МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

Г.Г. ШВАЧИЧ, Г.М. БАРТЕНЄВ,
А.І. ТІМОШКІН, В.В. ТОЛСТОЙ

ОСНОВИ ДИСКРЕТНОЇ МАТЕМАТИКИ

Частина ІІ. Основи булевої алгебри

**Затверджено на засіданні Вченої ради академії
як навчальний посібник. Протокол № 1 від 29.01.2013**

Дніпропетровськ НМетАУ 2013

УДК 543. 211/.205+543.4
Швачич Г.Г., Бартенєв Г.М., Тімошкін А.І., Толстой В.В. Основи дискретної математики. Частина ІІ. Основи булевої алгебри: Навч. посібник (російською мовою). – Дніпропетровськ: НМетАУ, 2013. – 50 с.

Навчальний посібник органічно поєднує теоретичні положення дискретної математики з їх використанням для розв'язку практичних завдань відповідного напрямку.

Наведені приклади розв'язування типових задач і задач для самостійної роботи студентів з розділу дисципліни «Основи дискретної математики».

Призначений для студентів напряму 6.050101 – комп'ютерні науки.

Іл. 6. Табл. 7. Бібліогр.: 7 найм.

Друкується за авторською редакцією.

Відповідальний за випуск Г.Г. Швачич, канд. техн. наук, проф.

Рецензенти: А.В. Поліщук, канд. техн. наук, доц. (УДХТУ)
О.Г. Холод, канд. техн. наук, доц. (Державний Університет ім. А. Нобеля)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Логика - очень древняя наука. Ещё в античные времена была известна *формальная логика*, позволяющая делать заключения о правильности какого-либо суждения не по его фактическому содержанию, а только по форме его построения. Например, уже в древности был известен **закон исключения третьего**. Его содержательная трактовка была такова: «Во время своих странствований Платон ***был*** в Египте **ИЛИ** ***не был*** Платон в Египте». В такой форме это или любое другое выражение будут правильны (тогда говорили: *истинно*). Ничего другого быть не может: Платон либо был, либо не был в Египте - третьего не дано.

Другой закон логики **- закон непротиворечивости**. Если сказать: «Во время своих странствий Платон ***был*** в Египте **И** ***не был*** Платон в Египте», то очевидно, любое высказывание, имеющее такую форму, всегда будет *ложно*. Если из теории следуют два противоречащих друг другу вывода, то такая теория безусловно неправильная (ложная) и должна быть отвергнута.

Ещё один закон, известный в древности - **закон отрицания:** «Если **НЕ** верно, что Платон **НЕ *был*** в Египте, то значит, Платон ***был*** в Египте».

Формальная логика основана на «высказываниях». «Высказывание» - это основной элемент логики, определяемый как повествовательное предложение, относительно которого можно однозначно сказать, истинное или ложное утверждение оно содержит.

Например: Листва на деревьях опадает осенью. Земля прямоугольная.

Первое высказывание содержит истинную информацию, а второе - ложную. Вопросительное, побудительное и восклицательное предложения не являются высказываниями, так как в них ничего не утверждается и не отрицается.

Пример предложений, не являющихся высказываниями: Не пейте сырую воду! Кто не хочет быть счастливым?

Высказывания могут быть и такими: $2>1$, $H_2O+SO_3=H_2SO_4$. Здесь используются языки математических символов и химических формул.

Приведённые выше примеры высказываний являются *простыми.* Но из простых высказываний можно получить *сложные*, объединив их с помощью логических связок. Логические связки - это слова, которые подразумевают определённые логические связи между высказываниями. Основные логические связки издавна употребляются не только в научном языке, но и в обыденном, - это «и», «или», «не», «если ... то», «либо ... либо» и другие известные нам из русского языка связки. В рассмотренных нами трёх законах формальной логики использовались связки «и», «или», «не», «если ... то» для связи простых высказываний в сложные.

Высказывания бывают *общими, частными* и *единичными.* Общее высказывание начинается со слов: ***всё, все, всякий, каждый, ни один.*** Частное высказывание начинается со слов: ***некоторые, большинство*** и т.п. Во всех других случаях высказывание является единичным.

Формальная логика была известна в средневековой Европе, она развивалась и обогащалась новыми законами и правилами, но при этом вплоть до 19 века она оставалась обобщением конкретных содержательных данных, и её законы сохраняли форму высказываний на разговорном языке.

1. ОСНОВНЫЕ ПОНЯТИЯ АЛГЕБРЫ ЛОГИКИ

Для выполнения работ, связанных с функционально-логическим анализом и синтезом различных цифровых систем, современному инженеру необходимы знания математических основ их работы – алгебры логики и теории цифровых автоматов. В разработку основ алгебры логики значительный вклад внес ирландский математик Дж. Буль (1815-1864), поэтому данная алгебра называется алгеброй Буля или булевой алгеброй.

В алгебре логики (в отличие от обычной алгебры) переменные принимают значения из двухэлементного множества $E_2 = \{0,1\}$. Булевы переменные могут быть связаны функциональными зависимостями. Функция $f(x_1, x_2, ..., x_n)$, зависящая от n двоичных переменных $x_1, x_2, ..., x_n$ (где n – натуральное число), является булевой, если она определяет отображение

$$\underbrace{E_2 x E_2 x ... x E_2}_{n} \xrightarrow{f} E_2 \text{; где } E_2 = \{0,1\}.$$

Говоря по-другому, функцией алгебры логики (или логической функцией) от n переменных называется n-парная операция на $E_2 = \{0,1\}$. Алгебра, образованная множеством E_2 вместе со всеми возможными операциями на нем, называется алгеброй логики [1].

Набор $(\alpha_1, \alpha_2, ..., \alpha_n)$, где $\alpha_i \in \{0,1\}$, $1 \le i \le n$, значений n входных аргументов (переменных) в логической функции называется входным булевым или двоичным набором (вектором). Элементы этого вектора (набора) называются компонентами или координатами. Коротко вектор $(\alpha_1, \alpha_2, ..., \alpha_n)$ обозначается через $\tilde{\alpha}^n$ или $\tilde{\alpha}$ [2]. Число n называется длиной вектора $\tilde{\alpha}^n$. Нормой (или весом) $\left\|\tilde{\alpha}^n\right\|$ вектора $\tilde{\alpha}^n$ называется число его координат, равных 1, т.е. $\left\|\tilde{\alpha}^n\right\| = \sum_{i=1}^{n} \alpha_i$. Из определения функции $f(x_1, x_2, ..., x_n)$ следует, что для ее задания достаточно указать, какое значение функции соответствует каждому из входных векторов значений аргументов, т.е. составить таблицу (табл. 1.1).

Таблица 1.1

$x_1\ x_2 \dots x_{n-1}\ x_n$	$f(x_1, x_2, \dots, x_{n-1}, x_n)$
0 0 … 0 0	$f(0, 0, \dots, 0, 0)$
0 0 … 0 1	$f(0, 0, \dots, 0, 1)$
0 0 … 1 0	$f(0, 0, \dots, 1, 0)$
0 0 … 1 1	$f(0, 0, \dots, 1, 1)$
………………	………………
1 1 … 1 1	$f(1, 1, \dots, 1, 1)$

Легко видеть, что n аргументов (переменных) принимают 2^n различных значений. Для удобства употребляется стандартное расположение векторов [3]: если вектор рассматривать как запись числа в двоичном исчислении, то порядок следования векторов в таблице 1 соответствует естественному порядку следования чисел 0, 1, 2, …, 2^n-1. В качестве примера можно привести функцию $f(x_1,x_2, x_3)$ трех аргументов, задаваемую таблицей 1.2.

Таблица 1.2

$x_1\ x_2\ x_3$	$f(x_1, x_2, x_3)$
0 0 0	*0*
0 0 1	*1*
0 1 0	*1*
0 1 1	*0*
1 0 0	*1*
1 0 1	*0*
1 1 0	*0*
1 1 1	*1*

Векторы (наборы), на которых функция $f=1$, называются единичными векторами функции f, а множество единичных векторов – единичным множеством f. Соответственно, векторы, на которых $f=0$, называются нулевыми векторами *f[1]*.

Данная функция определена на всех возможных векторах логических аргументов и называется полностью определенной функцией. Множество всех полностью определенных логических функций, содержащее также константы 0 и 1, в литературе принято обозначать через P_2, а множество всех полностью определенных логических функций n переменных (или аргументов) – через $p_2(n)$.

Справедлива следующая теорема.

Т е о р е м а 1. Число $p_2(n)$ всех функций из P_2, зависящих от n аргументов (переменных) $x_1, x_2, \ldots, x_n$, равно 2^{2^n}.

Д о к а з а т е л ь с т в о. При любом фиксированном упорядочении векторов значений аргументов (переменных) $x_1, x_2, \ldots, x_n$ любая логическая функция $f(x_1,x_2,\ldots,x_n)$ из P_2 полностью определяется вектор-столбцом ее значений, т.е. вектор-столбцом длины 2^n. Следовательно, число $|p_2(n)|$ различных функций n аргументов из P_2 равно числу различных двоичных векторов длины 2^n, т.е. 2^{2^n} .

Функция, логические значения для которой определены только для части логических векторов аргументов, называется частично определенной булевой функцией. Векторы значений аргументов, для которых функция определена, называются информационными, а для которых не определена – неинформационными. Частично определенную булеву функцию можно сделать полностью определенной, если приписать неинформационным векторам значений аргументов произвольные значения функции. Часто доопределение булевой функции проводится так, чтобы упростить ее реализацию.

Говорят, что функция $f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ из P_2 зависит существенным образом от переменной (аргумента) x_i, если существуют такие значения $\alpha_1,\ldots,\alpha_{i-1},\alpha_{i+1},\ldots,\alpha_n$ переменных (аргументов) $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$, что

$$f(\alpha_1,\ldots,\alpha_{i-1},0,\alpha_{i+1},\ldots,\alpha_n) \neq f(\alpha_1,\ldots,\alpha_{i-1},1,\alpha_{i+1},\ldots,\alpha_n) \text{ [3].}$$

В этом случае переменная x_i называется существенной. Если x_i не является существенной переменной, то она называется несущественной или фиктивной.

Определение. Функции f_1 и f_2 называются равными, если функцию f_2 можно получить из f_1 путем добавления и изъятия фиктивных аргументов [3].

Функция $f(x_1, \dots, x_n)$ называется вырожденной тогда, когда она не зависит, по крайней мере, от одной из своих переменных. Функция называется невырожденной, если она зависит существенным образом от всех своих переменных [4].

Существуют два типа функций, которые не имеют существенных переменных: функции первого типа тождественно равны 0, а второго – 1.

