

## Глава 2

# МАТЕМАТИЧЕСКАЯ ЛОГИКА

### § 8

## Логика и компьютеры

*Ключевые слова:*

- логика
- формальная логика
- логическое высказывание
- алгебра логики
- логические переменные
- логическая операция
- операция «НЕ»
- операция «И»
- операция «ИЛИ»
- логическая функция

### Что такое высказывание?

В быту мы часто используем слова «логика», «логично». **Логика** (от древнегреческого *λογικός* — наука о рассуждении) — это наука о том, как правильно рассуждать, делать выводы, доказывать утверждения.

В естественном языке рассуждения связаны с самыми разными предметами и понятиями, и поэтому изучать их достаточно сложно. Древнегреческий философ **Аристотель** начал изучать общие правила построения правильных выводов из известной информации, которая считается истинной. Такая логика называется **формальной**, она изучает истинность и ложность *логических высказываний*, отвлекаясь от их содержания.



**Аристотель**  
(384–322 до н.э.)



**Логическое высказывание** — это повествовательное предложение, про которое можно однозначно сказать, истинно оно или ложно.

Используя это определение, проверим, можно ли считать логическими высказываниями следующие предложения:

- 1) Сейчас идёт дождь.
- 2) Вчера жирафы улетели на север.

- 3) Беги сюда!
- 4) Который час?
- 5) В городе  $N$  живёт более 2 миллионов человек.
- 6) У квадрата 10 сторон, и все разные.
- 7) История — интересный предмет.

Здесь высказываниями являются только предложения 1, 2 и 6, остальные не подходят под определение. Предложения 3 и 4 не повествовательные (призыв к действию и вопрос). Предложение 5 станет высказыванием только в том случае, если  $N$  заменить на название конкретного города. Утверждение 7 кто-то считает истинным, а кто-то — ложным (нет однозначности), его можно более строго сформулировать в виде «По мнению  $N$ , история — интересный предмет». Для того чтобы оно стало высказыванием, нужно заменить  $N$  на имя человека.

Используя определение, проверьте, являются ли логическими высказываниями предложения:

- 1) Лошади едят сено.
- 2) Карету мне, карету!
- 3) Где расположен Канин Нос?
- 4) Восемью три – двенадцать.
- 5) Программировать очень просто.



**Дж. Буль**  
(1815–1864)

Логика и компьютеры тесно связаны. В классической формальной логике высказывание может быть истинно или ложно, третий вариант исключается<sup>1)</sup>. Если обозначить истинное значение единицей, а ложное — нулём, то получится, что формальная логика изучает правила выполнения операций с нулями и единицами, т. е. с двоичными кодами. Оказалось, что всю обработку двоичных данных можно свести к выполнению логических операций.

Важный шаг в этом направлении сделал английский математик **Джордж Буль**. Буль впервые ввёл в науку двоичные переменные, принимающие только два значения — «истина» и «ложь», — и три основные логические операции: НЕ, И, ИЛИ. Кроме того, он предложил применить для исследования логичес-

<sup>1)</sup> Существуют неклассические логические системы, например трёхзначная логика, где кроме «истинно» и «ложно» есть ещё состояние «не определено» (или «возможно»).

ких высказываний методы алгебры. Позже этот раздел математики получил название *алгебры логики*, или *алгебры высказываний*. Ещё его называют *булевой алгеброй* (по имени Дж. Буля).



**Алгебра логики** — это математический аппарат, с помощью которого записывают, упрощают и преобразуют логические высказывания, вычисляют их значения.

Алгебра логики определяет правила выполнения операций с логическими значениями «ложь» и «истина». Если обозначить эти значения как 0 и 1, то получается, что с помощью алгебры логики можно описать алгоритмы работы с двоичными данными. Например, так можно построить запоминающие элементы и выполнять арифметические действия.



Используя дополнительные источники, выясните, как называлась основная научная работа Дж. Буля и в каком году она была написана. Сколько лет было тогда учёному?

## Простые и сложные высказывания

Высказывания бывают простые и сложные (составные). Простые высказывания нельзя разделить на более мелкие высказывания, например: «Сейчас идёт дождь» или «Форточка открыта». Сложные (составные) высказывания строятся из простых с помощью логических связок — **логических операций НЕ, И, ИЛИ**.

В алгебре логики высказывания обычно обозначаются латинскими буквами. Таким образом, мы уходим от конкретного содержания высказываний, нас интересует только их истинность или ложность. Например, можно обозначить буквой  $A$  высказывание «Сейчас идёт дождь», а буквой  $B$  — высказывание «Форточка открыта».

Так как высказывания могут быть истинными или ложными, введённые символы  $A$  и  $B$  можно рассматривать как **логические переменные**, которые могут принимать два возможных значения: «ложь» (0) и «истина» (1). Из них строятся сложные высказывания:

$\text{не } A$  = Неверно, что сейчас идёт дождь.

$A$  и  $B$  = Сейчас идёт дождь и открыта форточка.

$A$  или  $B$  = Сейчас идёт дождь или открыта форточка.

Если и другие логические операции, но НЕ, И и ИЛИ используются чаще всего. Оказывается, с их помощью можно выразить любую логическую операцию, поэтому эти три операции можно считать основными, **базовыми**, и говорят, что они составляют базис.

Различные устройства компьютера строятся на основе элементов, выполняющих логические операции НЕ, И, ИЛИ.

При введённых выше обозначениях  $A$  и  $B$  запишите на русском языке высказывания:

- а) **не**  $B$ ;                      б) (**не**  $A$ ) **и**  $B$ ;                      в)  $A$  **или** (**не**  $B$ ).

При тех же обозначениях запишите в символьном виде высказывания:

- а) «Неверно, что сейчас идёт дождь и открыта форточка».  
 б) «Неверно, что сейчас идёт дождь или закрыта форточка».

