Розділ 6

Основи теорії кодування

- Алфавітне й рівномірне кодування
- Достатні умови однозначності декодування. Властивості роздільних кодів
- Оптимальне кодування
- Коди, стійкі до перешкод. Коди Хеммінга

Кодування (у широкому розумінні) – це перехід від одного способу подання інформації до іншого, який дає змогу відновлювати початкову інформацію. Теорія кодування виникла в 40-х роках XX століття після появи робіт К. Шеннона (C. Shannon). У цій теорії досліджуються методи кодування, пов'язані з основними математичними моделями, які відображають істотні риси реальних інформаційних систем.

6.1. Алфавітне й рівномірне кодування

Без утрати загальності можна сформулювати задачу кодування так. Нехай задано алфавіт $A=\{a_1, ..., a_r\}$ зі скінченної кількості символів, які називають *буквами*. Скінченну послідовність букв $a_{i_1}a_{i_2}...a_{i_l}$ алфавіту A називають *словом* у цьому алфавіті. Для позначення слів будемо використовувати малі грецькі літери: $\alpha = a_{i_1}a_{i_2}...a_{i_l}$. Число l – кількість букв у слові α – називають *довжиною* слова α та позначають $l(\alpha)$. Множину всіх слів у алфавіті A позначають як A^*. Порожнє слово позначають λ; зазначимо, що $\lambda \notin A$, $\lambda \in A^*$, $l(\lambda)=0$.

Якщо слово α має вигляд $\alpha=\alpha_1\alpha_2$, то α_1 називають *початком* (*префіксом*) слова α, а α_2 – його *закінченням* (*постфіксом*). Якщо при цьому $\alpha_1\neq\lambda$ (відповідно $\alpha_2\neq\lambda$), то α_1 (відповідно α_2) називають *власним початком* (відповідно *власним закінченням*) слова α.

Нехай S – підмножина множини A^*: $S\subset A^*$; елементи множини S називають *повідомленнями*. Нехай також задано скінченний алфавіт $B=\{b_1, ..., b_q\}$. Як β позначимо слово в алфавіті B, як B^* – множину всіх слів у алфавіті B. Задамо відображення F, яке кожному слову $\alpha\in S$ ставить у відповідність слово $\beta=F(\alpha)$, $\beta\in B^*$. Слово β називають *кодом повідомлення* α. Перехід від слова α до його коду називають *кодуванням*.

Якщо $|B| = q$, то кодування називають *q-ковим*. Найчастіше використовують алфавіт $B = \{0, 1\}$, тобто *двійкове кодування*. Саме його ми й розглядатимемо в наступних підрозділах, випускаючи слово „двійкове".

Відображення F задають якимось алгоритмом. Є два способи задати відображення F.

1. *Алфавітне кодування* задають схемою (або таблицею кодів) σ:

$$a_1 \to \beta_1,$$
$$a_2 \to \beta_2,$$
$$\dots \dots \dots$$
$$a_r \to \beta_r,$$

   де $a_i \in A$, $\beta_i \in B^*$, $i = 1, \dots, r$. Схема σ задає відповідність між буквами алфавіту A та деякими словами в алфавіті B. Вона визначає алфавітне кодування так: кожному слову $\alpha = a_{i_1} a_{i_2} \dots a_{i_l}$ із повідомленням S поставлено у відповідність слово $\beta = \beta_{i_1} \beta_{i_2} \dots \beta_{i_l}$ — його *код*. Слова $\beta_1, \dots, \beta_r$ називають *елементарними кодами*.

2. *Рівномірне кодування з параметрами* k й n задають так. Повідомлення α розбивають на блоки довжиною k:

$$\alpha = (x_1 \dots x_k)\ (x_{k+1} \dots x_{2k})\ \dots\ (x_{mk+1} \dots x_{mk+s}),$$

   де $s \leqslant k$ (останній блок може бути коротшим, у такому разі спосіб його кодування спеціально обумовлюють), $x_i \in A$, $i = 1, \dots, mk + s$. Блоки довжиною k розглядають як „букви" якогось алфавіту (таких блоків, очевидно, r^k, бо алфавіт A складається з r букв) і кодують словами в алфавіті B довжиною n за *схемою рівномірного кодування* $\sigma_{k,n}$:

$$\alpha_1 \to \beta_1,$$
$$\alpha_2 \to \beta_2,$$
$$\alpha_3 \to \beta_3,$$
$$\dots \dots \dots$$
$$\alpha_{r^k} \to \beta_{r^k}.$$

Надлишковістю схеми $\sigma_{k,n}$ *на символ повідомлення* називають величину $R = (n-k)/k = (n/k) - 1$.

Отже, ми описали дві можливості задати відображення F: $S \to B^*$. Однозначне декодування можливе, якщо існує обернене відображення F^{-1}.

6.2. Достатні умови однозначності декодування. Властивості роздільних кодів

Розглянемо схему алфавітного кодування σ та різні слова, складені з елементарних кодів. Схему σ називають *роздільною*, якщо з рівності $\beta_{i_1} \dots \beta_{i_k} = \beta_{j_1} \dots \beta_{j_l}$ випливає, що $k = l$ та $i_t = j_t$ для кожного $t = 1, \dots, k$, тобто будь-яке слово, складене з елементарних кодів, можна єдиним способом розкласти на елементарні коди. Очевидно, що алфавітне кодування з роздільною схемою вможливлює однозначне декодування [35, 49].

Схему σ називають *префіксною*, якщо для будь-яких i та j ($i, j = 1, ..., r$; $i \neq j$) елементарний код β_i не є префіксом елементарного коду β_j.

ТЕОРЕМА 6.1. Префіксна схема є роздільною.

Доведення. Доведемо цю теорему від протилежного. Оскільки схема σ префіксна, то всі елементарні коди в ній попарно різні, тобто $\beta_i \neq \beta_j$, якщо $i \neq j$. Нехай кодування зі схемою σ не роздільне. Тоді існує таке слово $\beta \in B^*$, що можуть бути два розклади на елементарні коди:

$$\beta = \beta_{i_1} ... \beta_{i_k} = \beta_{j_1} ... \beta_{j_l}.$$

Нехай $\beta_{i_1} = \beta_{j_1}$, ..., $\beta_{i_{p-1}} = \beta_{j_{p-1}}$ але $\beta_{i_p} \neq \beta_{j_p}$. У такому разі одне зі слів β_{i_p}, β_{j_p} являє собою префікс іншого, а це суперечить тому, що схема σ префіксна.

Властивість префіксності достатня, але не необхідна умова роздільності схеми.

Приклад 6.1. Нехай $A = \{a, b\}$, $B = \{0, 1\}$, σ: $a \to 0$, $b \to 01$. Схема σ не префіксна, але роздільна. Справді, перед кожним входженням 1 в слові β_2 безпосередньо є 0. Це дає змогу виділити всі пари (01). Частина слова, що залишилась, складається із символів 0.

Нехай $\beta = b_{i_1} b_{i_2} ... b_{i_n}$ — слово з B^*. Позначимо як $\tilde{\beta}$ слово, одержане оберненням слова β, тобто $\tilde{\beta} = b_{i_n} b_{i_{n-1}} ... b_{i_1}$. Позначимо як $\tilde{\sigma}$ схему

$$\begin{aligned} a_1 &\to \tilde{\beta}_1, \\ a_2 &\to \tilde{\beta}_2, \\ &\dots \dots \dots \\ a_r &\to \tilde{\beta}_r \end{aligned}$$

Приклад 6.2. Візьмемо схему σ з прикладу 6.1. Тоді $\tilde{\sigma}$ має такий вигляд: $a \to 0$, $b \to 10$. Схема $\tilde{\sigma}$ префіксна, тому за теоремою 6.1 вона роздільна.

Зауваження. Схеми σ та $\tilde{\sigma}$ водночас або роздільні, або ні.

Це зауваження дає змогу посилити теорему 6.1.

ТЕОРЕМА 6.2. Якщо або схема σ, або схема $\tilde{\sigma}$ префіксна, то обидві схеми σ та $\tilde{\sigma}$ роздільні.

Можна навести приклад такої роздільної схеми σ, що ні σ, ні $\tilde{\sigma}$ не префіксні. Отже, теорема 6.2 також дає достатню, але не необхідну умову роздільності схеми.

Нехай задано схему алфавітного кодування σ: $a_i \to \beta_i$, $a_i \in A$, $\beta_i \in B^*$, $i = 1, ..., r$. Для того, щоб схема алфавітного кодування була роздільною, необхідно, щоб довжини елементарних кодів задовольняли *нерівність Мак-Міллана* (B. McMillan).

ТЕОРЕМА 6.3 (нерівність Мак-Міллана). Якщо схема алфавітного кодування σ роздільна, то

$$\sum_{i=1}^{r}\frac{1}{2^{l_i}}\leqslant 1,\text{ де } l_i=l(\beta_i).$$

Доведення. Позначимо $L=\max(l_1, l_2, ..., l_r)$. Запишемо n-й степінь лівої частини нерівності:

$$\left(\sum_{i=1}^{r}2^{-l_i}\right)^n.$$

Розкривши дужки, отримаємо суму

$$\sum_{(i_1,...,i_n)}\left(2^{l_{i_1}+...+l_{i_n}}\right)^{-1},$$

де $(i_1, ..., i_n)$ — різні набори номерів елементарних кодів. Позначимо як $\nu(n, m)$ кількість доданків вигляду $1/2^m$, які входять у цю суму; тут $m=l_{i_1}+...+l_{i_n}$. Для деяких m може бути $\nu(n, m)=0$. Зведемо подібні члени й одержимо суму

$$\sum_{m=1}^{nL}\frac{\nu(n,m)}{2^m}.$$

Кожному доданку вигляду $(2^{l_{i_1}+...+l_{i_n}})^{-1}$ можна однозначно зіставити код (слово в алфавіті B) вигляду $\beta_{i_1}...\beta_{i_n}$. Воно складається з n елементарних кодів і має довжину m. Отже, $\nu(n, m)$ — це кількість деяких слів вигляду $\beta_{i_1}...\beta_{i_n}$ таких, що $l(\beta_{i_1}...\beta_{i_n})=m$. Позаяк схема σ роздільна, то $\nu(n, m)\leqslant 2^m$. Справді, коли припустити, що $\nu(n, m) > 2^m$, то за принципом коробок Діріхле існує два однакових слова $\beta_{i_1}...\beta_{i_n}=\beta_{j_1}...\beta_{j_n}$, які можна розкласти по-різному. Можна записати

$$\sum_{m=1}^{nL}\frac{\nu(n,m)}{2^m}\leqslant\sum_{m=1}^{nL}\frac{2^m}{2^m}=nL.$$

Отже, для кожного n виконується нерівність $\left(\sum_{i=1}^{r}2^{-l_i}\right)^n\leqslant nL$, тобто $\sum_{i=1}^{r}2^{-l_i}\leqslant\sqrt[n]{nL}$, звідки випливає

$$\sum_{i=1}^{r}2^{-l_i}\leqslant\lim_{n\to\infty}\sqrt[n]{nL}=1.$$

Зауваження. У теоремі 6.3 розглянуто нерівність Мак-Міллана для двійкового кодування. Для загального випадку q-кового кодування вона має вигляд

$$\sum_{i=1}^{r}\frac{1}{q^{l_i}}\leqslant 1,\text{ де } l_i=l(\beta_i).$$

Розглянемо ще один важливий факт.

ТЕОРЕМА 6.4. Якщо числа $l_1, ..., l_n$ задовольняють нерівність

$$\sum_{i=1}^{r} \frac{1}{2^{l_i}} \leqslant 1 \text{ (нерівність Мак-Міллана),}$$

то існує префіксна схема алфавітного кодування σ:

$$a_1 \to \beta_1,$$
$$\dots\dots\dots$$
$$a_r \to \beta_r,$$

де $l_1 = l(\beta_1), ..., l_r = l(\beta_r)$.

Доведення. Без утрати загальності можна вважати, що $l_1 \leqslant l_2 \leqslant ... \leqslant l_r$. Розіб'ємо множину $\{l_1, ..., l_r\}$ на класи еквівалентності так: l_i та l_j належать одному класу тоді й лише тоді, коли $l_i = l_j$. Нехай μ $(1 \leqslant \mu \leqslant r)$ – кількість класів, а $\lambda_1, ..., \lambda_\mu$ та $\nu_1, ..., \nu_\mu$ позначають відповідно представників та потужності класів. Можна також уважати, що $\lambda_1 < \lambda_2 < ... < \lambda_\mu$.

Нерівність Мак-Міллана можна переписати у вигляді

$$\sum_{i=1}^{\mu} \frac{\nu_i}{2^{\lambda_i}} \leqslant 1.$$

Вона породжує серію допоміжних нерівностей:

$$\frac{\nu_1}{2^{\lambda_1}} \leqslant 1, \text{ звідки випливає } \nu_1 \leqslant 2^{\lambda_1},$$

$$\frac{\nu_1}{2^{\lambda_1}} + \frac{\nu_2}{2^{\lambda_2}} \leqslant 1, \text{ звідки випливає } \nu_2 \leqslant 2^{\lambda_2} - \nu_1 2^{\lambda_2 - \lambda_1},$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

$$\frac{\nu_1}{2^{\lambda_1}} + \frac{\nu_2}{2^{\lambda_2}} + \cdots + \frac{\nu_\mu}{2^{\lambda_\mu}} \leqslant 1, \text{ звідки випливає}$$
$$\nu_\mu \leqslant 2^{\lambda_\mu} - \nu_1 2^{\lambda_\mu - \lambda_1} - \nu_2 2^{\lambda_\mu - \lambda_2} - ... - \nu_{\mu-1} 2^{\lambda_\mu - \lambda_{\mu-1}}.$$

Розглянемо слова довжиною λ_1 у двійковому алфавіті B. Оскільки $\nu_1 \leqslant 2^{\lambda_1}$, то можна вибрати з них ν_1 різних слів $\beta_1, ..., \beta_{\nu_1}$ довжиною λ_1. Вилучимо з подальшого розгляду всі слова з множини B^*, які починаються зі слів $\beta_1, ..., \beta_{\nu_1}$. Потім розглянемо множину слів у алфавіті B, що мають довжину λ_2 та не починаються зі слів $\beta_1, ..., \beta_{\nu_1}$. Таких слів, очевидно, $2^{\lambda_2} - \nu_1 2^{\lambda_2 - \lambda_1}$. Позаяк $v_2 \leqslant 2^{\lambda_2} - \nu_1 2^{\lambda_2 - \lambda_1}$, то з цієї множини можна вибрати ν_2 різних слів; позначимо їх як $\beta_{\nu_1+1}, ..., \beta_{\nu_1+\nu_2}$. Вилучимо з множини B^* слова, які починаються з $\beta_{\nu_1+1}, ..., \beta_{\nu_1+\nu_2}$, з подальшого розгляду. Продовжимо цей процес, поступово використовуючи допоміжні нерівності. Одержимо слова $\beta_1, ..., \beta_r$, де $r = \sum_{i=1}^{\mu} \nu_i$. Узявши їх як елементарні коди, отримаємо якусь схему алфавітного кодування σ. Вона, очевидно, префіксна. Крім того, $l(\beta_1) = l_1, ..., l(\beta_r) = l_r$.

Наслідок 1. Нерівність Мак-Міллана – необхідна й достатня умова існування алфавітного кодування з префіксною схемою та довжинами елементарних кодів, що дорівнюють $l_1, ..., l_r$.

Наслідок 2. Якщо існує роздільна схема алфавітного кодування із заданими довжинами елементарних кодів, то існує й префіксна схема з тими самими довжинами елементарних кодів.

Для доведення потрібно спочатку застосувати теорему 6.3, а потім – теорему 6.4.

6.3. Оптимальне кодування

Досі термін „код" було використано в загальноприйнятому розумінні (див. підрозділ 6.1). Однак у теорії кодування, а також у техніці слово *„код"* трактують також і як *множину елементарних кодів* [49]. Починаючи з цього місця, будемо використовувати слово „код" у двох значеннях. З контексту буде зрозуміло, яке саме з них ми маємо на увазі.

Нехай задано алфавіт $A = \{a_1, ..., a_r\}$ і ймовірності появи букв у повідомленні $P = (p_1, ..., p_r)$; тут p_i – імовірність появи букви a_i. Не втрачаючи загальності, можна вважати, що

$$p_1 \geqslant p_2 \geqslant ... \geqslant p_r > 0,$$

тобто можна одразу вилучити букви, які не можуть з'явитись у повідомленні, і впорядкувати букви за спаданням імовірностей їх появи. Крім того, $p_1 + p_2 + ... + p_r = 1$.

Для кожної роздільної схеми алфавітного кодування σ математичне сподівання коефіцієнта збільшення довжини повідомлення в разі кодування (позначають $l^{\sigma}_{\text{сер}}$) визначають так:

$$l^{\sigma}_{\text{сер}}(P) = \sum_{i=1}^{r} p_i l_i,$$

де $l_i = l(\beta_i)$, $i = 1, ..., r$, і називають *середньою довжиною* кодування за схемою σ для розподілу ймовірностей P.

Уведемо величину $l_* = \inf\limits_{\sigma} l^{\sigma}_{\text{сер}}$, де нижню грань беруть за всіма роздільними схемами σ. Легко довести, що

$$1 \leqslant l_* \leqslant \lceil \log_2 r \rceil,$$

де верхню оцінку дає схема з елементарними кодами з однаковою довжиною $\lceil \log_2 r \rceil$. Звідси випливає, що для побудови кодів, у яких величина $l_{\text{сер}}$ близька до l_*, можна не враховувати коди з більшим $l_{\text{сер}}$, ніж $\lceil \log_2 r \rceil$. Отже, будемо розглядати лише схеми з $p_i l_i \leqslant \lceil \log_2 r \rceil$. Позначимо $p_* = \min(p_1, ..., p_r)$, тоді

$$l_i \leqslant \lceil \log_2 r \rceil / p_*$$

для всіх $i = 1, 2, ..., r$.

Звідси випливає, що є лише скінченна кількість варіантів значень $l_{сер}$, для яких $l_* \leqslant l_{сер} \leqslant \lceil \log_2 r \rceil$. Отже, значення l_* досягається на якійсь схемі σ; його можна визначити як

$$l_* = \min_{\sigma} l_{сер}^{\sigma}.$$

Коди, визначені схемою σ з $l_{сер} = l_*$, називають *кодами з мінімальною надлишковістю* (або *оптимальними кодами*) для розподілу ймовірностей P. За наслідком 2 з теорем 6.3 та 6.4 існує алфавітне кодування з префіксною схемою, яке дає оптимальні коди. У зв'язку з цим, будуючи оптимальні коди, можна обмежитися префіксними схемами.

Проста ідея побудови коду, близького до оптимального, належить американському математикові Р. Фано (R. Fano). Сформулюємо алгоритм кодування за методом Фано.

1. Упорядковуємо букви алфавіту A за спаданням імовірностей їх появи в повідомленні.
2. Розбиваємо множину букв, записаних у зазначеному порядку, на дві послідовні (без перестановок букв) частини так, щоб сумарні ймовірності кожної з них були якомога близькими одна до одної. Кожній букві з першої частини приписуємо символ 0, другої — символ 1. Далі те саме робимо з кожною частиною, якщо вона містить принаймні дві букви. Процедуру продовжуємо доти, доки всю множину не буде розбито на окремі букви.

Приклад 6.3. Нехай задано розподіл імовірностей $P = (0.4, 0.15, 0.15, 0.15, 0.15)$. Побудуємо код за методом Фано. Розв'язок подано в табл. 6.1.

Таблиця 6.1

Буква алфавіту A	Імовірність появи букви	Розбиття множини букв			Елементарний код
a_1	0.4	0	0		00
a_2	0.15		1		01
a_3	0.15	1	0		10
a_4	0.15		1	0	110
a_5	0.15			1	111

Середня довжина коду дорівнює

$$l_{сер}^{F} = 0.4 \cdot 2 + 0.15 \cdot 2 + 0.15 \cdot 2 + 0.15 \cdot 3 + 0.15 \cdot 3 = 2.3;$$

тут верхній індекс F означає, що код отримано методом Фано.

Алгоритм Фано має просту інтерпретацію за допомогою бінарного дерева. Від кореня відходять два ребра, одне з яких позначено символом 0, друге — символом 1. Ці два ребра відповідають розбиттю множини всіх букв на дві майже рівноймовірні частини, одній з яких поставлено у відповідність символ 0, а другій — символ 1. Ребра, що виходять із вершин наступного рівня, відповідають розбиттю одержаних частин знову на дві майже рівноймовірні послідовні частини.

Цей процес продовжують доти, доки множину букв не буде розбито на окремі букви. Кожний листок дерева відповідає якомусь елементарному коду. Щоб виписати цей код, потрібно пройти шлях від кореня до відповідного листка.

Зазначимо, що незалежно від способу кодування кожному бінарному дереву відповідає набір двійкових елементарних кодів. У такому разі дерево називають *кодовим*. Якщо елементарні коди відповідають листкам кодового дерева, то відповідна схема алфавітного кодування є префіксною (отже, забезпечується однозначність декодування).

Приклад 6.4. На рис. 6.1. зображено колове дерево, отримане методом Фано для розподілу ймовірностей із прикладу 6.3.

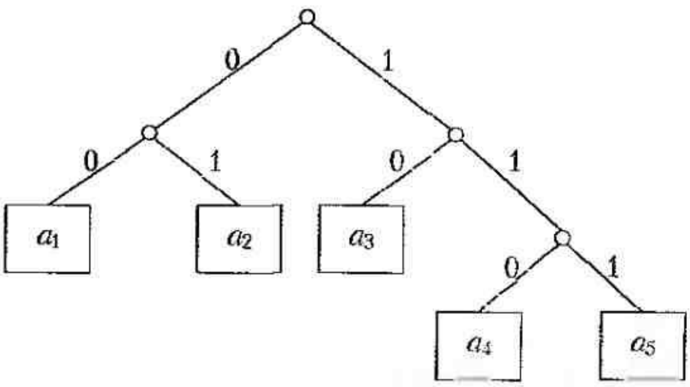

Рис. 6.1

Що ймовірніша поява букви, то швидше вона утворить „самостійну" підмножину, й, отже, її буде закодовано коротшим елементарним кодом. Тому метод Фано доволі ефективний. Але чи завжди він дає змогу отримати оптимальний код? Виявляється, що ні. Поданий тут спосіб побудови оптимального коду потребує тонших міркувань. Передусім сформулюємо й доведемо деякі допоміжні твердження.

ЛЕМА 6.1. В оптимальному коді букву з меншою ймовірністю її появи не може бути закодовано коротшим словом. Інакше кажучи, для оптимального коду з того, що $p_i < p_j$, випливає, що $l_i \geqslant l_j$.

Доведення. Припустимо протилежне: нехай є дві букви a_i й a_j такі, що $p_i < p_j$ і $l_i < l_j$. Поміняємо місцями β_i та β_j у схемі кодування. Тоді середня довжина елементарних кодів зміниться на

$$p_i l_j + p_j l_i - p_i l_i - p_j l_j = (p_i - p_j) l_j - (p_i - p_j) l_i = (p_i - p_j)(l_j - l_i) < 0,$$

тобто зменшиться, що суперечить оптимальності коду.

ЛЕМА 6.2. Якщо код оптимальний, то можна так перенумерувати букви алфавіту $A = \{a_1, ..., a_r\}$ і відповідні елементарні коди $\beta_1, \beta_2, ..., \beta_r$, що $p_1 \geqslant p_2 \geqslant ... \geqslant p_r$, і при цьому $l_1 \leqslant l_2 \leqslant ... \leqslant l_r$.

Доведення. Якщо $p_i > p_{i+1}$, то з леми 6.1 випливає, що $l_i \leqslant l_{i+1}$. Якщо ж $p_i = p_{i+1}$, але $l_i > l_{i+1}$, то поміняємо місцями нумерацію букв a_i, a_{i+1} і відповідних елементарних кодів. Повторюючи цю процедуру потрібну кількість разів, одержимо бажану нумерацію.