Определение. Булева функция $f(x_1, \dots, x_n)$ называется симметрической относительно переменных $x_1, \dots, x_k$ $((2 \le k \le n))$, если для любой подстановки $\begin{pmatrix} 1 \dots k \\ j_1 \dots j_k \end{pmatrix}$ имеет место равенство

$$f(x_1, \dots, x_k, x_{k+1}, \dots, x_n) = f(x_{j_1}, \dots, x_{j_k}, x_{k+1}, \dots, x_n).$$

Аналогично вводится понятие функции, симметрической относительно произвольных переменных $x_{i_1}, \dots, x_{i_k}$ [3].

Очевидно, что функции, тождественно равные константам 0 и 1, являются симметрическими относительно любой совокупности своих переменных.

В алгебре логики большую роль играют логические функции одной и двух переменных, на основе которых, применяя принцип суперпозиции, можно выразить все логические функции от произвольного числа переменных.

2. ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ

Логических функций одной переменной – четыре. Они приведены в таблице 2.1 (так как $n=1$, то $p(1)=2^2=4$)

Таблица 2.1

Логические функции одной переменной

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

Функции f_0 и f_3 – это константы 0 и 1соответственно; их значения не зависят от значений переменной x, поэтому переменная x для них является несущественной. Функция f_1 носит название функции повторения входной переменной x или тождественной функции (значение функции f_1 повторяет значение переменной x). Функция f_2 называется отрицанием x, или инверсией x, и обозначается x, $\neg x$, x', $\sim x$. Значение этой функции противоположно значению входной переменной x.

Функцию $f_2 = x$ еще называют функцией НЕ, которая обладает следующими свойствами:

1) двойная инверсия (а также произвольное четное количество инверсий) некоторой переменной x равна самой переменной

$$\overline{\overline{x}} = x;$$

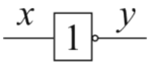
2) тройная инверсия (а также произвольное нечетное количество инверсий) некоторой переменной x равна инверсии этой переменной

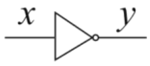
$$\overline{\overline{\overline{x}}} = \overline{x};$$

3) если дано произвольное логическое равенство, то инверсия обеих его частей не нарушает этого равенства, т.е. если $f(x_1) = f(x_2)$, то $\overline{f(x_1)} = \overline{f(x_2)}$.

Схемное обозначение (УГО) логического элемента, реализующего функцию НЕ, таково:

в отечественной литературе в англоязычной литературе





3. ФУНКЦИИ ДВУХ ПЕРЕМЕННЫХ

Логических функций двух переменных – 16; они приведены в таблице 3.1 (так как $n=2$, то $p(2)=2^{2^2}=2^4=16)$

Таблица 3.1

Логические функции двух переменных

$x_1 x_2$	φ_0	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9	φ_{10}	φ_{11}	φ_{12}	φ_{13}	φ_{14}	φ_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Функции φ_0 и φ_{15} – константы 0 и 1, т.е. функции с двумя несущественными (фиктивными) аргументами. Формально эти функции отличаются от f_0 и f_3 в таблице 2.1: все функции из таблицы 2.1 – это унарные операции на E_2, а все функции из таблицы 3.1 – бинарные операции на E_2. Однако, поскольку они отличаются лишь несущественными переменными (аргументами), то они являются равными. Функции φ_3, φ_5, φ_{10}, φ_{12} – это функции с одним фиктивным (несущественным) аргументом. При этом функция φ_3 – это повторение входной переменной x_1, функция φ_5 – повторение входной переменной x_2. Функция φ_{10} – это отрицание переменной x_2, а функция φ_{12} – это отрицание переменной x_1. В функциях φ_3 и φ_{12} фиктивна переменная x_2, в функциях φ_5 и φ_{10} – переменная x_1.

Итак, из 16 функций двух аргументов шесть функций имеют фиктивные аргументы. Доля функций, имеющих фиктивные аргументы, с ростом *n* (числа аргументов) убывает и стремится к нулю [1].

Функцию $\varphi_2 = x_1 \nrightarrow x_2$ называют прямой антиимпликацией или запретом x_2.

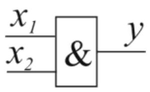
Функцию $\varphi_4 = x_1 \nleftarrow x_2$ называют обратной антиимпликацией или запретом x_1. Функция $\varphi_{11} = x_1 \leftarrow x_2$ носит название обратной импликации. Функция же $\varphi_{13} = x_1 \rightarrow x_2$ носит название импликации. Оставшиеся шесть функций двух аргументов φ_1, φ_6, φ_7, φ_8, φ_9 и φ_{14} наиболее часто используются в технических приложениях булевой алгебры.

Функцию $\varphi_1 = x_1 \cdot x_2$ называют логическим умножением, конъюнкцией, функцией И, совпадением, пересечением. Для ее обозначения используются символы: • (точка), ^, &, ∩. Для данной функции выполняются следующие свойства:

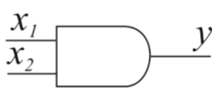
$$x_1 \cdot 0 = 0,\ x_1 \cdot 1 = x_1,\ x_1 \cdot x_1 = x_1,\ x_1 \cdot \overline{x_1} = 0.$$

Схемное обозначение (УГО) логического элемента, реализующего функцию И, таково:

в отечественной литературе



в англоязычной литературе

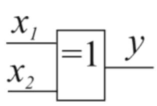


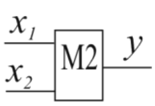
Функцию $\varphi_6 = x_1 \oplus x_2$ называют ИСКЛЮЧАЮЩИМ ИЛИ, а также функцией неравнозначности, разноименности, суммой по модулю 2. Для ее обозначения используются символы $\oplus$, Δ, $\not\equiv$. Эта функция равна 1, когда значения ее переменных различны, и равна 0, когда они равны. Данная функция обладает следующими свойствами:

$$x_1 \oplus 0 = x_1,\ x_1 \oplus 1 = \overline{x_1},\ x_1 \oplus x_1 = 0,\ x_1 \oplus \overline{x_1} = 1.$$

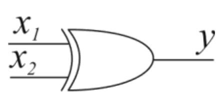
Схемное обозначение (УГО) логического элемента, реализующего функцию ИСКЛЮЧАЮЩЕЕ ИЛИ, таково:

в отечественной литературе





в англоязычной литературе

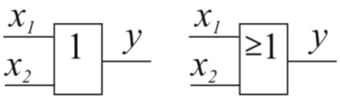


Функцию $\varphi_7 = x_1 \vee x_2$ называют логическим сложением, дизъюнкцией, функцией ИЛИ, суммой, объединением. Для ее обозначения используются символы $\vee$, +, $\cup$. Функция φ_7 равна 0 тогда и только тогда, когда оба аргумента равны нулю ($x_1 = x_2 = 0$). Для данной функции выполняются следующие соотношения:

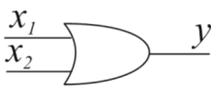
$$x_1 \vee 0 = x_1,\; x_1 \vee 1 = 1,\; x_1 \vee x_1 = x_1,\; x_1 \vee \overline{x_1} = 1.$$

Схемное обозначение логического элемента, реализующего функцию ИЛИ, таково:

в отечественной литературе



в англоязычной литературе

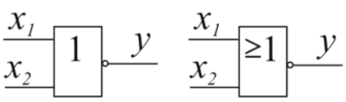


Функцию $\varphi_8 = x_1 \downarrow x_2$ называют стрелкой Пирса, или функцией ИЛИ-НЕ. Для ее обозначения используются символы $\downarrow$, 0. Функция φ_8 равна 1 тогда и только тогда, когда оба аргумента равны нулю $(x_1 = x_2 = 0)$. Для данной функции выполняются следующие свойства:

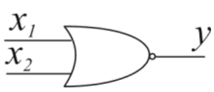
$$x_1 \downarrow 0 = \overline{x_1},\; x_1 \downarrow 1 = 0,\; x_1 \downarrow x_1 = \overline{x_1},\; x_1 \downarrow \overline{x_1} = 0.$$

Схемное обозначение логического элемента, реализующего функцию ИЛИ-НЕ, таково:

в отечественной литературе



в англоязычной литературе

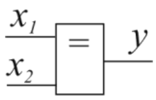


Функцию $\varphi_9 = x_1 \sim x_2$ называют эквивалентностью, или равнозначностью. Для ее обозначения используются символы ~, $\equiv$, $\rightleftarrows$, $\leftrightarrow$. Функция φ_9 равна 1 тогда и только тогда, когда оба аргумента x_1 и x_2 принимают одинаковые значения, и равна нулю тогда и только тогда, когда аргументы принимают различные значения (см. табл. 3.1). Для данной функции выполняются следующие свойства:

$$x_1 \sim 0 = \overline{x_1},\; x_1 \sim 1 = x_1,\; x_1 \sim x_1 = 1,\; x_1 \sim \overline{x_1} = 0.$$

Схемное обозначение логического элемента, реализующего функцию РАВНОЗНАЧНОСТЬ, таково:

в отечественной литературе



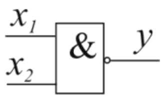
в англоязычной литературе

Функцию $\varphi_{14} = x_1 / x_2$ называют штрихом Шеффера, а также функцией И-НЕ, несовместимостью. Функция φ_{14} равна 0 тогда и только тогда, когда оба аргумента равны единице $(x_1=x_2=1)$. Для данной функции выполняются следующие свойства:

$$x_1 / 0 = 1,\ x_1 / 1 = \overline{x_1},\ x_1 / x_1 = \overline{x_1},\ x_1 / \overline{x_1} = 1.$$

Схемное обозначение логического элемента, реализующего функцию И-НЕ, таково:

в отечественной литературе



в англоязычной литературе

4. ФОРМУЛЫ. РЕАЛИЗАЦИЯ ФУНКЦИЙ ФОРМУЛАМИ

Исходя из наиболее простых функций одной и двух переменных, можно строить формулы. Дадим индуктивное определение формулы [3].

Определение. Пусть μ (греч. «мю») – некоторое (не обязательно конечное) подмножество функций из P_2.

1. Базис индукции. Каждая функция $f(x_1,...,x_m)$ из μ называется **формулой над** μ.

2. Индуктивный переход. Пусть $f_0(x_1,...,x_m)$ – функция из μ и $A_1,...,A_m$ – выражения, являющиеся либо формулами над μ, либо символами переменных из U (где U – исходный алфавит переменных (аргументов)). Тогда выражение $f_0(A_1,...,A_m)$ называется формулой над μ.

В дальнейшем формулы будут обозначаться заглавными греческими буквами с квадратными скобками, в которых перечисляются функции, необходимые для их построения. Так

$$\Delta[f_1,...,f_s]$$

означает, что формула Δ построена из $f_1,...,f_s$. Если функция *f* соответствует формуле Δ (т.е. формула Δ задает способ вычисления функции *f*), то говорят, что формула Δ реализует функцию *f*.

