ГЛАВА 4. ЛОГИКА

4.1 Логика высказываний

Высказывание – это утверждение или повествовательное предложение, которое может быть либо истинным, либо ложным.
Значением истинного высказывания является «И» – истина, ложного – «ложь».

Повелительные («Войдите, пожалуйста»), вопросительные («Который час?») и бессмысленные предложения («Сумма пяти и восемнадцати»), в которых ничего не утверждается, не являются высказываниями.

Предметом логики является анализ различных логических связей и методы построения на их основе правильных логических рассуждений.

Способы построения новых высказываний из заданных с помощью логических связок и способы установления истинности высказываний, построенных таким образом, изучаются в логике высказываний.

Основные логические связки – это связки: ***и, или, не, если … то…***, которые в логике высказываний имеют специальные названия и обозначения. Иногда к ним добавляют еще две связки ***либо …, либо … (или …, или …); если, и только если (тогда и только тогда).***

Для одной и той же связки в разных источниках используются разные названия и обозначения, которые приведены в таблице 4.1.

Таблица 4.1

Связка	Название	Обозначение	Высказывание, полученное с помощью связки	Математическая запись
1. ***и***	конъюнкция (или логическое умножение)	&, $\wedge$, $\cdot$	A ***и*** B	$A \& B$, $A \wedge B$, $A \times B, AB$
2. ***или***	дизъюнкция	$\vee$, +	A ***или*** B	$A \vee B$, $A + B$
3. ***не***	отрицание, инверсия	$\neg$, —	***не*** A	$\bar{A}, \neg A$
4. ***если …, то …***	импликация	$\rightarrow$, $\supset$	***если*** A, ***то*** B (A ***влечет*** B)	$A \rightarrow B$, $A \supset B$
5. ***либо …, либо …***	исключающее «или», неравнозначность	$\oplus$, Δ, $\neq$	***либо*** A, ***либо*** B	$A \oplus B$, $A \Delta B$
6. ***если и только если***	эквивалентность, равнозначность	$\equiv$, ~, $\leftrightarrow$	A, ***если и только если*** B	$A \equiv B$, $A \sim B$

В последней колонке таблице 4.1 записаны *формулы*, или выражения логики высказываний. С помощью букв *A, B, C, ...* обозначающих высказывания, связок и скобок можно построить разнообразные формулы.

Исследование свойств таких формул и способов установления их истинности и является основным предметом логики высказываний.

Существуют два подхода к построению логики высказываний, которые образуют два варианта этой логики: ***алгебру логики*** и ***исчисление высказываний***.

4.1.1 Алгебра логики. Булевы функции.

Алгебра логики рассматривает *логические формулы* как алгебраические выражения, которые можно преобразовать по определенным правилам. Знаки операций обозначают логические операции (логические связки).

В формулах алгебры логики *переменные* – это высказывания. Они принимают только два значения – *ложь* и *истина*, которые обозначаются либо 0 и 1, либо Л и И, либо *false* и *true*.

<u>Двоичное (бинарное) множество</u> $B = \{0,1\}$, где логический символ 0 означает – «ложь», логический символ 1 означает – «истина».

Каждая формула задает ***логическую функцию (булева функция)***: функцию от логических переменных, которая сама может принимать только два логических значения.

Таким образом функция n аргументов f есть $f: B^n \to B$.

<u>Единичным набором</u> значений аргументов называется набор, на котором функция равна 1. Множество единичных наборов функции f называется ***единичным множеством***.

<u>Нулевым набором</u> значений аргументов называется набор, на котором функция равна 0. Множество нулевых наборов функции f называется ***нулевым множеством***.

Множество всех булевых функций (функций алгебры логики) n переменных будем обозначать $P_2(n)$.

Все функции n переменных можно пронумеровать. Количество всех булевых функции n переменных находится по формуле $|P_2(n)| = 2^{2^n}$.

<u>Например</u>,

булевых функции 1 переменной $|P_2(1)| = 2^{2^1} = 4$,

булевых функции 2 переменных $|P_2(1)| = 2^{2^2} = 16$, булевых функции 3 переменных $|P_2(1)| = 2^{2^3} = 256$.

Булев вектор — это последовательность булевых констант, называемых *компонентами* булева вектора. Договоримся обозначать булевы векторы греческими буквами, а компоненты вектора – латинскими с указанием номеров компонент.

<u>П р и м е р ы.</u> $\alpha = a_1a_2\ldots a_6 = 010101, \beta = b_1b_2\ldots b_8 = 11110000$.

Длиной булева вектора назовем количество его компонент, а *весом* вектора – количество компонент, равных единице.

<u>П р и м е р</u>. Длина булева вектора $\alpha = 101010$ равна шести, а вес – трем.

<u>Способы задания булевых функций</u>

1) Задание булевой функции таблицей истинности.

Функция задается в виде таблицы истинности, представляющей собой совокупность всех наборов переменных и соответствующих им значений функции.

Булева функция n переменных может быть задана таблицей, состоящая из двух частей: в левой части перечисляются все наборы значений аргументов в естественном порядке, то есть по возрастанию значений чисел, представляемых этими векторами, а в правой части – значения булевой функции на соответствующих наборах и 2^n строк (или из двух строк и 2^n столбцов, в зависимости от удобства размещения таблицы).

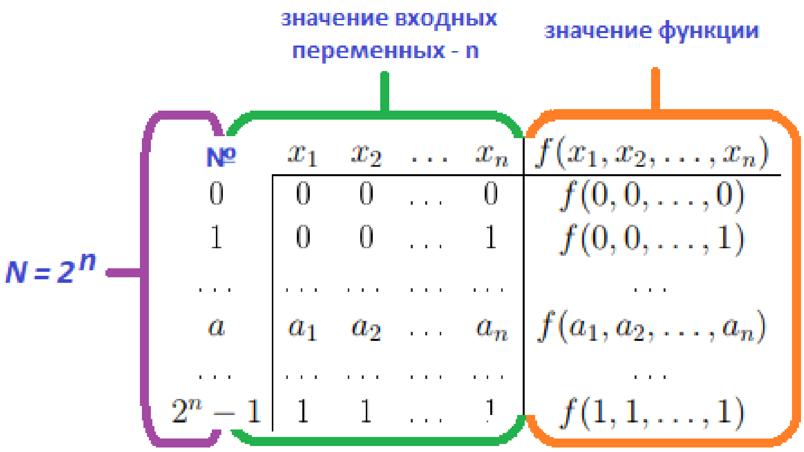


Рисунок 4.1. Задание булевой функции таблицей истенности

<u>П р и м е р</u>:

Таблица функций $f(x)$ одной переменной:

x	<u>Константа 0:</u> $f(x)=0$	<u>Тождество:</u> $f(x)=x$	<u>Отрицание:</u> $f(x)=\overline{x}$	<u>Константа 1:</u> $f(x)=1$
0	0	0	1	1
1	0	1	0	1

Таблица функций двух переменных $f(x_1,x_2)$, соответствующих основным логическим связкам:

Таблица 4.2

		Дизъюнкция	Конъюнкция	Импликация	Эквивалентность (равнозначность)	Неравнозначность (сложение по модулю 2)	Штрих Шеффера (НЕ – И)	Стрелка Пирса (НЕ – ИЛИ)
x_1	x_2	$x_1 \vee x_2$	$x_1 \wedge x_2$	$x_1 \rightarrow x_2$	$x_1 \sim x_2$	$x_1 \oplus x_2$	$x_1 \mid x_2$	$x_1 \downarrow x_2$
0	0	0	0	1	1	0	1	1
0	1	1	0	1	0	1	1	0
1	0	1	0	0	0	1	1	0
1	1	1	1	1	1	0	0	0

Таблица функций трех переменных $f(x_1,x_2,x_3)$, называемая мажоритарной (или функцией голосования): она принимает значение 1 на тех и только тех наборах, в которых единиц больше, чем нулей (major – больший).

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

Значение любой логической формулы, содержащей знаки этих функций, на заданном наборе значений переменных можно вычислить, используя эти таблицы.

<u>Интерпретацией</u> формулы логики высказываний называется набор значений высказываний, входящих в нее.

2) Задание булевой функции характеристическими множествами.

Так называются два множества:

M_f^1, состоящее из всех наборов, на которых функция принимает значение 1, то есть $M_f^1 = \{\alpha \in B^n : f(\alpha)=1\}$;

M_f^0, состоящее из всех наборов, на которых функция принимает значение 0, то есть $M_f^0 = \{\alpha \in B^n : f(\alpha) = 0\}$.

Пример (для функции $y = f(x_1, x_2, x_3)$ заданной таблично).

$M_f^1 = \{011, 101, 110, 111\}$, $M_f^0 == \{000, 001, 110, 100\}$.

3) Задание булевой функции координатным методом (карта Карно, диаграмма Вейча, матрица Грея).

При этом способе дискретный автомат задается с помощью карты его состояний, которая известна как карта Карно или диаграмма Вейча или матрица Грея.

Карта Карно или диаграмма Вейча — это специального вида таблица, которая позволяет упростить процесс поиска минимальных форм и успешно применяется, когда число переменных не превосходит шести. На рисунке 4.2 представлены карты двух, трёх и четырёх переменных. Расположение групп переменных x_i не имеет значения, необходимо лишь, чтобы каждая ячейка отличалась от любой соседней лишь на одну переменную. Согласно принятой форме построения карт соседними ячейками также считаются ячейки первой и последней строк, ячейки первого и последнего столбцов. Число ячеек карты равно числу возможных комбинаций значений логических переменных и в каждую ячейку записывается значение логической функции, соответствующее данному набору переменных [20].

