Логические основы информатики

Логические операции
Логические функции и выражения
Основные законы алгебры логики
Способы описания логических функций

Логика

От греческого слова λόγος — «слово», «мысль», «понятие», «рассуждение».

Логика — наука о законах и формах мышления, познания мира, способах построения суждений, доказательств и опровержений.

Основная цель логики – исследование того, как из одних утверждений можно выводить другие.

Логика – наука, изучающая методы установления истинности или ложности одних высказываний (утверждений) на основе истинности или ложности других высказываний.

Исторически логика изучалась как часть философии.

Сейчас логика также изучается как часть математики, информатики.

Логика появилась примерно в IV в. до н.э. в Древней Греции, ее создателем считается Аристотель.

Аристотелевская (или традиционная) логика для анализа правильного мышления использует естественный язык,

а **символическая** логика, появившаяся в XIX в., использует искусственный язык символов, подобный языку математики.

Алгебра логики

-это раздел математической логики, изучающий логические высказывания и методы установления их истинности или ложности с помощью алгебраических методов.

Основоположник - английский математик **Джордж Буль** (1815-1864), изучал логику мышления математическими методами и разработал алгебраические методы решения традиционных логических задач.

В работе Буля «Исследование законов мышления» была изложена алгебра логики высказываний, основанная на трех операциях

And (И), **Or** (ИЛИ) и **Not** (НЕ).

Эту алгебру называют **булевой алгеброй**.

<u>Позволяет описывать принципы построения и работы логических схем компьютеров, использующих двоичную систему счисления.</u>

Под **высказыванием** понимают повествовательное утвердительное предложение, о котором имеет смысл говорить истинно оно или ложно в данных условиях места и времени.

Предложения

“Какое сегодня число?”, “x=20” не являются высказываниями.

Предложения

“Параллелограмм имеет четыре вершины”,

“Число 6 делится на 2 и на 3”

являются истинными высказываниями.

Предложения

“Париж – столица Англии”, “карась не рыба”, “47< 16”

являются ложными.

Высказывание, представляющее собой одно утверждение, принято называть **простым** или **элементарным**.

Обозначают элементарные высказывания малыми буквами латинского алфавита: *a*, *b*, *c*, *d*, *p*, *q*, *r*, …., а логическое значение высказывания обозначается цифрами 1 (истина) и 0 (ложь).

Высказывания, которые получаются из элементарных с помощью грамматических связок *не*, *и*, *или*, *если*…, *то*…., *тогда и только тогда*, принято называть **сложными** или **составными**.

истина	ложь
true	false
да	нет
1	0

Логические переменные используются в алгебре логики для описания и формализации различных утверждений и высказываний.

Истинность или ложность составных высказываний зависит от истинности или ложности простых высказываний.

В информатике используются логические операции, которые позволяют составлять новые высказывания путём соединения более простых.

№	Операция	Обозначение
1	Отрицание (инверсия, логическое НЕ)	$\neg A$ $\overline{A}$ не A not A
2	Конъюнкция (логическое умножение, логическое И)	$A \wedge B$ A & B $A \cdot B$ A и B A and B
3	Дизъюнкция (логическое сложение, логическое ИЛИ, включающее ИЛИ)	$A \vee B$ $A + B$ A или B A or B
4	Исключающая дизъюнкция (исключающее ИЛИ, сложение по модулю 2, строгая дизъюнкция)	$A \oplus B$ A xor B
5	Импликация (следование)	$A \rightarrow B$ $A \Rightarrow B$ A – посылка B – следствие
6	Эквиваленция (равнозначность)	$A \leftrightarrow B$, $A \equiv B$, $A \sim B$

Логические переменные		Логические операции					
		отрицание	конъюнкция	дизъюнкция	исключающая дизъюнкция	импликация	эквиваленция
A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \oplus B$	$A \rightarrow B$	$A \leftrightarrow B$
0	0	1	0	0	0	1	1
0	1	1	0	1	1	1	0
1	0	0	0	1	1	0	0
1	1	0	1	1	0	1	1

Порядок выполнения операций указывается скобками.

Скобки можно опускать, придерживаясь следующего порядка действий:

- конъюнкция выполняется раньше, чем все остальные операции,
- дизъюнкция выполняется раньше, чем импликация и эквивалентность.