Формулы Δ и Γ(«гамма») над μ называются эквивалентными, если соответствующие им функции f_Δ и f_Γ равны. Запись $\Delta=\Gamma$ в дальнейшем будет означать, что формулы Δ и Γ эквивалентны.

Примеры эквивалентных формул:

$$(x \to y) = (\bar{y} \to \bar{x}),$$

$$0 = (x \& \bar{x}).$$

5. ЗАКОНЫ АЛГЕБРЫ ЛОГИКИ

Логические функции И и ИЛИ обладают рядом свойств, подобных свойствам обычных алгебраических операций умножения и сложения. Анализируя таблицы истинности функций И и ИЛИ, легко убедиться в справедливости для них сочетательного (ассоциативного), переместительного (коммутативного) и распределительного (дистрибутивного) законов.

Сочетательные (ассоциативные) законы:

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

$$x_1 \cdot x_2 \cdot x_3 = x_1 \cdot (x_2 \cdot x_3) = x_2 \cdot (x_1 \cdot x_3) = x_3 \cdot (x_1 \cdot x_2).$$

Переместительные (коммутативные) законы:

$$x_1 \vee x_2 = x_2 \vee x_1; \quad x_1 \cdot x_2 = x_2 \cdot x_1$$

Следует отметить, что переместительные законы справедливы для дизъюнкции и конъюнкции любого числа переменных.

Распределительные законы:

а) дистрибутивность конъюнкции относительно дизъюнкции (распределительный закон 1-го рода):

$$x_1(x_2 \vee x_3) = x_1 \cdot x_2 \vee x_1 \cdot x_3;$$

б) дистрибутивность дизъюнкции относительно конъюнкции (распределительный закон 2-го рода):

$$x_1 \vee (x_2 \cdot x_3) = (x_1 \vee x_2)(x_1 \vee x_3).$$

Также нетрудно убедиться в справедливости законов де Моргана (законов инверсии):

$$\overline{x_1 \cdot x_2 \cdot \ldots \cdot x_m} = \overline{x_1} \vee \overline{x_2} \vee \ldots \vee \overline{x_m};$$

$$\overline{x_1 \vee x_2 \vee \ldots \vee x_m} = \overline{x_1} \cdot \overline{x_2} \cdot \ldots \cdot \overline{x_m}.$$

При решении логических задач приходится иметь дело с выражениями, включающими действия отрицания, конъюнкции и дизъюнкции в любом их сочетании. Возникает вопрос: в какой последовательности следует производить различные операции? По аналогии с арифметическими операциями, отрицание является старшей логической операцией (операцией первой ступени), конъюнкция является средней логической операцией (операцией второй ступени), а дизъюнкция – младшей логической операцией (операцией третьей ступени).

При выполнении совместных логических операций необходимо пользоваться следующими правилами:

1) если в выражении имеются операции только одной ступени, то их необходимо производить в том порядке, в котором они написаны;

2) если в выражении имеются операции различных ступеней, то сначала необходимо производить операции первой ступени, затем – второй ступени и, наконец, – третьей.

Для изменения приоритетов производства логических операций используют скобки.

6. ПРАВИЛА УПРОЩЕНИЯ ЛОГИЧЕСКИХ ВЫРАЖЕНИЙ

До того как рассматривать правила упрощения логических выражений, рассмотрим следующие понятия алгебры логики: элементарная конъюнкция (дизъюнкция), конституента единицы (нуля), ранг элементарной конъюнкции (дизъюнкции), соседние элементарные конъюнкции (дизъюнкции).

Зафиксируем перечень входных переменных $X = \{x_1, x_2, ..., x_n\}$.

Конъюнкция переменных из набора (множества) X или их отрицаний, т.е. конъюнкция вида $\tilde{x}_{i1} \cdot \tilde{x}_{i2} \cdot ... \cdot \tilde{x}_{ik}$, где

$$\tilde{x}_{ij} = \begin{cases} x_{ij} \\ \bar{x}_{ij} \end{cases} (j = 1,2,...,k),$$

называется элементарной, если в ней каждая буква встречается не более одного раза [5]. К элементарным конъюнкциям относятся также выражения, состоящие из одной буквы (с отрицанием или без отрицания), называетмые одноместными элементарными конъюнкциями. Константу 1 можно рассматривать как нульместную элементарную конъюнкцию. Так, элементарными конъюнкциями являются $1, \bar{x}_1, x_1 \cdot x_2, \bar{x}_1 \cdot \bar{x}_2 \cdot x_3$, а выражения $x_1 \cdot x_2 \cdot x_1$, $x_1 \cdot \bar{x}_1$, $\overline{x_1 \cdot x_2}$ ими не являются по определению. Определение элементарной дизъюнкции дается аналогично двойственным образом [5].

Конституента единицы (нуля) – элементарная конъюнкция (дизъюнкция), в которой содержатся все *n* переменных в прямом или инверсном виде. Конституента единицы (нуля) еще называется минтермом (макстермом).

Ранг элементарной конъюнкции (дизъюнкции) - это количество переменных в прямом или инверсном виде, содержащихся в элементарной конъюнкции (дизъюнкции). Так, $\bar{x}_1 \cdot x_2$ - элементарная конъюнкция второго ранга, а $x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4$ - элементарная конъюнкция четвертого ранга.

Две элементарные конъюнкции (дизъюнкции) одного и того же ранга называются соседними тогда и только тогда, когда они являются функциями одних и тех же переменных и отличаются только знаком инверсии одной из переменных. Так, например, конъюнкции $x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4$ и $x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4$ являются соседними, так как они являются функциями одних и тех же переменных и отличаются знаком отрицания только одной переменной x_2.

Перейдем теперь к рассмотрению правил упрощения логических выражений. Это следующие правила: правило склеивания для соседних элементарных конъюнкций, правило склеивания для соседних элементарных дизъюнкций, правило поглощения для двух элементарных конъюнкций, правило поглощения для двух элементарных дизъюнкций.

Правило склеивания для соседних элементарных конъюнкций Дизъюнкция двух соседних конъюнкций некоторого ранга *p* заменяется одной элементарной конъюнкцией ранга *p-1*, являющейся общей частью исходных операндов дизъюнкции. Данное правило является следствием распределительного закона 1-го рода.

$$x_1 \cdot x_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee x_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 = x_1 \cdot x_2 \cdot \bar{x}_3,\ \bar{x}_1 \cdot x_2 \vee x_1 \cdot x_2 = x_2.$$

Правило склеивания для соседних элементарных дизъюнкций Конъюнкция двух соседних элементарных дизъюнкций некоторого ранга *p* заменяется одной элементарной дизъюнкцией ранга *p-1*, являющейся общей частью исходных операндов конъюнкции. Данное правило является следствием распределительного закона 2-го рода.

$$(x_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4) \cdot (x_1 \vee x_2 \vee \bar{x}_3 \vee x_4) = x_1 \vee x_2 \vee \bar{x}_3,$$

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

Правило поглощения для элементарных конъюнкций

Логическая сумма двух элементарных конъюнкций разных рангов, из которых одна является собственной частью другой, заменяется конъюнкцией, имеющей меньший ранг. Данное правило представляет собой следствие распределительного закона 1-го рода.

$$x_1 \cdot x_2 \cdot \bar{x}_3 \cdot x_4 \vee \bar{x}_3 \cdot x_4 = \bar{x}_3 \cdot x_4; \; \bar{x}_2 \vee x_1 \cdot \bar{x}_2 = \bar{x}_2.$$

Кроме этого, выполняются равенства:

а) $x_1 \vee \bar{x}_1 \cdot x_2 = x_1 \vee x_2$; б) $\bar{x}_1 \vee x_1 \cdot x_2 = \bar{x}_1 \vee x_2$.

Доказательство равенства а):

$$x_1 \vee \bar{x}_1 \cdot x_2 = x_1 \cdot (x_2 \vee \bar{x}_2) \vee \bar{x}_1 \cdot x_2 = x_1 \cdot x_2 \vee x_1 \cdot \bar{x}_2 \vee \bar{x}_1 \cdot x_2 = x_1 \cdot x_2 \vee$$
$$\vee x_1 \cdot \bar{x}_2 \vee x_1 \cdot x_2 \vee \bar{x}_1 \cdot x_2 = x_1 \cdot (x_2 \vee \bar{x}_2) \vee x_2 \cdot (x_1 \vee \bar{x}_1) = x_1 \vee x_2.$$

Доказательство равенства б):

$$\bar{x}_1 \vee x_1 \cdot x_2 = \bar{x}_1 \cdot (x_2 \vee \bar{x}_2) \vee x_1 \cdot x_2 = \bar{x}_1 \cdot x_2 \vee \bar{x}_1 \cdot \bar{x}_2 \vee x_1 \cdot x_2 = \bar{x}_1 \cdot x_2 \vee \bar{x}_1 \cdot \bar{x}_2 \vee$$
$$\vee \bar{x}_1 \cdot x_2 \vee x_1 \cdot x_2 = \bar{x}_1 \cdot (x_2 \vee \bar{x}_2) \vee x_2 \cdot (\bar{x}_1 \vee x_1) = \bar{x}_1 \vee x_2.$$

Правило поглощения для элементарных дизъюнкций

Конъюнкция двух элементарных дизъюнкций разных рангов, из которых одна является собственной частью другой, заменяется дизъюнкцией, имеющей меньший ранг. Данное правило является следствием распределительного закона 2-го рода и широко используется при упрощении логических выражений.

$$(x_1 \vee x_2 \vee \bar{x}_3 \vee x_4) \cdot (\bar{x}_3 \vee x_4) = \bar{x}_3 \vee x_4;$$

$$\bar{x}_2 \cdot (x_1 \vee \bar{x}_2) = \bar{x}_2.$$

Правило обобщенного склеивания:

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

Доказательство:

$$xz \vee y\bar{z} \vee xy = xz \vee y\bar{z} \vee xyz \vee xy\bar{z} = xz \vee y\bar{z}.$$

Законы алгебры логики могут быть доказаны двумя способами:

1) аналитическим;

2) табличным (с помощью таблицы истинности).

Рассмотрим широко используемый в алгебре логики принцип двойственности.

Определение. Функция $[f(x_1,...,x_n)]^*$, равная $f(\bar{x}_1,...,\bar{x}_n)$, называется двойственной функцией к функции $f(x_1,...,x_n)$.

Очевидно [3], что таблица истинности для двойственной функции (при выбранном порядке векторов) получается из таблицы истинности для функции $f(x_1,...,x_n)$ инвертированием (т.е. заменой 0 на 1 и 1 на 0) столбца функции и его переворачиванием (см. табл. 6.1).