З нерівності $l_1 \leqslant l_2 \leqslant \ldots \leqslant l_r$ випливає, що букву a_r (найменш імовірну) закодовано словом β_r із найбільшою довжиною.

ЛЕМА 6.3. В оптимальному коді є два елементарні коди з найбільшою довжиною l_r, які відрізняються лише останніми символами.

Доведення. Припустимо, що це не так. Тоді можна відкинути останній символ елементарного коду β_r, не порушуючи властивості префіксності. При цьому, очевидно, зменшиться середня довжина елементарного коду. Це суперечить твердженню, що код оптимальний.

ТЕОРЕМА 6.5. Існує такий оптимальний код, у якому елементарні коди двох найменш імовірних букв a_{r-1} і a_r відрізняються лише останніми символами.

Доведення. За лемою 6.3 знайдеться елементарний код β_t, який має ту саму довжину, що й β_r, і відрізняється від нього лише останнім символом. Із лем 6.1 і 6.2 випливає, що $l_t = l_{t-1} = \ldots = l_r$. Якщо $t \neq r-1$, то можна поміняти місцями β_t та β_{r-1}, не порушуючи нерівності $l_1 \leqslant l_2 \leqslant \ldots \leqslant l_r$.

Теорема 6.5 дає змогу розглядати лише такі схеми алфавітного кодування, у яких елементарні коди β_{r-1} і β_r (для двох найменш імовірних букв a_{r-1} і a_r) мають найбільшу довжину й відрізняються тільки останніми символами. Це означає, що листки β_{r-1} і β_r кодового дерева оптимального коду мають бути з'єднані в одній вершині попереднього рівня (рис. 6.2).

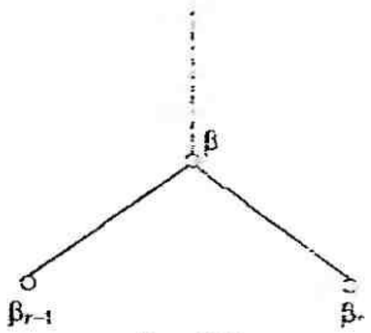

Рис. 6.2

Розглянемо новий алфавіт $A_1 = \{a_1, \ldots, a_{r-2}, a\}$ із розподілом імовірностей $P_1 = (p_1, \ldots, p_{r-2}, p)$, де $p = p_{r-1} + p_r$. Його одержано з алфавіту

$$A = \{a_1, \ldots, a_{r-2}, a_{r-1}, a_r\}$$

об'єднанням двох найменш імовірних букв a_{r-1} та a_r в одну букву a з імовірністю $p = p_{r-1} + p_r$. Говорять, що A_1 отримано *стисненням* алфавіту A.

Нехай для алфавіту A_1 побудовано схему σ_1 з елементарними кодами $\beta_1, \beta_2, \ldots, \beta_{r-2}, \beta$. Інакше кажучи, є якесь кодове дерево з листками $\beta_1, \beta_2, \ldots, \beta_{r-2}, \beta$. Схемі σ_1 можна поставити у відповідність схему σ з елементарними кодами $\beta_1, \beta_2, \ldots, \beta_{r-2}, \beta_{r-1}, \beta_r$ для початкового алфавіту A, узявши $\beta_{r-1} = \beta 0$, $\beta_r = \beta 1$ (тобто елементарні коди β_{r-1} та β_r одержують з елементарного коду β приписуванням справа відповідно 0 та 1).

Процедуру переходу від σ_1 до σ називають *розщепленням*.

ТЕОРЕМА 6.6. Якщо схема σ_1 оптимальна для алфавіту A_1, то схема σ оптимальна для алфавіту A.

Доведення. Легко переконатись, що середні довжини $l_{\text{сер}}^{\sigma}$ та $l_{\text{сер}}^{\sigma_1}$ елементарних кодів у схемах σ та σ_1 пов'язані співвідношенням $l_{\text{сер}}^{\sigma} = l_{\text{сер}}^{\sigma_1} + p$.

Припустимо, що схема σ не оптимальна для алфавіту A, тобто існує схема Σ, для якої $l_{\text{сер}}^{\Sigma} < l_{\text{сер}}^{\sigma}$. Нехай її елементарні коди — $\beta_1', \beta_2', ..., \beta_{r-2}', \beta_{r-1}', \beta_r'$. Можна вважати, що листки β_{r-1}' і β_r' кодового дерева схеми Σ відповідають двом найменш імовірним буквам алфавіту A й відрізняються лише останніми символами.

Розглянемо схему Σ_1 для алфавіту A_1. Елементарні коди схеми Σ_1 — $\beta_1', \beta_2', ..., \beta_{r-2}', \beta'$. Елементарний код β' одержують відкиданням останнього символу від β_r' (або від β_{r-1}', результат буде тим самим). Середні довжини кодів $l_{\text{сер}}^{\Sigma}$ й $l_{\text{сер}}^{\Sigma_1}$ пов'язані співвідношенням $l_{\text{сер}}^{\Sigma} = l_{\text{сер}}^{\Sigma_1} + p$.

Зі співвідношень $l_{\text{сер}}^{\sigma} = l_{\text{сер}}^{\sigma_1} + p$, $l_{\text{сер}}^{\Sigma} = l_{\text{сер}}^{\Sigma_1} + p$ та $l_{\text{сер}}^{\Sigma} < l_{\text{сер}}^{\sigma}$ випливає, що $l_{\text{сер}}^{\Sigma_1} < l_{\text{сер}}^{\sigma_1}$. Це суперечить оптимальності схеми σ_1 для алфавіту A_1.

Із теореми 6.6 випливає такий метод побудови оптимальної схеми алфавітного кодування. Спочатку послідовно стискають алфавіт A до отримання алфавіту з двох букв, оптимальна схема кодування для якого очевидна: першу букву кодують символом 0, другу — символом 1. Потім послідовно розщеплюють одержану схему. Очевидно, що отримана в результаті схема префіксна.

Описаний метод кодування запропоновано 1952 р. американським математиком Д. Хаффманом (D. Huffman).

Приклад 6.5. Нехай задано розподіл імовірностей P=(0.4, 0.15, 0.15, 0.15, 0.15), як і в *прикладі 6.3*. Побудуємо код методом Хаффмана. Розв'язок наведено в *табл. 6.2*.

Таблиця 6.2

Букви алфавіту A	***Імовірність та позначення елементарних кодів***							
	Початковий алфавіт *A*		**Стиснуті алфавіти**					
			A_1		A_2		A_3	
a_1	0.4	1	0.4	1	0.4	1	0.6	0
a_2	0.15	010	0.3	00	0.3	00	0.4	1
a_3	0.15	011	0.15	010	0.3	01		
a_4	0.15	000	0.15	011				
a_5	0.15	001						

Кожний з алфавітів A_1 – A_3 одержують стисненням попереднього алфавіту. Алфавіт A_3 складається з двох букв, тому оптимальна схема містить лише два елементарні коди — символи 0 і 1. Послідовне розщеплення дає оптимальну схему для початкового алфавіту A (у процесі розщеплення потрібно рухатися проти стрілок).

Середня довжина побудованого коду, яка дорівнює $l_{\text{сер}}^{H} = 0.4 \cdot 1 + 4 \cdot 0.15 \cdot 3 = 2.2$, як це випливає з попереднього, мінімально можлива для даного розподілу ймовірностей P. Тут індекс H означає, що код отримано методом Хаффмана.

У прикладі 6.3 одержано код методом Фано із середньою довжиною $l_{\text{сер}}^{F} = 2.3$. Отже, метод кодування Фано не завжди дає оптимальний код.

За допомогою алгоритму Хаффмана можна безпосередньо побудувати кодове дерево оптимального коду, яке називають ***деревом Хаффмана*** [50]. Бінарне дерево, що відповідає оптимальному коду, будують знизу вверх, починаючи з $|A| = r$ листків, і виконують $r-1$ злиття. Злиття полягає в побудові нового дерева з двох наявних дерев (або листків) із найменшими ймовірностями. При цьому листок або дерево з більшою ймовірністю утворює ліве піддерево; сума ймовірностей лівого та правого піддерев дорівнює ймовірності отриманого дерева (її записують у корінь). На ребро до лівого піддерева поміщають 0, до правого – 1. Розподіл імовірностей перед кожним злиттям упорядковують за спаданням.

Алгоритм Хаффмана належить до жадібних алгоритмів [23].

Приклад 6.6. Побудуємо дерево Хаффмана для розподілу ймовірностей $P=(0.34, 0.18. 0.17, 0.16, 0.15)$. Оскільки є всього п'ять букв, то для побудови дерева потрібно зробити чотири злиття. На кожному кроці зливають два піддерева з найменшими ймовірностями. Одержують нове дерево із сумарною ймовірністю лівого та правого піддерев і впорядковують розподіл імовірностей за спаданням. Листки зображають прямокутниками, у яких букви та їх імовірності відокремлюють двокрапкою (рис. 6.3). Внутрішні вершини зображають кружечками, у яких зазначають суми ймовірностей їхніх синів. Динаміку роботи алгоритму відображено на рис. 6.4.

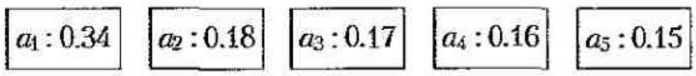

Рис. 6.3

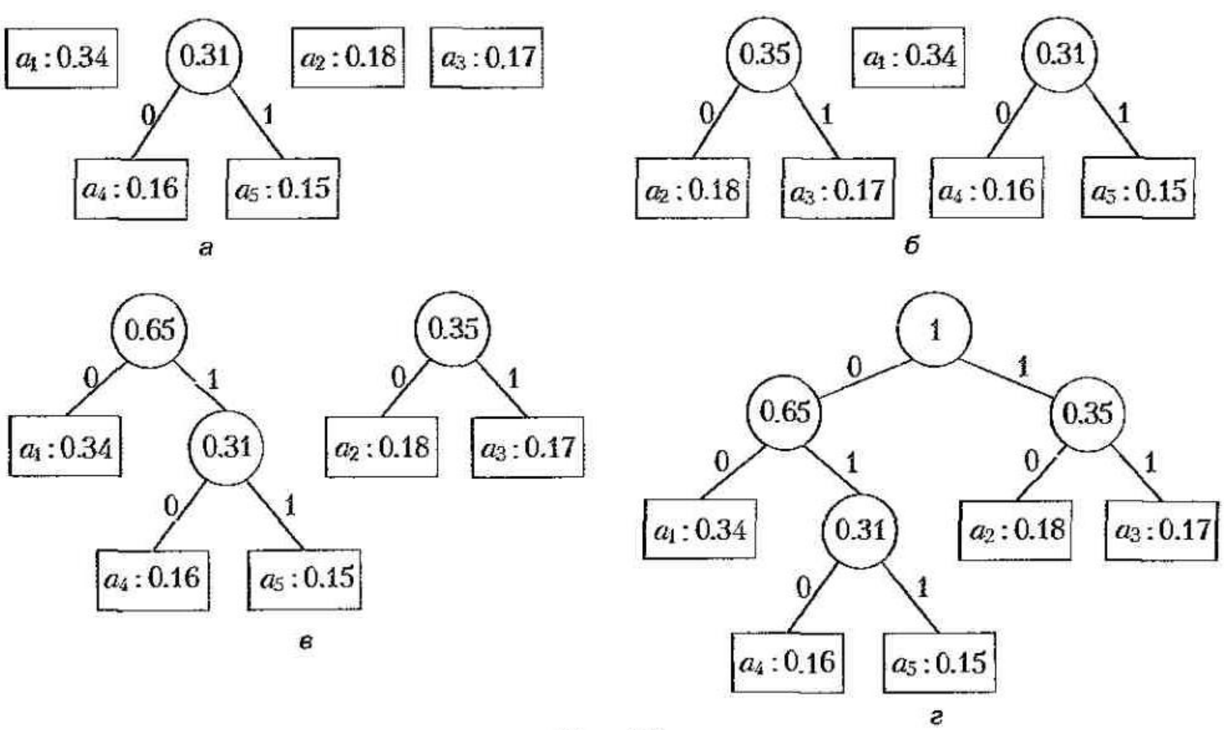

Рис. 6.4

Щоб отримати елементарний код для певної букви, потрібно пройти єдиний шлях від кореня до відповідного листка, записуючи послідовність бітів. Одержимо таку оптимальну схему алфавітного кодування за заданого розподілу ймовірностей: $a_1 \to 00$, $a_2 \to 10$, $a_3 \to 11$, $a_4 \to 010$, $a_5 \to 011$.

Узагалі кажучи, оптимальна схема не єдина. Так, у табл. 6.3 для розподілу ймовірностей $P=(0.4,\ 0.2,\ 0.2,\ 0.1,\ 0.1)$ наведено три різні оптимальні схеми ($l^1_{\text{сер}} = l^2_{\text{сер}} = l^3_{\text{сер}} = l_* = 2.2$).

Таблиця 6.3

Буква алфавіту	Імовірність	Схема 1	Схема 2	Схема 3
a_1	0.4	11	1	1
a_2	0.2	10	011	01
a_3	0.2	01	010	001
a_4	0.1	001	001	0001
a_5	0.1	000	000	0000

6.4. Коди, стійкі до перешкод. Коди Хеммінга

Розглянемо один частковий випадок рівномірного двійкового кодування, коли $A=B=\{0, 1\}$ [27]. Розглянемо схему рівномірного кодування $\sigma_{k,n}$ із параметрами k, n:

$$\begin{gathered}\alpha_1 \to \beta_1,\\ \alpha_2 \to \beta_2,\\ \alpha_3 \to \beta_3,\\ \dots\ \dots\ \dots\\ \alpha_{2^k} \to \beta_{2^k},\end{gathered}$$

де α_i, β_i — відповідно слова довжиною k та n, $n > k$. Говорять, що *схему* $\sigma_{k,n}$ *задає код* $V=\{\beta_1, \beta_2, \beta_3, \dots, \beta_{2^k}\} \subset E_2^n$ (див. підрозділ 6.3). Слова в алфавіті $\{0, 1\}$ – це впорядковані набори з нулів і одиниць (їх називають також двійковими векторами). Зазначимо, що кількість усіх слів α_i довжиною k в алфавіті $\{0, 1\}$ дорівнює 2^k.

Із методичних міркувань у цьому підрозділі нам зручно використовувати такі позначення. Елементи множини E_2^n (двійкові вектори довжиною n) позначатимемо великими латинськими буквами X, Y, Z, ..., а їх компоненти – відповідними малими буквами з індексами. Зокрема, елементарні коди позначатимемо як традиційно (β_1, β_2, β_3, ...), так і X, Y, Z, ..., залежно від контексту.

Нормою $\|X\|$ двійкового вектора $X=x_1x_2\dots x_n$ називають число, яке дорівнює кількості його одиничних компонент. Отже,

$$\|X\| = \sum_{i=1}^{n} x_i.$$

Нагадаємо, що операцію $\oplus$ – „альтернативне або" над двійковими розрядами (бітами) – означують так: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$ (див. підрозділ 1.1). Цю операцію називають також *додаванням за модулем 2* (mod 2). Якщо X та Y – двійкові вектори, то $X \oplus Y$ – *порозрядне* додавання за модулем 2.

Припустимо, що в каналі зв'язку діє *джерело адитивних перешкод*, яке описують множиною $P(n, t)$. Її елементи – двійкові вектори-помилки $x_1x_2\dots x_s$, у яких норма будь-якого фрагмента $x_i x_{i+1} \dots x_{i+l-1}$ не більша ніж t, якщо довжина фрагмента $l \leqslant n$, (тобто на n переданих поспіль двійкових символів припадає не більше ніж t помилок). Це означає, що коли на вході каналу зв'язку передано повідомлення α, то на виході можна отримати будь-яке слово з множини $\{\alpha \oplus \gamma \mid \gamma \in P(n, t), l(\alpha) = l(\gamma)\}$, де $\alpha \oplus \gamma$ – порозрядне додавання за mod 2.

Позаяк проблема локалізації інформації (тобто розділення закодованого повідомлення на елементарні коди) у моделі рівномірного кодування тривіальна, то виявлення помилок полягає у відшуканні незбігу локалізованої групи n символів ні з яким елементарним кодом. Якщо через помилку елементарний код перейде в інший елементарний код, то помилку не буде виявлено. Іноді можна виправити помилку. Якщо групу бітів локалізовано правильно, то для цього необхідно й достатньо, щоб помилкова група була „синонімом" єдиного елементарного коду.

Канал зв'язку називають *надійним*, якщо будь-які помилки можна виявити чи виправити відповідно до заданої мети декодування. Далі наведено головні положення побудови кодів, які забезпечують надійність найпростіших каналів зв'язку.

Віддаллю Хеммінга називають функцію $\rho(X,Y)$ двох змінних, означену на множині E_2^n:

$$\rho(X, Y) = \sum_{i=1}^{n} (x_i \oplus y_i)$$

(вона дорівнює кількості розрядів, у яких вектори X та Y не збігаються).

Скалярний добуток векторів $X, Y \in E_2^n$ означають так:

$$\langle X, Y \rangle = \sum_{i=1}^{n} x_i y_i;$$

він дорівнює кількості розрядів, у яких X та Y збігаються й дорівнюють 1.

Легко перевірити такі співвідношення:

$$\rho(X, 0) = \|X\| = \sum_{i=1}^{n} x_i, \tag{6.1}$$

де 0 – n-вимірний вектор із нульовими компонентами;

$$\rho(X, Y) = \|X \oplus Y\|, \tag{6.2}$$

де $X \oplus Y$ – порозрядне додавання за mod 2;

$$\rho(X \oplus Z, Y \oplus Z) = \rho(X, Y), \tag{6.3}$$

$$\rho(X, Y) = \|X\| + \|Y\| - 2\langle X, Y \rangle. \tag{6.4}$$

Для віддалі Хеммінга виконуються *аксіоми метрики*:

- $\rho(X, Y) \geqslant 0$, причому $\rho(X, Y) = 0$ в тому й лише в тому разі, якщо $X = Y$;
- $\rho(X, Y) = \rho(Y, X)$;
- $\rho(X, Y) + \rho(Y, Z) \geqslant \rho(X, Z)$ (нерівність трикутника).

Метрика Хеммінга – зручне математичне поняття для формулювання умов надійності кодування в разі адитивних помилок. Нехай схему $\sigma_{k,n}$ задано кодом $V = \{\beta_1, \beta_2, \beta_3, ..., \beta_{2^k}\}$. *Кодовою віддалю* для коду V називають величину

$$\rho(V) = \min\{\rho(X, Y) \mid X, Y \in V, X \neq Y\}.$$

ТЕОРЕМА 6.7. Якщо в каналі зв'язку діє джерело адитивних перешкод $P(n, t)$, то правдиві такі твердження.

1. Для виявлення будь-яких помилок необхідно й достатньо, щоб $\rho(V) > t$.
2. Для виправлення будь-яких помилок необхідно й достатньо, щоб $\rho(V) > 2t$.

Зауваження. Інакше кажучи, код може виявляти будь-які комбінації з t й меншої кількості помилок тоді й лише тоді, коли його кодова віддаль більша ніж t; код може виправляти будь-які комбінації з t й меншої кількості помилок тоді й лише тоді, коли його кодова віддаль більша ніж $2t$.

Доведення. 1. Нехай $\rho(V) > t$. Якщо $X \in V$, $Y \in P(n, t)$, $Y \neq 0$, то, використовуючи спочатку рівність (6.3), а потім – (6.1), можемо записати $\rho(X, X \oplus Y) = \rho(0, Y) = \|Y\| \leqslant t$. Отже, $X \oplus Y \notin V$, і помилку виявлено.

Навпаки, нехай $\rho(X, Y) \leqslant t$ й $X, Y \in V$, $X \neq Y$. Тоді, застосувавши рівність (6.2), маємо $\|X \oplus Y\| = \rho(X, Y) \leqslant t$; отже, $Z = X \oplus Y \in P(n, t)$. Звідси випливає, що $X \oplus Z = Y$, тобто помилку в елементарному коді Y виявити не можна.

2. Нехай $\rho(V) > 2t$. Якщо $X \in V$, $Z \in P(n, t)$, то X – єдиний елементарний код із V, який міг перейти внаслідок помилки в $X \oplus Z$. Справді, припустимо, що існує такий двійковий вектор $Y \neq X$, що $Y \in V$ та $Y \oplus Z_1 = X \oplus Z$ для якогось $Z_1 \in P(n, t)$. Додавши до обох частин останньої рівності $X \oplus Z_1$, отримаємо $X \oplus Y = Z_1 \oplus Z$. Але, згідно з рівністю (6.2), можемо записати $\|X \oplus Y\| = \rho(X, Y) > 2t$, а $\|Z_1 \oplus Z\| \leqslant \|Z_1\| + \|Z\| \leqslant 2t$. Одержали суперечність.

Навпаки, нехай $\rho(X, Y) \leqslant 2t$ для якихось різних $X, Y \in V$. Тоді $\|X \oplus Y\| \leqslant 2t$, й існують такі двійкові вектори Z_1, Z_2, що $\|Z_1\| \leqslant t$, $\|Z_2\| \leqslant t$ (тобто вони належать $P(n, t)$) й $X \oplus Y = Z_1 \oplus Z_2$. Додавши до обох частин останньої рівності $Y \oplus Z_1$, одержимо $X \oplus Z_1 = Y \oplus Z_2 = W$. Отже, у разі отримання спотвореного елементарного коду W неможливо виявити, що було передано насправді: X чи Y.

Доведене твердження має геометричну інтерпретацію.

Множину $S_t(X) = \{Z \mid \rho(X, Z) \leqslant t\}$ називають *кулею радіусом t з центром у точці X*.

ТЕОРЕМА 6.8. Якщо в каналі зв'язку діє джерело адитивних перешкод $P(n, t)$, то правдиві такі твердження.

1. Для виявлення будь-яких помилок необхідно й достатньо, щоб для будь-якого $X \in V$ куля $S_t(X)$ не містила інших елементарних кодів, окрім X.
2. Для виправлення будь-яких помилок необхідно й достатньо, щоб для будь-яких $X, Y \in V$ було виконано умову $S_t(X) \cap S_t(Y) = \varnothing$.

Рівномірне кодування $\sigma_{k,n}$: $\alpha_i \to \beta_i$ $(i = 1, 2, 3, ..., 2^k)$ називають *систематичним*, якщо можна виділити множину k розрядів $I = \{i_1, ..., i_k\} \subset \{1, 2, ..., n\}$, які називають *інформаційними*, так, що коли $\beta_i = x_1 ... x_n$ $(i = 1, 2, 3, ..., 2^k)$, то $\alpha_i = x_{i_1} ... x_{i_k}$. Решту розрядів у такому разі називають *контрольними*.

Рівномірне кодування $\sigma_{k,n}$: $\alpha_i \to \beta_i$ $(i = 1, 2, 3, ..., 2^k)$ називають *лінійним*, або *груповим*, якщо код $V = \{\beta_1, \beta_2, \beta_3, ..., \beta_{2^k}\}$ утворює підгрупу E_2^n щодо операції $\oplus$ порозрядного додавання векторів за модулем 2. У такому разі код V водночас являє собою лінійний підпростір простору E_2^n над полем E_2. Усі лінійні коди систематичні.

Лінійні коди можна задавати простіше, ніж коди загального типу. Достатньо зазначити твірні групи V. Їх можна подати матрицею з k рядками та n стовпцями $G(V)$ — базисом векторного простору V; $G(V)$ називають *породжувальною матрицею коду V*. Базис можна вибрати не єдиним способом, тому матрицю $G(V)$ означено неоднозначно. За допомогою породжувальної матриці $G = G(V)$ можна кодувати повідомлення. Якщо $\alpha_i = x_{i_1} ... x_{i_k}$ — повідомлення, то його код $\beta_i = \alpha_i G$.