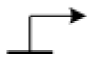
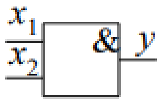
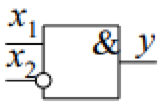
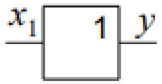
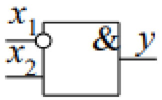
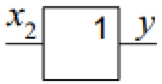
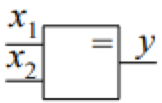
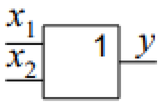
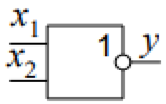
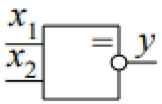
По убыванию приоритета:

$\neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow$

Функции одного аргумента

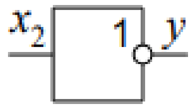
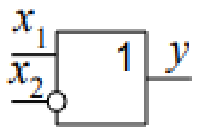
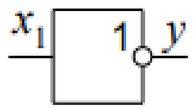
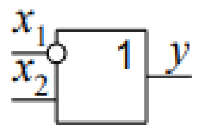
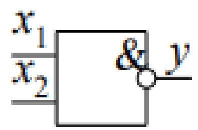
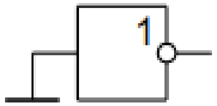
x	0	1	Функция	Название функции
Y_0	0	0	$y = 0$	Константа 0
Y_1	0	1	$y = x$	Переменная x
Y_2	1	0	$y = \overline{x}$	Инверсия x
Y_3	1	1	$y = 1$	Константа 1

Функции двух аргументов

x_1 / x_2	0 0 1 1 / 0 1 0 1	Функция	Название функции	Название элемента	Обозначение элемента
y_0	0 0 0 0	$y = 0$	Константа 0	Генератор нуля	
y_1	0 0 0 1	$y = x_1 \wedge x_2$ $y = x_1 \cdot x_2$	Конъюнкция	Логическое «И»	
y_2	0 0 1 0	$y = x_1 \Delta x_2$	Запрет по x_1	Запрет	
y_3	0 0 1 1	$y = x_1$	Переменная x_1	Повторитель	
y_4	0 1 0 0	$y = x_2 \Delta x_1$	Запрет по x_2	Запрет	
y_5	0 1 0 1	$y = x_2$	Переменная x_2	Повторитель	
y_6	0 1 1 0	$y = x_1 \oplus x_2$	Сумма по модулю 2	Исключающее «ИЛИ»	
y_7	0 1 1 1	$y = x_1 \vee x_2$ $y = x_1 + x_2$	Дизъюнкция	Логическое «ИЛИ»	
y_8	1 0 0 0	$y = x_1 \downarrow x_2$	Стрелка Пирса	Логическое «ИЛИ-НЕ»	
y_9	1 0 0 1	$y = x_1 \equiv x_2$	Эквивалент-ность	Исключающее «ИЛИ-НЕ»	

Продолжение:

Функции двух аргументов

x_1	0 0 1 1	Функция	Название функции	Название элемента	Обозначение элемента
x_2	0 1 0 1				
y_{10}	1 0 1 0	$y = \overline{x_2}$	Инверсия x_2	Логическое «НЕ»	
y_{11}	1 0 1 1	$y = x_2 \to x_1$	Импликация от x_2 к x_1	Импликатор	
y_{12}	1 1 0 0	$y = \overline{x_1}$	Инверсия x_1	Логическое «НЕ»	
y_{13}	1 1 0 1	$y = x_1 \to x_2$	Импликация от x_1 к x_2	Импликатор	
y_{14}	1 1 1 0	$y = x_1 \mid x_2$	Штрих Шеффера	Логическое «И-НЕ»	
y_{15}	1 1 1 1	$y = 1$	Константа 1	Генератор единицы	

Основные законы алгебры логики

Для логического выражения установлен порядок выполнения операций:

- вначале выполняются операции инверсии,
- затем операции конъюнкции
- в последнюю очередь операции дизъюнкции.

Если требуется изменить порядок выполнения операций, то используются скобки.

$$(x1 \vee x2) \wedge (x1 \vee x3)$$

При записи логических выражений допускается вид

$$(x1 + x2) \cdot (x1 + x3)\ .$$

Для операций конъюнкции и дизъюнкции справедливы следующие **законы алгебры логики**.

Символ + (сложение) - операция дизъюнкции (логическое «ИЛИ»),
символ умножения - операция конъюнкции (логическое «И»).