Таблица 6.1

x_1 x_2 x_3	$f(x_1, x_2, x_3)$	$[f(x_1, x_2, x_3)]^*$
0 0 0	1	0
0 0 1	1	1
0 1 0	0	0
0 1 1	0	1
1 0 0	0	1
1 0 1	1	1
1 1 0	0	0
1 1 1	1	0

В дальнейшем двойственную к $f(x_1,...,x_n)$ функцию будем обозначать через $f^*(x_1,...,x_n)$.

Нетрудно заметить, что среди функций $0, 1, x, \bar{x}, x_1 \& x_2, x_1 \vee x_2$ функция 0 двойственна 1, функция 1 двойственна функции 0, функция x двойственна функции x, функция $\bar{x}$ двойственна функции $\bar{x}$; функция $x_1 \& x_2$ двойственна функции $x_1 \vee x_2$, функция $x_1 \vee x_2$ двойственна $x_1 \& x_2$.

Из определения двойственности следует, что

$$f^{**} = (f^*)^* = f$$

т.е. функция *f* является двойственной к *f** (свойство взаимности) [3].

Обозначим через $x_1,...,x_n$ все различные символы переменных, встречающиеся в множествах $(x_{11},...,x_{1p_1}),...,(x_{m_1},...,x_{mp_m})$.

Теорема 2. Если

$$\Phi(x_1,\dots,x_n) = f(f_1(x_{11},\dots,x_{1p_1}),\dots,f_m(x_{m1},\dots,x_{mp_m})),\text{ то}$$

$$\Phi^*(x_1,\dots,x_n) = f^*(f_1^*(x_{11},\dots,x_{1p_1}),\dots,f_m^*(x_{m1},\dots,x_{mp_m})).$$

Доказательство:

$$\Phi^*(x_1,\dots,x_n) = \overline{\Phi}(\overline{x}_1,\dots,\overline{x}_n) = \overline{f}(f_1(\overline{x}_{11},\dots,\overline{x}_{1p_1}),\dots,f_m(\overline{x}_{m1},\dots,\overline{x}_{mp_m})) =$$

$$= \overline{f}(\overline{\overline{f}}_1(\overline{x}_{11},\dots,\overline{x}_{1p_1}),\dots,\overline{\overline{f}}(\overline{x}_{m_1},\dots,\overline{x}_{mp_m})) = \overline{f}(\overline{f}_1^*(x_{11},\dots,x_{1p_1}),\dots,\overline{f}_m^*(x_{m_1},\dots,x_{mp_m})) =$$
$$= f^*(f_1^*(x_{11},\dots,x_{1p_1}),\dots,f_m^*(x_{m1},\dots,x_{mp_m})).$$

Из теоремы следует

Принцип двойственности. Если формула $\Gamma = C[f_1,\dots f_s]$ реализует функцию $f(x_1,\dots,x_n)$, то формула $\Gamma = C[f_1^*,\dots f_s^*]$, т.е. формула, полученная из Γ заменой функций $f_1,\dots f_s$ соответственно на $f_1^*,\dots f_s^*$, реализует функцию $f^*(x_1,\dots,x_n)$.

Эту формулу будем называть формулой, двойственной к Γ, и обозначать через Γ^*. Итак,

$$\Gamma^* = C[f_1^*,\dots f_s^*].$$

Для формул над множеством $\mu = \{0, 1, \overline{x}, x_1 \& x_2, x_1 \vee x_2\}$ принцип двойственности формулируется следующим образом: для получения формулы Γ^*, двойственной к формуле Γ, нужно в формуле Γ всюду заменить 0 на 1, 1 на 0, & на $\vee$, $\vee$ на &. Или, если $\Gamma = C[0, 1, \overline{x}, x_1 \& x_2, x_1 \vee x_2]$, то $\Gamma^* = C[1, 0, \overline{x}, x_1 \vee x_2, x_1 \& x_2]$.

Примеры.

1) $\Gamma_1(x_1,x_2,x_3) = x_1 \& x_2 \& x_3$, $\Gamma_1^*(x_1,x_2,x_3) = x_1 \vee x_2 \vee x_3$;

2) $\Gamma_2(x_1,x_2,x_3) = x_1 \cdot x_2 \cdot x_3 \vee \overline{x}_1 \cdot \overline{x}_2 \cdot \overline{x}_3$;

$$\Gamma_2^*(x_1,x_2,x_3) = (x_1 \vee x_2 \vee x_3) \cdot (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3).$$

Из принципа двойственности также следует, что если $\Gamma(x_1,...,x_n) = \Delta(x_1,...,x_n)$, то $\Gamma^*(x_1,...,x_n) = \Delta^*(x_1,...,x_n)$.

П р и м е р. Из тождества $\overline{x_1 \& x_2 \& ... \& x_n} = \bar{x}_1 \vee \bar{x}_2 \vee ... \vee \bar{x}_n$ следует тождество $\overline{x_1 \vee x_2 \vee ... \vee x_n} = \bar{x}_1 \& \bar{x}_2 \& ... \& \bar{x}_n$.

7. РАЗЛОЖЕНИЕ БУЛЕВЫХ ФУНКЦИЙ ПО ПЕРЕМЕННЫМ. ОСНОВНЫЕ АНАЛИТИЧЕСКИЕ ФОРМЫ ОПИСАНИЯ БУЛЕВЫХ ФУНКЦИЙ

Если в качестве μ допустить некоторый запас элементарных функций, то любая ли функция булевой алгебры может быть выражена в виде формулы через эти элементарные функции?

Введем обозначение

$$x^\sigma = x\sigma \vee \bar{x}\bar{\sigma},$$

где σ – параметр, равный либо 0, либо 1. Очевидно, что

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

Легко видеть, что $x^\sigma = 1$ если и только если $x = \sigma$, т.е. значение «основания» равно значению «показателя» и $x^\sigma = 0$ если $x \neq \sigma$.

Т е о р е м а 3 (о разложении функций по переменным). Любую функцию алгебры логики $f(x_1,...,x_n)$ при любом m $(1 \le m \le n)$ можно представить в следующем виде:

$$f(x_1,...,x_m,x_{m+1},...,x_n) = \bigvee_{(\sigma_1,...,\sigma_m)} x_1^{\sigma_1} \& ... \& x_m^{\sigma_m} \& f(\sigma_1,...,\sigma_m,x_{m+1},...,x_n), \quad (*)$$

где дизъюнкция берется по всем 2^m наборам значений переменных $x_1,...,x_m$.

Это равенство называется разложением по переменным $x_1,...,x_m$.

Д о к а з а т е л ь с т в о. Теорема доказывается подстановкой в обе части равенства (*) произвольного набора $(\alpha_1,...,\alpha_m,\alpha_{m+1},...,\alpha_n)$.

Левая часть дает $f(\alpha_1,...,\alpha_n)$. Правая - $\bigvee_{(\sigma_1,...\sigma_m)} \alpha_1^{\sigma_1} \& ... \& \alpha_m^{\sigma_m} \& f(\sigma_1,...,\sigma_m,\alpha_{m+1},...,\alpha_n) = \alpha_1^{\alpha_1} \& ... \& \alpha_m^{\alpha_m} \&$
$\& f(\alpha_1,...,\alpha_m,\alpha_{m+1},...,\alpha_n) = f(\alpha_1,...,\alpha_n),$

поскольку $\alpha_i^{\sigma i} = 1$ только когда $\alpha_i = \sigma_i$. Из всех конъюнкций в 1 обратится только одна – та, в которой $\alpha_1 = \sigma_1,...,\alpha_m = \sigma_m$. Все остальные конъюнкции равны 0.

В качестве следствий имеем два специальных случая разложения.

1) Разложение по одной переменной:

$$f(x_1,...,x_{n-1},x_n) = x_n \& f(x_1,...,x_{n-1},1) \vee \bar{x}_n \& f(x_1,...,x_{n-1},0).$$

Функции $f(x_1,...,x_{n-1},0)$ и $f(x_1,...,x_{n-1},1)$ называются компонентами разложения.

2) Разложение по всем *n* переменным:

$$f(x_1,...,x_n) = \bigvee_{(\sigma 1,...\sigma n)} x_1^{\sigma 1} \& ... \& x_n^{\sigma n} \& f(\sigma_1,...,\sigma_n).$$

При $f(x_1,...,x_n) \not\equiv 0$ оно может быть преобразовано:

$$\bigvee_{(\sigma 1,...\sigma n)} x_1^{\sigma 1} \& ... \& x_n^{\sigma n} \& f(\sigma_1,...,\sigma_n) = \bigvee_{\substack{(\sigma 1,...\sigma n) \\ f(\sigma 1,...\sigma n)=1}} x_1^{\sigma 1} \& ... \& x_n^{\sigma n}.$$

В результате окончательно получаем:

$$f(x_1,...,x_n) = \bigvee_{\substack{(\sigma 1,...\sigma n) \\ f(\sigma 1,...\sigma n)=1}} x_1^{\sigma 1} \& ... \& x_n^{\sigma n} \quad (**).$$

Такое разложение носит название совершенной дизъюнктивной нормальной формы (совершенной д.н.ф.). Соотношение (**) приводит к важной теореме.

Теорема 4. Каждая функция алгебры логики может быть представлена в виде формулы через отрицание, конъюнкцию и дизъюнкцию.

Доказательство. 1) Пусть $f(x_1,...,x_n) \equiv 0$. Тогда, очевидно, $f(x_1,...,x_n) = x_1 \& \bar{x}_1$. 2) Пусть $f(x_1,...,x_n) \not\equiv 0$. Представим ее в виде совершенной д.н.ф.:

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

Таким образом, в обоих случаях функция f выражается в виде формулы через отрицание, конъюнкцию и дизъюнкцию.

Итак, любую функцию алгебры логики можно представить формулой над μ, взяв в качестве μ множество, состоящее из трех функций: отрицания, конъюнкции и дизъюнкции.

Данная теорема носит конструктивный характер, так как она позволяет для любой функции фактически построить формулу, ее реализующую в виде совершенной д.н.ф. Для этого в таблице для функции $f(x_1,...,x_n)\not\equiv 0$ отмечаем все строки $(\sigma_1,...,\sigma_n)$, в которых $f(\sigma_1,...,\sigma_n)=1$; далее для каждой такой строки образуем конъюнкцию $x_1^{\sigma_1}\&...\&x_n^{\sigma_n}$, и затем все полученные конъюнкции соединяем знаками дизъюнкции.

Примеры. 1) Найти совершенную д.н.ф. для функции $x_1\to x_2$. Имеется три набора *(0, 0)*, *(0, 1)* и *(1, 1)*, на которых эта функция равна *1*. Поэтому, $x_1\to x_2=x_1^0\&x_2^0\vee x_1^0\&x_2^1\vee x_1^1\&x_2^1=\bar{x}_1\&\bar{x}_2\vee\bar{x}_1\&x_2\vee x_1\&x_2$.

2) Найти совершенную д.н.ф. для функции, заданной таблицей 7.1.