Двоїстість, яка пов'язує ортогональні підпростори, дає ще один спосіб подання лінійних кодів. Вектори $X, Y \in E_2^n$ називають *ортогональними*, якщо $\langle X, Y\rangle = 0 (\mathrm{mod}\ 2)$. Ортогональний підпростір $V^{\perp} = \{X \mid \langle X, V\rangle = 0 (\mathrm{mod}\ 2)\}$ для лінійного коду V називають *двоїстим кодом* для V. Його розмірність дорівнює $n - k$, і код $V^{\perp}$ задає *двоїсту схему* алфавітного кодування $\sigma_{n-k,n}$. Матрицю $H(V) = G(V^{\perp})$ — породжувальну матрицю коду $V^{\perp}$ — називають *перевірною матрицею* коду V. Отже, маємо $V = \{X \mid \langle X, H\rangle = 0 (\mathrm{mod}\ 2)\}$ — *двоїстий спосіб подання лінійного коду V перевірною матрицею* $H = H(V)$. Легко перевірити, що коли $G(V) = [J_k A]$, де J_k — одинична $k \times k$ матриця, то $H(V) = [A^{\mathrm{T}} J_{n-k}]$, де A^{T} — транспонована матриця. Звідси випливає, що коли I — множина інформаційних розрядів для коду V, то як множину інформаційних розрядів для двоїстого коду $V^{\perp}$ можна взяти $\bar{I} = \{1, 2, ..., n\} \setminus I$.

Приклад 6.7. Для лінійного коду $V = \{000, 011, 101, 110\}$, який задає схему лінійного рівномірного кодування $\sigma_{2,3}$

$$00 \to 000,\ 01 \to 011,\ 10 \to 101,\ 11 \to 110$$

з інформаційними розрядами $I = \{1, 2\}$, маємо

$$G(V) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix},\ H(V) = [1 \ \ 1 \ \ 1],\ V^{\perp} = \{000, 111\}$$

і двоїсту схему $\sigma_{1,3}$:

$$0 \to 000,\ 1 \to 111.$$

ТЕОРЕМА 6.9. Для кодової віддалі лінійного коду виконується рівність

$$\rho(V) = \min\{\|X\| \mid X \in V, X \neq 0\}. \tag{6.5}$$

Доведення. Зазначимо, що нульовий вектор 0 міститься в будь-якому лінійному коді. Рівність (6.5) випливає з того, що

$$\min\{\rho(X,Y) \mid X, Y \in V, X \neq Y\} = \min\{\rho(0, X \oplus Y) \mid X \oplus Y \in V, X \oplus Y \neq 0\}.$$

Р. Хеммінг (R. Hamming) 1950 р. запропонував коди для виявлення та виправлення помилок у разі $t = 1$. Коди Хеммінга лінійні; вони мають найменшу надлишковість, можливу для даного k.

Коди Хеммінга $H_{\text{вияв}}(n)$ для виявлення помилок у каналі зв'язку із джерелом перешкод $P(n,1)$ означено для будь-якого n:

$$H_{\text{вияв}}(n) = \{X \mid X \in E_2^n,\ \|X\| = 0(\bmod 2)\}.$$

Код $H_{\text{вияв}}(n)$ лінійний. Справді, якщо X, $Y \in H_{\text{вияв}}(n)$, то з урахуванням рівностей (6.2) та (6.4) можемо записати

$$\|X \oplus Y\| = \|X\| + \|Y\| - 2\langle X, Y\rangle = -2\langle X, Y\rangle = 0(\bmod 2),$$

тобто $X \oplus Y \in H_{\text{вияв}}(n)$. Згідно зі співвідношенням (6.5), $\rho(H_{\text{вияв}}(n)) = 2$; отже, можна виявити будь-яку помилку в каналі з джерелом перешкод $P(n,1)$. Для коду $H_{\text{вияв}}(n)$ як інформаційні можна взяти будь-які $n-1$ розрядів, бо значення довільного розряду слова $x_1x_2\ldots x_n \in H_{\text{вияв}}(n)$ можна однозначно знайти за значеннями інших розрядів із рівняння $x_1 \oplus x_2 \oplus \ldots \oplus x_n = 0$. Маємо

$$H(H_{\text{вияв}}(n)) = [1 \quad 1 \quad \ldots \quad 1], \quad H_{\text{вияв}}^{\perp}(n) = \{00\ldots0,\ 11\ldots1\}.$$

Як одну з породжувальних матриць можна взяти

$$G(H_{\text{вияв}}(n)) = \begin{bmatrix} 1 & 0 & \ldots & 0 & 1 \\ 0 & 1 & \ldots & 0 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & 1 \end{bmatrix}.$$

Вибравши $I = \{1, 2, \ldots, n-1\}$, отримаємо відповідну схему рівномірного кодування $\sigma_{n-1,n}$:

$$x_1x_2\ldots x_{n-1} \to x_1x_2\ldots x_{n-1}x_n,$$

де $x_n = x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1}$. Надлишковість у разі використання коду $H_{\text{вияв}}(n)$ становить $R = (n-1)^{-1}$.

У цьому коді Хеммінга втілено ідею перевірки на парність.

Коди Хеммінга для виправлення помилок у каналі зв'язку із джерелом перешкод $P(n,1)$ будують для значень $n = 2^s - 1$ ($s = 2, 3, \ldots$). Код $H_{\text{вип}}(n)$ зручно задавати перевірною матрицею, яка має s рядків і $2^s - 1$ стовпців. Тут стовпці — усі можливі ненульові двійкові набори довжиною s. Їх зручно розміщувати так, щоб i-й зліва стовпець h_i був двійковим розкладом числа i (старші розряди — зверху):

$$H(H_{\text{вип}}(n)) = \begin{bmatrix} 0 & 0 & 0 & \ldots & 1 & 1 \\ 0 & 0 & 0 & \ldots & 1 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 1 & 1 & \ldots & 1 & 1 \\ 1 & 0 & 1 & \ldots & 0 & 1 \end{bmatrix}.$$

Таке розміщення стовпців перевірної матриці зумовлене вибором як контрольних розрядів тих, у яких номери являють собою степені двійки: $T=\{1, 2, 4, 8, 16, ..., 2^{s-1}\}$.

Приклад 6.8. Для значень $n=3$, $n=7$ маємо такі перевірні матриці коду $H_{\text{вип}}(n)$:

$$H(H_{\text{вип}}(3)) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix};$$

$$H(H_{\text{вип}}(7)) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

За допомогою перевірної матриці $H=H(H_{\text{вип}}(n))$ код Хеммінга $H_{\text{вип}}(n)$ можна задти так:

$$H_{\text{вип}}(n) = \{X \mid \langle X, H \rangle = 0 (\text{mod} 2), X \in E_2^n\} =$$
$$= \left\{ X \;\middle|\; x_1 \begin{pmatrix} h_{11} \\ h_{21} \\ \dots \\ h_{s1} \end{pmatrix} \oplus x_2 \begin{pmatrix} h_{12} \\ h_{22} \\ \dots \\ h_{s2} \end{pmatrix} \oplus \dots \oplus x_n \begin{pmatrix} h_{1n} \\ h_{2n} \\ \dots \\ h_{sn} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix} \right\}. \tag{6.6}$$

Код Хеммінга $H_{\text{вип}}(n)$ у разі $n=7$ називають (7, 4)-кодом Хеммінга, а в разі $n= 15$ – (15, 11)-кодом Хеммінга. Тут перше число в круглих дужках дорівнює довжині елементарного коду, а друге – кількості інформаційних розрядів.

Приклад 6.9. Закодуємо повідомлення *1001* за допомогою (7, 4)-коду Хеммінга. Запишемо макет коду, беручи до уваги розміщення контрольних розрядів: $x_1x_21x_4001$. Для відшукання значень контрольних розрядів використаємо умову (6.6) (доданки з нульовими коефіцієнтами випущено):

$$x_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \oplus x_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \oplus x_4 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

звідки $x_4 \oplus 1=0$, $x_2 \oplus 1 \oplus 1=0$, $x_1 \oplus 1 \oplus 1=0$. Отже, $x_1=0$, $x_2=0$, $x_4=1$, й отримаємо такий код заданого повідомлення: 0011001.

Доведемо, що $\rho(H_{\text{вип}}(n)) \geqslant 3$, тобто код $H_{\text{вип}}(n)$ забезпечує корекцію будь-яких помилок у каналі зв'язку із джерелом перешкод $P(n,1)$. Справді, якщо $X \neq 0$ та $X \in H_{\text{вип}}(n)$, то $\|X\| \neq 1$, оскільки всі стовпці перевірної матриці ненульові. Далі, $=\|X\| \neq 2$, бо якщо X має одиниці *тільки* у двох розрядах, скажімо $x_i=x_j=1$, $i \neq j$, то $h_i \oplus h_j = 0$ (тут h_i й h_j – стовпці перевірної матриці H), що рівносильно рівності $h_i=h_j$, а всі стовпці перевірної матриці різні. Отже, найменша вага ненульового вектора з $H_{\text{вип}}(n)$ не менша 3 і, відповідно до співвідношення (6.5), $\rho(H_{\text{вип}}(n)) \geqslant 3$.

Нехай в елементарному коді $X \in H_{\text{вип}}(n)$ виникла помилка в i-му розряді: код $X = x_1x_2 \dots x_i \dots x_n$ перетворився на $X' = X \oplus e_i = x_1x_2 \dots x_i \oplus 1 \dots x_n$ (тут e_i — вектор, i-та компонента якого дорівнює 1, а решта компонент — 0). Тоді

$$\langle X', H \rangle = \langle X \oplus e_i, H \rangle = \langle X, H \rangle \oplus h_i = h_i \pmod 2,$$

бо $\langle X, H \rangle = 0 \pmod 2$. Звідси випливає, що коли локалізовано послідовність символів $X \in E_2^n$, то достатньо визначити вектор $\langle X, H \rangle \pmod 2$. Якщо він нульовий, то помилки немає, а ні, то цей вектор являє собою двійковий розклад номера розряду, у якому виникла помилка.

Приклад 6.10. Нехай під час декодування повідомлення в (7, 4)-коді Хеммінга локалізовано послідовність $X = 0110010$. Знаходимо

$$\langle X, H \rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Доходимо висновку, що виникла помилка в сьомому розряді елементарного коду; виправляємо її (інвертуємо помилковий розряд): 0110011. Із відкоректованого коду виділяємо інформаційну групу розрядів (третій, п'ятий, шостий і сьомий). У результаті отримуємо 1011.

Надлишковість у разі використання коду $H_{\text{вип}}(n)$ зменшується зі зростанням n:

$$R = \frac{n}{k} - 1 = \frac{2^s - 1}{2^s - s - 1} - 1 = \frac{s}{2^s - s - 1}.$$

Якщо $n \neq 2^s - 1$, то будують *укорочений код Хеммінга.* Його задають перевірною матрицею, утвореною першими n стовпцями перевірної матриці $H_{\text{вип}}(n_1)$, де n_1 — найменше ціле із чисел, які більші ніж n і мають вигляд $2^s - 1$. Очевидно, що всі попередні міркування правильні й для вкороченого коду Хеммінга.

Контрольні запитання та завдання

1. Нехай числа 1, 2, 4, 17, 98 закодовано їх двійковими розкладами з мінімально можливою довжиною. Наприклад, код одиниці — 1, двійки — 10, четвірки — 100. Чи це кодування роздільне?
2. Задано коди. Для кожного роздільного коду V побудувати префіксний код із тим самим набором довжин елементарних кодів:

   а) $V = \{01, 10, 100, 111, 011\}$;

   б) $V = \{1, 10, 00, 0100\}$;

   в) $V = \{10, 101, 111, 1011\}$.
3. Для заданих розподілів імовірностей появи букв побудувати коди методом Фано:

   а) $P = \{0.6, 0.1, 0.09, 0.08, 0.07, 0.06\}$;

   б) $P = \{0.4, 0.4, 0.1, 0.03, 0.03, 0.02, 0.02\}$;

в) $P=\{0.3, 0.2, 0.2, 0.1, 0.1, 0.05, 0.05\}$;

г) $P=\{0.25, 0.2, 0.15, 0.15, 0.15, 0.1\}$;

д) $P=\{0.4, 0.18, 0.1, 0.1, 0.07, 0.06, 0.05, 0.04\}$;

е) $P=\{0.2, 0.2, 0.19, 0.12, 0.11, 0.09, 0.09\}$.

Для кожного розподілу визначити $l_{\text{сер}}^F$.

4. Для розподілів імовірностей задачі 3 побудувати оптимальні коди методом Хаффмана. Для кожного розподілу визначити $l_{\text{сер}}^H$ і порівняти з відповідним значенням $l_{\text{сер}}^F$. Указати розподіли, для яких метод Фано не дає оптимального коду.
5. Довести, що якщо ймовірності появи букв являють собою степені двійки $p_i = 2^{-n_i}$, то довжини відповідних елементарних кодів, одержаних за методами Фано та Хаффмана, збігаються й дорівнюють n_i.
6. За допомогою алгоритму Хаффмана стиснути наведений нижче текст. Визначити коефіцієнт стиснення (уважати, що в нестисненому тексті кожний символ закодовано одним байтом):

   а) THIS IS A SIMPLE EXAMPLE OF HUFFMAN ENCODING;

   б) МІНІМІЗАЦІЯ ЧАСУ ВИКОНАННЯ ПРОГРАМИ.
7. Для коду V з'ясувати, чи він лінійний, і знайти кодову віддаль $\rho(V)$:

   а) $V=\{000000, 011100, 100111, 111011\}$;

   б) $V=\{00001, 01110, 10010, 11101\}$;

   в) $V=\{00001, 01111, 10010, 11100\}$;

   г) $V=\{00011, 00001, 10110, 11110\}$.
8. Нехай G – породжувальна матриця лінійного коду. Знайти перевірну матрицю та двоїстий код:

   а) $G=\begin{bmatrix}1 & 0 & 1 & 0 & 1\\ 0 & 1 & 0 & 0 & 1\\ 1 & 0 & 0 & 1 & 0\end{bmatrix}$;

   б) $G=\begin{bmatrix}0 & 0 & 1 & 0 & 1 & 1\\ 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 1 & 0 & 1\end{bmatrix}$;

   в) $G=\begin{bmatrix}0 & 1 & 1 & 0 & 1 & 1 & 0 & 1\\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0\end{bmatrix}$.
9. За допомогою коду Хеммінга $H_{\text{вияв}}(n)$ для значення параметра $n=8$ закодувати повідомлення 0110111100110101111101.
10. Повідомлення закодовано за допомогою коду Хеммінга $H_{\text{вияв}}(8)$. На виході каналу зв'язку із джерелом адитивних перешкод $P(8, 1)$ отримано код 011010011001010011101110. Чи можна твердити, що під час його передавання виникла помилка? Якщо так, то в якій групі цифр?
11. Для повідомлень 1101 і 1011 побудувати (7, 4)-код Хеммінга, використовуючи перевірну матрицю H.

12. За допомогою (7, 4)-коду Хеммінга закодувати такі повідомлення:

    а) 110010111011; б) 100111010111.

    Для кодування використати перевірну матрицю H.
13. На виході каналу зв'язку із джерелом адитивних перешкод $P(7, 1)$ отримано такі комбінації в (7, 4)-коді Хеммінга:

    а) 1001001; б) 0110001; в) 0011111; г) 0110100.

    Застосувати корекцію кодів і декодувати ці повідомлення.
14. У каналі зв'язку із джерелом перешкод $P(7, 1)$ використано (7, 4)-код Хеммінга. Застосувати корекцію й декодувати повідомлення 1110010110011110001110010001.
15. Знайти породжувальну матрицю G для (7, 4)-коду Хеммінга.
16. Для повідомлень 1101 і 1011 побудувати (7, 4)-код Хеммінга, використовуючи породжувальну матрицю G (див. задачу 15).
17. За допомогою (7, 4)-коду Хеммінга закодувати такі повідомлення:

    а) 110010111011; б) 100111010111.

    Для кодування використати породжувальну матрицю G (див. задачу 15).
18. Побудувати код Хеммінга для виправлення помилки в одному розряді та виявлення помилки у двох розрядах у разі передавання 4-розрядної двійкової комбінації. Навести перевірну матрицю цього коду.

    Вказівка. Використати (7, 4)-код Хеммінга, додавши перевірку на парність (розширений (8, 4)-код Хеммінга).
19. Під час передавання за розширеним (8, 4)-кодом Хеммінга (див. задачу 18) отримано такі повідомлення:

    а) 00100001; б) 00110001.

    Що можна твердити у наведених випадках?
20. Записати перевірну матрицю (15, 11)-коду Хеммінга. За її допомогою закодувати повідомлення 10110110110. Показати процес виявлення помилки в п'ятому розряді коду, отриманого на виході каналу зв'язку.

Комп'ютерні проекти

Скласти програми із зазначеними вхідними даними та результатами.

1. Задано повідомлення. Закодувати його методом Фано.
2. Задано повідомлення. Закодувати його методом Хаффмана.
3. Задано схему алфавітного кодування й код повідомлення. Декодувати це повідомлення.
4. Задано повідомлення. Закодувати його за допомогою (7, 4)-коду Хеммінга.
5. У каналі зв'язку із джерелом перешкод $P(7, 1)$ використано (7, 4)-код Хеммінга. Задано код повідомлення (можливо, з помилками, зумовленими властивостями каналу зв'язку). Декодувати це повідомлення.

Розділ 7
Булеві функції

- Означення булевої функції. Реалізація функцій формулами
- Алгебри булевих функцій
- Спеціальні форми подання булевих функцій
- Повнота й замкненість
- Мінімізація булевих функцій
- Реалізація булевих функцій схемами з функціональних елементів

У цьому розділі розглянуто теорію булевих функцій. Апарат булевих функцій широко застосовують у математичній і технічній кібернетиці, зокрема для проектування мікропроцесорів, побудови логічних форм функцій вибору в теорії вибору та прийняття рішень тощо.

7.1. Означення булевої функції. Реалізація функцій формулами

Булевою називають функцію $f(x_1, ..., x_n)$ з областю значень $\{0, 1\}$, змінні $x_1, ..., x_n$ якої також набувають лише цих двох значень. Множину всіх булевих функцій позначають P_2, множину всіх булевих функцій від n змінних $- P_2(n)$.

Булеву функцію від n змінних називають *n-місною*. Область її визначення — множина E_2^n усіх можливих двійкових наборів довжиною n. Отже, область визначення n-місної булевої функції скінченна й складається з 2^n наборів. Для набору $(a_1, ..., a_n)$ у цьому розділі будемо використовувати позначення $\tilde{a}^n$ або $\tilde{a}$ (якщо довжина набору зрозуміла з контексту).

Щоб зробити викладення цього розділу незалежним, сформулюємо ще раз означення норми двійкового набору та віддалі Хеммінга між наборами. *Нормою* набору $\tilde{a}^n$ називають число $\|\tilde{a}^n\|$, яке дорівнює кількості його одиничних компонент. *Віддалю Хеммінга* між наборами $\tilde{a}^n$ і $\tilde{b}^n$ називають число $\rho(\tilde{a}^n, \tilde{b}^n)$, що дорівнює кількості компонент, у яких набори $\tilde{a}^n$ і $\tilde{b}^n$ різняться.

Набори $\tilde{a}^n$ і $\tilde{b}^n$ називають *сусідніми*, якщо $\rho(\tilde{a}^n, \tilde{b}^n) = 1$, і *протилежними*, якщо $\rho(\tilde{a}^n, \tilde{b}^n) = n$. Отже, сусідні набори різняться точно однією компонентою, а протилежні — усіма n компонентами. Наприклад, набори (0100) і (1100) сусідні, а (0100) і (1011) — протилежні.

Скінченність області визначення булевих функцій дає змогу задавати такі функції за допомогою таблиць. Розглянемо двійкові набори значень змінних як записи цілих чисел у двійковій системі числення. Це означає, що набір $\tilde{a}^n = (a_1, ..., a_n)$ ототожнюють із записом числа $a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + ... + a_{n-1} \cdot 2 + a_n$. Назвемо це число номером набору $\tilde{a}^n$. Наприклад, для тримісної булевої функції номером набору (110) — число $1 \cdot 2^2 + 1 \cdot 2 + 0 = 6$.

Номери наборів значень змінних n-місної булевої функції змінюються від 0 до $2^n - 1$. Розмістимо набори в стовпчик за зростанням їх номерів і покажемо значення функції на кожному наборі. Одержимо *таблицю булевої функції* (табл. 7.1).

Таблиця 7.1

x_1	x_2	...	x_{n-1}	x_n	$f(x_1, x_2, ..., x_{n-1}, x_n)$
0	0	...	0	0	$f(0, 0, ..., 0, 0)$
0	0	...	0	1	$f(0, 0, ..., 0, 1)$
0	0	...	1	0	$f(0, 0, ..., 1, 0)$
...	...	...	...	...	... ...
1	1	...	1	1	$f(1, 1, ..., 1, 1)$

Правий стовпець (стовпець значень функції) складається з 2^n нулів і одиниць. Отже, n-місних булевих функцій стільки, скільки наборів довжиною 2^n з 0 і 1. Тому справджується таке твердження.

ТЕОРЕМА 7.1. Кількість різних булевих функцій від n змінних дорівнює 2^{2^n}.

Далі ми завжди будемо вважати, що набори значень булевих функцій розміщено за зростанням їх номерів (від 0 до $2^n - 1$). Тому функцію $f(\tilde{x}^n) = f(x_1, ..., x_n)$ можна задати вектором $\tilde{y}_f = (y_0, y_1, ..., y_{2^n-1})$, у якому компонента y_i — це значення функції $f(\tilde{x}^n)$ на i-му наборі значень змінних, де $i = 0, 1, ..., 2^n - 1$.

Множину наборів, на яких булева функція $f(\tilde{x}^n)$ набуває значення 1, позначають N_f:

$$N_f = \{\tilde{a}^n \mid \tilde{a}^n \in E_2^n, f(\tilde{a}^n) = 1\}.$$

Очевидно, що множина N_f повністю визначає функцію f.

Змінну x_i функції $f(x_1, ..., x_{i-1}, x_i, x_{i+1}, ..., x_n)$ називають *істотною*, якщо існує такий набір $(a_1, ..., a_{i-1}, a_{i+1}, ..., a_n)$ значень решти змінних, що

$$f(a_1, ..., a_{i-1}, 0, a_{i+1}, ..., a_n) \neq f(a_1, ..., a_{i-1}, 1, a_{i+1}, ..., a_n).$$

Змінну, яка не є істотною, називають *неістотною* або *фіктивною*. Отже, змінна x_i функції $f(x_1, ..., x_{i-1}, x_i, x_{i+1}, ..., x_n)$ неістотна (фіктивна), якщо

$$f(x_1, ..., x_{i-1}, 0, x_{i-1}, ..., x_n) = f(x_1, ..., x_{i-1}, 1, x_{i+1}, ..., x_n)$$

для будь-яких значень решти змінних. Це означає, що зміна значення x_i в довільному наборі значень $x_1, ..., x_n$ не змінює значення функції. Тоді функція $f(x_1, ..., x_n)$ насправді залежить від $(n-1)$ змінної, тобто являє собою функцію $g(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)$. У такому разі можна говорити, що функцію g отримано з функції f *вилученням* фіктивної змінної, а функцію f отримано з g *введенням* фіктивної змінної. Функції f і g називають *рівними*, якщо функцію g можна одержати з f уведенням або вилученням фіктивних змінних.

Фіктивні змінні вилучають тому, що вони не впливають на значення функції, тому зайві. Проте іноді корисно вводити фіктивні змінні. Завдяки цьому будь-яку функцію n змінних можна зробити функцією довільної більшої кількості змінних. Тому можна вважати, що всі булеві функції зі скінченної множини $\{f_1, ..., f_s\}$ залежать від одних і тих самих змінних $x_1, ..., x_n$. Зокрема, твердження $|P_2(n)| = 2^{2^n}$ теореми 7.1 передбачає, що ми враховуємо всі булеві функції від n змінних, включаючи функції з фіктивними змінними.