1. Переместительный закон

$$x + y = y + x; \qquad x \vee y = y \vee x;$$
$$x \cdot y = y \cdot x; \qquad x \wedge y = y \wedge x.$$

2. Сочетательный закон

$$x + (y + z) = (x + y) + z;$$
$$(x \cdot y) \cdot z = x \cdot (y \cdot z).$$

3. Распределительный закон

$$(x + y) \cdot z = x \cdot z + y \cdot z;$$
$$x \cdot y + z = (x + z)(y + z).$$

4. Закон инверсии (формулы де Моргана)

$$\overline{x + y} = \bar{x} \cdot \bar{y}; \qquad \overline{x \cdot y} = \bar{x} + \bar{y}.$$

5. Закон двойного отрицания

$$\bar{\bar{x}} = x.$$

6. Закон исключенного третьего

$$x + 1 = 1; \qquad x + \bar{x} \cdot y = x + y;$$
$$x + \bar{x} = 1; \qquad x(\bar{x} + y) = x \cdot y.$$

7. Закон противоречия

$$x \cdot 0 = 0;$$
$$x \cdot \bar{x} = 0.$$

8. Закон идентичности

$$x \cdot x = x;$$
$$x + x = x.$$

9. Закон поглощения

$$x \cdot 1 = x;$$
$$x \cdot x \cdot x \cdot \ldots \cdot x = x;$$
$$x(x + y) = x;$$
$$x + x \cdot y = x;$$
$$x + x + \ldots + x = x;$$
$$x + x \cdot y + x \cdot z = x.$$

Следствия законов:

1. Если логическая сумма двоичных аргументов или функций содержит хотя бы одну пару взаимно инверсных слагаемых, то эта сумма всегда истинна

$$x + y + z \cdot y + \bar{x} = 1;$$

$$P + Q + L + \overline{Q} = 1.$$

2. Если логическое произведение двоичных аргументов или функций содержит хотя бы одну пару взаимно инверсных сомножителей, то это произведение всегда ложно

$$x \cdot y \cdot z \cdot \bar{x} = 0;$$

$$P \cdot Q \cdot L \cdot \overline{Q} = 0.$$

Формы логических функций

Элементарная конъюнкция – *минтерм* – образуется логическим умножением переменных и их отрицаний

$$P = x \cdot y \cdot \bar{z};$$

$$Q = \bar{x} \cdot y \cdot \bar{z}.$$

Элементарная дизъюнкция – *макстерм* – образуется логическим сложением переменных и их отрицаний

$$P = x + y + \bar{z};$$

$$Q = \bar{x} + \bar{y} + z.$$

Число переменных, составляющих минтерм или макстерм, называется *рангом*.

$P = x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4$ – элементарная конъюнкция четвертого ранга.

$M = x_1 + \bar{x}_2 + x_3$ – элементарная дизъюнкция третьего ранга.

Функция в дизъюнктивной нормальной форме (ДНФ) является логической суммой элементарных конъюнкций

$$P = \bar{x} \cdot y + x \cdot \bar{y} + x \cdot y \cdot z + \bar{x} \cdot y \cdot \bar{z}.$$

Функция в конъюнктивной нормальной форме (КНФ) является логическим произведением элементарных дизъюнкций

$$P = (x + y) \cdot (x + \bar{y}) \cdot (\bar{x} + y + z).$$

Если в состав логической формулы входят наборы аргументов одинакового ранга, то такая форма называется *совершенной* (СКНФ и СДНФ).

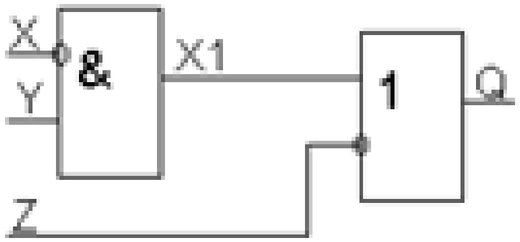
Требования к совершенной форме:

1. В функции не должно быть двух одинаковых конъюнкций (дизъюнкций).
2. Ни одна конъюнкция (дизъюнкция) не содержит двух одинаковых двоичных переменных.
3. Ни одна конъюнкция (дизъюнкция) не содержит двоичную переменную вместе с ее отрицанием.
4. Все конъюнкции (дизъюнкции) одного ранга.

Рассмотрим задачи, которые решаются с помощью алгебры логики.

Первая задача:
Существует логическая схема (или алгоритм) и необходимо описать её с помощью логического выражения, т.е. с помощью алгебры логики.

Пример:



$q = f(x, y, z) = ?$

Обозначим вспомогательную переменную через x_1

$x_1 = \bar{x} \cdot y$, $q = x_1 + \bar{z}$, в итоге $q = \bar{x} \cdot y + \bar{z}$ или

$q = \bar{x} \& y \vee \bar{z}$

Вторая задача:

По таблице истинности необходимо составить логическое высказывание.

Дано: таблица;

нужно получить формулу.