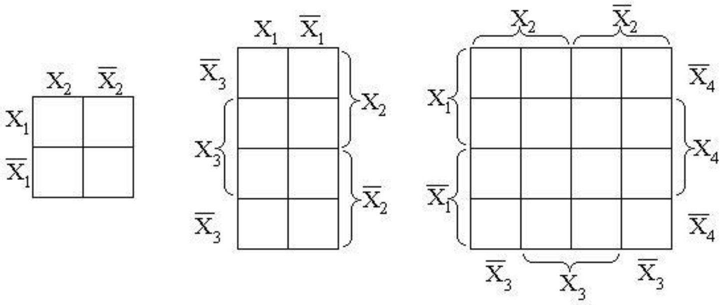


Рисунок 4.2. Карты двух, трёх и четырёх переменных.

Матрицей в коде Грея. Булево пространство размерности n представляется матрицей, состоящей из 2^s строк и 2^p столбцов, где s и p – целые числа, такие что $s + p = n$ и $s = p$ либо $s = p - 1$. Строкам матрицы поставлены в соответствие булевы векторы длины s (их называют *кодами строк*), а столбцам – булевы векторы длины p *(коды столбцов)* [20].

Коды столбцов упорядочены по следующему принципу:

– младшая компонента кодов, то есть компонента с меньшим номером, равна 0 в первой половине столбцов и равна 1 во второй их половине (например, если столбцов восемь, то младшая компонента принимает значения 00001111);

– следующая компонента равна 0 в первой четверти кодов и равна 1 во второй четверти, после чего значения симметрично повторяются, то есть равны 1 в третьей четверти и 0 в четвертой (в примере: 00111100) – симметрирование происходит в момент, когда предыдущая компонента меняет свое значение с 0 на 1;

– следующая компонента равна 0 в первой осьмушке кодов и равна 1 во второй их осьмушке, после чего ее значения дважды симметрично повторяются (в примере: 01100110) – симметрирование происходит в моменты, когда вторая компонента, а затем первая, меняют свои значения с 0 на 1;

– и так далее (за 1/8 следуют 1/16, 1/32, …).

Коды строк строятся аналогично.

Элемент матрицы в коде Грея, стоящий в i-й строке и j-м столбце, задает булев вектор, который получается приписыванием к коду строки i кода столбца j.

П р и м е р. Пусть $n = 5$. На левой матрице показан процесс построения кодов столбцов. Выделенная клетка задает булев вектор 10011 [20].

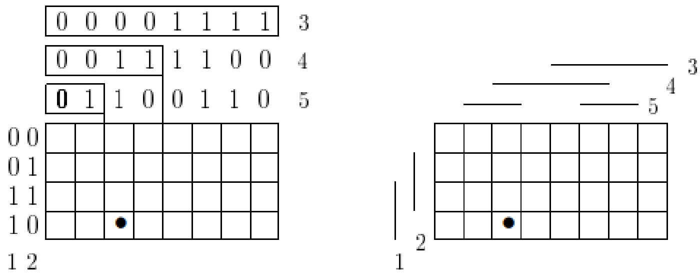


Рисунок 4.3. Процесс построения кодов столбцов.

Договоримся изображать коды условно: единицу – черточкой, а ноль – ее отсутствием: такой код более нагляден, да и быстрее рисуется (он показан на правой матрице предыдущего примера).

Алгоритм рисования кода Грея для столбцов матрицы [20].

Начало: задана матрица из 2^p столбцов.

Шаг 1: отступив от края матрицы один столбец, нарисуем черточку над двумя столбцами, затем, пропустив два столбца, нарисуем черточку над двумя следующими столбцами и так далее до конца матрицы.

Шаг 2: отступив от полученного ряда черточек немного вверх, начнем следующий ряд: нарисуем черту от середины первой до середины второй черточки предыдущего ряда, затем – от середины третьей до середины четвертой черточки предыдущего ряда и так далее до конца матрицы. Повторим шаг 2, пока это возможно (последнюю черту, которая начнется с середины матрицы, оборвем у правого края).

Конец.

П р и м е р. Матрица содержит 16 столбцов.

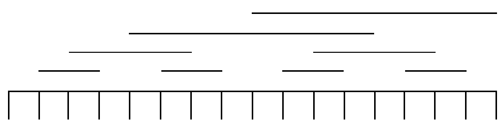


Рисунок 4.4. Алгоритм рисования кода Грея.

Далее будем называть матрицу в коде Грея матрицей Грея.

4) Графическое обозначение булевых операций.

Для изображения логических схем используются условные графические обозначения (УГО) элементарных логических функций или булевых операций, описывающие только выполняемую элементами функцию и не зависящие от его схемы

В настоящее время в мире существует несколько общепринятых стандартов условных обозначений. Наиболее распространенными являются американский стандарт milspec 806B и стандарт МЭК 117 – 15А, созданный Международной Электротехнической
Комиссией.

5) Задание булевой функции формулами.

4.1.2 Основные свойства и законы логических операций

1. Ассоциативность: а) $x_1(x_2x_3)=(x_1x_2)x_3$; б) $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$.

2. Коммутативность: а) $x_1x_2 = x_2x_1$; б) $x_1 \vee x_2 = x_2 \vee x_1$.

3. Дистрибутивность конъюнкции относительно дизъюнкции: $x_1(x_2 \vee x_3) = x_1x_2 \vee x_1x_3$.

4. Дистрибутивность дизъюнкции относительно конъюнкции:

$$x_1 \vee (x_2x_3) = (x_1 \vee x_2)(x_1 \vee x_3).$$

5. Идемпотентность: а) $xx = x$; б) $x \vee x = x$.

6. Двойное отрицание: $\overline{\overline{x}} = x$.

7. Свойства констант: а) $x \vee 1 = 1$; б) $x \vee 0 = x$; в) $x \wedge 1 = x$; г) $x \wedge 0 = 0$; д) $\overline{0} = 1$; е) $\overline{1} = 0$.

8. Правила де Моргана: а) $\overline{x_1x_2} = \overline{x}_1 \vee \overline{x}_2$; б) $\overline{x_1 \vee x_2} = \overline{x}_1\overline{x}_2$.

Эти свойства проверяются путем подстановки в обе части равенства поочередно всех наборов значений аргументов. С помощью основных свойств булевых операций доказываются следующие законы:

– закон поглощения

$$1)\ x \vee xy = x; \qquad 2)\ x \cdot (x \vee y) = x;$$

– закон простого склеивания

$$xy \vee x\overline{y} = x;$$

– закон расщепления

$$x = xy \vee x\overline{y};$$

– закон обобщенного склеивания

$$1)\ xz \vee y\overline{z} \vee xy = xz \vee y\overline{z}; \qquad 2)\ x \vee \overline{x}y = x \vee y.$$

Основные свойства булевых операций и законы применяются для ***эквивалентных преобразований*** одной равносильной формулы в другую.

4.1.3. Фиктивные переменные

Говорят, что булева функция $f(x_1, \ldots, x_i, \ldots, x_n)$ *существенно зависит от переменной* x_i, если выполняется условие

$$f(x_1, \ldots, x_i - 1, 0, x_i + 1, \ldots, x_n) \neq f(x_1, \ldots, x_i - 1, 1, x_i + 1, \ldots, x_n).$$

В этом случае также говорят, что переменная x_i *существенная*, в противном случае ее называют *фиктивной* переменной [20].

П р и м е р. Рассмотрим булеву функцию $f(x_1, x_2, x_3)$ и исследуем ее переменные x_1 и x_3.

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

x_2	x_3	$f(0, x_2, x_3)$	$\neq f(1, x_2, x_3)$
0	0	0	1
0	1	0	1
1	0	1	0
1	1	1	0

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

Рисунок 4.5. Пример исследования переменных на фиктивность

Из таблиц истинности видно, что переменная x_1 функции $f(x_1, x_2, x_3)$ существенная, так как $f(0, x_2, x_3) \neq f(1, x_2, x_3)$. Переменная x_3 фиктивная, так как $f(x_1, x_2, 0) = f(x_1, x_2, 1)$.

Очевидно, что для выявления фиктивных переменных можно не строить в явном виде таблиц истинности левой и правой частей неравенства, а сравнивать соответствующие части вектора-столбца значений функции.

Алгоритм распознавания фиктивной переменной по таблице истинности [20].

– для переменной x_1 сравниваются половины столбца значений функции: верхняя и нижняя, так как именно в верхней половине $x_1 = 0$, а в нижней $x_1 = 1$, если они совпадают, то переменная x_1 фиктивна;

– для переменной x_2 сравниваются четвертины столбца в каждой половине, так как именно в верхних четвертинах $x_2 = 0$, а в нижних $x_2 = 1$, если четвертины в каждой половине совпадают, то переменная x_2 фиктивна;

– и так далее (за четвертинами следуют 1/8, 1/16, …).