### Операция НЕ

**Операция НЕ** часто называется **отрицанием** или **инверсией**. В алгебре логики всего два возможных значения (0 и 1), поэтому логическое отрицание — это переход от одного значения к другому: от 1 к 0 или наоборот. Если высказывание  $A$  истинно, то НЕ  $A$  ложно, и наоборот.

Используя дополнительные источники, переведите на русский язык слово *inverse*, от которого произошло слово «инверсия».

Операцию НЕ обозначают чертой сверху, например:  $\bar{A}$ . В школьном алгоритмическом языке<sup>1)</sup> эта операция обозначается словом **не**, а в языке программирования Паскаль — английским словом **not**.

Используя дополнительные источники, найдите другие обозначения операции НЕ.

Операцию НЕ можно задать в виде таблицы (рис. 2.1).

Эта таблица состоит из двух частей: слева перечисляются все возможные значения исходной величины (их всего два — 0 и 1), а в последнем столбце записывают результат выполнения логической операции для каждого из этих вариантов. Такая таблица называется **таблицей истинности** логической операции. Таблица истинности задаёт логическую функцию.

$A$	не $A$
0	1
1	0

Рис. 2.1

<sup>1)</sup> Вспомним, что мы договорились называть его просто «алгоритмический язык».



**Логическая функция** — это правило преобразования входных логических значений в логическое значение-результат.



Используя таблицу истинности на рис. 2.1, определите, как можно упростить выражение **не(не A)**. Рассмотрите оба варианта: когда  $A = 0$  и когда  $A = 1$ . Сделайте вывод.

## Операция И

Из двух простых высказываний  $A$  и  $B$  (например,  $A =$  Сейчас идёт дождь,  $B =$  Форточка открыта) можно составить сложное высказывание  $A$  и  $B$ . Высказывание  $A$  и  $B$  истинно в том и только в том случае, когда оба высказывания,  $A$  и  $B$ , истинны одновременно.

Для понимания операции И можно представить себе простую схему, в которой для включения лампочки используются два выключателя, соединённых последовательно (рис. 2.2). Чтобы лампочка загорелась, нужно обязательно включить оба выключателя. Вместе с тем, чтобы выключить лампочку, достаточно выключить любой из них.

Операция И (в отличие от НЕ) выполняется с двумя логическими значениями. В алгоритмическом языке системы КуМир операция И обозначается словом **и**, а в Паскале — словом **and**.

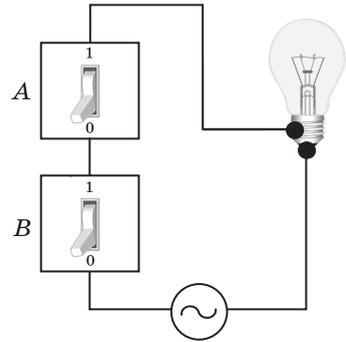


Рис. 2.2



Используя дополнительные источники, найдите другие обозначения операции И.

В таблице истинности операции И будет уже не один столбец с исходными данными, а два, мы обозначим исходные данные как  $A$  и  $B$ . Число строк также выросло, с 2 до 4, поскольку с помощью 2 бит можно записать 4 разных комбинации значений двух переменных: 00, 01, 10 и 11. Как следует из определения операции И, в последнем столбце будет всего одна единица, для варианта  $A = B = 1$  (оба высказывания,  $A$  и  $B$ , истинны одновременно) — рис. 2.3.

$A$	$B$	$A$ и $B$
0	0	0
0	1	0
1	0	0
1	1	1

Рис. 2.3

Из значений  $A$  и  $B$  в каждой строке этой таблицы составьте двоичное число и запишите его в десятичной системе счисления. Почему строки в таблице расположены именно так?



Легко проверить, что этот результат можно получить «обычным» умножением  $A$  на  $B$ , поэтому операцию  $\text{И}$  называют **логическим умножением**. Она часто обозначается знаком умножения (точкой):  $A \cdot B$ .

С точки зрения обычной математики, эта операция выбирает *наименьшее* из исходных значений. Математики используют ещё одно название операции  $\text{И}$  — **конъюнкция**.

Используя дополнительные источники, выясните, от какого слова произошло слово «конъюнкция» и что оно обозначает.



С помощью таблицы истинности можно упрощать логические выражения. Например, рассмотрим выражение  $A \text{ и } 1$ . По таблице истинности на рис. 2.3 получаем:

при  $A = 0$ :  $A \text{ и } 1 = 0 \text{ и } 1 = 0$

при  $A = 1$ :  $A \text{ и } 1 = 1 \text{ и } 1 = 1$ .

Можно заметить, что в любом случае результат совпадает с  $A$ , поэтому  $A \text{ и } 1 = A$ .

Используя таблицу истинности операции  $\text{И}$ , упростите выражения:

а)  $A \text{ и } 0$ ;      б)  $A \text{ и } A$ ;      в)  $A \text{ и } (\text{не } A)$ .



## Операция ИЛИ

Высказывание  $A$  **или**  $B$  (например, «Сейчас идет дождь или форточка открыта») истинно тогда, когда истинно хотя бы одно из входящих в него высказываний или оба одновременно.



В алгоритмическом языке операция **ИЛИ** обозначается словом **или**, а в языке Паскаль — английским словом **or**.

Используя дополнительные источники, найдите другие обозначения операции **ИЛИ**.



Для понимания операции **ИЛИ** можно представить себе схему с двумя выключателями, соединёнными параллельно (рис. 2.4).

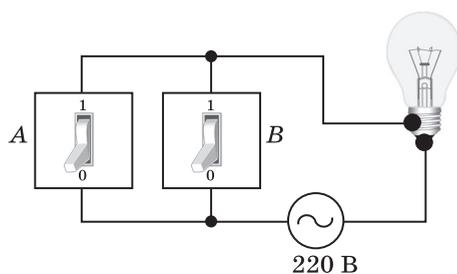


Рис. 2.4

Чтобы лампочка загорелась, достаточно включить хотя бы один из выключателей. Чтобы выключить лампочку, необходимо обязательно выключить оба. В таблице истинности будет только один ноль — для варианта  $A = B = 0$  (рис. 2.5).