Зі зростанням кількості змінних швидко збільшується кількість залежних від них булевих функцій. Наприклад, різних булевих функцій чотирьох змінних є $2^{2^4} \approx 65{,}5$ тис., а п'яти змінних — $2^{2^5} \approx 4$ млрд. Зі збільшенням кількості змінних таблиці для булевих функцій стають громіздкими, і ними незручно користуватись.

Розглянемо аналітичний метод подання булевих функцій, тобто зображення функцій формулами. Спочатку за допомогою табл. 7.2, 7.3 означають функції, які називають *елементарними*.

Таблиця 7.2

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

Таблиця 7.3

x_1	x_2	x_1x_2	$x_1 \vee x_2$	$x_1 \rightarrow x_2$	$x_1 \sim x_2$	$x_1 \oplus x_2$	$x_1 \mid x_2$	$x_1 \downarrow x_2$
0	0	0	0	1	1	0	1	1
0	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	1	0
1	1	1	1	1	1	0	0	0

Уведені елементарні функції мають такі назви:

- $f_1(x) = 0$ — *константа* 0;
- $f_2(x) = 1$ — *константа* 1;
- $f_3(x) = x$ — *тотожна функція*;
- $f_4(x) = \overline{x}$ — *заперечення* x, читають „не x";
- $f_5(x_1, x_2) = x_1x_2$ — *кон'юнкція*, читають „x_1 і x_2" (іноді використовують символи $\wedge$ та &);
- $f_6(x_1, x_2) = x_1 \vee x_2$ — *диз'юнкція*, читають „x_1 або x_2";
- $f_7(x_1, x_2) = x_1 \rightarrow x_2$ — *імплікація*, читають „із x_1 випливає x_2" (іноді використовують символ $\supset$);

- $f_8(x_1, x_2) = x_1 \sim x_2$ — *еквівалентність* (використовують також символ $\equiv$);
- $f_9(x_1, x_2) = x_1 \oplus x_2$ — *додавання за* mod2, читають також як альтернативне „або" („або, або");
- $f_{10}(x_1, x_2) = x_1 \mid x_2$ — *штрих Шеффера*;
- $f_{11}(x_1, x_2) = x_1 \downarrow x_2$ — *стрілка Пірса*.

За допомогою елементарних функцій можна подати будь-яку булеву функцію аналітично, тобто у вигляді формули. Нехай задано булеві функції $f(x)$ і $g(x)$. Говорять, що функцію $h(x) = g(f(x))$ отримано *підстановкою* f у g. По-перше, можливі будь-які підстановки булевих функцій замість змінних у багатомісні булеві функції $f_1, ..., f_m$. По-друге, можливі будь-які *перейменування* змінних; наприклад, перейменування x_3 в x_2 перетворює функцію $f(x_1, x_2, x_3, x_4)$ на функцію трьох змінних $f(x_1, x_2, x_2, x_4)$ (у такому разі говорять, що змінні x_2 та x_3 *ототожнено*). Функцію, одержану з функцій $f_1, ..., f_m$ якоюсь підстановкою їх одна в одну або перейменуванням змінних, називають ***суперпозицією функцій*** $f_1, ..., f_m$. Вираз, який описує цю суперпозицію та містить функціональні знаки, круглі дужки й символи змінних, називають *формулою*. Поняття суперпозиції дуже важливе, тому розглянемо його докладніше.

Нехай задано множину елементарних функцій $Q = \{f_1, ..., f_m\}$. Символи змінних $x_1, ..., x_n$ уважають ***формулами глибиною*** 0. Формула F має ***глибину*** $k+1$, якщо вона має вигляд $f_i(F_1, ..., F_{n_i})$, де $f_i \in Q$, n_i – кількість аргументів функції f_i, а $F_1, ..., F_{n_i}$ — формули, максимальна з глибин яких дорівнює k. $F_1, ..., F_{n_i}$ називають ***підформулами*** формули F, а функцію f_i – зовнішньою (головною) операцією формули F. Усі підформули формул $F_1, ..., F_{n_i}$ також називають підформулами формули F. Наприклад, $f_5(x_1, x_2)$ — це формула глибиною 1, а $f_9(f_6(x_3, x_1), f_5(x_1, f_9(x_1, x_2)))$ – формула глибиною 3, яка містить одну підформулу глибиною 2 та дві підформули глибиною 1.

Надалі будемо розглядати конкретні формули в більш звичному вигляді, коли знаки функцій стоять між аргументами (такий запис називають *інфіксним*).

Приклад 7.1. Якщо f_5 – кон'юнкція, f_6 – диз'юнкція, f_9 – додавання за mod2, то наведена вище формула має вигляд

$$(x_3 \vee x_1) \oplus (x_1 \wedge (x_1 \oplus x_2)).$$

Цю тримісну функцію задано не таблицею, а формулою. Вона являє собою суперпозицію функцій f_5, f_6 і f_9.

Формулу, побудовану описаним вище способом, тобто таку, що містить лише символи змінних, дужки та знаки функцій із множини Q, називають *формулою над* Q.

Приклад 7.2. Нехай $Q = \{\overline{x}, xy, x \vee y\}$; тоді вираз $\overline{(x \vee y)} z \vee t$ – формула над Q.

Щоб зменшити кількість дужок у формулах, уводять пріоритет операцій:

- заперечення;
- кон'юнкція;
- усі інші операції.

Крім того, домовляються, що символ заперечення відіграє роль дужок, якщо він є над частиною формули.

Можна й інакше означити поняття глибини. Наприклад, часто вважають, що розставлення заперечень над змінними не збільшує глибини; коли формула Q містить асоціативну операцію f, можна означити глибину так, що застосування f до формул з тою самою зовнішньою операцією f не збільшує глибини формули. Наприклад, формули $x_1(x_2 \vee x_3x_4)$ та $x_2x_1(x_2 \vee x_3x_4)$ мають одну й ту саму глибину 3; диз'юнктивні та кон'юнктивні нормальні форми (див. підрозділ 7.3) завжди мають глибину 2.

Із вигляду формули, у якій функцію f виражено як суперпозицію інших функцій, випливає очевидний спосіб її обчислення за таким правилом: формулу можна обчислити лише тоді, коли вже обчислено значення всіх її підформул. Отже, формула кожному набору значень аргументів ставить у відповідність значення функції, тому її можна вважати способом подання й обчислення функції. Зокрема, за формулою обчисленням її значень на всіх 2^n наборах можна відновити таблицю функції. Про формулу, що подає функцію, говорять також, що вона *реалізує* (*задає*) цю функцію.

Приклад 7.3. Функцію задано формулою $\overline{x} \to (\overline{z} \sim (y \oplus xz))$. Потрібно задати її таблицею. Процес розв'язування цієї задачі проілюстровано в табл. 7.4.

Таблиця 7.4

x	y	z	xz	$y \oplus xz$	$\overline{z}$	$\overline{z} \sim (y \oplus xz)$	$\overline{x}$	$\overline{x} \to (\overline{z} \sim (y \oplus xz))$
0	0	0	0	0	1	0	1	0
0	0	1	0	0	0	1	1	1
0	1	0	0	1	1	1	1	1
0	1	1	0	1	0	0	1	0
1	0	0	0	0	1	0	0	1
1	0	1	1	1	0	0	0	1
1	1	0	0	1	1	1	0	1
1	1	1	1	0	0	1	0	1

Подання функції формулою не єдине. Формули називають *еквівалентними* (*рівносильними*), якщо вони реалізують рівні булеві функції. Еквівалентність формул позначають символом $=$.

Приклад 7.4. Формули $x \to y$ і $(\overline{x} \vee y) \vee z\overline{z}$ еквівалентні: $x \to y = (\overline{x} \vee y) \vee z\overline{z}$. У цьому можна переконатись, побудувавши таблиці відповідних булевих функцій. Очевидно, що змінна z у другій формулі фіктивна.

Крім побудови таблиць є й інші методи доведення еквівалентності формул і побудови нових формул, рівносильних даним. Ці методи називають *рівносильними* (*еквівалентними*) *перетвореннями формул*.

7.2. Алгебри булевих функцій

Нехай функцію f_1 задано формулою F_1, а функцію f_2 – формулою F_2. Підстановка F_1 і F_2, наприклад, у диз'юнкцію $x_1 \vee x_2$ дає формулу $F_1 \vee F_2$. Узявши формулу Φ_1, рівносильну F_1 (тобто формула Φ_1 також реалізує функцію f_1) та формулу Φ_2, рівносильну F_2, отримаємо формулу $\Phi_1 \vee \Phi_2$, рівносильну формулі $F_1 \vee F_2$. Отже, диз'юнкцію можна розглядати як двомісну операцію на множині всіх булевих функцій. Ця операція кожній парі функцій f_1 і f_2, незалежно від вигляду формул, якими їх подано, однозначно ставить у відповідність функцію $f_1 \vee f_2$. Аналогічно, й інші булеві функції можна розглядати як операції на множині P_2 всіх булевих функцій. Наприклад, заперечення – одномісна операція, кон'юнкція та диз'юнкція – двомісні.

Множину P_2 всіх булевих функцій разом з уведеною на ній системою операцій називають *алгеброю булевих функцій* [11, 47].

Розглянемо дві алгебри. Алгебру $(P_2; \neg, \wedge, \vee)$ з операціями заперечення, кон'юнкції та диз'юнкції називають *алгеброю Буля*, а алгебру $(P_2; \wedge, \oplus)$ з операціями кон'юнкції та додавання за mod2 – *алгеброю Жегалкіна*. Формули цих алгебр будують зі знаків операцій, круглих дужок, символів змінних і констант 0 і 1. Знак кон'юнкції $\wedge$ у формулах обох алгебр зазвичай не пишуть.

Якщо немає дужок, пріоритет операцій у булевій алгебрі такий: заперечення, кон'юнкція, диз'юнкція. В алгебрі Жегалкіна спочатку виконується кон'юнкція, а потім – додавання за mod2. За наявності дужок спочатку виконуються операції всередині їх. У булевій алгебрі як дужки в разі заперечення виразів використовують сам символ заперечення.

Приклад 7.5. Замість $\overline{(x \vee y)} \vee (xy)$ можна написати $\overline{x \vee y} \vee xy$, а замість $(xy) \oplus (yz)$ – $xy \oplus yz$.

Одна з найважливіших задач – виявлення основних еквівалентностей. Ці еквівалентності називають *законами* відповідної алгебри.

Закони алгебри Буля:

- закони асоціативності $(xy)z = x(yz) = xyz$, $(x \vee y) \vee z = x \vee (y \vee z) = x \vee y \vee z$;
- закони комутативності $xy = yx$, $x \vee y = y \vee x$;
- дистрибутивний закон для кон'юнкції щодо диз'юнкції $(x \vee y)\, z = xz \vee yz$;
- дистрибутивний закон для диз'юнкції щодо кон'юнкції $xy \vee z = (x \vee z)(y \vee z)$;
- закон подвійного заперечення $\overline{\overline{x}} = x$;
- закони де Моргана $\overline{xy} = \overline{x} \vee \overline{y}$, $\overline{x \vee y} = \overline{x}\,\overline{y}$;
- закони ідемпотентності $xx = x$, $x \vee x = x$;
- закони поглинання $x \vee xy = x$, $x(x \vee y) = x$;
- співвідношення для констант $\overline{1} = 0$, $\overline{0} = 1$, $1x = x$, $0x = 0$, $1 \vee x = 1$, $0 \vee x = x$;
- закон виключеного третього $x \vee \overline{x} = 1$;
- закон суперечності $x\overline{x} = 0$.

Закони алгебри Жегалкіна:

- закони асоціативності $(xy)z = x(yz) = xyz$, $(x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus y \oplus z$;
- закони комутативності $xy = yx$, $x \oplus y = y \oplus x$;
- дистрибутивний закон для кон'юнкції щодо додавання *за* mod2 $x(y \oplus z) = xy \oplus xz$;
- співвідношення для констант $1x = x$, $0x = 0$, $x \oplus 0 = x$;
- закон ідемпотентності для кон'юнкції $xx = x$;
- закон зведення подібних членів у разі додавання за mod2 $x \oplus x = 0$.

Правильність цих еквівалентностей можна довести за допомогою таблиць.

Наведені еквівалентності справджуються й у разі підстановки замість змінних довільних булевих функцій (тобто формул, які подають ці функції). Важливо лише дотримуватися такого правила підставлення формули замість змінної: підставляючи формулу F замість змінної x, усі входження змінної x у цю еквівалентність потрібно водночас замінити формулою F. Наприклад, підставивши замість змінної x формулу xy, а замість $y - z \vee u$, з $\overline{xy} = \overline{x} \vee \overline{y}$ отримаємо $\overline{xy(z \vee u)} = \overline{xy} \vee \overline{z \vee u}$.

Закони алгебр Буля та Жегалкіна дають змогу доводити нові еквівалентності вже без таблиць, на основі тотожних перетворень.

Приклад 7.6. Доведемо рівності

$$\text{а) } (x \vee y)(z \vee u) = xz \vee yz \vee xu \vee yu;$$
$$\text{б) } xy \vee zu = (x \vee z)(y \vee z)(x \vee u)(y \vee u).$$

Двічі застосувавши закони дистрибутивності, одержимо:

$$\text{а) } (x \vee y)(z \vee u) = (x \vee y)z \vee (x \vee y)u = xz \vee yz \vee xu \vee yu.$$
$$\text{б) } xy \vee zu = (x \vee zu)(y \vee zu) = (x \vee z)(y \vee z)(x \vee u)(y \vee u).$$

Приклад 7.7. Доведемо, що $\overline{\overline{x}zy} = x \vee \overline{z} \vee \overline{y}$. Послідовно застосувавши закони де Моргана, подвійного заперечення й асоціативності, запишемо

$$\overline{\overline{x}zy} = \overline{\overline{x}z} \vee \overline{y} = \overline{\overline{x}} \vee \overline{z} \vee \overline{y} = x \vee \overline{z} \vee \overline{y}.$$

Наведемо еквівалентності, які дають змогу перетворити будь-яку формулу булевої алгебри в рівносильну до неї формулу алгебри Жегалкіна й навпаки:

- $\overline{x} = 1 \oplus x$;
- $x \vee y = x \oplus y \oplus xy$;
- $x \oplus y = \overline{x}y \vee x\overline{y}$.

За допомогою законів алгебр Буля та Жегалкіна можна спрощувати різні формули в цих алгебрах.

Приклад 7.8. Перетворимо $\overline{xy \vee \overline{z}}$ на рівносильну формулу алгебри Жегалкіна. Одержану формулу спростимо.

$$\overline{xy \vee \overline{z}} = (xy \vee (z \oplus 1)) \oplus 1 = (xy \oplus (z \oplus 1) \oplus xy(z \oplus 1)) \oplus 1 =$$
$$= xy \oplus z \oplus 1 \oplus xyz \oplus xy \oplus 1 = xyz \oplus z.$$

Запишемо формули $x \to y$ та $x \sim y$ за допомогою операцій алгебри Буля. Оскільки $x \sim y = (x \to y)(y \to x)$, то достатньо виразити формулу $x \to y$. Порівнявши таблицю для $x \to y$ із таблицею для $\overline{x} \vee y$, отримаємо еквівалентність $x \to y = \overline{x} \vee y$.

Розглянемо тепер іще одне важливе поняття. Функцію $f^*(x_1, ..., x_n)$ називають *двоїстою* до функції $f(x_1, ..., x_n)$, якщо $f^*(x_1, ..., x_n) = \overline{f}(\overline{x}_1, ..., \overline{x}_n)$. Візьмемо заперечення обох частин останньої рівності й підставимо $\overline{x}_1, ..., \overline{x}_n$ замість $x_1, ..., x_n$; одержимо $\overline{f}^*(\overline{x}_1, ..., \overline{x}_n) = \overline{\overline{f}}(\overline{\overline{x}}_1, ..., \overline{\overline{x}}_n) = f(x_1, ..., x_n)$. Звідси випливає, що функція f двоїста до f^*, тобто $(f^*)^* = f$. Отже, відношення двоїстості між функціями симетричне. Пари двоїстих функцій утворюють, наприклад, кон'юнкція та диз'юнкція, константи 0 і 1, додавання за mod2 й еквівалентність. Таблицю для двоїстої функції для вибраного нами порядку наборів отримують із таблиці для функції $f(x_1, ..., x_n)$ інвертуванням (тобто заміною 0 на 1 й 1 на 0) стовпця функції та його перевертанням.

З означення двоїстості зрозуміло, що для довільної функції двоїста функція визначається однозначно. Функція може бути двоїстою до самої себе. Тоді її називають *самодвоїстою*. Наприклад, заперечення $\overline{x}$ та $x \oplus y \oplus z$ – самодвоїсті функції.

Нехай функцію задано формулою F. Який вигляд має формула F^*, що задає двоїсту функцію? Відповідь на це питання дає така теорема.

ТЕОРЕМА 7.2. Якщо

$$F(x_1, ..., x_n) = f(f_1(x_1, ..., x_n), ..., f_s(x_1, ..., x_n)),$$

то

$$F^*(x_1, ..., x_n) = f^*(f_1^*(x_1, ..., x_n), ..., f_s^*(x_1, ..., x_n)).$$

Доведення.

$$\begin{aligned} F^*(x_1, ..., x_n) &= \overline{F}(\overline{x}_1, ..., \overline{x}_n) = \overline{f}(f_1(\overline{x}_1, ..., \overline{x}_n), ..., f_s(\overline{x}_1, ..., \overline{x}_n)) = \\ &= \overline{f}(\overline{\overline{f}}_1(\overline{x}_1, ..., \overline{x}_n), ..., \overline{\overline{f}}_s(\overline{x}_1, ..., \overline{x}_n)) = \overline{f}(\overline{f}_1^*(x_1, ..., x_n), ..., \overline{f}_s^*(x_1, ..., x_n)) = \\ &= f^*(f_1^*(x_1, ..., x_n), ..., f_s^*(x_1, ..., x_n)). \end{aligned}$$

З теореми 7.2 випливає таке твердження.

Принцип двоїстості. Якщо у формулі F, що реалізує функцію f, усі символи функцій замінити символами відповідних двоїстих функцій, то одержана формула F^* реалізує функцію f^*, двоїсту до f.

Інакше це твердження можна сформулювати так: функція, двоїста до суперпозиції функцій, дорівнює відповідній суперпозиції двоїстих функцій.

В алгебрі Буля принцип двоїстості має простіший вигляд.

Принцип двоїстості в алгебрі Буля. Якщо у формулі F, що реалізує функцію f, усі кон'юнкції замінити на диз'юнкції, диз'юнкції – на кон'юнкції, 1 – на 0, 0 – на 1, то отримаємо формулу F^*, яка реалізує функцію f^*, двоїсту до f.

Приклад 7.9. Знайдемо функцію, двоїсту до $f(x,y,z) = xy \vee \overline{z}(x \vee \overline{u}y)$.
За принципом двоїстості маємо $f^*(x,y,z) = (x \vee y)(\overline{z} \vee x(\overline{u} \vee y))$. Застосовуючи принцип двоїстості, слід ураховувати пріоритет операцій, тобто в разі потреби розставляти дужки. Так, у формулі $xy \vee \overline{z}(x \vee \overline{u}y)$ спочатку виконуються кон'юнкції xy та $\overline{u}y$ (дужки випущено за домовленістю про пріоритет операцій); отже, у формулі, що реалізує двоїсту функцію, спочатку мають виконуватися диз'юнкції $x \vee y$ та $\overline{u} \vee y$, тому потрібні дужки.

Якщо функції рівні, то й двоїсті до них функції також рівні. Це дає змогу отримувати нові еквівалентності за допомогою принципу двоїстості. Для цього потрібно від еквівалентності $F_1 = F_2$ перейти до $F_1^* = F_2^*$.

7.3. Спеціальні форми подання булевих функцій

Спеціальними формами подання булевих функцій є диз'юнктивні й кон'юнктивні нормальні форми та поліном Жегалкіна.

7.3.1. Диз'юнктивні нормальні форми

Уведемо позначення $x^\sigma = x\sigma \vee \overline{x}\overline{\sigma}$, де σ — параметр, який дорівнює 0 чи 1. Очевидно, що

$$x^\sigma = \begin{cases} \overline{x}, & \text{якщо } \sigma = 0, \\ x, & \text{якщо } \sigma = 1. \end{cases}$$

Зауважимо, що $\sigma^\sigma = 1$.

Зафіксуємо множину змінних $X = \{x_1, x_2, ..., x_n\}$.

Елементарною кон'юнкцією називають вираз $k = x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} ... x_{i_r}^{\sigma_r}$, де x_{i_j} — змінні з множини X, причому всі x_{i_j} різні. Число r називають *рангом* кон'юнкції. У разі $r = 0$ кон'юнкцію називають *порожньою* та вважають такою, що дорівнює 1.

Приклад 7.10. Елементарними кон'юнкції – це, наприклад 1, $\overline{x_1}, x_1x_2, x_2, \overline{x_1}\,\overline{x_2}x_3$, а вирази 0, $x_1x_2x_1, \overline{x_1}x_1, \overline{\overline{x_1}x_2}$ – не елементарні кон'юнкції.

Елементарну кон'юнкцію, яка містить усі змінні з множини X, називають *конституентою одиниці*. Інакше кажучи, конституента одиниці — це елементарна кон'юнкція з рангом n. Очевидно, що всіх різних конституент одиниці для фіксованої множини n змінних $x_1, x_2, ..., x_n$ стільки, скільки двійкових наборів з n компонентами, тобто 2^n.

Диз'юнктивною нормальною формою (*ДНФ*) називають диз'юнкцію $D = k_1 \vee k_2 \vee ...$ $... \vee k_s$ елементарних кон'юнкцій k_j, у якій k_j попарно різні.

Є алгоритм, який дає змогу для будь-якої формули булевої алгебри на основі тотожних перетворень знайти рівносильну до неї ДНФ. На першому його етапі формулу перетворюють на рівносильну, побудовану зі змінних і їх заперечень за допомогою самих лише кон'юнкцій і диз'юнкцій (тобто заперечення можуть стояти лише над змінними). Для цього використовують закони де Моргана та закон подвійного заперечення.

На другому етапі домагаються, щоб усі кон'юнкції виконувалися раніше, ніж диз'юнкції, для чого розкривають дужки на підставі дистрибутивного закону для кон'юнкції $(x \vee y)z = xz \vee yz$ або рівносильності $(x \vee y)(z \vee u) = xz \vee yz \vee xu \vee yu$ (див приклад 7.6). Далі з використанням співвідношень для констант і закону суперечності вилучають нулі та, виходячи із законів ідемпотентності, об'єднують рівні члени. На цьому процес отримання ДНФ закінчують.

Приклад 7.11. Зведемо формулу $\overline{x \vee z}(x \to y)$ до ДНФ. Застосовуючи сформульований алгоритм, можемо записати

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

Зазначимо, що ДНФ булевої функції не єдина; наприклад,

$$xz \vee y\overline{z} \vee xy = xz \vee y\overline{z}.$$

Досконалою диз'юнктивною нормальною формою (ДДНФ) називають ДНФ, у якій кожна елементарна кон'юнкція k_j $(j = 1, ..., s)$ — конституента одиниці.

ТЕОРЕМА 7.3. Будь-яку булеву функцію $f(x_1, ..., x_n) \neq 0$ можна єдиним способом подати в ДДНФ.

Доведення. Нехай задано функцію $f(x_1, ..., x_n) \neq 0$. Кожному двійковому набору $\tilde{a}_n = (a_1, a_2, ..., a_n)$ значень змінних відповідає єдина конституента одиниці $k = x_1^{a_1} x_2^{a_2} ... x_n^{a_n}$, яка перетворюється на цьому наборі в 1. Усі інші конституенти одиниці на цьому наборі перетворюється в 0. Наприклад, набору (0101) відповідає конституента $\overline{x_1}x_2\overline{x_3}x_4$.

