2}x_3$;
3. $f = x_1x_2\overline{x_4} + x_2x_3\overline{x_4} + x_1x_2x_3 + \overline{x_1}\,\overline{x_2}x_3 + \overline{x_1}x_3\overline{x_4}$;
4. $f = x_1x_2\overline{x_3} + x_1x_2x_4 + \overline{x_1}x_2\overline{x_4} + \overline{x_1}x_3$.

Таблиця 5.2 – Відповіді до тестових завдань 5.2

1.4	2.4	3.4	4.4	5.3	6.2	7.4	8.4	9.3	10.4

5.3 Основні поняття теорії інформації та кодування

1. Ентропія фізичної системи максимальна, коли стани системи розподілені за:

1 нормальним законом;
2 біноміальним законом;
3 законом Пуассона;
4 рівномірним законом.

2. Середня довжина кодової комбінації в оптимальних кодах обчислюється за формулою:

1. $l_{серед} = \sum_{i}^{n} l_i * \mathrm{p}_i$;
2. $l_{серед} = \sum_{i}^{n} l_i$;
3. $l_{серед} = \sum_{i}^{n} \mathrm{p}_i$;
4. $l_{серед} = \sum_{i}^{n} l_i + \mathrm{p}_i$.

3. Циклічний зсув вліво кодової комбінації циклічного коду $P_n(x) = x^7 + x^5 + x^4 + x^2 + x + 1$ буде таким:

1. 10010110;
2. 10101111;
3. 10010111;
4. 01101111.

4. Нехай отримана комбінація двійкового коду з перевіркою на парність, яка саме?

1. 10011010111;
2. 10001111000;
3. 11000001010;
4. 11110101111;

5. Нехай є числа 22 та 19. Кодовою відстанню між двійковими кодами, які зображають ці числа, є:

1. 2;
2. 3;
3. 1;
4. 4.

6. Циклічний зсув вправо кодової комбінації 100111001 циклічного коду буде таким:

1. 110011100;
2. 100101111;
3. 100101100;
4. 100101101.

7. Нехай є надлишковий код, який має дозволені кодові комбінації 1010, 1001, 0101; тоді мінімальна кодова відстань дорівнює:

1. 4;
2. 3;
3. 2;
4. 1.

8. Для виявлення *t*-кратних помилок надлишкового коду необхідно і достатньо, щоб виконувалось співвідношення:

1. $d_{\min} \geq 2t + 3$;
2. $d_{\min} \geq \log t + 1$;
3. $d_{\min} \geq t + 1$;
4. $d_{\min} \geq 2t + 1$.

9. Нехай є циклічний код. Многочлен $P_n(x) = x^7 + x^5 + x^4 + x^2 + x + 1$ відповідає кодовій комбінації:

1. 10010111;
2. 10011011;
3. 10010101;
4. 10110111.

10. Нехай є кодова комбінація 10000011, для цієї комбінації кодом з перевіркою на парність є:

1. 100000110;
2. 100000111;
3. 110011111;
4. 111100111.

Таблиця 5.3 – Відповіді до тестових завдань 5.3

1.4	2.1	3.4	4.3	5.1	6.1	7.3	8.3	9.4	10.2

СПИСОК ВИКОРИСТАНОЇ ТА РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Корченко А. Г. Построение систем защиты информации на нечетких множествах. Киев : «МК-Пресс», 2006. 320 с.
2. Ротштейн О. П. Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі. Вінниця : «УНІВЕРСУМ-Вінниця», 1999. 300 с.
3. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика : підручник. Львів : « Магнолія Плюс», 2005. 607 с.
4. Бондаренко М. Ф. Комп'ютерна дискретна математика : підручник. Харків : «Компанія СМІТ», 2004. 485 с.
5. Капітонова Ю. В. Основи дискретної математики : підручник. Київ : Наукова думка, 2002. 573 с.
6. Спекторський І. Я. Дискретна математика : навчальний посібник. Вид. 2-ге, К. : ІВЦ Видавництво Політехніка, 2004. 220 с.
7. Кондратенко Н. Р. Комп'ютерний практикум з дискретної математики. Вінниця : ВНТУ, 2010. 120 с.
8. Жураковський Ю. П., Полторак В. П. Теорія інформації та кодування : підручник. К. : Вища школа, 2001. 255 с.
9. Основи теорії інформації та кодування : підручник / за ред. І. В. Кузьміна. Хмельницький : ХНУ, 2009. 373 с.
10. Цымбал В. П. Теория информации и кодирования. К. : Вища школа, 1992. 264 с.
11. Кулик А. Я., Кривогубченко С. Г. Теорія інформації та кодування : навчальний посібник. Вінниця : ВНТУ, 2008. 145 с.
12. Бортник Г. Г., Кичак В. М. Основи теорії передачі інформації : навчальний посібник. Вінниця : ВДТУ, 2002. 128 с.
13. Лужецький В. А., Кожухівський А. Д., Войтович О. П. Основи інформаційної безпеки : навчальний посібник. Вінниця, ВНТУ, 2013. 221 с.
14. Вишня В. Б., Гавриш О. С., Рижков Е. В. Основи інформаційної безпеки : навчальний посібник. Дніпро : Дніпроп. держ. ун-т внутріш. справ, 2020. 128 с.

Електронне навчальне видання
комбінованого використання
Можна використовувати в локальному та мережному режимах

Кондратенко Наталія Романівна
Остапенко-Боженова Аліна Василівна

РОЗДІЛИ ДИСКРЕТНОЇ МАТЕМАТИКИ ДЛЯ ЗАДАЧ ЗАХИСТУ ІНФОРМАЦІЇ

Навчальний посібник

Рукопис оформлено: *Н. Р. Кондратенко*
А. В. Остапенко-Боженовою

Редактор *В. Дружиніна*

Оригінал-макет виготовлено *в РВВ ВНТУ*

Підписано до видання 28.12.2022
Гарнітура Times New Roman.
Зам. № Р2022-089

Видавець та виготовлювач -
Вінницький національний технічний університет,
Редакційно-видавничий відділ.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95, м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
Email: irvc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.