$A$	$B$	$A$ или $B$
0	0	0
0	1	1
1	0	1
1	1	1

Рис. 2.5

Операцию ИЛИ называют **логическим сложением**, потому что она похожа на обычное математическое сложение. Поэтому она часто обозначается знаком сложения:  $A+B$ . Единственное отличие — в последней строке таблицы истинности: в математике  $1+1$  равно 2, а в алгебре логики — единице.

Можно считать, что в результате применения операции ИЛИ из исходных значений выбирается *наибольшее*. Другое название этой операции — **дизъюнкция**.

Используя дополнительные источники, выясните, от какого слова произошло слово «дизъюнкция» и что оно обозначает.

Запишите в тетради ответы на следующие вопросы.

- Сколько строк в таблице истинности функции с двумя переменными?
- Сколько существует возможных вариантов распределения нулей и единиц в последнем столбце?
- Сколько можно придумать различных логических функций с двумя переменными?

Используя таблицу истинности операции И, упростите выражения:  
а)  $A$  или 0;    б)  $A$  или 1;    в)  $A$  или  $A$ ;    г)  $A$  или (не  $A$ ).

## Выводы

- Логическое высказывание — это повествовательное предложение, про которое можно однозначно сказать, истинно оно или ложно.
- Алгебра логики — это математический аппарат, с помощью которого записывают, упрощают и преобразуют логические высказывания, вычисляют их значения.
- Логическая функция — это правило преобразования входных логических значений в логическое значение-результат.
- Если высказывание  $A$  истинно, то **не**  $A$  ложно, и наоборот.
- Высказывание  $A$  и  $B$  истинно тогда и только тогда, когда оба высказывания,  $A$  и  $B$ , истинны одновременно.
- Высказывание  $A$  или  $B$  истинно тогда, когда истинно хотя бы одно из высказываний,  $A$  или  $B$ , или оба они истинны одновременно.
- С помощью операций НЕ, И, ИЛИ можно выразить любую логическую операцию, поэтому говорят, что эти три операции составляют базис.

Нарисуйте в тетради интеллект-карту этого параграфа.



## Вопросы и задания

1. Объясните значения слов «логика», «формальная логика», «алгебра логики».
2. Можно ли считать высказываниями эти предложения?
  - 1) Не плачь, девчонка!
  - 2) Почему я водовоз?
  - 3) Купите слоника!
  - 4) Клубника очень вкусная.
  - 5) Сумма  $X$  и  $Y$  равна 36.
3. Как вы думаете, зачем в курсе информатики изучается логика?
4. Почему в таблице истинности для операции НЕ две строки, а для других изученных операций — четыре?
5. Сколько строк в таблице истинности выражения с тремя переменными? С четырьмя? С пятью?
6. В каком порядке обычно записываются значения переменных в таблице истинности? Зачем это нужно?
7. В чём различие арифметического и логического сложения?
8. Выполните по указанию учителя задания в рабочей тетради.





## Подготовьте сообщение

- «Информатика и логика»
- «Логическая операция исключающее ИЛИ»
- «Логическая операция импликация»
- «Логическая операция эквиваленция»



## § 9 Логические элементы

### Ключевые слова:

- логический элемент
- логическая схема

### Условные обозначения

Все узлы компьютера построены на логических элементах — электронных схемах, которые выполняют логические операции. В принципе, для того чтобы собрать компьютер, достаточно только базовых логических элементов — НЕ, И и ИЛИ.

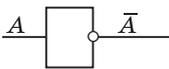


Рис. 2.6

Чтобы все инженеры-компьютерщики однозначно понимали схемы различных устройств, разработали государственный стандарт (ГОСТ 2.743-91), в котором определены условные обозначения логических элементов. **Элемент НЕ** обозначается прямоугольником с кружком на выходе (рис. 2.6).



Определите, какой сигнал будет на выходе в схемах, в которых последовательно соединяются: (а) два элемента НЕ; (б) три элемента НЕ (рис. 2.7).

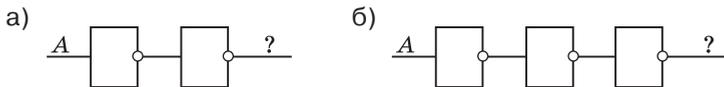


Рис. 2.7

Составьте таблицы истинности для обеих функций.

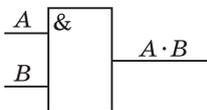


Рис. 2.8

**Элемент И** обозначается прямоугольником со знаком & в верхней части. У такого элемента два входа и один выход (рис. 2.8).

В иностранной литературе приняты другие условные обозначения для логических элементов. Используя дополнительные источники, выясните, как обозначаются за рубежом логические элементы НЕ, И и ИЛИ.

www

### Исследование логических элементов

Используя тренажёр (веб-страницу [element.htm](#)), постройте таблицу истинности элемента с кодовым именем NAND. Сравните её с таблицей истинности операции И. Как с помощью известных вам логических элементов собрать элемент NAND? Как бы вы его назвали?

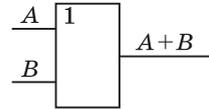


Рис. 2.9

**Элемент ИЛИ** обозначается прямоугольником с единицей в верхней части (рис. 2.9).

Используя тренажёр (веб-страницу [element.htm](#) на сайте поддержки учебника <http://kpolyakov.spb.ru/school/osnbook.htm>), постройте таблицу истинности элемента с кодовым именем NOR. Сравните её с таблицей истинности операции ИЛИ. Как с помощью известных вам логических элементов собрать элемент NOR? Как бы вы его назвали?

www

Используя тренажёр, постройте таблицы истинности элементов с кодовыми именами XOR и EQV. Как бы вы их назвали?

www

Как, используя элемент XOR и другие известные вам элементы, построить элемент EQV? Нарисуйте схему в тетради.