П р и м е р. Для булевой функции из предыдущего примера переменная x_1 существенна, так как верхняя половина столбца значений функции (0011) не равна нижней половине (1100). Переменная x_2 существенна, так как четвертины уже в первой половине различаются (00 и 11). Переменная x_3 фиктивна, так как осьмушки во всех четвертинах равны (0 и 0, 1 и 1, 1 и 1, 0 и 0).

Выявление фиктивных переменных можно ускорить, используя следующее очевидное утверждение.

Достаточное условие отсутствия фиктивных переменных. Если вес вектора-столбца значений функции нечетен, то функция не может содержать фиктивных переменных.

Алгоритм удаления фиктивной переменной x_i состоит в вычеркивании из таблицы истинности всех строк, в которых $x_i = 0$ (или всех строк, в которых $x_i = 1$), и столбца x_i [20].

Если переменная x_i функции $f(x_1, \ldots, x_n)$ фиктивна, то на наборах, соседних по i компоненте, функция принимает одинаковые значения.

П р и м е р (функция та же). Применив алгоритм для удаления фиктивной переменой x_3 (таблица слева), получаем результат (таблица справа).

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

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

Рисунок 4.6. Пример распознания фиктивной переменной по таблице истинности

Алгоритм распознавания фиктивной переменной по матрице Грея (основан на свойстве симметрии кода Грея) [20].

Переменная фиктивна тогда и только тогда, когда точки на матрице расположены симметрично относительно осей этой переменной. Упрощенная матрица – это одна из ее симметричных половин.

Пример (функция та же и представлена на левой матрице). Переменная x_3 функции фиктивна. Справа показан результат ее удаления.

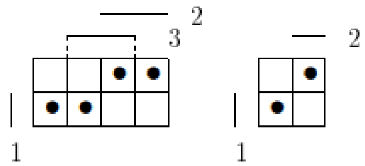


Рисунок 4.7. Пример распознания фиктивной переменной по матрице Грея

Булевы функции назовем *равными с точностью до фиктивных переменных*, если равны (в смысле, определенном ранее) функции, полученные из исходных удалением фиктивных переменных (и именно это расширенное толкование равенства функций мы будем иметь в виду во всех дальнейших рассуждениях).

4.1.4. Нормальные формы

4.1.4.1 ДНФ, КНФ

Пусть имеем множество переменных $X = \{x_1, x_2, \ldots, x_n\}$.

Элементарной конъюнкцией назовем конъюнкцию переменных множества X, в которую каждая переменная входит не более одного раза (с инверсией или без инверсии).

Примеры. Пусть $X = \{x_1, x_2, x_3, x_4\}$, тогда $x_1 x_2 x_4$, $x_1 x_3$, x_1, 1 являются элементарными конъюнкциями, а $x_1 x_1 x_2$, $x_1 x_2 x_2 x_4$ не являются.

Число переменных, образующих элементарную конъюнкцию, назовем ее ***рангом***.

<u>П р и м е р ы.</u> Ранги элементарных конъюнкций $x_1x_3x_4$ и 1 равны трем и нулю.

<u>Полной конъюнкцией</u> назовем элементарную конъюнкцию, состоящую из всех *n* переменных множества X, то есть конъюнкцию ранга n.

<u>П р и м е р.</u> При $n = 4$ конъюнкция $x_1 x_2 x_3 x_4$ – полная.

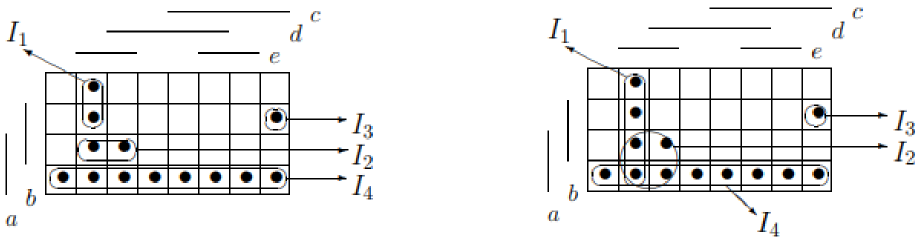
<u>Дизъюнктивной нормальной формой (ДНФ)</u> булевой функции $f(x_1, \ldots, x_n)$ назовем дизъюнкцию различных элементарных конъюнкций, задающую функцию $f(x_1, \ldots, x_n)$.

<u>П р и м е р.</u> $x_1x_2 \vee x_1x_2x_3 \vee x_1x_3x_4 = ДНФ$.

<u>Длиной</u> ДНФ назовем число ее конъюнкций, а ***рангом*** ДНФ – сумму рангов конъюнкций.

<u>П р и м е р.</u> Длина ДНФ из предыдущего примера равна трем, а ранг – восьми.

Пусть булева функция $f(x_1,\ldots,x_n)$ задана матрицей Грея. Выделим на ней достаточное множество интервалов. Запишем дизъюнкцию элементарных конъюнкций, заданных этими интервалами – получим ДНФ$_f$.



$$ДНФ_f = \underbrace{\bar a \bar c \bar d e}_{K_1} \vee \underbrace{a b \bar c e}_{K_2} \vee \underbrace{\bar a b c \bar d \bar e}_{K_3} \vee \underbrace{a \bar b}_{K_4}.$$

$$ДНФ'_f = \underbrace{\bar c \bar d e}_{K_1} \vee \underbrace{a \bar c e}_{K_2} \vee \underbrace{\bar a b c \bar d \bar e}_{K_3} \vee \underbrace{a \bar b}_{K_4}.$$

Рисунок 4.8. ДНФ по матрице Грея

<u>Элементарной дизъюнкцией</u> называется дизъюнкция переменных или их отрицаний, в которой каждая переменная встречается не более одного раза.

<u>Конъюнктивная нормальной формой (КНФ)</u> называется конъюнкция элементарных дизъюнкций.

<u>Элементарной конъюнкцией</u> называется конъюнкция переменных или их отрицаний, в которой каждая переменная встречается не более одного раза.

4.1.4.2 Представление логических функций в виде СДНФ, СКНФ

Дизъюнктивная форма будет ***совершенной (СДНФ),*** если каждая элементарная конъюнкция содержит все наименования переменных.

Правило получения СДНФ из вектор-столбца

1. Выбрать все единичные наборы значений аргументов.
2. Каждому единичному набору поставить в соответствие элементарную конъюнкцию всех переменных так, чтобы в конъюнкции переменная была с отрицанием, если в наборе она равна 0.
3. Соединить полученные элементарные конъюнкции знаком дизъюнкции.

П р и м е р. Применим алгоритм к функции (с переименованными аргументами) $f(x_1, x_2, x_3) = x_2 \sim x_1 x_3 \to x_2\, x_3$.

x_1	x_2	x_3	$f(x_1,x_2,x_3)$	
0	0	0	1	$x_1^0 x_2^0 x_3^0 = \overline{x}_1 \overline{x}_2 \overline{x}_3$
0	0	1	1	$x_1^0 x_2^0 x_3^1 = \overline{x}_1 \overline{x}_2 x_3$
0	1	0	0	
0	1	1	1	$x_1^0 x_2^1 x_3^1 = \overline{x}_1 x_2 x_3$
1	0	0	0	
1	0	1	1	$x_1^1 x_2^0 x_3^1 = x_1 \overline{x}_2 x_3$
1	1	0	1	$x_1^1 x_2^1 x_3^0 = x_1 x_2 \overline{x}_3$
1	1	1	1	$x_1^1 x_2^1 x_3^1 = x_1 x_2 x_3$

Рисунок 4.9. Получение СДНФ из вектор-столбца

Соединив полученные конъюнкции знаками дизъюнкции, имеем

$$СДНФ_f = \overline{x}_1\overline{x}_2\overline{x}_3 \vee \overline{x}_1\overline{x}_2 x_3 \vee \overline{x}_1 x_2 x_3 \vee x_1\overline{x}_2 x_3 \vee x_1 x_2 \overline{x}_3 \vee x_1 x_2 x_3 .$$

Конъюнктивная форма будет ***совершенной (СКНФ),*** если каждая элементарная дизъюнкция содержит все наименования переменных.

Правило получения СКНФ из вектор-столбца

1. Выбрать все нулевые наборы значений аргументов.
2. Каждому нулевому набору поставить в соответствие элементарную дизъюнкцию всех переменных так, чтобы в дизъюнкции переменная была с отрицанием, если в наборе она равна 1.
3. Соединить полученные элементарные дизъюнкции знаком конъюнкции.

П р и м е р. Применим алгоритм к рассмотренной в предыдущих примерах функции $f(x_1, x_2, x_3) = x_2 \sim x_1 x_3 \to x_2\, x_3$.

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	$x_1^1 \vee x_2^0 \vee x_3^1 = x_1 \vee \overline{x}_2 \vee x_3$
0	1	1	1	
1	0	0	0	$x_1^0 \vee x_2^1 \vee x_3^1 = \overline{x}_1 \vee x_2 \vee x_3$
1	0	1	1	
1	1	0	1	
1	1	1	1	

Рисунок 4.10. Получение СКНФ из вектор-столбца

Соединив полученные дизъюнкции знаками конъюнкции, имеем
$СКНФ_f = (x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee x_3)$.

<u>*Алгоритм преобразования ДНФ в СДНФ*</u> (основан на законах склеивания $y = xy\ \bar{x}y$ и идемпотентности $x\ x = x$).

Начало: задана ДНФ$_f$ функции $f(x_1, \ldots, x_n)$.