Таблица истинности:

№	x1x2x3	y
0	0 0 0	0
1	0 0 1	0
2	0 1 0	1
3	0 1 1	1
4	1 0 0	0
5	1 0 1	1
6	1 1 0	1
7	1 1 1	0

СДНФ: $y = \overline{x1}x2\overline{x3} + \overline{x1}x2x3 + x1\overline{x2}x3 + x1x2\overline{x3}$

СКНФ: $y = (\overline{x1} + \overline{x2} + \overline{x3})(\overline{x1} + \overline{x2} + x3)(x1 + \overline{x2} + \overline{x3})(x1 + x2 + x3)$

Совершенная дизъюнктивная нормальная форма (СДНФ)

Таблица истинности:

№	x1x2x3	y
0	0 0 0	0
1	0 0 1	0
2	0 1 0	1
3	0 1 1	1
4	1 0 0	0
5	1 0 1	1
6	1 1 0	1
7	1 1 1	0

Алгоритм построения СДНФ по таблице истинности:

1. Выбрать все строки таблицы, в которых значение функции **равно единице**.
2. Для каждой такой строки записать конъюнкцию всех переменных следующим образом: если значение некоторой переменной в этом наборе равно 1, то в конъюнкцию включить саму переменную, в противном случае — её отрицание.
3. Все полученные конъюнкции связать операциями дизъюнкции.

Важно: если на большей части строк таблицы истинности значение функции равно 0,
то для получения её логической формулы лучше построить СДНФ,
в противном случае — СКНФ.

СДНФ: $y = \overline{x1}x2\overline{x3} + \overline{x1}x2x3 + x1\overline{x2}x3 + x1x2\overline{x3}$

Совершенная конъюнктивная нормальная форма (СКНФ)

Таблица истинности:

№	x1x2x3	y
0	0 0 0	0
1	0 0 1	0
2	0 1 0	1
3	0 1 1	1
4	1 0 0	0
5	1 0 1	1
6	1 1 0	1
7	1 1 1	0

Алгоритм построения СКНФ по таблице истинности:

1. Выбрать все строки таблицы, в которых значение функции **равно нулю**.
2. Для каждой такой строки записать дизъюнкцию всех переменных следующим образом: если значение некоторой переменной в этом наборе равно 0, то в дизъюнкцию включить саму переменную, в противном случае — её отрицание.
3. Все полученные дизъюнкции связать операциями конъюнкции.

Важно: если на большей части строк таблицы истинности значение функции равно 1, то для получения её логической формулы лучше построить СКНФ.

СКНФ: $y = (\overline{x1} + \overline{x2} + \overline{x3})(\overline{x1} + \overline{x2} + x3)(x1 + \overline{x2} + \overline{x3})(x1 + x2 + x3)$

СДНФ - Выбрать все строки таблицы, в которых значение функции равно единице.

СКНФ - Выбрать все строки таблицы, в которых значение функции равно нулю.

Пример:

x	y	q		
				По единицам:
0	0	0		$q=\overline{x}\cdot y+x\cdot\overline{y}$
0	1	1		По нулям:
1	0	1		$q=(x+y)\cdot(\overline{x}+\overline{y})=x\cdot\overline{x}+x\cdot\overline{y}+y\cdot\overline{x}+y\cdot\overline{y}=\overline{x}\cdot y+x\cdot\overline{y}$
1	1	0		т.е. результат одинаков.

СДНФ - Выбрать все строки таблицы, в которых значение функции равно единице.

СКНФ - Выбрать все строки таблицы, в которых значение функции равно нулю.

Пример:

x	y	z	q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

по единицам:

$$q = \overline{x} \cdot y \cdot \overline{z} + \overline{x} \cdot y \cdot z + x \cdot \overline{y} \cdot z + x \cdot y \cdot \overline{z} + x \cdot y \cdot z$$

или по нулям:

$$q = (x + y + z) \cdot (x + y + \overline{z}) \cdot (x + \overline{y} + \overline{z})$$

По таблицам истинности составить логические выражения.
СДНФ - Выбрать все строки таблицы, в которых значение функции равно 1.

1.

x	y	z	q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

2.

x	y	z	q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

По таблицам истинности составить логические выражения.
СДНФ - Выбрать все строки таблицы, в которых значение функции равно 1.

3.

x	y	z	q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

4.

x	y	z	q
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

По таблицам истинности составить логические выражения.
СДНФ - Выбрать все строки таблицы, в которых значение функции равно 1.

5.

x	y	z	q
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

6.

x	y	z	q
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

По таблицам истинности составить логические выражения.

СДНФ - Выбрать все строки таблицы, в которых значение функции равно 1.

7.

x	y	z	q
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	1
1	1	1	1

8.

x	y	z	q
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

По таблицам истинности составить логические выражения.
СДНФ - Выбрать все строки таблицы, в которых значение функции равно 1.

9.

x	y	z	q
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	0
1	1	1	1

10.

x	y	z	q
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0