Сколько существует различных комбинаций значений трёх логических переменных? Сколько строк будет в таблице истинности логической схемы с тремя входами? В каком порядке вы их запишете?

Постройте таблицу истинности и запишите логическую функцию для каждой логической схемы (рис. 2.10).

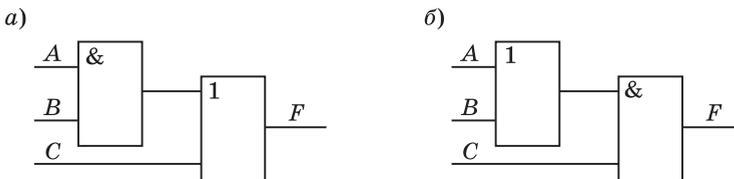


Рис. 2.10

Используя тренажёр (веб-страницу [element2.htm](#) на сайте поддержки учебника <http://kpolyakov.spb.ru/school/osnbook.htm>), постройте таблицы истинности схем с кодовыми именами KD и DK. Сравните их с таблицами истинности, полученными в предыдущем задании.

www

Используют также элементы, у которых некоторые входы или выходы отмечены кружками. Эти кружки обозначают дополнительные элементы НЕ, которые установлены на входе или на выходе. Например, схема на рис. 2.11, а может быть нарисована в подробной форме — так, как на рис. 2.11, б.

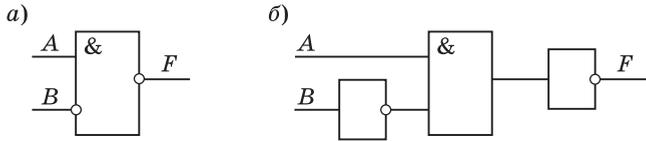


Рис. 2.11

Для каждой схемы (рис. 2.12) постройте таблицу истинности и запишите логическую функцию. Выполните работу в парах — один выполняет задания для первых трёх схем, другой — для последних трёх.

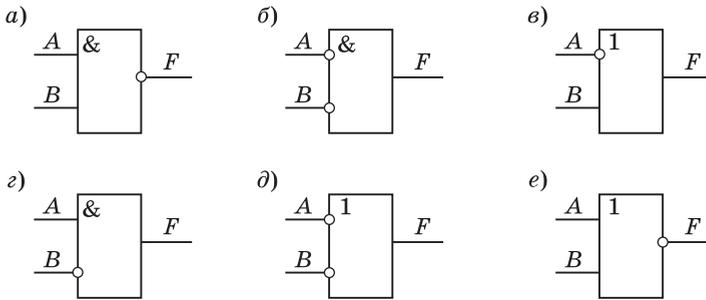
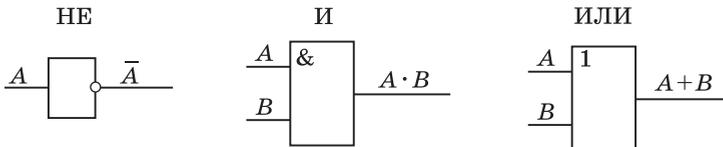


Рис. 2.12

Определите, какие логические схемы выполняют одну и ту же функцию. Запишите равенства, которые вы только что обнаружили.

## Выводы

- Все узлы компьютера построены на логических элементах — электронных схемах, которые выполняют логические операции.
- Для того чтобы собрать компьютер, достаточно только базовых логических элементов — НЕ, И, ИЛИ.
- Элементы, которые выполняют логические операции НЕ, И и ИЛИ, обозначаются на схемах логических устройств следующим образом:



- Кружок  $\circ$  на входе (или на выходе) элемента обозначает операцию НЕ, применённую к входному (или выходному) сигналу.

Нарисуйте в тетради интеллект-карту этого параграфа.

## Вопросы и задания

Выполните по указанию учителя задания в рабочей тетради.



## Практическая работа

Выполните практическую работу № 7 «Логические элементы».



## § 10

# Другие логические операции

### Ключевые слова:

- импликация
- исключающее ИЛИ
- эквиваленция

Мы изучили две логические операции с двумя переменными — И и ИЛИ. Существуют ли другие? И сколько их?

В таблице истинности любой логической операции с двумя переменными всего четыре строки, и эти таблицы отличаются только четырьмя значениями функции в последнем столбце. Поэтому можно придумать всего  $16 = 2^4$  различных логических операций с двумя переменными.

На уроках математики вы доказывали теоремы, в которых встречались выражения «если ..., то...», «тогда и только тогда, когда...». Эти связки тоже обозначают логические операции, с которыми мы сейчас и познакомимся.

## Импликация

Слово «**импликация**» означает «**следование**» — из одного утверждения следует другое. Если истинно первое, то истинно и второе. Например, возьмём высказывание

$X$  = Если идёт дождь, то Лена раскрывает зонтик.

Если ввести обозначения для простых высказываний:

$A$  = Идёт дождь,

$B$  = Лена раскрывает зонтик,

то можно записать высказывание  $X$  в символьном виде:

$$X = A \rightarrow B.$$

Стрелка вправо обозначает логическую операцию «импликация».

www

Используя дополнительные источники, выясните, от какого иностранного слова произошло слово «импликация». Что оно означает?

Теперь давайте разберёмся, когда такое высказывание будет истинно, а когда — ложно. Если дождь идёт ( $A = 1$ ) и Лена раскрывает зонтик ( $B = 1$ ), то импликация, очевидно, истинна. Если же при дожде Лена зонтик не раскрыла, импликация ложна (из  $A$  не следует  $B!$ ).

$A$	$B$	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

Рис. 2.13

А если дождя нет ( $A = 0$ )? Тогда о состоянии зонтика Лены мы ничего сказать не можем, потому что он может быть как закрыт, так и открыт. И в обоих случаях импликация будет истинна, потому что *не исключено*, что из  $A$  следует  $B$  (возможно, что, когда пойдёт дождь, Лена откроет зонтик). Говорят, что «из истины следует истина, а из лжи — что угодно».