Шаг 1: рассматриваем очередную конъюнкцию К исходной ДНФ$_f$. Если все конъюнкции исчерпаны, идем на конец.

Шаг 2: если конъюнкция K не является полной, то выбираем переменную x_i, которая не входит в K, и по закону склеивания заменяем K на дизъюнкцию двух конъюнкций: $K = Kx_i \vee K\bar{x}_i$ (в таком применении будем называть его законом *расклеивания соседних конъюнкций*). Если полученные слагаемые не являются полными конюнкциями, то применяем к каждой из них закон расклеивания (шаг 2) до тех пор, пока не получим из конъюнкции K дизъюнкцию полных конъюнкций. Идем на шаг 1.

Конец: на основании закона идемпотентности приводим подобные среди одинаковых конъюнкций – получаем СДНФ функции $f(x_1, \ldots, x_n)$.

<u>Пример.</u> Пусть $ДНФf = x_1x_2 \vee \bar{x}_1\bar{x}_2 \vee x_3$, $n = 3$. Применим закон расклеивания к каждой конъюнкции:

$$x_1x_2 = x_1x_2x_3 \vee x_1x_2\bar{x}_3$$

$$\bar{x}_1\bar{x}_2 = \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3$$

$$x_3 = x_1x_3 \vee \bar{x}_1x_3 = \underline{x_1x_2x_3} \vee x_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3 \vee \underline{\bar{x}_1\bar{x}_2x_3}$$

В последней строке подчеркнуты конъюнкции, совпадающие с полученными ранее. По закону идемпотентности они не войдут в совершенную ДНФ, которая примет вид

$$СДНФ_f = x_1x_2x_3 \vee x_1x_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3$$

4.1.4.3. Сокращенная, тупиковая, кратчайшая, минимальная и безызбыточная ДНФ

Элементарная конъюнкция K называется простой импликантой функции $f(x_1, \ldots, x_n)$,если она является импликантой этой функции, и не существует другой конъюнкции K', которая является импликантой функции $f(x_1, \ldots, x_n)$ и поглощает конъюнкцию K.

ДНФ, состоящая из всех простых импликант булевой функции $f(x_1,\ldots,x_n)$, называется ***сокращенной ДНФ*** этой функции (или *СокрДНФf*).

Преобразования ДНФ в сокращенную ДНФ на основе матриц Грея.

<u>Примеры</u>. Пусть задана функция $f(x,y,z)$ задана матрицей Грея (рис. 4.11) [20].

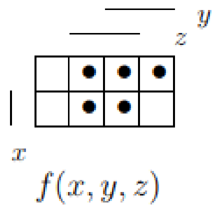
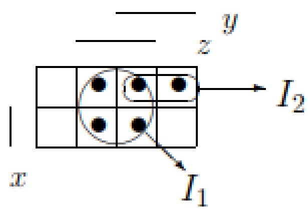


Рисунок 4.11. Пример – исходная функция $f(x,y,z)$ заданная матрицей Грея

Найдем сокращенные ДНФ функции $f(x,y,z)$



$$\text{СокрДНФ}_f = \underset{K_1}{z} \vee \underset{K_2}{\bar{x}y}.$$

Рисунок 4.12. Пример – сокращенная ДНФ функции $f(x,y,z)$

Даже если ДНФ сокращенная, ее можно минимизировать, то есть уменьшить количество входящих в нее элементарных конъюнкций.

<u>Пример.</u> Пусть сокращенная ДНФ функции имеет вид:

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

Тогда ее единичное множество может быть представлено в виде:

$$M_f = M_{xy} \cup M_{\bar{y}z} \cup M_{xz} = \{110,\ 111\} \cup \{001, 101\} \cup \{101, 111\}.$$

Заметим, что наборы, входящие в последнее подмножество, находятся так же в первом и во втором. Следовательно, *импликант* xz *– лишний.*

Сокращенная ДНФ, из которой удалены все лишние импликанты, называется ***<u>тупиковой</u>.***

Отношение покрытия между единичными наборами и импликантами ДНФ наглядно задается *таблицей покрытия.*

Строки таблицы соответствуют конъюнкциям ДНФ, столбцы – элементам единичного множества. На пересечении строки и столбца ставится пометка, если данная конъюнкция обращается в единицу данным набором значений аргументов (иначе говоря, данный *набор покрывается единичным множеством конъюнкции*).

<u>Пример.</u> Пусть ДНФ функции имеет вид:

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

Тогда ее единичное множество может быть представлено в виде:

$$M_f = M_y \cup M_{yz} = \{010, 011, 110, 111\} \cup \{011, 111\} = \{010, 011, 110, 111\} .$$

Построим таблицу покрытия:

	010	011	110	111
y	•	•	•	•
$\bar{y}z$	•	•		

Из таблицы видно, что вторая строчка – лишняя, то есть если ее убрать, все элементы единичного множества останутся покрыты. Значит, импликант $\bar{y}z$ – лишний. Таким образом, ДНФ можно упростить, убрав лишний импликант. Она приобретает вид $f(x,y,z) = y$, и в данном случае является тупиковой, так как оставшийся импликант – простой.

Метод Блейка-Порецкого – метод получения сокращенной ДНФ, содержащей все простые импликанты.

Пусть дана СДНФ функции [20].

1. Перенумеруем элементарные конъюнкции.

2. Осуществим попарно склеивание каждой конъюнкции с каждой, если это возможно. Под полученными конъюнкциями будем фиксировать номера.

3. Допишем к списку полученных конъюнкций те, которые не участвовали в склеивании (их номера не фиксировались).

4. Вернемся к п.1.

В результате получим сокращенную ДНФ, содержащую все простые импликанты.

<u>П р и м е р.</u> Дана СДНФ вида:

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

Получить с помощью метода Блейка-Порецкого сокращенную ДНФ, содержащую все простые импликанты.

П. 1. $f(x,y,z) = \underset{1}{\bar{x}\bar{y}\bar{z}} \vee \underset{2}{\bar{x}\bar{y}z} \vee \underset{3}{x\bar{y}\bar{z}} \vee \underset{4}{xy\bar{z}} \vee \underset{5}{xyz}$;

П. 2, 3. $f(x,y,z) = \underset{1,2}{\bar{x}\bar{y}} \vee \underset{1,3}{\bar{y}\bar{z}} \vee \underset{3,4}{x\bar{z}} \vee \underset{4,5}{xy}$;

П.4. $f(x,y,z) = \underset{1}{\bar{x}\bar{y}} \vee \underset{2}{\bar{y}\bar{z}} \vee \underset{3}{x\bar{z}} \vee \underset{4}{xy}$.

Так как больше склеивания произвести нельзя, сокращенная ДНФ имеет вид:

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

Построим таблицу покрытия:

	000	001	100	110	111
$\bar{x}\bar{y}$	•	•			
$\bar{y}\bar{z}$	•		•		
$x\bar{z}$			•	•	
xy				•	•

Из таблицы видно, что первую строку удалять нельзя, так как при этом останется не покрыт набор 001. Вторую строку можно удалить, так как наборы, составляющие единичное множество конъюнкции $\bar{y}\bar{z}$, содержаться также в единичных множествах других конъюнкций. Аналогично с третьей строкой. Последнюю строку также нельзя удалить, так как если это сделать, останется не покрыт набор 111.

Таким образом, дана функция имеет 2 тупиковые ДНФ:

$$F_1(x,y,z)=\bar{x}\bar{y}\vee x\bar{z}\vee xy;$$

$$F_2(x,y,z)=\bar{x}\bar{y}\vee \bar{y}\bar{z}\vee xy.$$

П р и м е р 2.

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

П. 1. $f(x,y,z)=\underset{1}{\bar{x}\bar{y}\bar{z}}\vee \underset{2}{x\bar{y}z}\vee \underset{3}{\bar{x}yz}\vee \underset{4}{xy\bar{z}}\vee \underset{5}{xyz}$;

П. 2, 3. $f(x,y,z)=\underset{2,5}{xz}\vee \underset{3,5}{yz}\vee \underset{4,5}{xy}\vee \underset{1}{\bar{x}\bar{y}\bar{z}}$;

П.4. $f(x,y,z)=\underset{1}{xz}\vee \underset{2}{yz}\vee \underset{3}{xy}\vee \underset{4}{\bar{x}\bar{y}\bar{z}}$;

Так как больше склеивания произвести нельзя, сокращенная ДНФ имеет вид: $f(x,y,z)=xy\vee yz\vee xz\vee \bar{x}\bar{y}\bar{z}$.

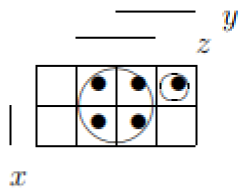
Построим таблицу покрытия:

	000	101	011	110	111
xy				•	•
yz			•		•
xz		•			•
$\bar{x}\bar{y}\bar{z}$	•				

Из таблицы видно, что при удалении любой ее строки появятся единичные наборы, не покрытые никаким единичным множеством. Следовательно, полученная сокращенная ДНФ является тупиковой.

Минимальной ДНФ (*МинДНФ$_f$*) булевой функции $f(x_1,\ldots,x_n)$ называется ДНФ наименьшего ранга из всех ДНФ, задающих функцию $f(x_1,\ldots,x_n)$.