Нехай f_i — значення функції f, якого вона набуває на i-му двійковому наборі $(i = 0, 1, ..., 2^n - 1)$, а k_i — конституента одиниці, що відповідає i-му набору. Доведемо рівність

$$f(x_1, ..., x_n) = f_0 k_0 \vee f_1 k_1 \vee ... \vee f_{2^n-1} k_{2^n-1}.$$

Для l-го набору $f_l = 0 \vee 0 \vee ... \vee 0 \vee f_l 1 \vee 0 \vee ... \vee 0 = f_l$. Нульові члени в диз'юнкції можна випустити. Отже, диз'юнкція конституент одиниці, що відповідають усім двійковим наборам, на яких булева функція набуває значення 1, являє собою ДДНФ функції $f(x_1, ..., x_n)$:

$$f(x_1, x_2, ..., x_n) = \bigvee_{\substack{(\sigma_1, \sigma_2, ..., \sigma_n) \\ f(\sigma_1, \sigma_2, ..., \sigma_n) = 1}} x_1^{\sigma_1} x_1^{\sigma_2} ... x_n^{\sigma_n}.$$

Для доведення єдиності ДДНФ скористаємося таким комбінаторним міркуванням. Знайдемо кількість ДДНФ від n змінних $x_1, ..., x_n$. Для цього будь-яким способом занумеруємо конституенти одиниці, їх буде 2^n. Кожній ДДНФ від змінних $x_1, ..., x_n$ можна таким взаємно однозначним способом поставити у відповідність набір із 2^n нулів і одиниць. На позиції з номерами тих конституент одиниці, які входять у ДДНФ, поставимо одиниці, а на решту позицій — нулі.

Нульовий набір при цьому не отримаємо, бо він відповідав би порожній ДДНФ. Отже, різних ДДНФ стільки, скільки існує наборів довжиною 2^n, відмінних від набору із самих лише нулів, тобто $2^{2^n}-1$. Функцій (окрім тотожного нуля) від змінних $x_1, ..., x_n$ також $2^{2^n}-1$. Кожну з них можна подати ДДНФ, до того ж єдиним способом.

Із доведення теореми 7.3 випливає, що для функції, заданої таблицею, ДДНФ будують так: для кожного набору, на якому функція набуває значення 1, будують відповідну йому конституенту одиниці; диз'юнкція всіх цих конституент — це ДДНФ даної функції.

Приклад 7.12. Побудуємо ДДНФ для функції, заданої табл. 7.5.

Таблиця 7.5

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	0
1	0	0
1	1	1

Функція набуває значення 1 на наборах (00) і (11), отже $f(x_1, x_2) = \overline{x_1}\,\overline{x_2} \vee x_1x_2$.

Будь-яку ДНФ можна звести до ДДНФ *розщепленням* кон'юнкцій, які містять не всі змінні: якщо кон'юнкція k не містить змінної x, то

$$k = k(x \vee \overline{x}) = kx \vee k\overline{x}.$$

Приклад 7.13. Перетворимо ДНФ $\overline{x}\,\overline{z} \vee \overline{x}yz$ на досконалу. Застосувавши розщеплення для кон'юнкції $\overline{x}\,\overline{z}$, одержимо

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

Якщо з формули F_1 за допомогою тотожних перетворень можна отримати формулу F_2, то з F_2 можна одержати F_1, обернувши ці перетворення.

ТЕОРЕМА 7.4. Для довільних двох рівносильних формул алгебри Буля F_1 і F_2 існує еквівалентне перетворення F_1 на F_2 за допомогою законів цієї алгебри.

Доведення. Перетворимо F_1 і F_2 на ДДНФ. Оскільки формули F_1 і F_2 рівносильні, то їх ДДНФ збігаються. Обернувши друге перетворення, матимемо такий ланцюжок перетворень: з F_1 одержимо ДДНФ, з ДДНФ — F_2.

Важливість цієї теореми полягає в тому, що набір законів алгебри Буля виявляється достатнім для довільних еквівалентних перетворень у ній.

7.3.2. Кон'юнктивні нормальні форми

Двоїстим способом, замінюючи в означеннях нулі одиницями й навпаки, диз'юнкції кон'юнкціями й навпаки, означають поняття елементарної диз'юнкції, конституенти нуля, кон'юнктивної нормальної форми, досконалої кон'юнктивної нормальної форми.

Нехай, як і раніше, зафіксовано множину $X = \{x_1, x_2, ..., x_n\}$. *Елементарною диз'юнкцією* називають вираз $d = x_{i_1}^{\sigma_1} \vee x_{i_2}^{\sigma_2} \vee ... \vee x_{i_r}^{\sigma_r}$, у якому всі x_{i_j} різні, $x_{i_j} \in X$. Число r називають *рангом* диз'юнкції. Якщо $r = 0$, диз'юнкцію називають *порожньою* та вважають такою, що дорівнює 0. Наприклад, елементарні диз'юнкції – це $x_1 \vee x_3 \vee \overline{x_7}, 0, \overline{x_1} \vee x_2$.

Кон'юнктивною нормальною формою (*КНФ*) називають кон'юнкцію $d_1 \wedge d_2 \wedge ... \wedge d_s$ елементарних диз'юнкцій d_j, у якій усі d_j різні.

Є алгоритм, який дає змогу для будь-якої формули булевої алгебри знайти рівносильну до неї КНФ. Перший етап цього алгоритму такий самий, як і для побудови ДНФ. На другому етапі домагаються, щоб усі диз'юнкції виконувалися раніше кон'юнкцій. Для цього потрібно скористатися дистрибутивним законом $x \vee yz = (x \vee y)(x \vee z)$ або наслідком із нього $xy \vee zu = (x \vee y)(x \vee u)(y \vee z)(y \vee u)$ (див. приклад 7.6). Потім на підставі співвідношень для констант і закону виключеного третього вилучають одиниці та на підставі законів ідемпотентності об'єднують рівні члени.

Приклад 7.14. Знайдемо КНФ для формули $\overline{x \vee z}\ (x \to y)$. Застосовуючи сформульований алгоритм, одержимо

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

Елементарну диз'юнкцію, яка містить усі змінні з множини X, називають *конституентою нуля*. Інакше кажучи, конституента нуля – це елементарна диз'юнкція з рангом n. Кожному двійковому набору $\tilde{b}^n = (b_1, b_2, ..., b_n)$ взаємно однозначно відповідає конституента нуля $x_1^{\overline{b}_1} \vee x_2^{\overline{b}_2} \vee ... x_n^{\overline{b}_n}$, яка перетворюється на ньому в 0. Усі інші конституенти нуля на цьому наборі перетворюються в 1. Наприклад, набору 0111 відповідає конституента нуля $x_1 \vee \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$.

Досконалою кон'юнктивною нормальною формою (ДКНФ) називають КНФ, у якій кожна елементарна диз'юнкція d_j ($j = 1, ..., s$) – конституента нуля.

Доведемо, що кожну булеву функцію $f(x_1, ..., x_n) \neq 1$ можна подати досконалою КНФ. Запишемо ДДНФ для f^* (зазначимо, що $f^* \neq 0$).

$$f^*(x_1, x_2, ..., x_n) = \bigvee_{\substack{(\tau_1, ..., \tau_n) \\ f^*(\tau_1, ..., \tau_n)=1}} x_1^{\tau_1} \wedge ... \wedge x_n^{\tau_n}.$$

Запишемо тотожність для двоїстих формул

$$(f^*)^*(x_1, x_2, ..., x_n) = \bigwedge_{\substack{(\tau_1, ..., \tau_n) \\ f^*(\tau_1, ..., \tau_n)=1}} (x_1^{\tau_1} \vee ... \vee x_n^{\tau_n}).$$

Ліва частина дорівнює $f(x_1, ..., x_n)$, а праву перетворюємо далі:

$$\bigwedge_{\substack{(\tau_1, ..., \tau_n) \\ f^*(\tau_1, ..., \tau_n)=1}} (x_1^{\tau_1} \vee ... \vee x_n^{\tau_n}) = \bigwedge_{\substack{(\tau_1, ..., \tau_n) \\ f(\overline{\tau}_1, ..., \overline{\tau}_n)=0}} (x_1^{\tau_1} \vee ... \vee x_n^{\tau_n}) =$$

$$= \bigwedge_{\substack{(\sigma_1, ..., \sigma_n) \\ f(\sigma_1, ..., \sigma_n)=0}} (x_1^{\overline{\sigma}_1} \vee ... \vee x_n^{\overline{\sigma}_n}).$$

Отже,

$$f(x_1, x_2, ..., x_n) = \bigwedge_{\substack{(\sigma_1, ..., \sigma_n) \\ f(\sigma_1, ..., \sigma_n) = 0}} (x_1^{\overline{\sigma}_1} \vee x_2^{\overline{\sigma}_2} \vee ... \vee x_n^{\overline{\sigma}_n}).$$

Звідси випливає, що ДКНФ за таблицею булевої функції f будують так. Виділяють набори, на яких функція має значення 0, і для кожного з них записують відповідну конституенту нуля. Кон'юнкція цих конституент нуля являє собою ДКНФ функції f.

Приклад 7.15. Побудуємо ДКНФ для функції, заданої табл. 7.5. Функція набуває значення 0 на наборах (01) і (10), отже $f(x_1, x_2) = (x_1 \vee \overline{x_2})(\overline{x_1} \vee x_2)$.

Зазначимо, що за допомогою тотожних перетворень будь-яку КНФ можна перетворити на ДКНФ. Якщо в якусь елементарну диз'юнкцію d не входить змінна x, то потрібно записати рівносильний вираз $d \vee x\overline{x}$ та застосувати дистрибутивний закон: $d \vee x\overline{x} = (d \vee x)(d \vee \overline{x})$. Після тривіальних перетворень отримаємо ДКНФ.

Приклад 7.16. Перетворимо КНФ $\overline{x}\overline{z}(\overline{x} \vee y)$ на ДКНФ. Розщепивши диз'юнкції, можемо записати

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

Зазначимо, що ДКНФ єдина.

7.3.3. Поліном Жегалкіна

Елементарну кон'юнкцію називають *монотонною*, якщо вона не містить заперечень змінних. Наприклад, $x_1x_2x_3$, x_1, 1 — монотонні кон'юнкції.

Формулу

$$P(\tilde{x}^n) = k_1 \oplus k_2 \oplus ... \oplus k_s,$$

де $k_1, k_2, ..., k_s$ — попарно різні монотонні кон'юнкції змінних із множини $X = \{x_1, x_2, ..., x_n\}$, називають *поліномом Жегалкіна*. Найбільший із рангів елементарних кон'юнкцій, що входять у поліном, називають ***степенем*** полінома. За окремим означенням 0 також уважатимемо поліномом Жегалкіна.

Приклад 7.17. Формули x, 1, $xyz \oplus xy \oplus z \oplus 1$ — поліноми Жегалкіна, а $x\overline{x}$, $xy \oplus yx \oplus x \oplus$ $\oplus\, 1$ — ні.

Щоб із будь-якої формули алгебри Жегалкіна одержати поліном Жегалкіна, достатньо розкрити дужки (за дистрибутивним законом), застосувати, якщо можливо, закон ідемпотентності для кон'юнкції та звести подібні члени.

ТЕОРЕМА 7.5. Будь-яку булеву функцію можна єдиним способом подати у вигляді полінома Жегалкіна.

Доведення. Нехай $f(x_1, ..., x_n)$ – довільна булева функція. Достатньо провести доведення для функцій, відмінних від констант, тому що 1 та 0 – поліноми Жегалкіна. Задамо функцію f досконалою ДНФ, тобто диз'юнкцією конституент одиниці $f = k_1 \vee k_2 \vee ... \vee k_s$. Замінивши знаки $\vee$ на $\oplus$, одержимо формулу $g = k_1 \oplus k_2 \oplus ... \oplus k_s$. Покажемо, що функції f і g рівні. На жодному наборі $(a_1, ..., a_n)$ значень змінних дві різні конституенти одиниці не можуть водночас перетворитися на 1, бо i-та конституента одиниці набуває значення 1 лише на i-му наборі. Отже, обчислюючи значення функції f і g, потрібно обчислити значення багатомісної диз'юнкції чи, відповідно, суми за mod2 членів, із яких тільки один може дорівнювати 1, а решта – обов'язково 0. Але тоді диз'юнкція та додавання за mod2 дають один і той самий результат (0, якщо всі члени дорівнюють 0, й 1, якщо один і тільки один член дорівнює 1).

У формулі $k_1 \oplus k_2 \oplus ... \oplus k_s$ за тотожністю $\bar{x} = 1 \oplus x$ замінимо всі заперечення змінних. Одержимо формулу алгебри Жегалкіна. Залишається лише розкрити дужки, застосувати закон ідемпотентності для кон'юнкції та звести подібні члени. Як результат отримаємо подання булевої функції f у вигляді полінома Жегалкіна.

Залишилося довести, що таке подання єдине. Для цього підрахуємо кількість поліномів Жегалкіна від n змінних $x_1, x_2, ..., x_n$. Кількість кон'юнкцій вигляду $x_{i_1} x_{i_2} ... x_{i_s}$ дорівнює кількості підмножин $\{i_1, i_2, ..., i_s\}$ множини $\{1, 2, ..., n\}$, тобто 2^n. Кожна з цих кон'юнкцій може входити в поліном із коефіцієнтом 0 або 1. Отже, кількість поліномів Жегалкіна від n змінних дорівнює кількості кортежів довжиною 2^n із компонентами 0 або 1, тобто 2^{2^n}, що дорівнює кількості всіх булевих функцій від тих самих змінних $x_1, x_2, ..., x_n$. Позаяк доведено, що будь-яку булеву функцію можна подати поліномом Жегалкіна, то звідси випливає єдиність цього подання.

Розглянемо методи побудови полінома Жегалкіна.

Метод невизначених коефіцієнтів. Для функції $f(x_1, ..., x_n)$ записують найзагальніший вигляд полінома Жегалкіна $P(x_1, ..., x_n)$ із невизначеними коефіцієнтами (їх 2^n). Зокрема, поліном від двох змінних має загальний вигляд

$$P(x, y) = c_0 \oplus c_1x \oplus c_2y \oplus c_3xy,$$

а від трьох змінних –

$$P(x, y, z) = c_0 \oplus c_1x \oplus c_2y \oplus c_3z \oplus c_4xy \oplus c_5xz \oplus c_6yz \oplus c_7xyz.$$

Для кожного двійкового набору $(a_1, ..., a_n)$ значень змінних записують 2^n рівнянь $f(a_1, ..., a_n) = P(a_1, ..., a_n)$. Розв'язавши їх, отримують коефіцієнти полінома $P(x_1, ..., x_n)$.

Приклад 7.18. Побудуємо поліном Жегалкіна для функції $f(x, y) = x \sim y$. Прирівняємо значення функції та полінома на всіх чотирьох наборах значень змінних і одержимо систему рівнянь відносно неозначених коефіцієнтів:

$$f(0, 0) = 1 = c_0,$$
$$f(0, 1) = 0 = c_0 \oplus c_2,$$
$$f(1, 0) = 0 = c_0 \oplus c_1,$$
$$f(1, 1) = 1 = c_0 \oplus c_1 \oplus c_2 \oplus c_3.$$

Розв'язавши її, визначаємо, що $c_0 = 1$, $c_1 = 1$, $c_2 = 1$, $c_3 = 0$ й, отже, $x \sim y = 1 \oplus x \oplus y$.

Побудова полінома Жегалкіна на основі рівносильних перетворень. Спочатку будують рівносильну формулу, у якій є лише операції кон'юнкції та заперечення, а потім всюди замінюють $\bar{x}$ на $1 \oplus x$. Після цього тривіальними перетвореннями отримують поліном Жегалкіна.

Приклад 7.19. Побудуємо поліном Жегалкіна для функції $f(x, y) = x \to y$. Використовуючи введені раніше еквівалентності, одержимо

$$x \to y = \bar{x} \vee y = \overline{x\bar{y}} = 1 \oplus x(1 \oplus y) = 1 \oplus x \oplus xy.$$

Побудова полінома Жегалкіна за ДДНФ булевої функції. Цей спосіб ґрунтується на доведенні теореми 7.5. Його доцільно застосовувати тоді, коли функцію задано ДДНФ або цю форму легко знайти.

Приклад 7.20. Побудуємо таким методом поліном Жегалкіна для функції $f(x, y, z) = xy\bar{z} \vee \bar{x}yz \vee xz$. Спочатку перетворимо цю ДНФ на досконалу:

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

Повторивши доведення теореми 7.5, маємо

$$xy\bar{z} \vee \bar{x}yz \vee xyz \vee x\bar{y}z = xy\bar{z} \oplus \bar{x}yz \oplus xyz \oplus x\bar{y}z =$$
$$= xy(1 \oplus z) \oplus (1 \oplus x)yz \oplus xyz \oplus x(1 \oplus y)z =$$
$$= xy \oplus xyz \oplus yz \oplus xyz \oplus xyz \oplus xz \oplus xyz = xy \oplus yz \oplus xz.$$

Поліном Жегалкіна має цікаву властивість, зручну для знаходження істотних змінних функції.

ТЕОРЕМА 7.6. Усі змінні булевої функції, які входять у її поліном Жегалкіна, істотні.

Доведення. Нехай змінна x_1 уходить у поліном Жегалкіна функції $f(x_1, x_2, ..., x_n)$. Згрупуємо члени, які містять x_1, і винесемо x_1 за дужки:

$$f(x_1, x_2, ..., x_n) = x_1 f_1(x_2, ..., x_n) \oplus f_2(x_2, ..., x_n).$$

Функція $f_1(x_2, ..., x_n) \neq 0$, оскільки в протилежному випадку змінна x_1 не входила б у поліном для функції f (унаслідок єдиності полінома Жегалкіна). Нехай на наборі $(a_2, ..., a_n)$ значення функції f_1 дорівнює 1. Тоді $f(x_1, a_2, ..., a_n) = x_1 \oplus \sigma$, де $\sigma = f_2(a_2, ..., a_n)$. Отже, зміна значення x_1 тоді, коли значення решти змінних задано набором $(a_2, ..., a_n)$, змінює значення функції $f(x_1, x_2, ..., x_n)$. Звідси випливає, що змінна x_1 істотна.

7.4. Повнота й замкненість

Ми розглянули два способи подання булевих функцій — табличний і формульний. Таблиця задає функцію безпосередньо як відповідність між двійковими наборами та значеннями функції на них. Вада табличного способу подання булевих функцій — його громіздкість.

Формула — значно компактніший спосіб подання функції, проте вона задає одну функцію через інші. Тому для довільної системи функцій Q виникає запитання: чи кожну булеву функцію можна подати формулою над Q? У попередньому підрозділі було отримано позитивну відповідь на це запитання для системи $Q_0 = \{\overline{x}, xy, x \vee y\}$. Справді, будь-яку булеву функцію $f \neq 0$ можна подати досконалою ДНФ. Якщо функція f тотожно дорівнює 0, маємо таке її подання через функції системи Q_0: $0 = x\overline{x}$. У цьому підрозділі показано, як розв'язати сформульовану проблему для довільної системи булевих функцій Q.

7.4.1. Функціонально повні системи

Систему булевих функцій Q називають *функціонально повною*, якщо довільну булеву функцію можна подати формулою над Q, тобто вона являє собою суперпозицію функцій із Q.

Приклад 7.21. Наведемо приклади повних систем.

1. Як уже було доведено, система $\{\overline{x}, xy, x \vee y\}$ повна.
2. Система $\{xy, x \oplus y, 1\}$ повна. Справді, якщо функція $f \neq 0$, то подамо її поліномом Жегалкіна; якщо $f = 0$, то $0 = x \oplus x$.

Очевидно, що не кожна система булевих функцій повна. Наприклад, система $\{\overline{x}, 0, 1\}$ неповна: за її допомогою не можна подати функцію двох змінних.

Дослідження повноти одних систем можна звести до дослідження повноти інших.

ТЕОРЕМА 7.7. Нехай задано дві системи булевих функцій Q_1 і Q_2, причому система Q_1 повна, і кожну її функцію можна виразити формулою через функції системи Q_2. Тоді система Q_2 також функціонально повна.

Доведення. Нехай f — довільна булева функція. Оскільки система Q_1 повна, то функцію f можна виразити формулою F_1, яка містить скінченну кількість функцій системи Q_1. Нехай це функції $g_1, g_2, ..., g_k$. За умовою теореми кожну з них функцій можна виразити формулою через скінченну кількість функцій системи Q_2; нехай це функції $h_1, h_2, ..., h_m$. Тому у формулі F_1 ми можемо виключити входження функцій $g_1, g_2, ..., g_k$, замінивши їх формулами, які містять функції $h_1, h_2, ..., h_m$. Одержимо формулу F_2, що містить лише функції $h_1, h_2, ..., h_m$. Отже, ми виразили функцію f як формулу F_2 через функції $h_1, h_2, ..., h_m$ системи Q_2. Позаяк функція f довільна, то система Q_2 функціонально повна.

Приклад 7.22. Розглянемо інші приклади функціонально повних систем.

3. Система $\{\bar{x}, xy\}$ повна, бо система 1 повна та $x \vee y = \overline{\bar{x}\bar{y}}$.
4. Система $\{\bar{x}, x \vee y\}$ повна, бо система 1 повна та $xy = \overline{\bar{x} \vee \bar{y}}$.
5. Система $\{x \mid y\}$ повна, бо система 3 повна та $\bar{x} = x \mid x$, $xy = \overline{x \mid y} = (x \mid y) \mid (x \mid y)$.

7.4.2. Замкнені класи

Множину K булевих функцій називають *замкненим класом*, якщо довільна суперпозиція функцій із K також належить K. Будь-яка система Q булевих функцій породжує замкнений клас, що складається з усіх функцій, які можна одержати суперпозиціями функцій із Q; його називають *замиканням системи функцій* Q. Замикання системи Q позначають $[Q]$. Очевидно, що коли K — замкнений клас, то $[K] = K$, а якщо Q — функціонально повна система, то $[Q] = P_2$.

Існує п'ять найважливіших замкнених класів:

- T_0 — функції, що зберігають 0;
- T_1 — функції, що зберігають 1;
- S — самодвоїсті функції;
- M — монотонні функції;
- L — лінійні функції.

Розглянемо ці класи.

Клас T_0. Булеву функцію $f(x_1, ..., x_n)$ називають *функцією, яка зберігає* 0, якщо $f(0, ..., 0) = 0$.

Наприклад, функції $0, x, xy, x \vee y, x \oplus y$ зберігають 0, тобто належать класу T_0, а функції $1, \bar{x}, x \to y$ — не зберігають, тобто не належать класу T_0. Таблиця значень функції з класу T_0 містить 0 у першому рядку. Отже, у класі T_0 є $2^{2^n-1} = 1/2 \cdot 2^{2^n}$ булевих функцій, що залежать від n змінних $x_1, ..., x_n$.

ТЕОРЕМА 7.8. T_0 — замкнений клас. Інакше кажучи, із функцій, що зберігають 0, суперпозицією можна одержати лише функції, які зберігають 0.

Доведення. Клас T_0 містить тотожну функцію. Отже, досить довести, що функція $F(x_1, ..., x_n) = f(f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n))$ зберігає 0, якщо функції $f, f_1, f_2, ..., f_m$ зберігають 0. Справді, $f(f_1(0, ..., 0), f_2(0, ..., 0), ..., f_m(0, ..., 0)) = f(0, ..., 0) = 0$.

Наслідок. Повна система функцій має містити хоча б одну функцію, яка не зберігає 0. Інакше кажучи, повна система функцій не може цілком міститись у замкненому класі T_0.

Клас T_1. Булеву функцію $f(x_1, ..., x_n)$ називають ***функцією, яка зберігає*** 1, якщо $f(1, ..., 1) = 1$.

Наприклад, функції $1, x, xy, x \vee y, x \to y$ належать класу T_1, а функції $0, \bar{x}, x \oplus y$ — ні.

ТЕОРЕМА 7.9. T_1 — замкнений клас.

Доведення. Клас T_1 містить тотожну функцію. Отже, досить довести, що функція $F(x_1, ..., x_n) = f(f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n))$ зберігає 1, якщо функції $f, f_1, f_2, ..., f_m$ зберігають 1. Справді, $f(f_1(1, ..., 1), f_2(1, ..., 1), ..., f_m(1, ..., 1)) = f(1, ..., 1) = 1$.