Таблица истинности операции импликация показана на рис. 2.13. Как видим, эта функция равна нулю только при одном значении исходных данных:  $1 \rightarrow 0 = 0$ , во всех остальных случаях она равна 1.

Обычно, говоря «если..., то...», мы имеем в виду причинно-следственную связь, когда одно вызывает другое. Например, «Если светит солнце, то лужи высыхают» (именно потому, что светит солнце!). Импликация не говорит о причине и следствии, а показывает *возможность* такой связи. Например, может быть истинной импликация «если сегодня вторник, то Эльбрус покрыт снегом».

Импликация часто используется при решении логических задач. Например, формулировку вида «если  $A$ , то  $B$ » можно записать как  $A \rightarrow B = 1$ .



Постройте таблицу истинности логической функции  $B \rightarrow A$ <sup>1)</sup>. Сравните её с таблицей истинности функции  $A \rightarrow B$ . Выполняется ли для импликации переместительный закон (если поменять местами  $A$  и  $B$ , то результат не изменяется)?



Постройте таблицу истинности логической функции  $\bar{A} + B$ . Сравните её с таблицами истинности известных вам функций с двумя переменными. Какую формулу вы сейчас доказали?

<sup>1)</sup> В этой и следующих трёх задачах сохраняйте порядок столбцов в таблице истинности: в первом столбце записывайте значение  $A$ , во втором — значение  $B$ .

Постройте таблицу истинности логической функции  $\bar{B} \rightarrow \bar{A}$ . Сравните её с таблицами истинности известных вам функций с двумя переменными. Какую формулу вы сейчас доказали?

Постройте таблицу истинности логической функции  $(A \rightarrow B)$  и  $(B \rightarrow A)$ . Как бы вы назвали эту функцию?



## Эквиваленция

Таблица истинности, которую вы построили в последнем задании, определяет логическую операцию «эквиваленция» (её также называют «равнозначность» или «логическое равенство»), которая обозначается знаком  $\leftrightarrow$ . Эта операция соответствует связке «тогда и только тогда». Высказывание  $A \leftrightarrow B$  истинно в том и только в том случае, когда  $A$  и  $B$  равны (рис. 2.14).

$A$	$B$	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Рис. 2.14

Используя дополнительные источники, выясните, какие знаки кроме  $\leftrightarrow$  также используются для обозначения эквиваленции.



Постройте таблицу истинности логической функции  $A \cdot B + \bar{A} \cdot \bar{B}$ . Сравните её с таблицами истинности известных вам функций с двумя переменными. Какую формулу вы только что доказали?



Постройте таблицу истинности логической функции  $(A + \bar{B}) \cdot (\bar{A} + B)$ . Сравните её с таблицами истинности известных вам функций с двумя переменными. Какую формулу вы только что доказали?



Постройте таблицу истинности логической функции  $A \cdot \bar{B} + \bar{A} \cdot B$ . Сравните её с таблицами истинности известных вам функций с двумя переменными. Какую формулу вы только что доказали?



## Исключающее ИЛИ

Функция, которую вы исследовали в последнем задании, называется **исключающее ИЛИ**. Её результат равен 1, если значения входных сигналов не равны (рис. 2.15).

Исключающее ИЛИ обозначается знаком  $\oplus$ . Смысл этой операции хорошо передаёт поговорка «либо пан, либо пропал»: возможен только один вариант из двух, но не оба одновременно.

$A$	$B$	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Рис. 2.16



Сравните таблицы истинности обычной операции «ИЛИ» и «исключающего ИЛИ».



Сравните таблицы истинности логических функций  $A \oplus B$  и  $A \leftrightarrow B$ . Какая формула связывает две эти операции?

Операция исключающее ИЛИ иначе называется **разделительной дизъюнкцией** (это значит «один или другой, но не оба вместе») или **сложением по модулю два**. Второе название связано с тем, что её результат равен остатку от деления арифметической суммы  $A + B$  на 2:

$$A \oplus B = (A + B) \bmod 2.$$

Здесь mod обозначает операцию взятия остатка от деления.



Составьте таблицы истинности логических функций  $A \oplus 0$ ,  $A \oplus 1$  и  $A \oplus A$ . Сравните значения каждой функции со столбцом  $A$ . Как можно упростить эти формулы?



Сравните таблицу истинности логической функции  $A \cdot \bar{B} + \bar{A} \cdot B$  (см. задание выше) с таблицей истинности операции исключающее ИЛИ. Какую формулу вы только что доказали?



Постройте таблицу истинности логической функции  $(A + B) \cdot (\bar{A} + \bar{B})$ . Сравните её с таблицами истинности известных вам функций с двумя переменными. Какую формулу вы только что доказали?



Составьте таблицу истинности логической функции  $(A \oplus B) \oplus B$ . Сравните столбец значений функции со столбцом  $A$ . Какую формулу можно записать в результате сравнения?

Из результатов выполнения последнего задания следует важный вывод: если два раза применить к значению  $A$  операцию исключающее ИЛИ с одним и тем же значением  $B$ , то мы восстановим исходное значение  $A$ . В этом смысле исключающее ИЛИ — *обратимая операция*.



Какие ещё обратимые логические операции вы знаете?



Используя дополнительные источники, выясните, в каких языках программирования есть логическая операция «исключающее ИЛИ» и как она обозначается.



Запишите в тетради формулы, с помощью которых можно представить операции импликацию, эквиваленцию и исключающее ИЛИ через базовые логические операции: НЕ, И и ИЛИ. Используйте результаты выполнения заданий в рабочей тетради.

## Шифрование

Формулу  $(A \oplus B) \oplus B = A$  можно использовать для шифрования данных. Пусть  $A$  и  $B$  — это двоичные коды одинаковой длины. Чтобы зашифровать данные  $A$  с использованием ключа  $B$ , надо применить операцию исключающее ИЛИ отдельно для каждого двоичного разряда  $A$  и  $B$ . Для расшифровки ещё раз применяется исключающее ИЛИ с тем же ключом  $B$ . Нужно отметить, что такой метод шифрования очень нестойкий: для достаточно длинных текстов его легко взломать.