Кратчайшей ДНФ (*КратДНФ$_f$*) булевой функции $f(x_1,\ldots,x_n)$ называется ДНФ наименьшей длины из всех ДНФ, задающих функцию $f(x_1,\ldots,x_n)$.



$$\text{КратДНФ}'_f = z \vee \overline{x}\, y\, \overline{z} \text{ (длины 2, ранга 4).}$$

Рисунок 4.13. Пример – кратчпйшая ДНФ функции $f(x,y,z)$

$$\text{МинДНФ}_f = \text{КратДНФ}_f = z \vee \overline{x}\, y \quad \text{(длины 2, ранга 3).}$$

ДНФ булевой функции $f(x_1,\ldots,x_n)$ называется ***безызбыточной ДНФ*** (Без ДНФ$_f$), если из нее нельзя удалить ни одной конъюнкции и ни одной переменной из конъюнкции так, чтобы она оставалась равносильной исходной ДНФ.

4.2 Алгебра Жегалкина

Полином Жегалкина (англ. Zhegalkin polynomial) — полином с коэффициентами вида 0 и 1, где в качестве произведения берётся конъюнкция, а в качестве сложения исключающее или.

Полином был предложен в 1927 году И. И. Жегалкиным в качестве средства для представления функций булевой логики.

Формально полином Жегалкина можно представить в виде

$$P(x_1 \ldots x_n) = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \ldots \oplus a_n x_n \oplus a_{12} x_1 x_2 \oplus a_{13} x_1 x_3 \oplus \ldots \oplus a_{1\ldots n} x_1 \ldots x_n$$

или в более формализованном виде как:

$$P(x_1 \ldots x_n) = a_0 \oplus \sum_{\substack{1<i_1<\ldots<i_k<n \\ 1\le k\le n}} a_{i_1\ldots i_k} x_{i_1\ldots i_k} = a_0 \oplus \sum_{\substack{1<i_1<\ldots<i_k<n \\ 1\le k\le n}} a_{i_1\ldots i_k} \prod_{j=1}^{k} x_{ij}$$

Заметим, что коэффициенты $a_{i_1\ldots i_k}$ принимают значения из множества $\{0,1\}$, причём если коэффициент равен нулю, то соответствующее слагаемое может быть опущено.

Полином Жегалкина, состоящий только из слагаемых с единичными коэффициентами (т. е. с опущенными слагаемыми с нулевыми коэффициентами), называется алгебраической нормальной формой (АНФ) соответствующей логической функции.

Алгеброй Жегалкина называется алгебра вида $AG = \left(P_2, \&, \oplus\right)$. В алгебре Жегалкина действуют тождества [20]:

1) коммутативность сложения по модулю 2

$$x \oplus y = y \oplus x;$$

2) ассоциативность сложения по модулю 2

$$(x \oplus y) \oplus z = x \oplus (y \oplus z);$$

3) дистрибутивность конъюнкции по отношению к сложению по модулю 2

$$(x \oplus y) \cdot z = xz \oplus yz,$$

4) $x \oplus x = 0,\ x \oplus 0 = x$

а также все тождества, истинные для конъюнкции.

Теорема Жегалкина: Каждая булева функция единственным образом представляется в виде полинома Жегалкина.

Полином Жегалкина называется *линейным*, если его степень не превышает единицы.

П р и м е р ы. $P_1=1$, $P_2 = x \oplus y \oplus 1$ – линейные полиномы.

Существует несколько способов построения полинома Жегалкина [20].

1. *По таблице истинности*

Пусть для функции $f\left(x_1 ... x_n\right)$ построена или задана таблица истинности. Запишем сначала данную функцию в виде полинома Жегалкина с неопределёнными коэффициентами. Затем по очереди подставляем всевозможные наборы в порядке увеличения количества единиц и находим коэффициенты с учётом того, что

$$a \oplus 1 = \bar{a}, a\, a \oplus 0 = a.$$

За каждую подстановку находим только один коэффициент.

П р и м е р: Дана функция $f\left(x_1, x_2, x_3\right) = \left(x_1 x_2 \vee x_3\right) \rightarrow \overline{x_2}$

Для этого необходимо построить сначала таблицу истинности для функции:

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

0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Тогда для функции $f\left(x_1,x_2,x_3\right)$ от трех переменных полином Жегалкина в общем виде задается:

$$P(x_1,x_2,x_3)=a_0\oplus a_1x_1\oplus a_2x_2\oplus a_3x_3\oplus a_{12}x_1x_2\oplus a_{13}x_1x_3\oplus a_{23}x_2x_3\oplus a_{123}x_1x_2x_3$$

x_1	x_2	x_3	$f(x_1,x_2,x_3)$		
0	0	0	$f\left(0,0,0\right)$	1	$a_0=1$
0	0	1	$f\left(0,0,1\right)$	1	$a_0\oplus a_3=1\oplus a_3=1\Rightarrow a_3=0$
0	1	0	$f\left(0,1,0\right)$	1	$a_0\oplus a_2=1\oplus a_2=1\Rightarrow a_2=0$
0	1	1	$f\left(0,1,1\right)$	0	$a_0\oplus a_2\oplus a_3\oplus a_{23}=1\oplus 0\oplus 0\oplus a_{23}=0\Rightarrow a_{23}=1$
1	0	0	$f\left(1,0,0\right)$	1	$a_0\oplus a_1=1\oplus a_2=1\Rightarrow a_1=0$
1	0	1	$f\left(1,0,1\right)$	1	$a_0\oplus a_1\oplus a_3\oplus a_{13}=1\oplus 0\oplus 0\oplus a_{13}=1\Rightarrow a_{13}=0$
1	1	0	$f\left(1,1,0\right)$	0	$a_0\oplus a_1\oplus a_2\oplus a_{12}=1\oplus 0\oplus 0\oplus a_{12}=0\Rightarrow a_{12}=1$
1	1	1	$f\left(1,1,1\right)$	0	$a_0\oplus a_1\oplus a_2\oplus a_3\oplus a_{12}\oplus a_{13}\oplus a_{23}\oplus a_{123}=$ $=1\oplus 0\oplus 0\oplus 0\oplus 1\oplus 0\oplus 1\oplus a_{123}=0\Rightarrow a_{123}=1$

Таким образом, полином Жегалкина выглядит так:

$$f\left(x_1,x_2,x_3\right)=1\oplus x_1x_2\oplus x_2x_3\oplus x_1x_2x_3$$

2) *<u>Метод треугольника</u>*

Требуется построить полином Жегалкина для функции f. Для примера, в качестве функции f возьмем функцию из предыдущего примера:

$$f\left(x_1,x_2,x_3\right)=\left(x_1x_2\vee x_3\right)\to\overline{x_2}=\left(11101100\right).$$

Полином Жегалкина в общем виде задается:

Алгоритм построения:

Шаг 1. Строим таблицу значений функции (строки в таблице идут в порядке возрастания двоичных кодов). Таблицу лучше разместить в левой части листа.

Шаг 2. Построение треугольника.

2.1. Для этого берем вектор значения функции и выписываем его напротив первой строки таблицы:

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

1 1 1 0 1 1 0 0

Рисунок 4.14. Алгоритм построения полинома Жегалкина методом треугольника. Шаг 2.1 - Построение треугольника

2.2. Далее заполняем треугольник, складывая попарно соседние значения по модулю 2, результат сложения выписываем ниже.

Продолжаем вычисления, пока в строке не останется лишь одна цифра.

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

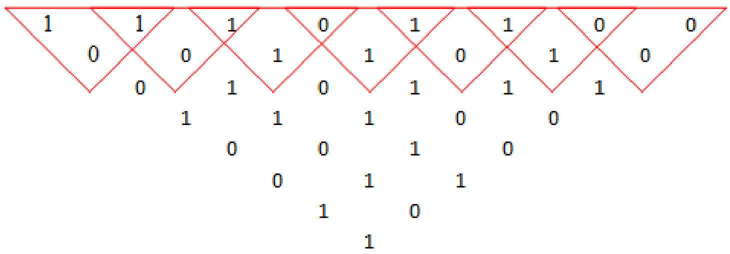


Рисунок 4.15. Алгоритм построения полинома Жегалкина методом треугольника. Шаг 2.2 - Построение треугольника

Шаг 3. Построение полинома Жегалкина.

Числа на левой стороне (выделены жирным шрифтом) треугольника есть коэффициенты полинома при монотонных конъюнкциях, соответствую наборам значений переменных.

Конъюнкции выписываются по двоичным наборам в левой части таблицы по принципу: если напротив переменной x_i стоит 1, то переменная входит в конъюнкцию; в противном случае переменная отсутствует в конъюнкции. Набору (0,0,0) соответствует константа 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	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

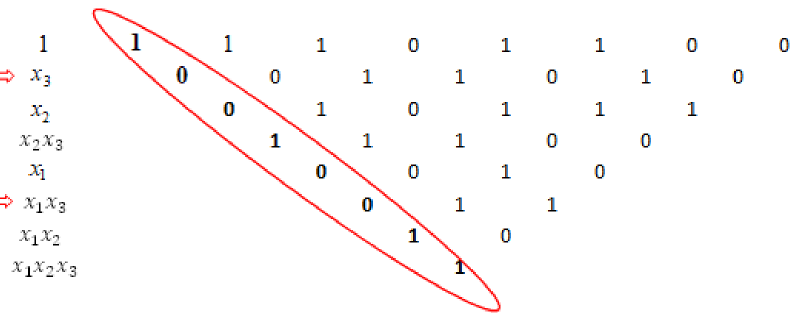


Рисунок 4.16. Алгоритм построения полинома Жегалкина методом треугольника. Шаг 3 - Построение полинома Жегалкина

Для построения полинома нужны только конъюнкции из строк с единицами на левой стороне треугольника. Это и есть конъюнкции, входящие в состав полинома Жегалкина. Осталось лишь выписать сам полином:

$$f\left(x_1, x_2, x_3\right) = 1 \oplus x_1x_2 \oplus x_2x_3 \oplus x_1x_2x_3$$

3) <u>Построение по СовДНФ</u>

Алгоритм построения полинома Жегалкина по СовДНФ (основан на доказательстве теоремы о существовании полинома Жегалкина) [20].