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

Таблица 7.1

x_1 x_2 x_3	$f(x_1$ x_2 $x_3)$
0 0 0	1
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1

Совершенная д.н.ф. есть выражение типа $\sum\Pi$, т.е. логическая сумма произведений $x_i^{\sigma i}$. Возможно ли для функций алгебры логики получить разложение типа $\Pi\sum$? Ответ на этот вопрос дает следующая теорема.

Т е о р е м а 5. Каждую функцию алгебры логики $f(x_1,...,x_n)$ при любом m $(1 \le m \le n)$ можно представить в следующем виде:

$$f(x_1,...,x_m,x_{m+1},...,x_n) = \underset{(\sigma_1,\sigma_2,...\sigma_m)}{\&}(x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee ... \qquad (***)$$
$$... \vee x_m^{\bar{\sigma}_m} \vee f(\sigma_1,\sigma_2,...,\sigma_m,x_{m+1},...,x_n)),$$

где конъюнкция берется по всевозможным наборам $(\sigma_1,\sigma_2,...,\sigma_m)$ из E_2^m.

Как и при доказательстве теоремы 3 рассмотрим произвольный набор значений переменных $(\alpha_1,\alpha_2,...,\alpha_n)$ и покажем, что левая и правая части соотношения (***) принимают на нем одно и то же значение. Левая часть дает $f(\alpha_1,\alpha_2,...,\alpha_n)$. Правая часть –

$$\underset{(\sigma_1,\sigma_2,...\sigma_m)}{\&}(\alpha_1^{\bar{\sigma}_1} \vee \alpha_2^{\bar{\sigma}_2} \vee ... \vee \alpha_m^{\bar{\sigma}_m} \vee f(\sigma_1,\sigma_2,...,\sigma_m,\alpha_{m+1},...,\alpha_n)) =$$
$$= \alpha_1^{\bar{\alpha}_1} \vee \alpha_2^{\bar{\alpha}_2} \vee ... \vee \alpha_m^{\bar{\alpha}_m} \vee f(\alpha_1,\alpha_2,...,\alpha_m,\alpha_{m+1},...,\alpha_n) = f(\alpha_1,\alpha_2,...,\alpha_n).$$

В качестве следствий имеем два специальных случая разложения:

1) Разложение по одной переменной:

$$f(x_1,...,x_{n-1},x_n) = (x_n^{\bar{\sigma}_n} \vee f(x_1,...,x_{n-1},\sigma_n))\&(x_n^{\sigma_n} \vee f(x_1,...,x_{n-1},\bar{\sigma}_n)) =$$
$$= (x_n^{\bar{0}} \vee f(x_1,...,x_{n-1},0))\&(x_n^{0} \vee f(x_1,...,x_{n-1},\bar{0})) = (x_n \vee f(x_1,...,x_{n-1},0))\&$$
$$\&(\bar{x}_n \vee f(x_1,...,x_{n-1},1)).$$

2) Разложение по всем n переменным

$$f(x_1,x_2,...,x_n) = \underset{(\sigma_1,\sigma_2,...\sigma_n)}{\&}(x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee ... \vee x_n^{\bar{\sigma}_n} \vee f(\sigma_1,\sigma_2,...,\sigma_n)).$$

При $f(x_1,x_2,...,x_n) \not\equiv 1$ оно может быть преобразовано:

$$\underset{(\sigma_1,\sigma_2,...\sigma_n)}{\&}(x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee ... \vee x_n^{\bar{\sigma}_n} \vee f(\sigma_1,\sigma_2,...,\sigma_n)) =$$
$$= \underset{\substack{(\sigma_1,\sigma_2,...\sigma_n) \\ f(\sigma_1,\sigma_2,...\sigma_n)=0}}{\&}(x_1^{\bar{\sigma}_1} \vee x_2^{\bar{\sigma}_2} \vee ... \vee x_n^{\bar{\sigma}_n}).$$

В результате окончательно получаем:

$$f(x_1,x_2,...,x_n)=\mathop{\&}_{\substack{(\sigma_1,\sigma_2,..\sigma_n)\\ f(\sigma_1,\sigma_2,..\sigma_n)=0}}(x_1^{\bar\sigma_1}\vee x_2^{\bar\sigma_2}\vee...\vee x_n^{\bar\sigma_n});$$

где конъюнкция берется по всем наборам $(\sigma_1,\sigma_2,...,\sigma_n)\in E_2^n$, на которых функция $f(x_1,x_2,...,x_n)$ обращается в 0. Такое разложение носит название совершенной конъюнктивной нормальной формы (совершенной к.н.ф.) функции $f(x_1,x_2,...,x_n)$.

Алгоритм перехода от таблицы истинности логической функции к ее записи в виде СКНФ:

1) в таблице для функции $f(x_1,x_2,...,x_n)(f\not\equiv 1)$ отмечаем все наборы $(\sigma_1,\sigma_2,...,\sigma_n)$, на которых $f(\sigma_1,\sigma_2,...,\sigma_n)=0$;

2) для каждого такого набора образуем логическую сумму $x_1^{\bar\sigma_1}\vee x_2^{\bar\sigma_2}\vee...\vee x_n^{\bar\sigma_n}$ (конституенту нуля – макстерм);

3) все полученные дизъюнкции (макстермы) соединяем знаками конъюнкции.

П р и м е р. Получить СКНФ логической функции φ_9, заданной следующей таблицей истинности:

Таблица 7.2

Номер набора	x_1	x_2	φ_9
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

Функция φ_9 принимает нулевое значение на первом и втором наборах: (01) и (10). Для этих наборов записываются макстермы:

$$x_1^{\bar 0}\vee x_2^{\bar 1}=x_1^1\vee x_2^0=x_1\vee\bar x_2;\ x_1^{\bar 1}\vee x_2^{\bar 0}=x_1^0\vee x_2^1=\bar x_1\vee x_2;$$

соединяем знаком конъюнкции полученные макстермы и получаем СКНФ для φ_9:

$$\varphi_9 = (x_1 \vee \bar{x}_2)(\bar{x}_1 \vee x_2).$$

Правильность полученного аналитического выражения можно проверить, если подставить в него значения переменных x_1 и x_2, соответствующие наборам, на которых функция φ_9 обращается в нуль. На векторе (01) $\varphi_9 = (0 \vee 0)(1 \vee 1) = 0$; на векторе (10) $\varphi_9 = (1 \vee 1)(0 \vee 0) = 0$. Так как на этих векторах $\varphi_9 = 0$, следовательно СКНФ записана правильно.

Кроме двух основных (СДНФ) и (СКНФ) аналитических форм представления функций в булевой алгебре имеются дизъюнктивные и конъюнктивные нормальные формы (ДНФ и КНФ), а также минимальные дизъюнктивные и конъюнктивные нормальные формы (МДНФ и МКНФ).

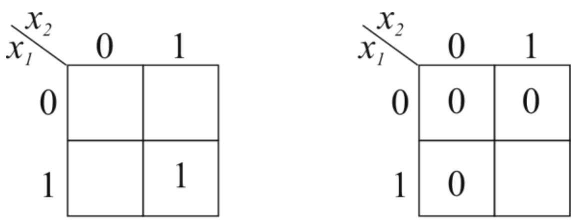
Для ДСНФ (СДНФ) характерны конъюнкции одинакового ранга. В ДНФ конъюнкции могут быть различного ранга. ДНФ/КНФ называется минимальной, если она содержит наименьшее число букв по сравнению со всеми ДНФ/КНФ, имеющимися для данной функции.

8. СЛОВЕСНАЯ И ТАБЛИЧНАЯ ФОРМЫ ОПИСАНИЯ БУЛЕВЫХ ФУНКЦИЙ

Произвольная булева функция может быть выражена в словесной форме. Например, функция $\varphi_1 = x_1 \cdot x_2$ может быть словесно представлена так: $\varphi_1 = 1$ тогда и только тогда, когда оба аргумента x_1 и x_2 равны единице. Эта же функция выражается табличной формой (таблицей истинности или картой Карно).

Карта Карно на два аргумента - это квадрат, разделенный на четыре ячейки: по одной на каждый входной набор [6]. Строки карты ассоциированы с аргументом x_1, столбцы – с аргументом x_2. Следовательно, находящаяся слева вверху ячейка соответствует вектору (0,0) или минтерму $(\bar{x}_1 \cdot \bar{x}_2)$, а находящаяся справа внизу ячейка соответствует вектору (1,1) или минтерму $(x_1 \cdot x_2)$ (рис. 8.1).

Представление булевой функции с помощью карты Карно осуществляется в соответствии с таблицей истинности. Если функция $\varphi_1 = x_1 \cdot x_2 = 1$ на векторе (1,1), то этот факт отражается на этой карте проставлением в правую нижнюю ячейку единицы (рис. 8.1, а). Остальные ячейки остаются незаполненными. В карте Карно могут проставляться нули в те ячейки, на векторах которых функция равна нулю. На рисунке 8.1, б дан пример заполнения карты Карно для функции $\varphi_1 = 0$.



а) б)

Рис. 8.1. Карта Карно на две переменные: $\varphi_1 = 1$ (а); $\varphi_1 = 0$ (б).

В случае трех аргументов в карте Карно имеется восемь ячеек: по одной на каждый входной вектор (рис. 8.2).

x_1 \ x_2x_3	00	01	11	10
0				
1				

Рис. 8.2. Карта Карно на три переменные

Аргумент x_1 ассоциирован с двумя строками карты, а аргументы x_2 и x_3 - с четырьмя столбцами. Карту заполняют таким образом, что два соседних столбца всегда ассоциированы либо с самим аргументом, либо с его отрицанием. Например, первые два столбца ассоциированы с отрицанием аргумента x_2 (т.е. с $\bar{x}_2$), второй и третий – с аргументом x_3, третий и четвертый – с аргументом x_2, а четвертый и первый – с отрицанием аргумента x_3 (т.е. с $\bar{x}_3$). Таким образом, любые две рядом находящиеся ячейки оказываются соседними, и их координаты отличаются только одним аргументом. Кроме этого, соседними являются ячейки, находящиеся в первом и последнем столбцах карты. Так как для четырех аргументов имеется 16 входных векторов, то в соответствующей карте Карно имеется 16 ячеек (рис. 8.3).

x_1x_2 \ x_3x_4	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Рис. 8.3. Карта Карно на четыре аргумента

Каждая ячейка пронумерована в соответствии с порядковым номером входного вектора. Для булевых функций с числом аргументов $n > 6$ карты Карно становятся слишком большими и неудобными для практического использования.