Например, пусть Алиса хочет секретно передать Борису число 9 в виде четырёхбитной двоичной цепочки 1001. Для шифрования они заранее договорились использовать ключ — двоичную цепочку 0101. Алиса шифрует каждый бит отдельно, выполняя операцию исключающее ИЛИ с соответствующим битом ключа, в результате получается цепочка 1100 (рис. 2.16).

Данные	1	0	0	1
Ключ	0	1	0	1
Результат	1	1	0	0

Рис. 2.16

Борис, получив цепочку 1100, применяет для расшифровки тот же ключ 0101 и получает исходное сообщение — 1001 (рис. 2.17).

Зашифрованные данные	1	1	0	0
Ключ	0	1	0	1
Результат	1	0	0	1

Рис. 2.17

*Работа в парах.* Договоритесь с напарником, какой четырёхбитный ключ вы будете использовать для шифровки. Зашифруйте год, когда произошло какое-нибудь известное историческое событие (четырёхбитный код каждой цифры отдельно с тем же ключом), и передайте напарнику зашифрованное сообщение. Попросите его назвать это событие.



## Выводы

- Импликация  $A \rightarrow B$  истинна во всех случаях, кроме  $A = 1$  и  $B = 0$ .
- Эквиваленция  $A \leftrightarrow B$  истинна тогда и только тогда, когда  $A$  и  $B$  имеют одинаковые значения.
- Исключающее ИЛИ  $A \oplus B$  принимает значение «истина» тогда и только тогда, когда значения  $A$  и  $B$  различны.

- Операция исключающее ИЛИ обратима: если взять любое значение  $A$  и выполнить дважды исключающее ИЛИ с любой постоянной, то получится исходное значение  $A$ . Эту особенность можно использовать для простого шифрования.
- Логические функции  $A \rightarrow B$ ,  $A \leftrightarrow B$  и  $A \oplus B$  можно выразить через базовые логические операции НЕ, И и ИЛИ:

$$A \rightarrow B = \bar{A} + B;$$

$$A \leftrightarrow B = A \cdot B + \bar{A} \cdot \bar{B} = (A + \bar{B}) \cdot (\bar{A} + B);$$

$$A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B = (A + B) \cdot (\bar{A} + \bar{B}).$$



Нарисуйте в тетради интеллект-карту этого параграфа.

### Вопросы и задания

1. Известно, что высказывание «Если  $X$  делится на 6, то  $X$  делится на 15» истинно. Будет ли верно это высказывание для чисел 7, 15, 18, 90?
2. Известно, что высказывание «Если Алиса ходила в поход, то Борис тоже ходил» ложно. Кто ходил в поход?
3. В лесу живут два медведя. Известно, что в берлоге всегда находится один медведь, а два медведя в берлогу не помещаются. С помощью какой логической операции можно сформулировать это высказывание?
4. Выполните по указанию учителя задания в рабочей тетради.



### Подготовьте сообщение

- а) «Логическая операция штрих Шеффера»
- б) «Логическая операция стрелка Пирса»
- в) «Частотный анализ текста»

### Практическая работа

Выполните практическую работу № 8 «Шифрование».

## § 11 Логические выражения

**Ключевые слова:**

- формализация
- логическое выражение
- таблица истинности
- вычислимое выражение
- тождественно истинное выражение
- тождественно ложное выражение
- равносильные выражения
- логическая схема

## Формализация

Обозначив простые высказывания буквами — логическими переменными и используя логические операции, можно записать любое высказывание в виде логического выражения.

**Логическое выражение** — это выражение, результат вычисления которого — логическое значение (истина или ложь).

Например, пусть система сигнализации должна дать аварийный сигнал, если вышли из строя два из трёх двигателей самолёта. Обозначим высказывания:

$A$  = Первый двигатель вышел из строя;  
 $B$  = Второй двигатель вышел из строя;  
 $C$  = Третий двигатель вышел из строя;  
 $X$  = Аварийная ситуация.

Тогда логическое высказывание  $X$  можно записать в виде логического выражения (**логической формулы**):

$$X = (A \text{ и } B) \text{ или } (A \text{ и } C) \text{ или } (B \text{ и } C).$$

Это выражение может быть записано с помощью других обозначений:

$$X = (A \cdot B) + (A \cdot C) + (B \cdot C). \quad (*)$$

Таким образом, мы выполнили *формализацию*.

**Формализация** — это переход от конкретного содержания высказываний к записи с помощью формального языка.

При вычислении логических выражений установлен такой порядок выполнения операций:

- 1) действия в скобках;
- 2) отрицание (НЕ);
- 3) логическое умножение (И), слева направо;
- 4) логическое сложение (ИЛИ), слева направо.

Можно ли убрать скобки в выражении (\*). Почему?

Уберите лишние скобки в логических выражениях:

а)  $X = (A + (B \cdot C) \cdot (A + C));$

б)  $X = (A + \bar{B}) \cdot (\bar{C} \cdot A) \cdot (A + (\bar{B} + \bar{C})).$



Вычислите значение логического выражения  $X = (A \cdot B + C) \cdot (\bar{A} + \bar{C})$  при:

а)  $A = 0, B = 0, C = 1$ ;

б)  $A = 0, B = 1, C = 1$ ;

в)  $A = 1, B = 1, C = 0$ .

## Таблицы истинности

Любую логическую функцию можно задать с помощью *таблицы истинности*, которая показывает, чему равно значение логического выражения при всех возможных комбинациях значений исходных переменных. Построим таблицу истинности для выражения

$$X = A \text{ и не } B \text{ или не } A \text{ и } B,$$

которое можно также записать в виде

$$X = A \cdot \bar{B} + \bar{A} \cdot B.$$



Сколько строк в таблице истинности выражения с двумя переменными?