Начало. Задана совершенная ДНФ функции $f\left(x_1,\ldots,x_n\right)$.

Шаг 1. Заменяем каждый символ дизъюнкции на символ дизюнкции с исключением.

Шаг 2. Заменяем каждую переменную с инверсией x равносильной формулой $x\oplus 1$.

Шаг 3. Раскрываем скобки.

Шаг 4. Вычеркиваем из формулы пары одинаковых слагаемых.

Конец. Получен полином Жегалкина функции $f\left(x_1,\ldots,x_n\right)$.

<u>П р и м е р.</u> Найдем полином Жегалкина мажоритарной булевой функции по ее совершенной ДНФ.

$$СовДНФ = \bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz = \bar{x}yz \oplus x\bar{y}z \oplus xy\bar{z} \oplus xyz =$$

$$= \left(1\oplus x\right)yz \oplus x\left(1\oplus y\right)z \oplus xy\left(1\oplus z\right) \oplus xyz =$$

$$= yz \oplus \cancel{xyz} \oplus xz \oplus \cancel{xyz} \oplus xy \oplus \cancel{xyz} \oplus \cancel{xyz} = yz \oplus xz \oplus xy = P$$

4) <u>Построение по ДНФ</u>

Алгоритм построения полинома Жегалкина по ДНФ (основан на равносильности $K_1 \vee K_2 = K_1 \oplus K_2 \oplus K_1K_2$) [20].

Начало. Задана произвольная ДНФ функции $f\left(x_1,\ldots,x_n\right)$.

Шаг 1. Разбиваем ДНФ на пары конъюнкций, предпочтительно ортогональных (если число конъюнкций нечетно, одна из них остается без пары).

Шаг 2. Заменяем дизъюнкцию каждой пары конъюнкций $K_1 \vee K_2$ формулой $K_1 \oplus K_2 \oplus K_1K_2$ или формулой $K_1 \oplus K_2$, если K_1 и K_2 ортогональны.

Шаг 3. В полученной формуле находим очередную дизъюнкцию A_1 к A_2 и заменяем ее формулой $A_1 \oplus A_2 \oplus A_1A_2$. Повторяем шаг 3 до тех пор, пока это возможно.

Шаг 4. Заменяем каждую переменную с инверсией x равносильной формулой $x \oplus 1$,.

Шаг 5. Раскрываем скобки.

Шаг 6. Вычеркиваем из формулы пары одинаковых слагаемых.

Конец. Получен полином Жегалкина функции $f(x_1,\ldots,x_n)$.

Пример. Найдем полином Жегалкина мажоритарной функции по ДНФ.

$$ДНФ = xy\bar{z} \vee xz \vee yz = (xy\bar{z} \vee xz) \vee yz = (xy\bar{z} \oplus xz) \vee yz =$$

$$= (xy\bar{z} \oplus xz) \oplus yz \oplus (xy\bar{z} \oplus xz)yz = xy\bar{z} \oplus xz \oplus yz \oplus xyz =$$

$$= xy(1 \oplus z) \oplus xz \oplus yz \oplus xyz = xy \oplus \cancel{xyz} \oplus xz \oplus yz \oplus \cancel{xyz} = xy \oplus xz \oplus yz = P$$

4.3 Двойственность

Функция $f^*(x_1,x_2,\ldots,x_n) = \overline{f(\bar{x}_1,\bar{x}_2,\ldots,\bar{x}_n)}$ называется ***двойственной*** функцией к функции $f(x_1,x_2,\ldots,x_n)$.

Чтобы получить двойственную функцию из исходной, надо каждой переменной придать отрицание на самом нижнем уровне формулы, затем общее отрицание всей формуле (на самом верхнем уровне) и привести к ДНФ (упростить).

У двойственной функции на противоположных наборах принимаются противоположные значения: если $f(0,0,1)=1$, то $f^*(1,1,0)=0$.

Функция $f(x_1,x_2,\ldots,x_n)$ называется ***самодвойственной,*** если $f(x_1,x_2,\ldots,x_n) = f^*(x_1,x_2,\ldots,x_n)$.

Самодвойственными являются функции $x, \bar{x}, xy \vee xz \vee yz$.

Вектор-столбец самодвойственной функции антисимметричен относительно своей середины (при лексикографическом порядке аргументов).

Принцип двойственности

Если в формуле F, представляющей функцию f все знаки функций заменить на знаки двойственных функций, то получится формула F^*, представляющая функцию f^*, двойственную к f.

П р и м е р. Получить функцию, двойственную к $f(x,y,z)$ с помощью определения двойственной функции. Выяснить, является ли функция самодвойственной.

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

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

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

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

Вывод: функция не самодвойственна.

4.4 Монотонные функции

Булева функция $f(x_1,\ldots,x_n)$ называется *монотонной* (*принадлежит классу* M), если для любой пары наборов α и β таких, что $\alpha \le \beta$, выполняется условие $f(\alpha) \le f(\beta)$ (назовем его *условием монотонности*).

Рассмотрим задачу: выяснить будет ли функция f монотонной? Для решения этой задачи удобно начертить диаграмму частично упорядоченного множества B^3 и в вершинах этой диаграммы проставить значения функции $f(x_1,x_2,x_3)$. Такая диаграмма приведена на рисунке 4.17, значения функции $f(x_1,x_2,x_3)$ заключены в квадрат, чтобы их отличать от обозначения вершин множества B^3.

X_1	X_2	X_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

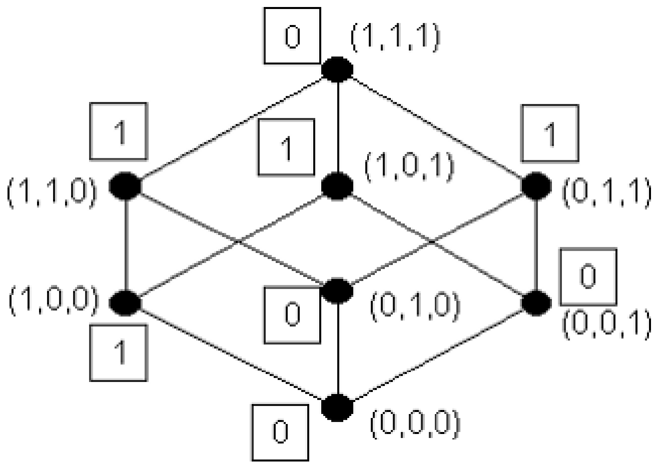


Рисунок 4.17. Диаграмма частично упорядоченного множества

Монотонность функции означает, что если в какой-то вершине диаграммы она принимает значение 1, то и всюду выше функция принимает то же значение 1. Функция $f(x_1,x_2,x_3)$ монотонной не является, так как в вершине (1,1,0) она принимает значение 1, а в вершине (1,1,1), которая выше первой, принимает значение 0.

Алгоритм распознавания монотонной булевой функции (основан на утверждении о условии немонотонности) [20].

Начало. Задана таблица истинности булевой функции.

Шаг 1. Сравниваем значения функции на наборах, соседних по первой переменной, то есть верхнюю половину столбца значений функции (вектор φ_1) с нижней половиной (вектор ψ_1). Если условие $\varphi_1 \le \psi_1$ нарушено, то функция не монотонна, идем на конец.

Шаг 2. Сравниваем значения функции на наборах, соседних по второй переменной, то есть верхние четвертины столбца значений функции (векторы φ'_2, φ''_2) с нижними четвертинами (векторами ψ'_2, ψ''_2) в каждой половине. Если хотя бы одно из условий $\varphi'_2 \le \varphi'_2$ и $\varphi''_2 \le \varphi''_2$ нарушено, то функция не монотонна, идем на конец.

Шаги $3 - n$. Аналогично сравниваем восьмые, шестнадцатые части, и так далее. Если ни одно из проверяемых условий не нарушено, то функция монотонна.

Конец.

<u>П р и м е р.</u> Рассмотрим две булевых функции (первая – мажоритарная).

x_1	x_2	x_3	$f(x_1,x_2,x_3)$	$g(x_1,x_2,x_3)$		
0	0	0	0	0		
0	0	1	0	1	φ_2'	
0	1	0	0	1	ψ_2'	
0	1	1	1	0		φ_1
1	0	0	0	0		ψ_1
1	0	1	1	1	φ_2''	
1	1	0	1	1	ψ_2''	
1	1	1	1	1		

Проверим на монотонность мажоритарную функцию. Сравниваем половины столбца значений: $\varphi_1 = 0001 \le 0111 = \psi_1$. Сравниваем четвертины: $\varphi'_2 = 00 \le 01 = \psi'_2$, $\varphi''_2 = 01 \le 11 = \psi''_2$. Сравниваем осьмушки : $0 \le 0$, $0 \le 1$, $0 \le 1$, $1 \le 1$. Следовательно, мажоритарная функция монотонна.