Наслідок. Повна система функцій має містити хоча б одну функцію, яка не зберігає 1. Інакше кажучи, повна система функцій не може цілком міститись у замкненому класі T_1.

Легко довести, що клас T_1 має $2^{2^n-1} = 1/2 \cdot 2^{2^n}$ функцій, що залежать від n змінних $x_1, ..., x_n$. Зазначимо, що існують булеві функції, які зберігають і 0, і 1; водночас є функції, які не зберігають ні 0, ні 1. Наприклад, xy, $x \vee y$, $x \oplus y \oplus z \in T_0 \cap T_1$, а $x | y$, $x \downarrow y \notin T_0 \cup T_1$.

Клас S. Нагадаємо означення самодвоїстої функції: функцію називають *самодвоїстою*, якщо вона двоїста до самої себе: $f^* = f$. Наприклад, функції x, $\overline{x}$, $x \oplus y \oplus z$ самодвоїсті, а функції xy, $x \vee y$, $x \to y$ — ні.

Пригадавши означення двоїстої функції, можна дати таке означення самодвоїстої функції, еквівалентне до попереднього. Булеву функцію $f(x_1, ..., x_n)$ називають *самодвоїстою*, якщо вона набуває протилежних значень на протилежних наборах значень змінних: $f(x_1, x_2, ..., x_n) = \overline{f}(\overline{x_1}, \overline{x_2}, ..., \overline{x_n})$. Звідси, зокрема, випливає, що самодвоїсту функцію можна повністю задати її значеннями на першій половині наборів. Отже, кількість самодвоїстих функцій від n змінних $x_1, ..., x_n$ дорівнює кількості двійкових наборів довжиною 2^{n-1}, тобто $2^{2^{n-1}} = \sqrt{2^{2^n}}$.

ТЕОРЕМА 7.10. Клас S самодвоїстих функцій замкнений.

Доведення. Оскільки тотожна функція належить класу S, достатньо довести, що функція $F(x_1, ..., x_n) = f(f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n))$ самодвоїста, якщо функції $f, f_1, f_2, ..., f_m$ самодвоїсті. Застосувавши принцип двоїстості, отримаємо:

$$F^*(x_1, ..., x_n) = f^*(f_1^*(x_1, ..., x_n), ..., f_m^*(x_1, ..., x_n)) =$$
$$= f(f_1(x_1, \ldots, x_n), ..., f_m(x_1, \ldots, x_n)) = F(x_1, \ldots, x_n).$$

Наслідок. Повна система булевих функцій має містити хоча б одну несамодвоїсту функцію.

ЛЕМА 7.1 (про несамодвоїсту функцію). Із несамодвоїстої функції $f(x_1, ..., x_n)$ підстановкою функцій x та $\overline{x}$ можна отримати несамодвоїсту функцію однієї змінної, тобто константу.

Доведення. Нехай $f(x_1, ..., x_n) \notin S$. Отже, існує хоча б один набір $\tilde{a}^n = (a_1, ..., a_n)$ значень змінних такий, що $f(a_1, ..., a_n) = f(\overline{a_1}, ..., \overline{a_n})$. За цим набором означимо допоміжні функції $\varphi_i(x)$, $i = 1, 2, ..., n$:

$$\varphi_i(x) = \begin{cases} x, & \text{якщо } a_i = 0, \\ \overline{x}, & \text{якщо } a_i = 1. \end{cases}$$

Легко переконатись, що ці функції мають властивість $\varphi_i(0) = a_i$, $\varphi_i(1) = \overline{a}_i$ ($i = 1, 2, ..., n$). Розглянемо функцію $\Phi(x) = f(\varphi_1(x), ..., \varphi_n(x))$. Її отримано з функції $f(x_1, ..., x_n)$ підстановкою x та $\overline{x}$. Функція $\Phi(x)$ — константа, бо

$$\Phi(0) = f(\varphi_1(0), ..., \varphi_n(0)) = f(a_1, ..., a_n) =$$
$$= f(\overline{a}_1, ..., \overline{a}_n) = f(\varphi_1(1), ..., \varphi_n(1)) = \Phi(1).$$

Клас M. Уведемо на множині E_2^n усіх n-місних двійкових наборів відношення часткового порядку. Нехай $\tilde{a}^n = (a_1, ..., a_n)$, $\tilde{b}^n = (b_1, ..., b_n)$ — двійкові набори; $\tilde{a}^n \leqslant \tilde{b}^n$, якщо $a_i \leqslant b_i$ для всіх $i = 1, 2, ..., n$. Наприклад, (010) $\leqslant$ (110), а набори (010) і (100) порівняти не можна.

Функцію $f(\tilde{x}^n) = f(x_1, ..., x_n)$ називають *монотонною*, якщо для будь-яких двійкових наборів $\tilde{a}^n$ і $\tilde{b}^n$ із того, що $\tilde{a}^n \leqslant \tilde{b}^n$, випливає, що $f(\tilde{a}^n) \leqslant f(\tilde{b}^n)$. Наприклад, 0, 1, x, xy, $x \vee y$ — монотонні функції, а $\overline{x}$, $x \oplus y$, $x \to y$ — немонотонні.

Щоб перевірити функцію на монотонність безпосередньо за означенням, потрібно проаналізувати таблицю функції, що може виявитися досить громіздкою справою. Проте часто досить легко виявити, що функція немонотонна.

ТЕОРЕМА 7.11. Нехай $f(\tilde{x}^n)$ — монотонна функція. Якщо ця функція не зберігає 0, то вона тотожно дорівнює 1. Якщо функція $f(\tilde{x}^n)$ не зберігає 1, то вона тотожно дорівнює 0.

Доведення. Для будь-якого набору $\tilde{a}^n = (a_1, ..., a_n)$ виконується умова $(0, ..., 0) \leqslant$ $\leqslant (a_1, ..., a_n)$. Далі, $1 = f(0, ..., 0) \leqslant f(a_1, ..., a_n)$. Отже, $f(\tilde{a}^n) = 1$, тобто $f(\tilde{x}^n)$ — константа 1. Друге твердження теореми можна довести аналогічно.

Наслідок. Якщо функція $f \notin T_0 \cap T_1$ — не константа, то вона немонотонна.

Отже, всі монотонні функції, окрім констант 0 і 1, належать $T_0 \cap T_1$. Зауважимо, що існують немонотонні функції, які належать $T_0 \cap T_1$, наприклад $x \oplus y \oplus z$.

ТЕОРЕМА 7.12. Клас M монотонних функцій замкнений.

Доведення. Оскільки тотожна функція належить класу M, то для доведення замкненості M досить довести, функція $F(x_1, ..., x_n) = f(f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n))$ монотонна в разі монотонності функцій $f, f_1, ..., f_m$. Нехай $\tilde{a}^n \leqslant \tilde{b}^n$; тоді $f_i(\tilde{a}^n) \leqslant$ $\leqslant f_i(\tilde{b}^n)$ для всіх $i = 1, 2, ..., n$. Звідси випливає, що

$$F(\tilde{a}^n) = f(f_1(\tilde{a}^n), ..., f_m(\tilde{a}^n)) \leqslant f(f_1(\tilde{b}^n), ..., f_m(\tilde{b}^n)) = F(\tilde{b}^n).$$

Отже, $F(\tilde{a}^n) \leqslant F(\tilde{b}^n)$.

Наслідок. Повна система булевих функцій має містити хоча б одну немонотонну функцію.

ЛЕМА 7.2 (про немонотонну функцію). Із немонотонної функції підстановкою констант 0, 1 і функції x можна отримати функцію $\overline{x}$.

Доведення. Нехай $f(\tilde{x}^n) \notin M$. Отже, існують такі два набори $\tilde{a}_n = (a_1, ..., a_n)$, $\tilde{b}_n = (b_1, ..., b_n)$ значень змінних $x_1, ..., x_n$, що $\tilde{a}^n \leqslant \tilde{b}^n$, але $f(\tilde{a}_n) > f(\tilde{b}_n)$, тобто $f(\tilde{a}_n) = 1$, $f(\tilde{b}_n) = 0$.

Якщо $\tilde{a}_n$ та $\tilde{b}_n$ відрізняються k компонентами, то в цих компонентах у наборі $\tilde{a}_n$ є нулі, а в наборі $\tilde{b}_n$ — одиниці (бо $\tilde{a}^n \leqslant \tilde{b}^n$). Підставимо у функцію $f(\tilde{x}^n)$ замість змінних, яким у наборах $\tilde{a}_n$ та $\tilde{b}_n$ відповідають однакові значення, просто ці значення, а на місце решти k змінних — функцію x. Тоді отримаємо функцію однієї змінної $g(x)$. З урахуванням того, що $f(\tilde{a}_n) = 1$, $f(\tilde{b}_n) = 0$, одержимо $g(0) =$ $= f(\tilde{a}_n) = 1$, $g(1) = f(\tilde{b}_n) = 0$. Отже, $g(x) = \overline{x}$.

Проілюструємо процес отримання функцій $\bar{x}$ на конкретному прикладі. Нехай $f = x \to y$. Очевидно, що $x \to y \notin M$. Далі, $(0, 0) \leqslant (1, 0)$, $0 \to 0 = 1$, $1 \to 0 = 0$. Отже, $g(x) = x \to 0 = \bar{x}$.

Клас L. Булеву функцію $f(\tilde{x}^n) = f(x_1, x_2, ..., x_n)$ називають *лінійною*, якщо її поліном Жегалкіна має вигляд $f(\tilde{x}^n) = c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus ... \oplus c_nx_n$. Це поліном першого степеня. або *лінійний* поліном. Він не має багатомісних кон'юнкцій x_ix_j, $x_ix_jx_k$ тощо. Коефіцієнти c_0, c_1, ..., c_n лінійного полінома можуть утворювати довільний набір значень з $n + 1$ нулів і одиниць. Унаслідок єдиності полінома Жегалкіна різним наборам коефіцієнтів відповідають різні булеві функції. Отже, є 2^{n+1} лінійних булевих функцій від n змінних. Приклади лінійних булевих функцій — 0, 1, x, $\bar{x}$, $x \sim y$, а функції $x \to y$, $x \vee y$ нелінійні.

Неважко переконатись, що серед лінійних функцій самодвоїсті ті, поліном Жегалкіна яких містить непарну кількість змінних, а несамодвоїсті ті, поліном Жегалкіна яких містить парну кількість змінних. Наприклад, функції x, $x \oplus 1$, $x \oplus y \oplus z$, $x \oplus y \oplus z \oplus 1$ самодвоїсті, а функції $x \oplus y$, $x \oplus y \oplus 1$ несамодвоїсті.

Серед лінійних функцій монотонні лише три функції: 0. 1, x.

ТЕОРЕМА 7.13. Клас L лінійних функцій замкнений.

Доведення. Множина L усіх лінійних функцій — замкнений клас, бо підстановка формул вигляду $c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus ... \oplus c_nx_n$ у формулу такого самого вигляду знову дає формулу того самого вигляду.

Наслідок. Повна система булевих функцій має містити хоча б одну нелінійну функцію.

ЛЕМА 7.3 (про нелінійну функцію). Якщо функція $f(\tilde{x}^n)$ нелінійна, то кон'юнкцію двох змінних можна подати як суперпозицію констант 0, 1, заперечення $\bar{x}$ і функції $f(\tilde{x}^n)$.

Доведення. Нехай $f \notin L$. Тоді поліном Жегалкіна функції f містить кон'юнкції змінних. Виберемо серед них кон'юнкцію з найменшим рангом $r \geqslant 2$; нехай це

$$k = x_{i_1}x_{i_2}x_{i_3}...x_{i_r}, \quad r \geqslant 2.$$

Надамо значення $x_{i_3} = ... = x_{i_r} = 1$, а всім змінним x_j, які не входять у кон'юнкцію k, надамо значення $x_j = 0$. Підстановка цих констант у поліном перетворить кон'юнкцію k на $x_{i_1}x_{i_2}$, а решту кон'юнкцій — на 0. При цьому функція f набере вигляду

$$\varphi(x_{i_1}, x_{i_2}) = x_{i_1}x_{i_2} \oplus \alpha x_{i_1} \oplus \beta x_{i_2} \oplus \gamma,$$

де α, β, γ — коефіцієнти, що дорівнюють 0 чи 1 залежно від конкретної функції $f(\tilde{x}^n)$. Розглянемо функцію

$$\begin{aligned} F(x, y) &= \varphi(x \oplus \beta, y \oplus \alpha) = \\ &= (x \oplus \beta)(y \oplus \alpha) \oplus \alpha(x \oplus \beta) \oplus \beta(y \oplus \alpha) \oplus \gamma = \\ &= xy \oplus \alpha x \oplus \beta y \oplus \alpha\beta \oplus \alpha x \oplus \alpha\beta \oplus \beta y \oplus \alpha\beta \oplus \gamma = \\ &= xy \oplus \alpha\beta \oplus \gamma. \end{aligned}$$

Отже, $F(x, y) = xy \oplus \alpha\beta \oplus \gamma$. Цю функцію отримано суперпозицією констант 0 і 1, заперечення $\bar{x} = x \oplus 1$ та функції $f(\tilde{x}^n) \notin L$. Якщо $\alpha\beta \oplus \gamma = 0$, то $F(x, y) = xy$; якщо $\alpha\beta \oplus \gamma = 1$, то $\overline{F}(x, y) = xy$.

7.4.3. Критерій функціональної повноти системи булевих функцій

Розглянемо критерій функціональної повноти — теорему про функціональну повноту, доведену Е. Постом (E. Post) 1921 р.

ТЕОРЕМА 7.14. Для того щоб система булевих функцій Q була функціонально повною, необхідно й достатньо, щоб вона містила:

1) функцію, яка не зберігає 0;
2) функцію, яка не зберігає 1;
3) несамодвоїсту функцію;
4) немонотонну функцію;
5) нелінійну функцію.

Інакше кажучи, для повноти системи Q необхідно й достатньо, щоб для кожного з п'яти замкнених класів T_0, T_1, S, M, L вона містила функцію, яка цьому класу не належить.

Доведення. Необхідність випливає з того, що класи T_0, T_1, S, M, L замкнені (наслідки з теорем 7.8–7.10, 7.12, 7.13).

Достатність. Уведемо для функцій із системи Q такі позначення: f_i — функція, що не зберігає 0; f_j — функція, що не зберігає 1; f_k — несамодвоїста функція; f_m — немонотонна функція; f_l — нелінійна функція.

Доведення достатності проведемо у два етапи.

Перший етап. Одержимо константи 0 і 1. Розглянемо функцію f_i: $f_i(0, ..., 0) = 1$. Можливі два випадки.

1. Якщо $f_i(1, ..., 1) = 1$, то функція $\varphi(x) = f_i(x, ..., x)$ тотожно дорівнює 1, бо $\varphi(0) = f_i(0, ..., 0) = 1$, $\varphi(1) = f_i(1, ..., 1) = 1$. Із функції f_j у цьому разі можна отримати константу 0, оскільки $f_j(\varphi(x), ..., \varphi(x)) = f_j(1, ..., 1) = 0$.
2. Якщо $f_i(1, ..., 1) = 0$. Тоді функція $\varphi(x) = f_i(x, ..., x)$ — заперечення: $\varphi(x) = \bar{x}$, бо $\varphi(0) = f_i(0, ..., 0) = 1$, $\varphi(1) = f_i(1, ..., 1) = 0$. Із несамодвоїстої функції f_k та побудованої функції $\bar{x}$ за лемою 7.1 про несамодвоїсту функцію одержимо константу. Другу константу отримаємо, використавши $\bar{x}$, тому що $\bar{0} = 1$, $\bar{1} = 0$.

Отже, в обох випадках ми одержали константи 0 і 1.

Другий етап. Використовуючи константи 0, 1 та немонотонну функцію f_m, за лемою 7.2 про немонотонну функцію одержимо $\bar{x}$. Потім за допомогою констант 0, 1, функцій $\bar{x}$ і $f_l \notin L$ за лемою 7.3 про нелінійну функцію отримаємо кон'юнкцію двох змінних xy. Отже, через функції системи Q ми виразили $\bar{x}$ та xy. Позаяк система $\{\bar{x}, xy\}$ повна, то за теоремою 7.7 система Q також повна.

Щоб перевірити, чи виконуються для скінченної системи функцій $\{f_1, ..., f_q\}$ умови теореми Поста, складають *таблицю Поста*. Її рядки позначають функціями системи, а стовпці – назвами п'яти основних замкнених класів. У клітках таблиці Поста ставлять знак „+" або „–" залежно від того, чи належить функція відповідному замкненому класу. Для повноти системи функцій необхідно й достатньо, щоб у кожному стовпці таблиці Поста стояв хоча б один знак „–".

Приклад 7.23. Дослідимо, чи функціонально повна система $\{x \to y, 0\}$. Побудувавши таблицю Поста (табл. 7.6), переконуємось, що система функціонально повна.

Таблиця 7.6

	T_0	T_1	S	M	L
$x \to y$	–	+	–	–	–
0	+	–	–	+	+

Мінімальну повну систему функцій (тобто таку, вилучення з якої довільної функції робить систему неповною) називають *базисом*. Із теореми Поста випливає, що базис не може містити більше п'яти функцій. Насправді можна довести точніше твердження.

ТЕОРЕМА 7.15. Максимальна кількість функцій базису дорівнює 4.

Доведення. Функція $f_t \notin T_0$ або несамодвоїста, тобто $f_t(0, ..., 0) = f_t(1, ..., 1) = 1$ (випадок 1 доведення теореми 7.14), або не зберігає 1 та немонотонна (випадок 2 доведення теореми 7.14). Отже, повна або система $\{f_t, f_j, f_m, f_l\}$, або $\{f_t, f_k, f_l\}$. Константу 4 у цій теоремі зменшити неможливо. Система з чотирьох функцій $\{f_1, f_2, f_3, f_4\}$, де $f_1 = 0$, $f_2 = 1$, $f_3 = xy$, $f_4 = x \oplus y \oplus z$ повна. Із неї не можна вилучити жодної функції без порушення повноти. Справді, f_1 – єдина функція цієї системи, що не зберігає 1, f_2 – єдина функція, що не зберігає 0, f_3 – єдина нелінійна функція, f_4 – єдина немонотонна. Несамодвоїсті тут три функції: f_1, f_2, f_3.

7.4.4. Послаблена функціональна повнота

Систему булевих функцій Q називають *послаблено функціонально повною*, якщо будь-яку булеву функцію можна подати формулою над системою $Q \cup \{0, 1\}$, тобто суперпозицією констант 0, 1 і функцій із системи Q.

ТЕОРЕМА 7.16. Для того щоб система булевих функцій була послаблено функціонально повною, необхідно й достатньо, щоб вона містила хоча б одну нелінійну та хоча б одну немонотонну функцію.

Доведення. Якщо є константи 0 і 1, ми маємо функцію, що не зберігає 0 (функцію 1), функцію, що не зберігає 1 (функцію 0) і несамодвоїсту функцію (обидві функції 0 і 1). Водночас константи 0 і 1, очевидно, лінійні та монотонні. Застосування теореми Поста завершує доведення.

7.5. Мінімізація булевих функцій

Мінімізацією булевої функції називають відшукання найпростішого її подання у вигляді суперпозиції функцій якоїсь функціонально повної системи.

7.5.1. Головні результати

Розглянемо лише спрощення ДНФ, тому що за принципом двоїстості з методів спрощення ДНФ можна отримати методи спрощення КНФ.

Мінімальною ДНФ булевої функції називають її ДНФ, що складається з найменшої можливої кількості букв. При цьому кожну букву враховують стільки разів, скільки вона зустрічається в ДНФ. Наприклад, ДНФ $\overline{x}y \vee x\overline{y} \vee xz$ складається з шести букв.

Елементарну кон'юнкцію $k = x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} \dots x_{i_r}^{\sigma_r}$ називають *імплікантою* булевої функції $f(x_1, \dots, x_n)$, якщо на довільному наборі значень змінних, на якому $k = 1$, значення функції f також дорівнює 1. Інакше кажучи, k – імпліканта функції f, якщо функція $k \to f$ тотожно дорівнює 1 (тобто виключено можливість $k = 1$, $f = 0$). Тут $x_{i_1}, x_{i_2}, \dots, x_{i_r}$ – якісь зі змінних $x_1, \dots, x_n$.

Приклад 7.24. Елементарна кон'юнкція $k = x\overline{y}\,\overline{z}$ – імпліканта функції

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

бо в разі $k = 1$ значення функції f дорівнює 1.

Елементарну кон'юнкцію k називають *простою імплікантою* булевої функції f, якщо k – імпліканта функції f, а елементарна кон'юнкція, одержана з k вилученням довільної букви, – не імпліканта. Диз'юнктивну нормальну форму, що складається з усіх простих імплікант булевої функції, називають її *скороченою диз'юнктивною нормальною формою* (*СДНФ*).

ТЕОРЕМА 7.17. СДНФ $D_{\text{скор}}$ булевої функції f задає цю функцію, тобто $f = D_{\text{скор}}$.

Доведення. Якщо $f(\tilde{x}^n) = 0$, то очевидно, що функція f не має жодної простої імпліканти. Нехай тепер $f(\tilde{x}^n) \neq 0$. Розглянемо такий довільний набір $\tilde{a}^n = (a_1, \dots, a_n)$, що $f(\tilde{a}^n) = 1$. Елементарна кон'юнкція $k = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$, яка входить у ДДНФ функції f, – імпліканта цієї функції. Якщо імпліканта k не проста, то можна вилучити з неї хоча б одну букву так, щоб одержана елементарна кон'юнкція k_1 була імплікантою. Якщо імпліканта k_1 знову не проста, то вилучимо ще одну букву так, що отримана елементарна кон'юнкція k_2 буде імплікантою функції f. Продовжуючи цей процес, за скінченну кількість кроків ми одержимо просту імпліканту k'. За побудовою імпліканта k' набуває на наборі $\tilde{a}^n$ значення 1. Оскільки формула $D_{\text{скор}}$ складається з усіх простих імплікант функції f, то вона має диз'юнктивним членом імпліканту k', і формула $D_{\text{скор}}$ набуває значення 1 на наборі $\tilde{a}^n$.

З іншого боку, нехай формула $D_{\text{скор}}$ на якомусь наборі $\tilde{b}^n$ набуває значення 1. Тоді якась проста імпліканта k_0 з $D_{\text{скор}}$ на цьому наборі дорівнює 1. Оскільки k_0 – імпліканта функції f, то $f(\tilde{b}^n) = 1$.

Отже, значення функції f і формули $D_{скор}$ на будь-якому наборі значень змінних збігаються.

Зазначимо, що СДНФ булевої функції f єдина, бо множина всіх простих імплікант булевої функції визначається однозначно (адже СДНФ являє собою диз'юнкцію їх усіх).

Зв'язок між мінімальною та скороченою ДНФ виражає така теорема.

ТЕОРЕМА 7.18. Мінімальну ДНФ булевої функції f можна одержати з її СДНФ вилученням деяких елементарних кон'юнкцій.

Доведення. Потрібно довести, що мінімальна ДНФ D_{min} довільної булевої функції f являє собою диз'юнкцію її простих імплікант (можливо, не всіх). Припустимо, що імпліканта k_1 із D_{min} не проста. Тоді з неї можна вилучити хоча б одну букву так, щоб отримана елементарна кон'юнкція k_2 також була імплікантою функції f. Окрім того, імпліканта k_2 набуває значення 1 на всіх тих наборах, на яких набуває значення 1 імпліканта k_1. Отже, у ДНФ D_{min} імпліканту k_1 можна замінити на k_2 й отримати ДНФ D_*, яка також задає функцію f. За побудовою формула D_* має менше букв, ніж D_{min}. Одержали суперечність.

Диз'юнктивну нормальну форму $D_{туп}$, яка задає функцію f (тобто $f = D_{туп}$), називають ***тупиковою ДНФ*** цієї функції, якщо:

а) кожна елементарна кон'юнкція з $D_{туп}$ – проста імпліканта функції f;

б) вилучення з формули $D_{туп}$ довільного диз'юнктивного члена призводить до ДНФ D_1, яка не задає функцію f, тобто $f \neq D_1$.