Будем вычислять выражение по частям: добавим в таблицу истинности дополнительные столбцы  $A \cdot \bar{B}$  и  $\bar{A} \cdot B$ , а потом выполним операцию ИЛИ с этими значениями (рис. 2.18).

$A$	$B$	$A \cdot \bar{B}$	$\bar{A} \cdot B$	$X$
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

Рис. 2.18

Из этой таблицы истинности видно, что при некоторых значениях переменных значение  $X$  истинно, а при некоторых — ложно. Такие выражения называют **вычислимыми**.

Высказывание «Вася — школьник или он не учится в школе» всегда истинно (для любого Васи). Выражение, истинное при любых значениях переменных, называется **тождественно истинным** или **тавтологией**.

Высказывание «Сегодня безветрие, и дует сильный ветер» никогда не может быть истинным. Соответствующее логическое

выражение всегда ложно, оно называется **тождественно ложным** или **противоречием**.

Выполните формализацию высказываний, о которых шла речь в последних двух абзацах. Запишите формулы для упрощения каждого из них.

Постройте самостоятельно таблицу истинности логического выражения  $X = (A + B) \cdot (\bar{A} + \bar{B})$ . Сравните её с таблицей на рис. 2.18. Истинно ли высказывание «Разные логические выражения могут определять одну и ту же логическую функцию»?

Если два выражения принимают одинаковые значения при всех значениях переменных, они называются **равносильными** или **тождественно равными**. Равносильные выражения определяют одну и ту же логическую функцию, т. е. при одинаковых исходных данных приводят к одинаковым результатам.

Сколько строк в таблице истинности выражения с тремя переменными?

Постройте таблицу истинности логического выражения  $X = A \cdot B + A \cdot C + B \cdot C$ .

Предположим, что нам известна только часть таблицы истинности для функции трёх переменных (рис. 2.19).

<i>A</i>	<i>B</i>	<i>C</i>	<i>F</i>
0	0	1	1
0	1	1	0
1	1	1	0

**Рис. 2.19**

Как вы думаете, можно ли по такой части таблицы истинности определить логическую функцию? Почему?

Всего в таблице истинности функции от трёх переменных  $2^3 = 8$  строк, для каждой из них нужно знать, чему равно значение функции. В нашем примере пять значений функции неизвестны, причём каждое из них может быть равно 0 или 1, т. е. у нас есть 5 свободных бит.



Сколько различных значений можно закодировать с помощью 5 бит? Как вы нашли это значение?

Итак, таблице на рис. 2.19 могут соответствовать 32 различные логические функции. Проверим, подходят ли следующие варианты:

- а)  $A + B + \bar{C}$ ;    б)  $\bar{A} \cdot \bar{C} + B$ ;    в)  $A + \bar{B} \cdot C$ ;  
 г)  $A \cdot B \cdot C$ ;    д)  $A \cdot B + C$ .

Прежде всего, заметим, что в столбце значений функции два нуля и одна единица. Следовательно, вариант а) не подходит, потому что цепочка операций ИЛИ со всеми переменными (или их инверсиями, обратными значениями) даст только один ноль — в случае, когда все слагаемые равны нулю.

Проверяем вариант б), подставляя значения переменных  $A$ ,  $B$  и  $C$  сначала из первой строки таблицы, потом из второй и третьей. В первой строке получаем  $\bar{A} \cdot \bar{C} + B = 1 \cdot 1 + 0 = 1$ , этот результат совпадает со значением функции в этой строке. Для второй строки  $\bar{A} \cdot \bar{C} + B = 1 \cdot 1 + 1 = 1$ , значит, эта функция тоже не подходит.



Проверьте самостоятельно, подходят ли остальные варианты.



Скажите без вычислений, сколько нулей и сколько единиц должно быть в последнем столбце таблицы истинности функций с тремя переменными:

- а)  $\bar{A} + \bar{B} + \bar{C}$ ;    б)  $\bar{A} \cdot \bar{C} \cdot B$ ;    в)  $A \cdot \bar{B} \cdot \bar{C}$ ;  
 г)  $\bar{A} + B + C$ ;    д)  $\bar{A} + \bar{B} + C$ .

Теперь предположим, что нам известна часть таблицы какой-то логической функции, причём с пропусками (рис. 2.20).

$A$	$B$	$C$	$F$
0		1	0
1	0		1
		1	1

Рис. 2.20

Выясним, какие из следующих функций могут соответствовать этой таблице:

- а)  $\bar{A} + B + \bar{C}$ ;    б)  $\bar{A} \cdot C \cdot B$ ;    в)  $A + \bar{B} + \bar{C}$ ;  
 г)  $A \cdot \bar{B} \cdot C$ ;    д)  $A + B + \bar{C}$ .

Во-первых, обратим внимание, что в столбце значений функции две единицы, поэтому сразу делаем вывод, что это не могут быть цепочки двух операций И (ответы б) и г) неверные). Остались три цепочки из операций ИЛИ, причём для верхней строки (при  $A = 0$  и  $C = 1$ ) мы должны получить 0 при каком-то выборе неизвестного значения  $B$ . Подставляем  $A = 0$  и  $C = 1$  в формулы-кандидаты:

$$\text{а) } 1 + B + 0; \quad \text{в) } 0 + \bar{B} + 0; \quad \text{д) } 0 + B + 0.$$

Видим, что в случае а) сумма не может быть равна нулю, это неверный ответ. А варианты в) и д) подходят: в первом из них нужно в первой строке поставить в пустую ячейку 1, а во втором — 0.

Кратко решение можно записать так:

- 1) поскольку в столбце значений функции один ноль, это не может быть цепочка операций И; остаются только цепочки операций ИЛИ;
- 2) для того чтобы получить нулевое значение функции в первой строке таблицы, нужно, чтобы переменная  $A$  (равная 0 в этой строке) входила в логическую сумму без инверсии (к ней не должна применяться операция НЕ) а переменная  $C$  (равная 1) — с инверсией;
- 3) этим условиям удовлетворяют функции в) и д).