Аналитические формы представления булевых функций такие, как СДНФ и СКНФ, могут оказаться слишком громоздкими. Для их упрощения может быть использована более компактная числовая форма записи, предусматривающая только десятичные эквиваленты двоичным входным векторам, на которых функция принимает значение единицы (для СДНФ) и нуля (для СКНФ). Эти десятичные эквиваленты соединяются знаками дизъюнкции (для СДНФ) и конъюнкции (для СКНФ).

Пример. Составить в числовой форме СДНФ функции φ_6.

Решение. Поскольку из аналитической формы (СДНФ) и таблицы истинности следует, что $\varphi_6 = 1$ на первом и втором наборах, то числовая форма СДНФ принимает вид:

$$\varphi_6 = 1 \vee 2 = \vee(1,2).$$

Пример. Составить в числовой форме СКНФ функции φ_6.

Решение. Поскольку из аналитической формы (СКНФ) и таблицы истинности следует, что $\varphi_6 = 0$ на нулевом и третьем наборах, то числовая форма СКНФ принимает вид:

$$\varphi_6 = 0 \cdot 3 = \&(0,3).$$

9. ГЕОМЕТРИЧЕСКОЕ И КУБИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ БУЛЕВЫХ ФУНКЦИЙ

В геометрическом смысле каждый двоичный набор $(\alpha_1, \alpha_2, ..., \alpha_n)$ можно рассматривать как n-мерный вектор, определяющий точку n-мерного двоичного пространства [6]. Исходя из этого, все множество двоичных векторов, на которых определена функция n аргументов, представляется в виде вершин n-мерного куба. Закрашивая вершины, в которых функция принимает единичные значения, получают ее геометрическое представление. Например, функция $y = \vee(0,4,5,6,7)$ геометрически представляется так, как показано на рисунке 9.1.

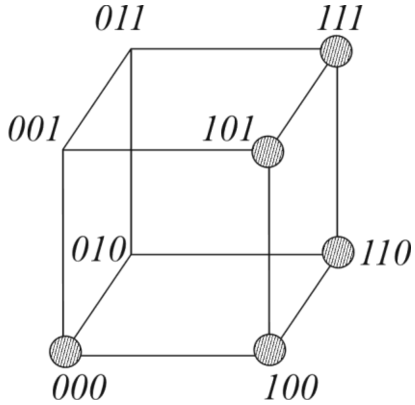


Рис. 9.1. Геометрическое представление (нулевые кубы) булевой функции

На основе геометрических представлений строятся кубические представления булевых функций, в которых применяются элементы *n*-мерных кубов. Каждую вершину, на которой функция принимает единичное значение, называют 0-кубом или нулевым кубом. Множество 0-кубов булевой функции формирует ее нулевой кубический комплекс K_0. Например, для булевой функции $y = \vee(0,4,5,6,7)$ имеется следующий нулевой кубический комплекс: $K_0 = \{000,100,101,110,111\}$. В этом случае нулевой кубический комплекс K_0 образуют пять 0-кубов, соответствующих определенным вершинам трехмерного куба (рис. 9.1). Если два 0-куба из комплекса отличаются только одной координатой, то два таких 0-куба формируют один единичный куб (1-куб). Он представляется общими координатами 0-кубов, а на месте координаты, по которой отличаются 0-кубы, ставится символ _ (пробел), обозначающий независимую координату. Например, два 0-куба 100 и 110 отличаются только второй координатой и формируют 1-куб 1_0, которому соответствует ребро трехмерного куба. Все множество единичных кубов булевой функции формирует ее единичный кубический комплекс K_1. Для булевой функции $y = \vee(0,4,5,6,7)$ единичный кубический комплекс K_1 состоит из пяти 1-кубов: $K_1 = \{\_00,10\_,1\_1,1\_0,11\_\}$ и фиксирует все множество ребер, на которых функция y принимает единичные значения (рис. 9.2).

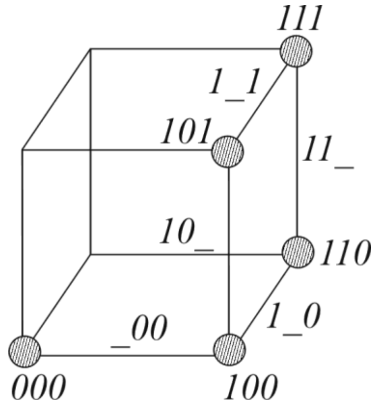


Рис. 9.2. Кубическое представление (единичные кубы) булевой функции

Если два единичных куба из комплекса K_1 содержат общую независимую координату и различаются только одной координатой, то они формируют один двоичный куб (2-куб). Его запись состоит из общих компонент единичных кубов, а координата, принимающая различные значения в единичных кубах, обозначается как независимая координата _ (пробел). Например, два единичных куба (1_0 и 1_1) формируют один двоичный куб (1_ _). Для рассматриваемой булевой функции y комплекс K_2 имеет вид $K_2 = \{1\_\ \_\}$ и состоит из одного двоичного куба, соответствующего грани трехмерного куба (рис. 9.3).

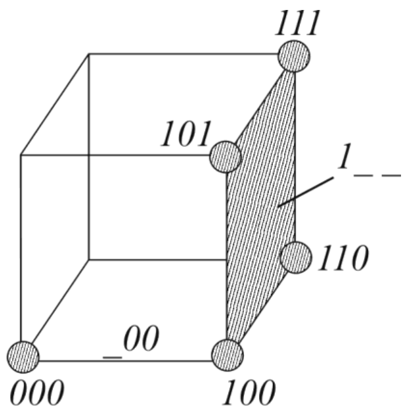


Рис. 9.3. Кубическое представление (двоичный куб) булевой функции

Размерность куба определяется числом независимых координат (пробелов в кубе). Объединение кубических комплексов $K_0, K_1, K_2, ..., K_n$

функции $f(x_1, x_2, ..., x_n)$ формирует кубический комплекс функции $K(f)$. Для рассматриваемого примера комплекс $K(f)$ соответствует объединению комплексов K_0, K_1 и K_2.

10. МИНИМИЗАЦИЯ БУЛЕВЫХ ФУНКЦИЙ

Наиболее эффективно задача минимизации булевых функций решается на основе использования кубических представлений этих функций. Каждая булева функция $f(x_1, x_2, ..., x_n)$ рассматривается как множество $П$ кубов, принадлежащих комплексу $K(f)$, и таких, что каждая вершина комплекса K_0 заключена по крайней мере в один куб множества $П$. Полученное таким образом множество $П$ называется покрытием булевой функции $П(f)$. Каждому избранному покрытию $П(f)$ соответствует своя ДНФ функции f. Например, функции $f = \vee(0,4,5,6,7)$ соответствует комплекс

$$K(f) = \{000, 100, 101, 110, 111, \_00, 10\_, 1\_1, 1\_0, 11\_, 1\_\,\_\}.$$

В покрытие данной логической функции могут быть включены различные совокупности кубов, принадлежащих $K(f)$ и охватывающих все вершины n-мерного куба, в которых $f = 1$:

$$\begin{Bmatrix} 0\,0\,0 \\ 1\_\,\_ \end{Bmatrix}, \begin{Bmatrix} \_\,0\,0 \\ 1\_\,\_ \end{Bmatrix}, \begin{Bmatrix} \_\,0\,0 \\ 1\_\,1 \\ 1\_\,0 \end{Bmatrix}, \begin{Bmatrix} \_\,0\,0 \\ 1\,0\,\_ \\ 1\,1\,\_ \end{Bmatrix} \text{ и др.}$$

От выбранного покрытия логической функции легко перейти к ее ДНФ. Например, для второго покрытия можно записать: $f = \bar{x}_2 \cdot \bar{x}_3 \vee x_1$.

Для оценки сложности логических функций вводится понятие цены покрытия $(Ц_n)$, равной сумме цен всех кубов, составляющих покрытие $П(f)$:

$$Ц_n = \sum_{K_i \in П(f)} Ц_{K_i}.$$

Цена r_i-куба, соответствующего элементарной конъюнкции функции n переменных, определяется как разность $n - r_i = Ц_{k_i}$ и соответствует количеству букв в элементарной конъюнкции. Например, цена второго покрытия будет равна $Ц_n = (3-1)+(3-2) = 3$, цена первого покрытия равна 4, третьего и четвертого равна 6. Покрытие, имеющее минимальную цену, назовем минимальным покрытием функции в смысле Квайна, а ДНФ, определенную по минимальному покрытию – МДНФ. Поскольку цена второго покрытия имеет минимальное значение из всех возможных, то соответствующая ему ДНФ $f = \bar{x}_2 \cdot \bar{x}_3 \vee x_1$ является МДНФ.

Из большого числа различных приемов и методов минимизации булевых функций рассмотрим два наиболее типичных:

1) табличный (с помощью карт Карно);

2) метод Квайна и Мак-Класки.

Во всех методах проводят минимизацию булевой функции, представленной в СДНФ или СКНФ. В общем случае процесс минимизации состоит из трех этапов.

Этап 1 – переход от СДНФ (СКНФ) к сокращенной ДНФ (КНФ) при помощи склеивания конституент. Под сокращенной формой функции понимается ДНФ (КНФ), членами которой являются только несклеивающиеся элементарные конъюнкции (дизъюнкции). Члены сокращенной ДНФ (КНФ) в булевой алгебре называются простыми импликантами. Не исключается случай, когда СДНФ (СКНФ) тождественно равна сокращенной форме рассматриваемой функции.

Этап 2 – переход от сокращенной ДНФ (КНФ) к тупиковой дизъюнктивной (конъюнктивной) нормальной форме – ТДНФ (ТКНФ). Тупиковой ДНФ (КНФ) называется такая ДНФ (КНФ), членами которой служат простые импликанты, среди которых нет ни одной лишней. Лишним называется такой член формы функции, удаление которого из формы не влияет на значение истинности этой функции. Покрытие логической функции f соответствующее ее тупиковой ДНФ называется неприводимым.

Правило проверки сокращенной ДНФ (КНФ) на лишние члены: для каждого члена сокращенной ДНФ (КНФ) определяется такой вектор значений истинности его переменных, который обращает данный член в единицу (нуль). Далее находится значение суммы (произведения) остальных членов формы на этом векторе. Если сумма (произведение) равна единице (нулю), то проверяемый член – лишний. В противном случае проверяемый член не является лишним. Возможны случаи, когда в сокращенной форме не имеется лишних членов. Тогда сокращенная ДНФ (КНФ) тождественно равна тупиковой. Название «тупиковая форма» говорит о том, что дальнейшая минимизация булевой функции в рамках нормальных форм уже невозможна.

Этап 3 – переход от тупиковой формы исследуемой функции к ее минимальной форме. Этот этап не является регулярным и требует определенного опыта в поиске путей дальнейшего упрощения функции. Эти пути связаны с уменьшением числа операций отрицания (на основе закона де-Моргана) или числа конъюнкций и дизъюнкций (на основе распределительных законов).