ТЕОРЕМА 7.19. Мінімальна ДНФ булевої функції являє собою її тупикову ДНФ.

Доведення цієї теореми випливає безпосередньо з означень мінімальної та тупикової ДНФ.

Існують тупикові, але не мінімальні ДНФ; одна й та сама булева функція f може мати декілька різних мінімальних ДНФ.

Із теорем 7.18 і 7.19 випливає, що відшукання мінімальних ДНФ можна поділити на два етапи:

1. Побудова скороченої ДНФ.
2. Побудова всіх тупикових ДНФ і вибір із них мінімальних.

7.5.2. Методи побудови скороченої ДНФ

Один із методів знаходження СДНФ булевої функції запропонував 1952 р. Куайн (W. Quine). Згідно з цим методом до ДДНФ булевої функції послідовно застосовують такі рівносильності:

$$ku \vee k\bar{u} = k \vee ku \vee k\bar{u} \text{ (неповне склеювання)},$$
$$ku \vee k = k \text{ (поглинання члена } ku\text{)},$$

де k – елементарна кон'юнкція, u – змінна. Говорять, що члени ku та $k\bar{u}$ *склеюються* по змінній u та в результаті дають k. Склеювання називають *неповним*, оскільки члени ku та $k\bar{u}$ залишаються в правій частині.

Алгоритм Куайна

Наведемо кроки алгоритму.

Крок 1. Булеву функцію $f(x_1, ..., x_n)$, яку потрібно мінімізувати, записати в ДДНФ; позначити її f_0. Виконати $i := 0$.

Крок 2. Якщо до ДНФ f_i не можна застосувати жодного неповного склеювання, то зупинитись: f_i – СДНФ. Інакше на основі ДНФ f_i побудувати ДНФ f_{i+1} за таким правилом: у формі f_i виконати всі неповні склеювання, які можна застосувати до елементарних кон'юнкцій із рангом $n-i$, а потім вилучити всі елементарні кон'юнкції з рангом $n-i$, до яких можна застосувати поглинання.

Крок 3. Виконати $i := i + 1$ і перейти до кроку 2.

Отже, в алгоритмі Куайна, починаючи з ДДНФ f_0 будують послідовність ДНФ $f_0, f_1, f_2, ...$ доти, доки не отримають СДНФ.

ТЕОРЕМА 7.20. Для будь-якої булевої функції f результат застосування алгоритму Куайна до ДДНФ цієї функції – її СДНФ.

Приклад 7.25. Побудуємо СДНФ булевої функції, яку задано досконалою ДНФ f_0:

$$f_0 = \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}\,\bar{z} \vee x\bar{y}z \vee xyz.$$

Виконаємо $i := 0$. Застосувавши неповне склеювання до членів 1 і 2, 2 та 5, 3 та 4, 4 та 5, одержимо

$$f_0' = \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}\,\bar{z} \vee x\bar{y}z \vee xyz \vee \bar{x}y \vee yz \vee x\bar{y} \vee xz.$$

Після п'ятикратного застосування поглинання отримаємо $f_1 = \bar{x}y \vee yz \vee x\bar{y} \vee xz$.
Виконаємо $i := 1$. Оскільки жодне неповне склеювання не застосовне до функції f_1, то ми маємо кінцевий результат, й f_1 — СДНФ.

Мак-Класкі (E. McCluskey) удосконалив алгоритм Куайна [5].

Алгоритм Мак-Класкі

Наведемо кроки алгоритму.

Крок 1. Записати булеву функцію, яку потрібно мінімізувати, у ДДНФ.

Крок 2. Упорядкувати змінні й записати їх у кожній елементарній кон'юнкції у вибраному порядку. Після цього подати кожну елементарну кон'юнкцію послідовністю з 1, 0 і — (рисок): на i-й позиції записати 1, якщо i-та змінна входить до елементарної кон'юнкції без заперечення, 0 — якщо вона входить із запереченням, і риску, якщо зовсім не входить. Наприклад, елементарні кон'юнкції xyz, $x\bar{z}$, $x\bar{u}$ записують відповідно у вигляді 111–, 1–0–, 1—0.

Крок 3. Розбити двійкові вирази, які відповідають елементарним кон'юнкціям, на класи за кількістю одиниць і розмістити списки цих класів за зростанням кількості одиниць. Наприклад, для ДДНФ $f_0 = \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}\bar{z} \vee x\bar{y}z \vee$ $\vee xyz$ із прикладу 7.25 отримаємо список із п'яти елементів, розбитих на три класи:

$$\begin{array}{c} 010 \\ 100 \\ \hline 011 \\ 101 \\ \hline 111 \end{array}$$

Крок 4. Виконати всі можливі склеювання $ku \vee k\bar{u} = k$. Їх можна застосовувати лише до тих елементів списку, які містяться в сусідніх класах. Склеювані елементи знаходять у сусідніх класах простим порівнянням: ці елементи мають різнитися точно однією позицією, і в цій позиції має бути не риска (–).

Крок 4 повторюють доти, доки можна застосувати склеювання. Якщо помістити до одного класу всі імпліканти, отримані з двох сусідніх класів, то на черговому повторенні кроку 4 нам знову доведеться порівнювати лише елементи із сусідніх класів. Склеювані елементи позначають *; надалі вони не увійдуть у список простих імплікант. Попередній список після опрацювання має такий вигляд:

$$\begin{array}{c} {}^{*}010 \\ {}^{*}100 \\ \hline {}^{*}011 \\ {}^{*}101 \\ \hline {}^{*}111 \end{array} \qquad \begin{array}{c} 01- \\ 10- \\ \hline -11 \\ 1-1 \end{array}$$

Далі неможливо застосувати склеювання. Непозначеними залишилися чотири елементи: 01–; 10–; –11; 1–1. Отже, множина всіх простих імплікант функції f_0 така: $\{\bar{x}y, x\bar{y}, yz, xz\}$, а її СДНФ — $\bar{x}y \vee x\bar{y} \vee yz \vee xz$.

Метод Блейка. У розглянутих алгоритмах Куайна та Мак-Класкі відшукання СДНФ починається з ДДНФ. Проте часто доводиться будувати СДНФ функції, заданої довільною ДНФ (а не ДДНФ). Тоді доцільно застосувати метод Блейка (A. Blake).

Цей метод ґрунтується на використанні рівносильності *узагальненого склеювання*:

$$AC \vee B\bar{C} = AC \vee B\bar{C} \vee AB,$$

де A, B, C — довільні формули. Якщо в цій рівносильності $AB \neq 0$, то говорять, що до членів AC та $B\bar{C}$ можна застосувати *нетривіальне* узагальнене склеювання.

Доведемо рівносильність узагальненого склеювання, використовуючи закони алгебри Буля:

$$AC \vee B\bar{C} = AC \vee ABC \vee B\bar{C} \vee AB\bar{C} =$$
$$= AC \vee B\bar{C} \vee AB(C \vee \bar{C}) = AC \vee B\bar{C} \vee AB.$$

ТЕОРЕМА 7.21. Виконавши в будь-якій ДНФ булевої функції f усі можливі узагальнені склеювання, а потім – усі поглинання, одержимо СДНФ цієї функції.

Доведення ґрунтується на тому, що після багатократного застосування рівносильності узагальненого склеювання до довільної ДНФ булевої функції можна отримати будь-яку її просту імпліканту.

Метод Блейка полягає в тому, що в довільній ДНФ заданої булевої функції спочатку виконують усі допустимі узагальнені склеювання (причому одержані внаслідок узагальнених склеювань члени беруть участь у нових узагальнених склеюваннях). Після цього виконують поглинання, тобто вилучають диз'юнктивні члени у вигляді *AB*, якщо є диз'юнктивні члени *A* чи *B*. У результаті отримаємо СДНФ.

Приклад 7.26. Знайдемо методом Блейка СДНФ булевої функції $f = x\overline{y} \vee \overline{x}yz \vee yz$. Очевидно, що перший і другий члени формули f можна піддати узагальненим склеюванням як по x, так і по y. Але члени, які виникають унаслідок цих склеювань, дорівнюють нулю. Нетривіальне узагальнене склеювання тут можливе лише для першого та третього членів формули f. Застосувавши його, одержимо ДНФ $f_1 = x\overline{y} \vee \overline{x}yz \vee yz \vee xz$.

У цій формі нетривіальне узагальнене склеювання можна застосувати до першого та третього, а також до другого й четвертого членів, проте обидва ці склеювання дають члени, які вже є у формі f_1. Отже, можна вважати, що у формі f_1 виконано всі можливі в ній узагальнені склеювання. Виконавши елементарне поглинання (член $\overline{x}yz$ поглинається членом yz), отримаємо СДНФ $f_2 = x\overline{y} \vee yz \vee xz$.

Іще один метод побудови СДНФ розробив Нельсон (R. Nelson). Ним зручно користуватись, коли функцію задано довільною КНФ.

Метод Нельсона. Цей метод обґрунтовує така теорема.

ТЕОРЕМА 7.22. Розкривши в будь-якій КНФ булевої функції всі дужки відповідно до дистрибутивного закону й виконавши всі поглинання, одержимо СДНФ цієї функції.

Для доведення достатньо переконатись, що, розкривши дужки в довільній КНФ булевої функції f, можна отримати будь-яку наперед задану просту імпліканту k цієї функції.

Приклад 7.27. Розглянемо булеву функцію f, задану КНФ $f = (x \vee \overline{y})(\overline{x} \vee z)(x \vee y \vee \overline{z})$. Розкривши дужки, отримаємо $f = xz \vee xzy \vee \overline{x}\,\overline{y}\,\overline{z} \vee x\overline{y}z$. Виконавши поглинання, одержимо СДНФ булевої функції f:

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

Отже, ми описали методи Куайна, Мак-Класкі, Блейка та Нельсона. Їх застосовують для отримання СДНФ булевої функції – першого етапу мінімізації.

7.5.3. Побудова тупикових ДНФ

На другому етапі мінімізації знаходять усі тупикові ДНФ, із яких вибирають мінімальні. Основний апарат для виконання другого етапу – ***імплікантна таблиця*** булевої функції. Це прямокутна таблиця, рядки якої позначено різними простими імплікантами функції f, а стовпці — наборами значень змінних (або відповідними їм конституентами одиниці), на яких функція набуває значення 1. Якщо якась проста імпліканта k_p перетворюється в 1 на наборі $\tilde{a}^n$, котрий позначає якийсь стовпець імплікантної таблиці, то на перетині рядка, позначеного k_p, і стовпця, позначеного набором $\tilde{a}^n$ (або відповідною йому конституентою одиниці k), ставлять зірочку. Тоді говорять, що проста імпліканта *накриває* одиницю булевої функції. Якщо стовпці імплікантної таблиці позначено конституентами одиниці, то очевидно, що таблицю слід заповнювати за таким правилом: на перетині рядка k_p та стовпця k імплікантної таблиці тоді й лише тоді ставлять зірочку, коли імпліканта k_p становить частину конституенти k (можливо, збігається з нею).

Вище методом Куайна для функції $f = \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}\bar{z} \vee x\bar{y}z \vee xyz$ ми знайшли СДНФ $f = \bar{x}y \vee x\bar{y} \vee xz \vee yz$. Ця функція має чотири прості імпліканти $\bar{x}y$, $x\bar{y}$, xz, yz і набуває значення 1 на п'яти наборах (010), (011), (100), (101), (111), котрим відповідають конституенти одиниці $\bar{x}y\bar{z}$, $\bar{x}yz$, $x\bar{y}\bar{z}$, $x\bar{y}z$, xyz. За сформульованим правилом будуємо імплікантну таблицю для цієї функції (табл. 7.7).

Таблиця 7.7

k_p \ k	$\bar{x}y\bar{z}$	$\bar{x}yz$	$x\bar{y}\bar{z}$	$x\bar{y}z$	xyz
$\bar{x}y$	*	*			
$x\bar{y}$			*	*	
xz				*	*
yz		*			*

Тупикові ДНФ можна будувати безпосередньо за імплікантою таблицею. Якщо в стовпці лише одна зірочка, то просту імпліканту, яка позначає рядок із цією зірочкою, потрібно вибрати обов'язково. Множину таких простих імплікант називають *ядром* булевої функції. У розглянутому прикладі ядро утворюють прості імпліканти $\bar{x}y$ та $x\bar{y}$. Імпліканти ядра входять у будь-яку тупикову ДНФ, але вони можуть накривати лише частину одиниць булевої функції. Стосовно імплікантної таблиці зручно говорити, що прості імпліканти *накривають конституенти*, які відповідають цим одиницям. Вилучимо з імплікантної таблиці стовпці, що мають зірочки на перетині з рядками, позначеними імплікантами ядра. Після цього методом перебору можна знайти мінімальні системи простих імплікант, що накривають решту конституент одиниці. Таким способом ми знайдемо всі тупикові ДНФ, із яких потрібно вибирати мінімальні. У нашому прикладі єдина конституента, що лишається не накритою імплікантами ядра, — це xyz. Її може накрити як імпліканта xz, так і імпліканта yz. Отримаємо дві тупикові ДНФ: $f_1 = \bar{x}y \vee x\bar{y} \vee xz$ і $f_2 = \bar{x}y \vee x\bar{y} \vee yz$. Обидві вони мінімальні (мають по шість букв).

Зазначимо, що метод перебору практично можна застосовувати лише для відносно простих імплікантних таблиць, а також тоді, коли потрібно знайти не всі, а тільки одну тупикову ДНФ. Для відшукання всіх тупикових ДНФ у разі складних таблиць можна застосувати метод, запропонований 1956 р. Петріком (S. Petrick).

Опишемо коротко цей метод [11].

Крок 1. Прості імпліканти позначають великими латинськими буквами. Для табл. 7.7 це має такий вигляд:

$$A - \bar{x}y; \quad B - x\bar{y}; \quad C - xz; \quad D - yz.$$

Крок 2. Для кожного стовпця імплікантної таблиці будують диз'юнкцію букв, які відповідають рядкам із зірочками в цьому стовпці.

Крок 3. Записують кон'юнкцію отриманих диз'юнкцій. Для табл. 7.7 одержимо вираз $A(A \vee D)\, B(B \vee C)(C \vee D)$. Його називають *кон'юнктивним поданням* імплікантної таблиці.

Крок 4. В отриманому на кроці 3 кон'юнктивному поданні імплікантної таблиці розкривають усі дужки за дистрибутивним законом. Одержаний вираз називають *диз'юнктивним поданням* імплікантної таблиці.

Крок 5. До отриманого на кроці 4 диз'юнктивного подання імплікантної таблиці застосовують усі можливі поглинання $A \vee AB = A$ та усувають усі повторення $AA = A$, $A \vee A = A$. Отриманий вираз називають *зведеним диз'юнктивним поданням* імплікантної таблиці. У ході його одержання з кон'юнктивного подання можна застосовувати різні перетворення згідно з тотожностями булевої алгебри, які не містять заперечень, іще до отримання звичайного (не зведеного) диз'юнктивного подання, тобто кроки 4 та 5 можна об'єднати й одразу шукати зведене диз'юнктивне подання імплікантної таблиці.

Крок 6. Прості імпліканти, позначення яких уходять у будь-який фіксований диз'юнктивний член зведеного диз'юнктивного подання імплікантної таблиці, утворюють тупикову ДНФ. Щоб отримати всі тупикові ДНФ, потрібно розглянути всі диз'юнктивні члени цього подання. Продовжуючи розгляд табл. 7.7, можемо записати:

$$A(A \vee D)\, B(B \vee C)(C \vee D) = AB(C \vee D) = ABC \vee ABD.$$

Очевидно, що кон'юнкції ABC відповідає тупикова ДНФ $\bar{x}y \vee x\bar{y} \vee xz$, а кон'юнкції ABD – $\bar{x}y \vee x\bar{y} \vee yz$.

7.5.4. Властивості скороченої ДНФ

Розглянемо важливі властивості СДНФ.

ТЕОРЕМА 7.23. Якщо СДНФ булевої функції не містить жодної букви, яка входить до неї водночас із запереченням і без заперечення, то вона являє собою мінімальну ДНФ.

Доведення. До ДНФ f, яка не має жодної букви водночас із запереченням і без заперечення, не можна застосувати рівносильність узагальненого склеювання. Це саме стосується диз'юнкції довільної кількості членів цієї форми. Якщо f — СДНФ, то її можна відновити за допомогою узагальненого склеювання з мінімальної ДНФ, що являє собою якусь частину f. Отже, у цьому разі мінімальна ДНФ збігається з СДНФ.

ТЕОРЕМА 7.24. Скорочена ДНФ монотонної функції f не містить заперечень змінних.

Доведення. Нехай k_p – проста імпліканта функції f, що містить заперечення змінних. Позначимо як k кон'юнкцію всіх змінних, що входять в імпліканту k_p без заперечень. Побудуємо набір $\tilde{a}^n = (a_1, ..., a_n)$ значень змінних так: усім змінним, які входять у кон'юнкцію k, надамо значення 1, а решті змінних – значення 0. Цілком очевидно, що імпліканта k_p, а, отже, і функція f має на цьому наборі значення 1. З іншого боку, кон'юнкція k перетворюється в 1 лише на наборах $\tilde{b}^n \geqslant \tilde{a}^n$ (зокрема, і на наборі $\tilde{a}^n$). Але на всіх таких наборах $\tilde{b}^n$ функція f, за означенням монотонності, також дорівнює 1. Отже, k – імпліканта функції f, а імпліканта k_p не проста. Отримали суперечність.

Наслідок. Скорочена ДНФ монотонної функції являє собою водночас і її мінімальну ДНФ.

7.5.5. Метод карт Карно побудови мінімальних ДНФ

Для відшукання мінімальних ДНФ функцій невеликої кількості змінних (не більше шести) можна застосувати метод карт Карно [52]. Цей візуальний метод запропонував 1953 р. Карно (M. Karnaugh), виходячи з дещо раніше опублікованої праці Вейча (E. Veitch). Розглянемо метод карт Карно для функцій трьох і чотирьох змінних. Карта Карно для функції трьох змінних складається з $2^3 = 8$ комірок. Кожному з наборів значень аргументів відповідає одна комірка. Кожна комірка зображає відповідну конституенту одиниці (рис. 7.1).

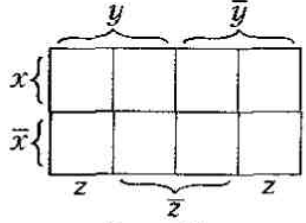

Рис. 7.1

Якщо на певному наборі значень аргументів функція дорівнює 1, то у відповідній комірці записують 1.

Приклад 7.28. Побудуємо карту Карно для функції

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

На рис. 7.2 зображено комірки, які відповідають конституентам одиниці функції f, а на рис. 7.3 – заповнену карту.

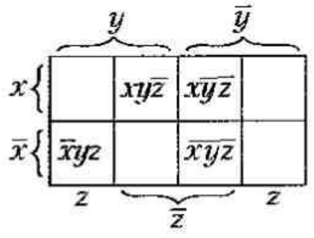

Рис. 7.2

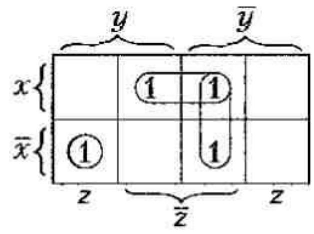

Рис. 7.3

Рівносильні склеювання $ku \vee k\bar u = k$ можна застосувати тільки до конституент (у загалі кажучи, до елементарних кон'юнкцій) зі змінними, у яких усі степені, окрім однієї, збігаються. Наприклад, $xy\bar z \vee x\bar y\bar z$ можна склеїти, оскільки степені x і z збігаються (x^1 та z^0), а степені змінної y різні. Такі конституенти називаються *сусідніми*.

Уважають, що в карті Карно для трьох змінних ліва та права межі тотожні (карту згорнуто в циліндр). Тоді всі сусідні конституенти мають на карті сусідні комірки. Блоку з двох сусідніх комірок відповідає елементарна кон'юнкція, що являє собою спільну частину двох конституент і містить на одну букву менше. Прямокутному блоку з чотирьох сусідніх комірок відповідає елементарна кон'юнкція, що являє собою спільну частину відповідних чотирьох конституент і містить на дві букви менше. Тому *об'єднувати можна прямокутні блоки на карті Карно, які містять сусідні дві, чотири, або – у загальному випадку – 2^i одиниць.*

Відшукання мінімальної ДНФ функції за допомогою карти Карно здійснюють так. Знаходять найбільші блоки на карті, і накривають усі одиниці найменшою кількістю блоків, першими використовуючи найбільші блоки. Можливо, це можна зробити не одним способом.

Приклад 7.28 (*закінчення*). На рис. 7.3 зображено покриття одиниць карти Карно блоками. Отримаємо мінімальну ДНФ $f_{\min} = x\bar z \vee \bar y\bar z \vee \bar x yz$.

Приклад 7.29. Розглянемо функцію

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

Карту Карно для неї зображено на рис. 7.4. Блоку з чотирьох одиниць відповідає елементарна кон'юнкція $\bar y$ із рангом 1, а блоку з двох одиниць – елементарна кон'юнкція $\bar x z$ із рангом 2. Отже, $f_{\min} = \bar y \vee \bar x z$.

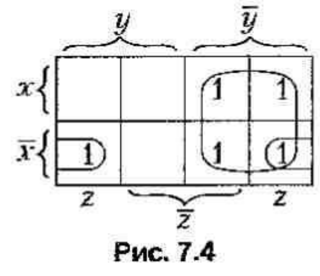

Рис. 7.4

Приклад 7.30. Карту Карно для функції

$$f(x,y,z) = xyz \vee xy\overline{z} \vee x\overline{y}z \vee x\overline{y}\,\overline{z} \vee \overline{x}yz \vee \overline{x}\,\overline{y}z \vee \overline{x}\,\overline{y}\,\overline{z}$$

зображено на рис. 7.5. Мінімальна ДНФ цієї функції $f_{\min} = x \vee z \vee \overline{y}$.

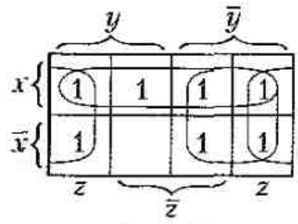

Рис. 7.5

У карті Карно для чотирьох змінних ототожнюють ліву та праву, а також верхню та нижню сторони (рис. 7.6).

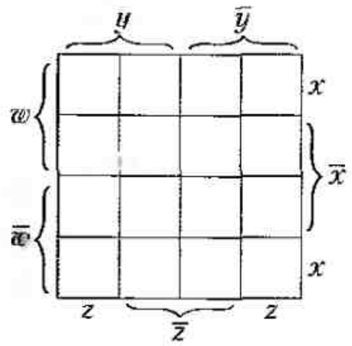

Рис. 7.6

Приклад 7.31. Знайти мінімальну ДНФ для функції

$$f(w,x,y,z) = wxy\overline{z} \vee wx\overline{y}\,\overline{z} \vee w\overline{x}yz \vee w\overline{x}y\overline{z} \vee w\overline{x}\,\overline{y}\,\overline{z} \vee$$
$$\vee \overline{w}xyz \vee \overline{w}xy\overline{z} \vee \overline{w}x\overline{y}\,\overline{z} \vee \overline{w}x\overline{y}z \vee \overline{w}\,\overline{x}y\overline{z} \vee \overline{w}\,\overline{x}\,\overline{y}\,\overline{z}.$$

Карту Карно для неї зображено на рис. 7.7. Отже,

$$f_{\min} = \overline{z} \vee \overline{w}x \vee w\overline{x}y.$$

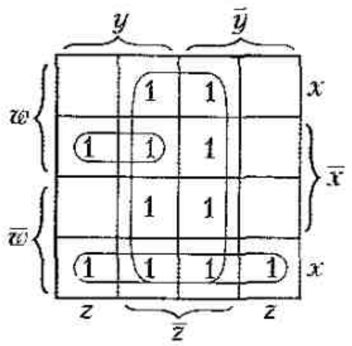

Рис. 7.7

У цьому прикладі можна сформувати блоки й інакше. Тоді одержимо іншу мінімальну ДНФ.