Проверим на монотонность функцию $g(x_1, x_2, x_3)$. Сравниваем половины столбца значений: $\varphi_1 = 0110 \le 0111 = \psi_1$. Сравниваем четвертины: так как $\varphi'_2 = 01$ не предшествует $\psi'_2 = 10$, функция $g(x_1, x_2, x_3)$ не монотонна.

Из элементарных булевых функций монотонными являются, например, конъюнкция и дизъюнкция. Не являются монотонными, например, штрих Шеффера и стрелка Пирса.

4.5 Линейные булевы функции

Булева функция называется ***<u>линейной</u>*** (*принадлежит классу L*), если ее полином Жегалкина линеен.

Полином Жегалкина линейной функции $f(x_1, \ldots, x_n)$ имеет вид:

$$P = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \ldots a_n x_n,$$

П р и м е р.

Мажоритарная функция не является линейной: степень ее полинома Жегалкина ($xy \oplus xz \oplus yz$) равна 2.

Из элементарных булевых функций линейными являются, например, инверсия и эквивалентность. Не являются линейными, например, штрих Шеффера и стрелка Пирса, $x \to y$, $x \vee y$, $x \wedge y$.

Докажем, например, нелинейность импликации $f(x,y) = x \to y$. Предположим противное. Пусть $f(x,y)$ – линейная функция. Это означает, что $x \to y = a_0 + a_1 x + a_2 y$ для некоторых $a_0, a_1, a_2 \in B$. Составим таблицу, задающую функции, стоящие в левой и правой части последнего равенства (см. таблицу).

Таблица

x	**y**	$x \to y$	$a_0 + a_1 x + a_2 y$	**Следствие**
0	0	1	a_0	$a_0 = 1$
0	1	1	$1 + a_2$	$a_2 = 0$
1	0	0	$1 + a_1$	$a_1 = 1$
1	1	1	0	Противоречие

Противоречие показывает, что $x \to y$ – нелинейная функция.

4.5.1 Класс булевых функций, сохраняющих константу 0

Булева функция *сохраняет константу 0* (*принадлежит классу* T^0*)*, если на наборе из всех нулей функция принимает значение ноль [20].

П р и м е р. Мажоритарная булева функция сохраняет константу 0 (принадлежит классу T^0):

X_1	X_2	X_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Из элементарных булевых функций классу T^0 принадлежат, например, конъюнкция и тождественная функция; не принадлежат классу T^0, например, штрих Шеффера и стрелка Пирса.

Утверждение о числе булевых функций класса T^0. Число различных булевых функций, зависящих от n переменных и сохраняющих константу 0, равно 2^{2n-1}.

4.5.2 Класс булевых функций, сохраняющих константу 1

Булева функция *сохраняет константу 1* (*принадлежит классу* T^1), если на наборе из всех единиц функция принимает значение единица [20].

П р и м е р. Мажоритарная булева функция сохраняет константу 1. Из элементарных булевых функций таковыми являются, например, дизъюнкция и конъюнкция. Не сохраняют константу 1, например, штрих Шеффера и стрелка Пирса.

Утверждение о числе булевых функций класса T^1. Число различных булевых функций, зависящих от n переменных и сохраняющих константу 1, равно 2^{2n-1}.

4.6 Функциональная полнота систем

Функционально полной называется такая система функций Σ, через функции которой можно выразить любую логическую функцию.

Множество функций N называется ***функционально полной системой*** (ФПС), если любая булева функция представима суперпозицией функций из N.

Под суперпозицией в отношении булевых функций понимается подстановка одних функций в другие вместо их аргумента.

Договоримся опускать аргументы при перечислении функций множества N и рассматривать термин "система" в данном контексте как синоним множества.

П р и м е р 1. Множество $\mathrm{N}_1 = \{\wedge, \vee, ^-\}$ является функционально полной системой, так как любую булеву функцию, кроме константы 0, можно представить совершенной ДНФ, то есть суперпозицией функций из N_1 а константу 0 – формулой $x\overline{x}$.

П р и м е р 2. Множество $\mathrm{N}_2 = \{\wedge, \oplus, 1\}$ является ФПС, так как любую булеву функцию можно представить полиномом Жегалкина, то есть суперпозицией функций из N_2, а полином 0 – формулой $1 \oplus 1$.

Функционально полной в слабом смысле называется такая система функций Σ, которая становится функционально полной, если к ней добавить константы 0 и 1.

Например, $\Sigma = \{\wedge, \oplus\}$ – функционально полна в слабом смысле. Эта система функций алгебры Жегалкина. Для того, чтоб с ее помощью можно было записать все полиномы Жегалкина, необходимо добавить константу 1. Это означает, что $\Sigma' = \Sigma \cup \{1\}$ – функционально полна (в сильном смысле).

Теорема 1 о функциональной полноте.

Для того чтобы система функций Σ была функционально полна в слабом смысле, необходимо и достаточно, чтобы она содержала хотя бы одну немонотонную и хотя бы одну нелинейную функцию.

Теорема 2 о функциональной полноте (теорема Поста).

Для того чтобы система функций Σ была функционально полна (в сильном смысле), необходимо и достаточно, чтобы она содержала

хотя бы одну немонотонную,

хотя бы одну нелинейную,

хотя бы одну несамодвойственную,

хотя бы одну не сохраняющую 0,

хотя бы одну не сохраняющую 1 функцию.

4.7 Исчисление высказываний

Исчисление высказываний - раздел математической логики, в котором формально-аксиоматическим методом изучаются сложные (составные) высказывания, составленные из простых (элементарных, не анализируемых) высказываний с помощью логических связок «и», «или», «если..., то» и «неверно, что».

Описание всякого исчисления включает в себя описание символов этого исчисления (алфавита), формул, являющихся конечными конфигурациями символов, и определение выводимых формул.

Алфавит исчисления высказываний состоит из символов трех категорий:

1) символы первой категории: $x, y, ..., x_1, x_2, ...$. Эти символы будем называть переменными высказываниями;

2) символы второй категории: $\wedge, \vee, \rightarrow, \neg$. Они носят общее название логических связок;

3) третью категорию составляет пара символов $(\ )$, называемая скобками.

Других символов исчисление высказывания не имеет. *Формулы исчисления высказываний* представляют собой последовательности символов алфавита исчисления высказываний. Для обозначения формул будем пользоваться большими буквами латинского алфавита. Эти буквы не являются символами исчисления. Они представляют собой только условные обозначения формул.

Определение формулы исчисления высказываний.

1. Всякая переменная $x, y, z, \ldots$ является формулой.

2. Если A и B – формулы, то слова $(A \vee B)$, $(A \wedge B)$, $(A \rightarrow B)$, $\neg A$ – также формулы.

3. Никакая другая строчка символов не является формулой.

Переменные высказывания будем называть элементарными формулами.

Одновременно с понятием формулы вводится понятие *подформулы* или части формулы.

1.Подформулой элементарной формулы является только она сама.

2.Если формула имеет вид A, то ее подформулами являются: она сама, формула A и все подформулы формулы A.

3.Если формула имеет вид $(A*B)$ (здесь и в дальнейшем под символом * будем понимать любой из трех символов $\wedge, \vee, \rightarrow$), то ее подформулами являются: она сама, формулы A и B и все подформулы формул A и B.

Введем в запись формул некоторые упрощения. Будем опускать в записи формул скобки по тем же правилам, что и в алгебре высказываний.

Следующим этапом в построении исчисления высказываний является выделение класса доказуемых формул.

Определение доказуемых формул имеет тот же характер, что и определение формулы.

Сначала определяются исходные доказуемые формулы (аксиомы), а затем определяются правила вывода, которые позволяют из имеющихся доказуемых формул получить новые доказуемые формулы.

Образование доказуемой формулы из исходных доказуемых формул путем применения правил вывода, называется выводом данной формулы из аксиом.

Система аксиом исчисления высказываний.

Система аксиом исчисления высказываний состоит из 11 аксиом, которые делятся на четыре группы.

I_1 $x \to (y \to x)$;

I_2 $(x \to (y \to z)) \to ((x \to y) \to (x \to z))$;

II_1 $x \wedge y \to x$;

II_2 $x \wedge y \to y$;

II_3 $(z \to x) \to ((z \to y) \to (z \to x \wedge y))$;

III_1 $x \to x \vee y$;

III_2 $y \to x \vee y$;

III_3 $(x \to z) \to ((y \to z) \to (x \vee y \to z))$;

VI_1 $(x \to y) \to (\neg y \to \neg x)$;

VI_2 $x \to \neg(\neg x)$;

VI_3 $\neg(\neg x) \to x$.

Правила вывода

1. **Правило подстановки**. Если формула A доказуема в исчислении высказываний, x переменная, B – произвольная формула исчисления высказываний, то формула, полученная в результате замены в формуле A переменной x всюду, где она входит, формулой B, является также доказуемой формулой.