10.1. МИНИМИЗАЦИЯ ПРИ ПОМОЩИ КАРТЫ КАРНО

Карта Карно является таблицей для представления булевых функций в форме СДНФ и СКНФ. Расположение клеток в таблице позволяет легко фиксировать склеивающиеся между собой члены. В других методах минимизации процесс поиска склеивающихся членов оказывается наиболее трудоемким, так как при осуществлении операции склеивания необходимо сравнивать все возможные пары членов исходной формы функции. В табличном методе Карно поиск склеивающихся членов формы максимально упрощен. Каждой клетке карты Карно соответствует вершина куба.

Склеивающиеся между собой конституенты находятся в соседних ячейках. Минимальное покрытие выбирается на основе анализа различных вариантов покрытий минимизируемой булевой функции.

Правила минимизации при помощи карт Карно заключается в следующем:

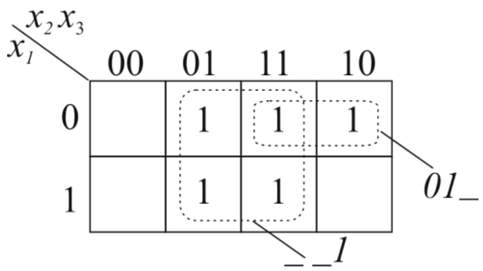
1. 2^k соседних клеток, содержащих единицы/нули и расположенных по вертикали либо по горизонтали, а также в виде прямоугольника либо квадрата, соответствуют одной элементарной конъюнкции/дизъюнкции, ранг которой r меньше ранга конституенты n на k единиц. Чем больше клеток в прямоугольнике, тем проще выражаемый им член формы булевой функции – импликанта (имплицента).

2. В любой карте Карно соседними клетками, к которым можно применить правило склеивания, являются не только рядом находящиеся ячейки, но и клетки, стоящие на противоположных концах любой строки и любого столбца (эти клетки становятся смежными, если карту свернуть по горизонтали или вертикали в цилиндр. При свертывании по горизонтали и вертикали образуется тороидальная поверхность).

Таким образом, процесс минимизации сводится к отысканию наиболее крупных прямоугольников из 2^k соседних заполненных ячеек. Следует стремиться к тому, чтобы каждая заполненная ячейка входила в какой-либо прямоугольник. Импликанта, соответствующая некоторому прямоугольнику заполненных ячеек, содержит символы тех переменных, значения которых совпадают у всех объединенных ячеек. Формула, получающаяся в результате минимизации булевой функции с помощью карт Карно, содержит сумму стольких элементарных произведений, сколько прямоугольников имеется в покрытии.

Пример. Минимизировать с помощью карты Карно булеву функцию вида $y = \vee(1, 2, 3, 5, 7)$.

Решение. Поскольку исходная функция задана для трех переменных в числовой форме, для минимизации выбираем карту Карно на три переменные, и на указанных в исходной функции векторах записываем в соответствующие ячейки единицы. Так как исходных наборов пять, то и карта Карно должна содержать пять единиц. После заполнения карты Карно производится склеивание конституент соседних ячеек. Получаем покрытие, состоящее из двух прямоугольников.



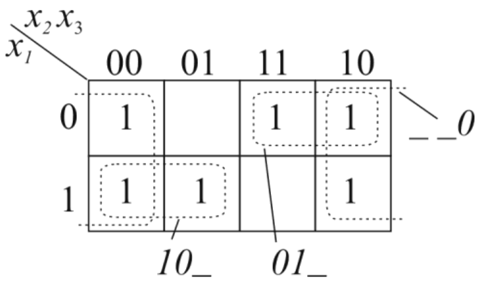
В первый прямоугольник входят четыре ячейки, находящиеся в средней части карты [6]. Эти ячейки покрываются переменной x_3 (т.е. склеиваются по переменным x_1 и x_2) и формируют двоичный куб $\_\_1$. Во второй прямоугольник входят две ячейки (*011* и *010*), которые склеиваются по переменной x_3 и формируют единичный куб *01_*. В результате имеем минимальное покрытие, состоящее из двух кубов:

$$П(y_{МДНФ}) = \begin{Bmatrix} \_\_1 \\ 01\_ \end{Bmatrix}$$

Тогда МДНФ записывается в виде $y_{МДНФ} = x_3 \vee \bar{x}_1 \cdot x_2$. Цена получившегося покрытия $Ц_{П_{МДНФ}} = 3$. Для сравнения можно указать, что цена покрытия исходной формы функции $Ц_{П_{СДНФ}} = 15$.

П р и м е р. Минимизировать с помощью карты Карно функцию $y = \vee(0,2,3,4,5,6)$.

Р е ш е н и е. Выбираем карту Карно на три переменные. Отмечаем на карте Карно исходную функцию y, т.е. проставляем единицы в ячейки, соответствующие единичным входным наборам функции.



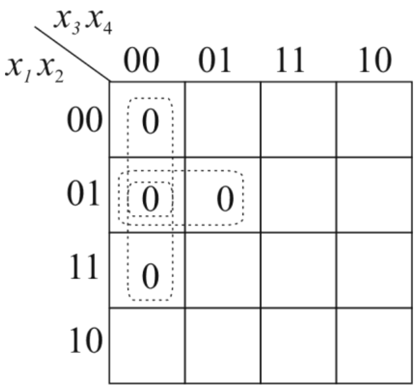
После заполнения карты Карно проводим склеивание конституент соседних ячеек. Получается минимальное покрытие исходной функции, состоящее из трех кубов:

$$П(y_{МДНФ}) = \begin{Bmatrix} \_\_0 \\ 01\_ \\ 10\_ \end{Bmatrix}.$$

Тогда МДНФ функции записывается в следующем виде:

$$y_{МДНФ} = \bar{x}_3 \vee \bar{x}_1 \cdot x_2 \vee x_1 \cdot \bar{x}_2 .$$

Иногда, чтобы найти форму с наименьшей ценой покрытия, необходимо минимизировать с помощью карты Карно как СДНФ, так и СКНФ булевой функции. Пусть дана СКНФ булевой функции в виде $y_{СКНФ} = \&(0,4,5,12)$. После заполнения карты Карно проводим склеивание конституент соседних ячеек.



Получаем минимальное покрытие СКНФ функции, состоящее из трех кубов:

$$\Pi(y_{МКНФ}) = \begin{Bmatrix} \_100 \\ 010\_ \\ 0\_00 \end{Bmatrix}.$$

Цена полученного покрытия $Ц_{\Pi_{МКНФ}} = 9$. МКНФ функции имеет вид:

$$y_{МКНФ} = (\bar{x}_2 \vee x_3 \vee x_4) \cdot (x_1 \vee \bar{x}_2 \vee x_3) \cdot (x_1 \vee x_3 \vee x_4).$$

Заполним теперь карту Карно для данной функции единицами и проведем склеивание конституент соседних ячеек.

x_1x_2 \ x_3x_4	00	01	11	10
00		1	1	1
01			1	1
11		1	1	1
10	1	1	1	1

Получим теперь минимальное покрытие СДНФ функции, состоящее из четырех кубов:

$$\Pi(y_{МДНФ}) = \begin{Bmatrix} 1\_\_1 \\ \_0\_1 \\ 10\_\_ \\ \_\_1\_ \end{Bmatrix}.$$

Цена полученного покрытия $Ц_{\Pi_{МДНФ}} = 7$. МДНФ функции имеет вид:

$$y_{МДНФ} = x_1x_4 \vee \bar{x}_2x_4 \vee x_1 \cdot \bar{x}_2 \vee x_3.$$

МДНФ функции получилась более экономичной, чем МКНФ.

10.2. МЕТОД МИНИМИЗАЦИИ КВАЙНА И МАК-КЛАСКИ

Отличительные особенности метода Квайна и Мак-Класки следующие:

1) метод минимизации легко реализуется на компьютере;

2) практически отсутствуют ограничения на количество переменных минимизируемой функции;

3) для уменьшения количества требуемых сравнений все конституенты группируются по признаку одинакового количества в них единиц. Это позволяет ограничиться проверкой возможности склеивания элементов только соседних групп.

Основные этапы минимизации логических функций методом Квайна и Мак-Класки [7]:

Этап 1. Нахождение простых импликант. Сначала все нулевые кубы распределяются на группы с одинаковым количеством единиц в них. Затем попарно сравниваются между собой все нулевые кубы соседних групп. Если два нулевых куба различаются только одной координатой, то они формируют единичный куб. Нулевые кубы, образующие единичный куб, отмечаются. После построения всех единичных кубов, формирующих комплекс K_1, выполняется построение двоичных кубов и т.д. При построении r-кубов все (r-1)-кубы, формирующие r-кубы, также отмечаются. Этап завершается, когда ни один ($r+1$)-куб не может быть построен. Все неотмеченные кубы комплекса $K(f)$ являются простыми импликантами и формируют покрытие $\Pi(f)$ функции f.

Этап 2. Построение таблицы покрытий. Строки таблицы покрытий соответствуют простым импликантам, а столбцы – нулевым кубам (конституентам единицы) минимизируемой функции. На пересечении i строки и столбца j проставляется отметка (единица), если импликанта i покрывает конституенту j. Импликанта покрывает конституенту, если она отличается от нее только независимыми координатами.

Этап 3. Нахождение минимального покрытия функции. Для определения минимального покрытия необходимо удалить из таблицы покрытий лишние простые импликанты. С этой целью в методе Квайна и Мак-Класки осуществляется следующий алгоритм решения задачи покрытия:

1. Выделение ядра Квайна. Если в каком-либо столбце таблицы покрытий имеется только одна единица, то импликанта, стоящая в соответствующей строке, является обязательной, т.е. существенной, и входит в ядро Квайна, а, следовательно, в минимальное покрытие, поскольку не используя ее, невозможно покрыть все конституенты.

Определение. Максимальный куб N относительно множества $K_0(f)$ называется ядровым, если существует такая точка $\tilde{\alpha}$ из $K_0(f)$, что $\tilde{\alpha} \in N$ и $\tilde{\alpha}$ не принадлежит никакому другому максимальному (относительно $K_0(f)$) кубу.

Определение. Множество всех простых импликант функции f, соответствующее множеству всех ядровых кубов относительно $K_0(f)$, называется ядром Квайна.

2. Вычеркивание строк и столбцов, покрываемых импликантами, входящими в ядро Квайна. Если в полученной таблице покрытий имеются столбцы, содержащие по одной единице, то операцию, описываемую в п. 1, следует повторить.

3. Поглощение лишних столбцов (сжатие по столбцам). Из матрицы вычеркивается тот столбец, в который полностью входит любой другой столбец. Говоря другими словами, если в таблице покрытий имеется такая пара столбцов d_i и d_j, что $d_i \subseteq d_j$, то столбец d_j вычеркивается, поскольку покрытие вычеркнутого столбца может осуществляться за счет покрытия оставшегося столбца.