Інколи доводиться працювати з булевими функціями, заданими не для всіх наборів значень аргументів. Тоді в карті Карно використовують позначку d для тих комірок, які відповідають наборам із невідомим значенням булевої функції (на цих наборах функцію можна довизначити довільно). Тепер під час формування найбільших блоків можна вважати, що в деяких (або всіх) комірках із літерою d містяться одиниці.

Приклад 7.32. Задати булеву функцію чотирьох змінних, визначену на наборах, які відповідають двійковому коду десяткових цифр 0, 1, 2, ..., 9, диз'юнктивною нормальною формою, що містить найменшу кількість букв. Значення функції дорівнює 1, якщо набір відповідає цифрам, більшим або рівним 5, і дорівнює 0, якщо набір відповідає цифрам, меншим ніж 5. Шукану функцію $F(w, x, y, z)$ задано табл. 7.8. Відповідну карту Карно зображено на рис. 7.8. Отже, $F(w, x, y, z) = w \vee xy \vee xz$.

Таблиця 7.8

Цифра	w	x	y	z	$F(w,x,y,z)$
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1

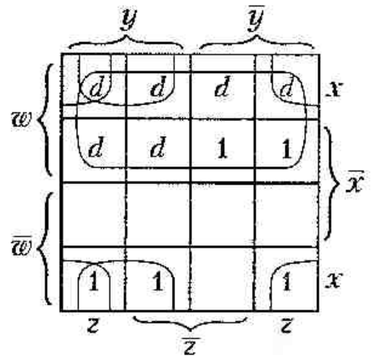

Рис. 7.8

7.6. Реалізація булевих функцій схемами з функціональних елементів

Під *функціональним елементом* розуміють пристрій із такими властивостями: він має $n \geqslant 1$ впорядкованих відростків зверху — *входів* — і один відросток знизу — *вихід*; на входи цього пристрою можуть подаватися сигнали, які мають значення 0 і 1; на кожному наборі сигналів на входах у той самий момент, коли вони надійшли, пристрій видає на виході один із сигналів (0 або 1); набір сигналів на входах однозначно задає сигнал на виході, тобто якщо в різні моменти на входи надійшли однакові набори сигналів, то в ці моменти на виході буде один і той самий сигнал. Кожному функціональному елементу з n входами співвідносять булеву функцію від n змінних $f(\tilde{x}^n)$ таким способом. Входу з номером i $(1 \leqslant i \leqslant n)$ ставлять у відповідність змінну x_i та з кожним набором $\tilde{a}^n = (a_1, ..., a_n)$ значень змінних співвідносять число $f(\tilde{a}^n)$, яке дорівнює 0 чи 1 залежно від сигналу на виході в разі подання цього набору сигналів на входи функціонального елемента. Щодо функції $f(\tilde{x}^n)$ говорять, що даний функціональний елемент *реалізує* її.

Далі ми розглядатимемо лише функціональні елементи, зображені на рис. 7.9, які реалізують відповідно булеві функції заперечення, кон'юнкцію та диз'юнкцію.

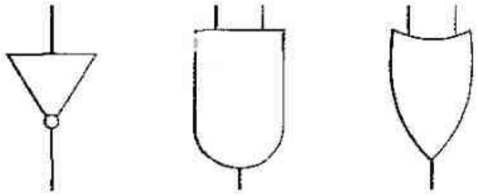

Рис. 7.9

Тепер означимо поняття схеми з функціональних елементів і її входи й вихід.

1. Кожний функціональний елемент являє собою схему з функціональних елементів із тими самими входами та виходом, що й у цього елемента.
2. Якщо S_1 – схема з функціональних елементів і два її входи з'єднано (рис. 7.10), то одержана конструкція S являє собою схему з функціональних елементів. Входи схеми S – усі не з'єднані входи S_1 і ще один вхід, який відповідає двом з'єднаним входам схеми S_1.

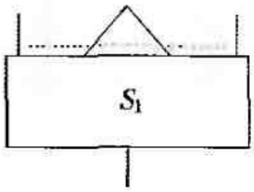

Рис. 7.10

3. Якщо S_1 і S_2 – дві схеми з функціональних елементів, то конструкція S, яку одержано з'єднанням якогось входу схеми S_2 з виходом схеми S_1, також являє собою схему з функціональних елементів (рис. 7.11). Входи схеми S – усі входи схеми S_1 і всі входи схеми S_2, окрім того, який з'єднано з виходом схеми S_1. Вихід схеми S – вихід схеми S_2.

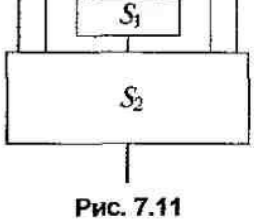

Рис. 7.11

4. Якщо в схемі з функціональних елементів S_1 її вхід з'єднати з виходом якогось функціонального елемента схеми S_1 без утворення циклу для жодного функціонального елемента (тобто вихід жодного функціонального елемента не має бути з'єднано з його ж входом – можливо, через інші елементи схеми S_1), то отримана конструкція являє собою схему з функціональних елементів (приклад такої конструкції наведено на рис 7.12).

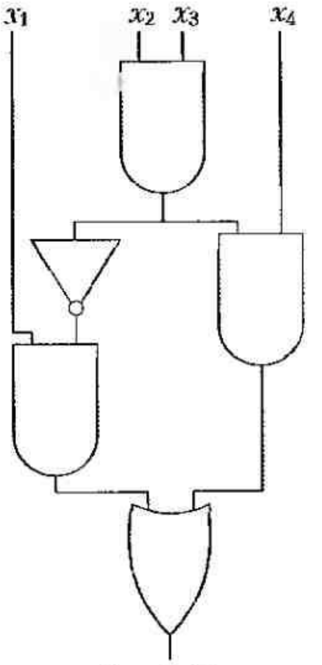

Рис. 7.12

Означення схеми з функціональних елементів завершено. Згідно з ним схема з функціональних елементів – це математичний об'єкт. Вона має різні технічні застосування.

Очевидно, що схема з функціональних елементів реалізує певну булеву функцію. Оскільки в ній допустимі з'єднання елементів, які відповідають суперпозиціям функцій системи $\{\bar{x}, x \wedge y, x \vee y\}$, а ця система повна, то будь-яку булеву функцію можна реалізувати схемою з функціональних елементів, зображених на рис. 7.9.

Під час побудови схем використовують викладені в підрозділі 7.5 методи мінімізації булевих функцій.

Приклад 7.33. Побудувати схему з функціональних елементів, яка реалізує булеву функцію

$$f = x\bar{y}z \vee x\bar{y}\bar{z} \vee \bar{x}yz \vee \bar{x}\bar{y}z \vee \bar{x}\bar{y}\bar{z}.$$

За методом карт Карно знаходимо мінімальну ДНФ (див. приклад 7.29) $f_{\min} = \bar{y} \vee \bar{x}z$. Відповідну схему наведено на рис. 7.13.

Зауважимо, що мінімізація булевих функцій не вичерпує всіх можливостей мінімізації схем. Наприклад, схема, зображена на рис. 7.12, реалізує булеву функцію $f(\tilde{x}^4) = x_1\overline{x_2x_3} \vee$ $\vee x_2x_3x_4$. Вона має п'ять елементів. Це менше, ніж символів операцій у будь-якій формулі булевої алгебри, яка реалізує цю функцію.

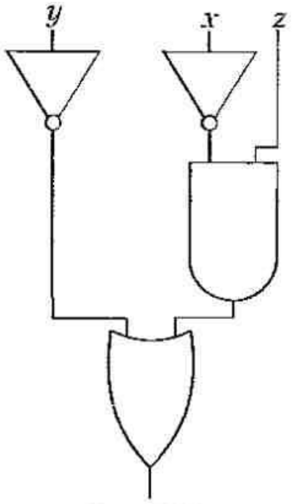

Рис. 7.13

Нехай S — схема з елементів кон'юнкції, диз'юнкції та заперечення, яка реалізує булеву функцію f, а $L_f(S)$ — кількість її елементів. Нехай також $L(f) = \min L_f(S)$, де мінімум узято за всіма схемами S, які реалізують функцію f. Нарешті, уведемо функцію $L(n) = \max L(f)$, де максимум узято за всіма булевими функціями від n змінних.

ТЕОРЕМА 7.25 (Шеннона–Лупанова). Для функції $L(n)$ виконується співвідношення

$$L(n) \sim \frac{2^n}{n},$$

причому для довільного $\varepsilon > 0$ кількість булевих функцій f, для яких $L(f) \leqslant (1-\varepsilon) \cdot 2^n / n$, прямує до 0 зі зростанням n.

Зауваження. 1. Функцію $L(n)$ називають *функцією Шеннона* (на честь американського математика К. Шеннона). 2. Символ ~ означає асимптотичну рівність: $\lim\limits_{n\to\infty}(nL(n)/2^n) = 1$. 3. Зміст другого твердження теореми полягає в тому, що зі зростанням n майже всі булеві функції реалізуються зі складністю, близькою до верхньої межі, тобто $L(n)$.

Теорема 7.25 свідчить, що більшість булевих функцій (у разі $n \to \infty$) мають складні мінімальні схеми. З огляду на побудову схем практичну цінність має лише достатньо вузький клас булевих функцій. Тому разом з універсальними методами побудови схем [49] потрібно мати й такі, що призначені для певних класів булевих функцій, тому що в них краще враховано їх властивості. Розглянемо один підхід до реалізації достатньо вузького класу функцій. Покажемо, як схеми з функціональних елементів можна використовувати для додавання двох цілих додатних чисел, заданих у двійковій системі числення. Спочатку побудуємо схему, яка обчислює $x + y$, де x та y — біти. Входи схеми — x та y; на кожний із них може бути поданий сигнал 0 або 1. Наша конструкція буде об'єднанням двох схем (матиме два виходи). Вихід s дає суму бітів у даному розряді, а вихід c використано для біта перенесення в наступний розряд. У табл. 7.9 наведено значення на входах і виходах схеми, яку називають *півсуматором*. Саму схему наведено на рис. 7.14. Зазначимо, що з табл. 7.9 маємо: $c = xy$; $s = x\bar{y} \vee \bar{x}y = (x \vee y)\overline{xy}$.

Таблиця 7.9

Вхід		Вихід	
x	y	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

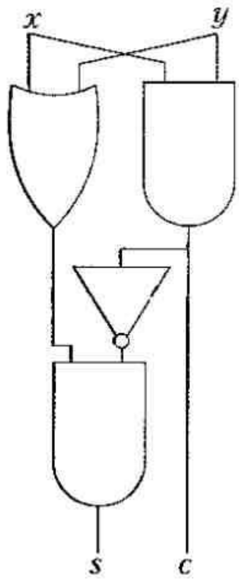

Рис. 7.14

Щоб урахувати біт перенесення c, використовують схему, яку називають *повним суматором*. Її входи – x, y і біт перенесення з попереднього розряду c_i. Виходів два: сума s у даному розряді та нове перенесення c_{i+1} (у наступний розряд). Відповідні булеві функції задано в табл. 7.10.

Таблиця 7.10

Вхід			Вихід	
x	y	c_i	s	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Подамо функції s і c_{i+1} у ДДНФ: $s = \bar{x}\bar{y}c_i \vee \bar{x}y\bar{c}_i \vee x\bar{y}\bar{c}_i \vee xyc_i$; $c_{i+1} = \bar{x}yc_i \vee x\bar{y}c_i \vee xy\bar{c}_i \vee xyc_i$. Замість того, щоб будувати повні суматори безпосередньо за зазначеними формулами, використаємо півсуматори (рис. 7.15).

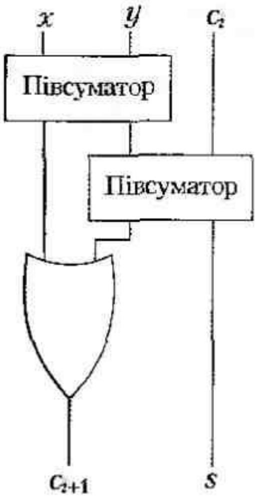

Рис. 7.15

Нарешті, на рис. 7.16 показано, як повні суматори та півсуматор можна використати для додавання трирозрядних цілих чисел $x_3x_2x_1$ та $y_3y_2y_1$ у двійковій системі числення. Результат – двійкове число $s_4s_3s_2s_1$. Зазначимо, що s_4 (старший розряд суми) збігається з c_3.

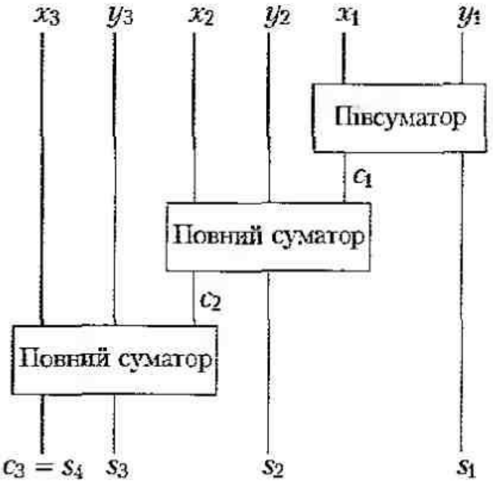

Рис. 7.16

Аналогічно будують схему для додавання n-розрядних двійкових чисел $x_nx_{n-1}\dots x_1$ та $y_ny_{n-1}\dots y_1$.

Контрольні запитання та завдання

1. Довести рівносильність наведених нижче формул F та G. Використати таблиці та тотожні перетворення.

   а) $F = \overline{x}\,\overline{z} \vee xy \vee x\overline{z}$, $G = xyz \vee \overline{z}$;

   б) $F = (x \to y) \to (x\overline{y} \oplus (x \sim \overline{y}))$, $G = (x \vee y)(\overline{x} \vee \overline{y})$;

   в) $F = xy \oplus xz \oplus yz$, $G = xy\overline{z} \vee xz \vee yz$.

2. За принципом двоїстості побудувати формулу, яка реалізує функцію, двоїсту до f:

   а) $f = xy \vee yz \vee xt \vee zt$; б) $f = x \vee y(zt \vee 0) \vee \overline{x}yt$;

   в) $f = (\overline{x} \vee y) \oplus ((x \downarrow y) \mid (\overline{x} \sim yz))$.

3. Перейти від заданої ДНФ до ДДНФ:

   а) $f(\tilde{x}^3) = x_1 \vee \overline{x_2}x_3$; б) $f(\tilde{x}^3) = x_1\overline{x_2} \vee \overline{x_1}x_3$;

   в) $f(\tilde{x}^3) = x_1 \vee \overline{x_1}x_2 \vee \overline{x_2}x_3$; г) $f(\tilde{x}^3) = \overline{x_1}x_2 \vee x_2\overline{x_3} \vee \overline{x_1}\,\overline{x_3}$.

4. За допомогою тотожних перетворень побудувати ДДНФ:

   а) $f(\tilde{x}^3) = (x_1 \vee x_2\overline{x_3})(x_1 \vee x_3)$; б) $f(\tilde{x}^3) = (x_1 \vee \overline{x_2x_3})(x_1x_2 \vee \overline{x_1x_3})$;

   в) $f(\tilde{x}^4) = (x_1 \vee \overline{x_2})x_3x_4 \vee \overline{x_1x_4}$.

5. Перетворити ДНФ задачі 3 на КНФ.

6. Побудувати ДКНФ для функцій задачі 5.

7. Методом невизначених коефіцієнтів знайти поліном Жегалкіна для функцій:

   а) $f(\tilde{x}^3) = (11111000)$; б) $f(\tilde{x}^3) = (00110100)$; в) $f(\tilde{x}^3) = (00111001)$.

8. За допомогою тотожних перетворень побудувати поліном Жегалкіна для функцій:

   а) $f(\tilde{x}^3) = (x_1 \mid x_2) \downarrow x_3$; б) $f(\tilde{x}^3) = (x_1 \to x_2)(x_2 \downarrow x_3)$;

   в) $f(\tilde{x}^3) = ((x_1 \to x_2) \vee \overline{x_3}) \mid x_1$.

9. Скориставшись властивістю полінома Жегалкіна, знайти істотні змінні функцій:

   а) $f(\tilde{x}^2) = (x_1 \oplus x_2)(x_1 \downarrow x_2)$;

   б) $f(\tilde{x}^3) = (((x_3 \to x_2) \vee x_1)(x_2 \to x_1)x_3\overline{x_1}) \oplus x_3$;

   в) $f(\tilde{x}^3) = ((x_1 \vee x_2)(x_1 \vee \overline{x_3}) \to (\overline{x_1} \to x_2\overline{x_3}))x_2$.

10. Довести, що якщо $f(\tilde{x}^n)$ — лінійна функція, відмінна від константи, то $|N_f| = 2^{n-1}$. Чи правильне обернене твердження?

11. Чи можна з функції $f = xy \vee xz \vee yz$ за допомогою суперпозиції одержати функцію xy?

12. Довести, що серед лінійних функцій самодвоїсті ті, поліном Жегалкіна яких має непарну кількість змінних, а несамодвоїсті – парну.

13. Довести, що серед лінійних функцій монотонні лише константи 0, 1 і тотожна функція x.

14. Використовуючи критерій повноти, з'ясувати, чи функціонально повна система Q:

   а) $Q = \{x \vee y, x \oplus y, 1\}$;

   б) $Q = \{x \oplus y \oplus z, x \vee y, 0, 1\}$;

   в) $Q = \{x \vee y, \overline{x}\}$;

   г) $Q = \{x \to y, x \oplus y, 1\}$;

   д) $Q = \{xy \vee xz \vee yz, \overline{x}\}$;

е) $Q = \{x \to y, 0\}$;

ж) $Q = \{xy \vee xz \vee yz, 0, 1\}$;

и) $Q = \{x \to y, x \to \bar{y}z\}$;

к) $Q = \{x\bar{y}, \bar{x} \sim yz\}$;

л) $Q = \{0, 1, x(y \sim z) \vee \bar{x}(y \oplus z)\}$;

м) $Q = \{(01101001), (10001101), (00011100)\}$;

н) $Q = (S \setminus M) \cup (L \setminus (T_0 \cup T_1))$;

п) $Q = (S \cap M) \cup (L \setminus M) \cup (T_0 \setminus S)$;

р) $Q = (M \setminus (T_0 \cap T_1)) \cup (L \setminus S)$.

15. Які неповні системи задачі 14 слабко функціонально повні?

16. Для кожної з наведених нижче функцій методом Куайна побудувати СДНФ і за імплікантною таблицею знайти всі тупикові та мінімальні ДНФ:

а) $\bar{x}yz \vee \bar{x}\bar{y}z$;

б) $xyz \vee xy\bar{z} \vee \bar{x}yz \vee \bar{x}y\bar{z}$;

в) $xy\bar{z} \vee x\bar{y}z \vee x\bar{y}\bar{z} \vee \bar{x}yz \vee \bar{x}\bar{y}z$;

г) $xyz \vee x\bar{y}z \vee x\bar{y}\bar{z} \vee \bar{x}yz \vee \bar{x}y\bar{z} \vee \bar{x}\bar{y}\bar{z}$;

д) $N_f = \{(000), (001), (011), (100), (111)\}$;

е) $N_f = \{(001), (011), (101), (110)\}$;

ж) $N_f = \{(001), (011), (100), (110)\}$;

и) $N_f = \{(000), (010), (011), (100)\}$.

17. Для кожної з функцій задачі 16 побудувати СДНФ методом Мак-Класкі.

18. Знайти мінімальні ДНФ для функцій задачі 16 методом карт Карно.

19. Для кожної з функцій, наведених нижче, побудувати СДНФ методом Мак-Класкі та знайти всі мінімальні ДНФ:

а) $wxyz \vee wx\bar{y}z \vee wx\bar{y}\bar{z} \vee w\bar{x}y\bar{z} \vee w\bar{x}\bar{y}z$;

б) $wxy\bar{z} \vee wx\bar{y}z \vee \bar{w}x\bar{y}\bar{z} \vee w\bar{x}yz \vee \bar{w}\bar{x}y\bar{z} \vee \bar{w}\bar{x}\bar{y}z$;

в) $wxy\bar{z} \vee wx\bar{y}z \vee w\bar{x}\bar{y}z \vee w\bar{x}\bar{y}\bar{z} \vee \bar{w}x\bar{y}z \vee \bar{w}\bar{x}y\bar{z} \vee \bar{w}\bar{x}\bar{y}z \vee wxyz$;

г) $wxyz \vee wxy\bar{z} \vee wx\bar{y}z \vee w\bar{x}yz \vee w\bar{x}y\bar{z} \vee \bar{w}xyz \vee \bar{w}\bar{x}yz \vee \bar{w}\bar{x}y\bar{z} \vee \bar{w}\bar{x}\bar{y}z$.

20. Знайти мінімальні ДНФ для функцій задачі 19 методом карт Карно.

21. Для кожної з функцій, наведених нижче, побудувати СДНФ методом Блейка:

а) $\bar{w}\bar{x} \vee w\bar{y}z \vee x\bar{y}z$; б) $\bar{w}\bar{x}y \vee w\bar{x}z \vee yz$;

в) $wx \vee \bar{w}y \vee w\bar{x}yz \vee \bar{w}xyz$.

22. Для кожної з функцій, наведених нижче, побудувати СДНФ методом Нельсона:

а) $(x \vee y \vee \bar{z})(\bar{x} \vee y \vee z)(\bar{y} \vee \bar{z})$; б) $(w \vee z)(x \vee \bar{y} \vee \bar{z})(\bar{w} \vee \bar{x} \vee \bar{y})$;

в) $(w \vee \bar{x} \vee \bar{y})(\bar{w} \vee z)(w \vee y \vee \bar{z})$.

23. Побудувати схеми з функціональних елементів кон'юнкції, диз'юнкції та заперечення, які реалізують функції:

   а) $xy \vee xz \vee yz$; б) $\overline{x \vee \overline{y}z}$; в) $x\overline{y}z \vee xy\overline{z} \vee \overline{x}yz$;

   г) $x\overline{y}\,\overline{z} \vee \overline{x}y\overline{z} \vee \overline{x}\,\overline{y}z \vee xyz$; д) $(x \vee y)(\overline{z} \vee \overline{y})$;

   е) $x \to y\overline{z}$; ж) $x \sim \overline{y}z$; и) $\overline{x}\,\overline{y} \to \overline{y} \vee \overline{z}$.

24. Побудувати схему з функціональних елементів кон'юнкції, диз'юнкції та заперечення, яка має 4 входи. На входи подаються комбінації двійкового коду десяткових цифр 0, 1, 2, ..., 9. На виході має бути 1 в тому й лише в тому разі, коли комбінації на входах відповідають одноцифровим числам:

   а) непарним;

   б) таким, що не діляться на 3;

   в) таким, що не дорівнюють 4, 5, 6.

   Скористатись методом карт Карно.

Комп'ютерні проекти

Скласти програми із зазначеними вхідними даними та результатами.

1. Булеву функцію n змінних задано таблицею. Побудувати її ДДНФ.
2. Булеву функцію n змінних задано таблицею. Побудувати її ДКНФ.
3. Булеву функцію n змінних задано таблицею. Побудувати формулу, яка подає її лише за допомогою операцій $\neg$ та $\wedge$.
4. Булеву функцію n змінних задано таблицею. Побудувати формулу, яка подає її лише за допомогою операцій $\neg$ та $\vee$.
5. Булеву функцію n змінних задано таблицею. Побудувати формулу, яка подає її лише за допомогою операції $|$.
6. Булеву функцію n змінних задано таблицею. Побудувати формулу, яка подає її функцію лише за допомогою операції $\downarrow$.
7. Булеву функцію трьох змінних задано таблицею. Побудувати її карту Карно.
8. Булеву функцію чотирьох змінних задано таблицею. Побудувати її карту Карно.
9. Булеву функцію n змінних задано таблицею. Побудувати її СДНФ методом Мак-Класкі.