Операция замены в формуле A переменной x формулой B носит название подстановки и символически записывается так:

$$\int_x^B (A).$$

Если A – доказуемая формула, то будем писать $\vdash A$. Тогда правило подстановки можно записать схематически следующим образом:

$$\frac{\vdash A}{\vdash \int_x^B (A)}.$$

И читается эта запись так: «Если формула A доказуема, то доказуема формула $\int_x^B (A)$.

2. **Правило заключения**. Второе правило исчисления высказываний: ***Modus Ponens.*** «Если A, то B; но A истинно, следовательно, истинно B».

Если формулы A и $A \to B$ доказуемы в исчислении высказываний, то формула B также доказуема. Схематическая запись этого правила имеет вид:

$$\frac{\vdash A; \vdash A \to B}{\vdash B}$$

а) Всякая аксиома является доказуемой формулой.

б) Формула, полученная из доказуемой формулы путем применения подстановки вместо переменной x произвольной формулы B есть доказуемая формула.

в) Формула B, полученная из доказуемых формул A и $A \to B$ путем применения правила заключения, есть доказуемая формула.

г) Никакая другая формула исчисления высказываний не считается доказуемой.

4.8 Булева алгебра и переключательные схемы

Первый опыт применения булевой алгебры был связан с релейными цепями. Хотя эти цепи уже не используются в ЭВМ, они находят применение в ряде систем управления движением.

Контактная цепь – устройство из проводов и контактов, связывающих два полюса. Любой контакт может быть либо замкнут, либо разомкнут. Контакты будем обозначать x_1, x_2, x_3, …

$$x_i = \begin{cases} 0, & \text{если контакт разомкнут} \\ 1, & \text{если контакт замкнут.} \end{cases}$$

Функция, реализуемая контактной цепью, принимает значение 1, если контур между двумя полюсами замкнут, и 0 – в противном случае. Основные операции булевой алгебры: конъюнкцию, дизъюнкцию и отрицание можно реализовать следующими контактными цепями.

Релейно-контактные схемы (или переключательные схемы) широко используются в технике автоматического управления.

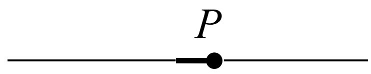
Под переключательной схемой понимают схематическое изображение некоторого устройства, состоящее из следующих элементов:

1) ***переключателей*** (ключей);

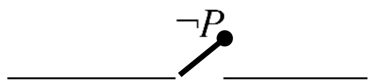
2) соединяющих их ***проводников***;

3) ***входов*** в схему и ***выходов*** из нее (полюсов).

Простейшая схема содержит один переключатель P и имеет один вход и один выход. Переключателю P ставится в соответствие истинное высказывание P, гласящее «переключатель P замкнут».



Переключателю $\neg P$ ставится в соответствие истинное высказывание: «переключатель P разомкнут» или «переключатель $\neg P$ замкнут».

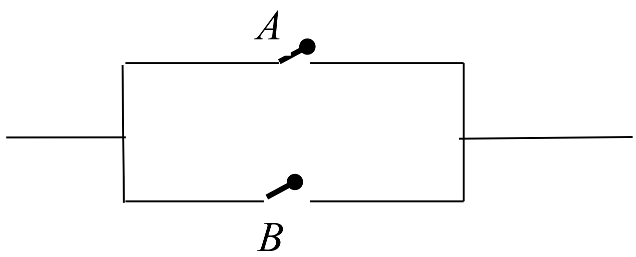


Таким образом, когда P замкнут, $\neg P$ – разомкнут и наоборот.

Если высказывание P истинно, то переключатель P замкнут – схема пропускает ток, если ложно – не пропускает. Следовательно, любому высказыванию может быть поставлена в соответствие переключательная схема с двумя полюсами (двухполюсная схема).

Конъюнкции высказываний A и B соответствует ***последовательное*** соединение переключателей A и B:

Дизъюнкции высказываний A и B соответствует ***параллельное*** соединение переключателей A и B:



Так как любая формула логики высказываний может быть записана в виде ДНФ или КНФ, то ясно, что любой формуле можно сопоставить схему переключателей. Причем, упрощение формулы ведет к упрощению схемы.

П р и м е р.

Имеется длинный коридор (или высокая башня) во всей длине которого установлены светильники. На обоих концах коридора установлены выключатели. Как они должны быть устроены, чтобы можно было включить свет на одном конце коридора, и, пройдя коридор, выключить свет на другом его конце. Построить схему переключателей.

Итак, переключателей должно быть два. Пусть, например, свет не горит, и оба переключателя разомкнуты. Тогда, переключатель A замыкается, свет горит. Коридор пройден, переключатель B замыкается, свет не горит. Оба переключателя замкнуты – свет не горит. И наоборот.

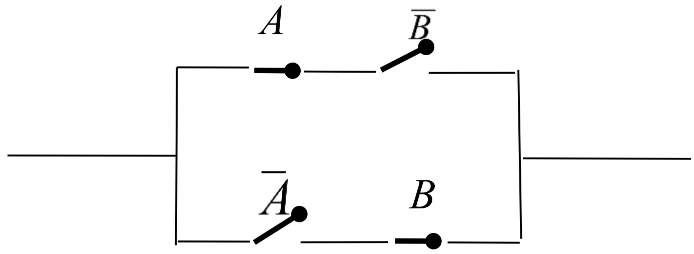
Составим таблицу истинности.

A	B	$F(A, B)$
0	0	0
1	0	1
0	1	1
1	1	0

Построим СДНФ данной логической функции.

$$F(A, B) = A\bar{B} \vee \bar{A}B.$$

Соответствующая схема переключателей имеет вид:



Комбинационные элементы – электронные компоненты, техническая реализация которых может быть основана на использовании различных физических явлений: магнитных, явлений в полупроводниках и т. д. Они являются основными компонентами компьютеров.

Все комбинационные элементы имеют один или более входов и ***один выход***. Каждый вход может принимать одно из двух значений (обычно низкое или высокое напряжение).

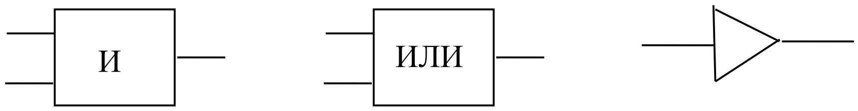
Наиболее важные типы комбинационных элементов приведены в таблице 1.

Различные комбинационные элементы могут быть связаны друг с другом в цепи так, что выход одних является входом других.

Таблица 1

Элементы	Конъюнкция $x_1 \wedge x_2$	Дизъюнкция $x_1 \vee x_2$	Отрицание $\bar{x}$

Обозначения



Такие цепи называются ***комбинационными схемами*** (логическими сетями).

Входы одних логических элементов можно подсоединить к выходам других элементов и получить логическую сеть. Логическая сеть реализует некоторую булеву функцию. В случае контактных цепей выражение для функции с наименьшим числом вхождений переменных дает реализацию цепи с наименьшим количеством контактов. Простота логической сети может определяться как числом логических элементов, так и числом входов логических элементов. Кроме того, логические элементы характеризуются запаздыванием, поэтому быстродействие сети пропорционально числу ступеней сети (т.е. максимальному числу логических элементов, через которые проходит входной сигнал без учета инверторов).

Представление функции в ДНФ дает двухступенчатую быстродействующую реализацию. Поэтому, для упрощения логической сети можно использовать минимизацию булевой функции методом Квайна, но при этом ввести понятие «стоимости» простой импликанты.

В задаче минимизации числа логических элементов стоимость простой импликанты, содержащей две и более переменные, принимается равной 1, а стоимость простой импликанты из одной переменной – 0 (т.к. для ее реализации не требуется логический элемент «и»).

В задаче минимизации общего числа входов логических элементов стоимость простой импликанты складывает из входов логического элемента «и», реализующего импликанту, и входов логического элемента «или». При построении минимальной ДНФ по множеству простых импликант, среди простых импликант выбирается множество с суммарной минимальной стоимостью простых импликант.

Так как штрих Шеффера и стрелка Пирса являются функционально полными системами, возможно описание выходов комбинационных схем с помощью каждого из этих элементов.

<u>Пример.</u>

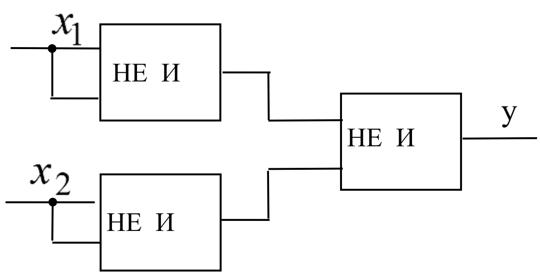
Построить комбинационную схему в базисе «штрих Шеффера», реализующую дизъюнкцию $x_1 \vee x_2$.

Так как $x_1 \mid x_2 = \overline{x_1 x_2}$. А отрицание $\overline{x} = x \mid x$, то дизъюнкция

$$x_1 \oplus x_2 = \overline{\overline{x}_1 \overline{x}_2} = \overline{x}_1 \mid \overline{x}_2 = (x_1 \mid x_1) \big| (x_2 \mid x_2).$$

Обозначим комбинационный элемент, соответствующий функции «штрих Шеффера» обозначим в виде:

Тогда соответствующая схема приобретает вид:



Очевидно, данная схем более сложная, чем та, что могла быть построена в базисе $\{\wedge, \vee, \neg\}$.