4. Поглощение лишних строк (сжатие по строкам). Если в таблице покрытий есть такая пара строк q_i и q_j, что $q_i \supseteq q_j$, то строка q_j вычеркивается (строка q_j поглощается строкой q_i).

5. Последовательное применение двух преобразований (сжатие по столбцам и строкам). В итоге исходная таблица покрытий преобразуется в неприводимую («циклическую»), импликанты которой должны входить в тупиковое или минимальное покрытие функции.

Далее простые импликанты, соответствующие строкам, которые входят в тупиковое или минимальное покрытие, соединяются знаками дизъюнкции и формируют ТДНФ или МДНФ булевой функции.

П р и м е р. Необходимо минимизировать логическую функцию четырех переменных $y = \vee(1,2,4,5,7,8,10,12,14,15)$ методом Квайна и Мак-Класки.

Р е ш е н и е. Отыщем простые импликанты в соответствии с этапом 1 алгоритма минимизации булевой функции:

$$K_0 = \left\{ \begin{array}{c} 0001 \\ 0010 \\ 0100 \\ 0101 \\ 0111 \\ 1000 \\ 1010 \\ 1100 \\ 1110 \\ 1111 \end{array} \right\} \begin{array}{l} 1 \\ 2 \\ 4 \\ 5 \\ 7 \\ 8 \\ 10 \\ 12 \\ 14 \\ 15 \end{array}$$

I гр	II гр	III гр	IV гр
0001	*0101*	*0111*	*1111*
0010	*1010*	*1110*	
0100	*1100*		
1000			

Осуществляется склеивание нулевых кубов:

0_01	*01_1*	*_111*
_010	*1_10*	*111_*
*010_*	*11_0*	
_100		
10_0		
1_00		

1_ _0
1_ _0

Осуществляется склеивание единичных кубов. Покрытие данной функции имеет вид:

$$\Pi(y) = \left\{ \begin{matrix} 0\_01 \\ \_010 \\ 010\_ \\ \_100 \\ 01\_1 \\ 1\_\_0 \\ \_111 \\ 111\_ \end{matrix} \right\}$$

Далее строится таблица покрытий функции *y*:

	0001	*0010*	*0100*	*0101*	*0111*	*1000*	*1010*	*1100*	*1110*	*1111*
1 0_01	*1*			*1*						
2 _010		*1*					*1*			
*3 010_*			*1*	*1*						
4 _100			*1*					*1*		
5 01_1				*1*	*1*					
6 1_ _0						*1*	*1*	*1*	*1*	
7 _111					*1*					*1*
*8 111_*									*1*	*1*

В ядро Квайна входят первая, вторая и шестая импликанты. Отмечаем столбцы, соответствующие конституентам, покрываемым этими импликантами (т.е. закрашиваем их). Далее строим таблицу, содержащую незакрашенные столбцы и импликанты, не вошедшие в ядро Квайна.

Далее производится операция поглощения строк: четвертая строка поглощается третьей строкой или наоборот. Пятая строка поглощается седьмой и восьмая строка поглощается седьмой строкой. После этого производится операция поглощения столбцов: десятый столбец поглощается 5-м в новой таблице.

	3 *0100*	*5* *0111*	*10* *1111*
*3 010_*	*1*		
4 _100	~~*1*~~		
5 01_1		~~*1*~~	
7 _111		*1*	*1*
*8 111_*			~~*1*~~

	3	*5*	*10*
3	*1*		
7		*1*	~~*1*~~

	3	*5*
3	*1*	
7		*1*

Окончательно получается следующая диагональная матрица-таблица, импликанты которой входят в минимальное или тупиковое покрытие исходной формы:

$$\Pi(y_{МДНФ}) = \begin{Bmatrix} 1 \\ 2 \\ 6 \\ 3 \\ 7 \end{Bmatrix} = \begin{Bmatrix} 0\_01 \\ \_010 \\ 010\_ \\ 1\_\_0 \\ \_111 \end{Bmatrix}$$

Рассмотрим еще один пример минимизации булевой функции.

П р и м е р. Необходимо минимизировать булеву функцию четырех переменных $y = \vee(0,1,4,5,6,7,9,11,12,15)$ методом Квайна и Мак-Класки.

Р е ш е н и е. Отыщем простые импликанты в соответствии с этапом 1 алгоритма минимизации

«0»	«1»	«2»	«3»	«4»
0000	*0001*	*0101*	*0111*	*1111*
	0100	*0110*	*1011*	
		1001		
		1100		

Осуществим склеивание нулевых кубов

*000_*	*0_01*	*01_1*	*_111*
0_00	*_001*	*011_*	*1_11*
	*010_*	*10_1*	
	01_0		
	_100		

Далее осуществим склеивание единичных кубов.

*0_0_*	*01_ _*
*0_0_*	*01_ _*

Построим таблицу покрытий заданной функции *y*.

	1 *0000*	*2* *0001*	*3* *0100*	*4* *0101*	*5* *0110*	*6* *0111*	*7* *1001*	*8* *1011*	*9* *1100*	*10* *1111*	
1. _001		*1*					*1*				
2._100			*1*						*1*		*
3.10_1							*1*	*1*			
4._111						*1*				*1*	
5.1_11								*1*		*1*	
*6.0_0_*	*1*	*1*	*1*	*1*							*
7.01_ _			*1*	*1*	*1*	*1*					*

Легко видеть, что в ядро Квайна должны входить импликанты *_100, 0_0_, 01_ _* (им соответствуют строки 2, 6, 7 таблицы покрытий). Следовательно, необходимо исключить строки 2, 6, 7 из дальнейшего рассмотрения (вычеркнуть). Далее, анализируя зачеркнутые строки, выявляем столбцы, в которых имеются единицы, и исключаем их из рассмотрения (вычеркиваем).

	7	*8*	*10*
1	*1*		
3	*1*	*1*	
4			*1*
5		*1*	*1*

Это столбцы 1, 2, 3, 4, 5, 6, 9. В результате получаем таблицу, состоящую из четырех строк и трех столбцов. Проведем сжатие полученной таблицы. Сжатие по столбцам невозможно, поэтому проведем сжатие по строкам. Первая строка поглощается третьей. Четвертая строка поглощается пятой. Получаем таблицу, состоящую из двух строк и трех столбцов.

	7	*8*	*10*
3	*1*	*1*	
5		*1*	*1*

Далее проводим сжатие по столбцам. Из таблицы вычеркивается восьмой столбец, поскольку он поглощается седьмым и десятым. После этого получается неприводимая таблица. Таким образом, минимальное покрытие функции *y* имеет вид:

$$\Pi(y_{МДНФ}) = \begin{Bmatrix} 3 \\ 5 \\ 2 \\ 6 \\ 7 \end{Bmatrix}$$

На основании покрытия записываем МДНФ:

$$y = x_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_4 \vee x_1 x_3 x_4 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_1 x_2$$

10.3. АЛГОРИТМ ПЕТРИКА

Теперь перейдем к построению алгоритма для нахождения всех тупиковых дизъюнктивных нормальных форм заданной функции. Этот алгоритм называется алгоритмом Петрика. Будем исходить из покрытия множества $K_0(f)$ функции f системой всех его максимальных кубов $K^1, K^2, ..., K^m$. Пусть $K_0(f) = \{P_1, ..., P_\lambda\}$, где P_j - некоторая точка 0-комплекса булевой функции *f*. (Считается, что $f \not\equiv 1$). Составляется таблица [3], в которой

$$\sigma_{ij} = \begin{cases} 0,\ если P_j \notin K^i & (i=1,2,...,m) \\ 1,\ если P_j \in K^i & (j=1,2,...,\lambda) \end{cases}$$

	P_1	…	P_j	…	P_λ
K^1	σ_{11}	…	σ_{1j}	…	$\sigma_{1\lambda}$
…	…	…	…	…	…
K^i	σ_{i1}	…	σ_{ij}	…	$\sigma_{i\lambda}$
…	…	…	…	…	…
K^m	σ_{m1}	…	σ_{mj}	…	$\sigma_{m\lambda}$

Очевидно, что в каждом столбце содержится хотя бы одна единица. Для каждого j $(1 \le j \le \lambda)$ находится множество E_j всех номеров строк, в которых столбец P_j содержит единицы. Пусть $E_j = \{e_{j1}, ..., e_{j\mu(j)}\}$. Составляется выражение $\mathop{\&}\limits_{j=1}^{\lambda}(e_{j1} \vee e_{j2} \vee ... \vee e_{j\mu(j)})$, и осуществляется преобразование $\& \vee \to \vee \&$. При этом символы $e_{j1}, e_{j2}, ..., e_{j\mu(j)}$ рассматриваются как булевы величины. После этого в полученном выражении ликвидируются поглощаемые или дублирующие члены, т.е. совершаются преобразования вида

$$A \cdot B \vee A = A$$

$$A \vee A = A$$

Получаем выражение $\vee \&'$, являющееся частью выражения $\vee \&$. Каждое слагаемое в $\vee \&'$ будет определять неприводимое покрытие.

ЛИТЕРАТУРА

1. Кузнецов О. П., Адельсон-Вельский Г. М. Дискретная математика для инженера. – М.: Энергоатомиздат, 1988.
2. Гаврилов Г. П., Сапоженко А. А. Задачи и упражнения по курсу дискретной математики. – М.: Наука. Гл. ред. физ.-мат. лит., 1992.
3. Яблонский С. В. Введение в дискретную математику. – М.: Наука. Гл. ред. физ.-мат. лит., 1986.
4. Бохманн Д., Постхоф Х. Двоичные динамические системы: Пер. с нем. – М.: Энергоатомиздат, 1986.
5. Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. Алгебра. Языки. Программирование. – К.: «Наукова думка», 1978.
6. Савельев П. В., Коняхин В. В. Математические основы проектирования комбинационных схем. – МИЭТ, М., 1989.
7. Проектирование цифровых вычислительных машин: Учебное пособие/Под ред. С. А. Майорова. – М.: Высшая школа, 1972.

Навчальне видання

Швачич Геннадій Григорович

Бартенєв Георгій Михайлович

Тімошкін Андрій Іванович

Толстой Віктор Володимирович

ОСНОВИ ДИСКРЕТНОЇ МАТЕМАТИКИ

Частина ІІ. Основи булевої алгебри.

Навчальний посібник

(російською мовою)

Тем. план 2013, поз. 247

Підписано до друку 18.06.2013. Формат 60x84 1/16. Папір друк. Друк плоский. Облік.-вид. арк. 2,94. Умов. друк. арк. 2,90. Тираж 100 пр. Замовлення №

Національна металургійна академія України
49600, м. Дніпропетровськ-5, пр. Гагаріна,4

Редакційно-видавничий відділ НМетАУ