Известна часть таблицы какой-то логической функции с пропусками (рис. 2.21).



$A$	$B$	$C$	$F$
0		1	0
1	0		1
		1	0

Рис. 2.21

Выясните, какие из следующих функций могут соответствовать этой таблице:

$$\begin{array}{lll} \text{а) } A \cdot \bar{B} + \bar{C}; & \text{б) } \bar{A} \cdot C \cdot B; & \text{в) } A + \bar{B} + \bar{C}; \\ \text{г) } A \cdot \bar{B} \cdot C; & \text{д) } A + B + \bar{C}. \end{array}$$

## Составление условий

Логические выражения часто используются при решении математических задач с помощью компьютеров.

Построим условия (логические выражения), соответствующие заштрихованным областям на числовой оси (рис. 2.22).

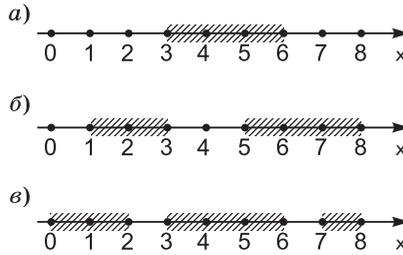


Рис. 2.22

На рисунке 2.22, *a* выделен отрезок  $[3; 6]$ . Для того чтобы определить такую область, нужно ограничить значение  $x$  с двух сторон: оно должно быть больше или равно трём и меньше или равно шести, причем эти два условия должны выполняться *одновременно*, т. е. их нужно связать с помощью операции И:

$$(3 \leq x) \text{ и } (x \leq 6).$$

Область на рис. 2.22, *б* — это объединение двух отрезков. Мы можем отдельно записать условия для каждого отрезка и связать их с помощью операции ИЛИ:

$$(1 \leq x) \text{ и } (x \leq 3) \quad \text{или} \quad (5 \leq x) \text{ и } (x \leq 8).$$



Запишите условие, которое определяет область на рис. 2.22, *в*.

Теперь рассмотрим задачу с областью на плоскости. Запишем условие, соответствующее заштрихованной части (рис. 2.23).

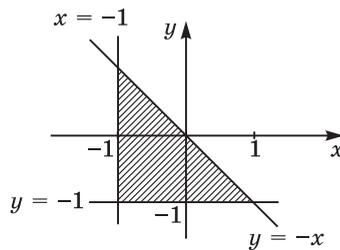


Рис. 2.23

Требуется составить логическое выражение, зависящее от переменных  $x$  и  $y$ , которое будет равно 1 (истинно) внутри заштрихованной области и равно 0 вне её.

Выделенная область находится справа от вертикальной линии  $x = -1$ , поэтому должно выполняться условие  $x \geq -1$ , которое определяет полуплоскость (рис. 2.24).

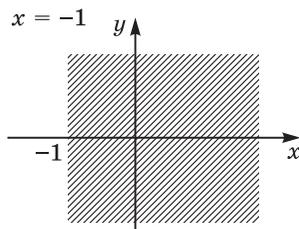


Рис. 2.24

Кроме того, все точки области находятся выше горизонтальной линии  $y = -1$  и ниже наклонной прямой  $y = -x$ , что даёт ещё два условия:  $y \geq -1$  и  $y \leq -x$ . Все три простых условия должны выполняться одновременно, поэтому их нужно связать с помощью двух операций И:

$$(x \geq -1) \text{ и } (y \geq -1) \text{ и } (y \leq -x).$$

Запишите условие, которое определяет области на рисунках (рис. 2.25).

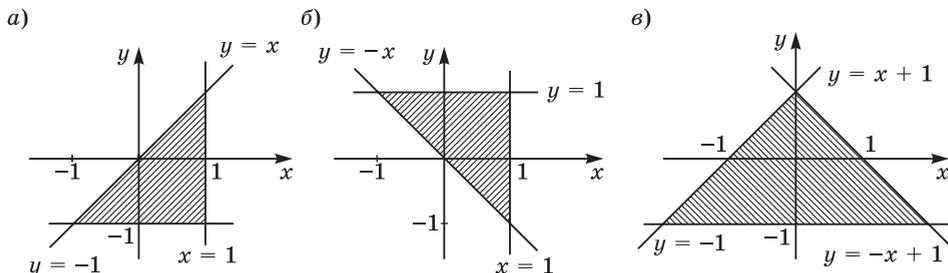


Рис. 2.25

Построим логическое выражение для области на рис. 2.26.

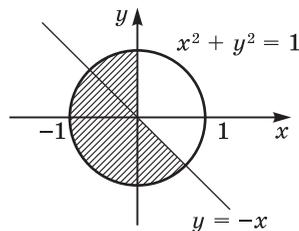


Рис. 2.26

Можно разделить заштрихованную область на две части (рис. 2.27).

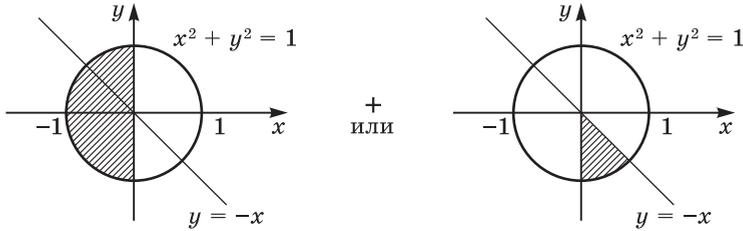


Рис. 2.27

Обе области находятся внутри круга радиуса 1 с центром в начале координат, т. е. в области  $x^2 + y^2 \leq 1$ . Первая область описывается выражением

$$(x^2 + y^2 \leq 1) \text{ и } (x \leq 0),$$

а вторая — выражением

$$(x^2 + y^2 \leq 1) \text{ и } (x \geq 0) \text{ и } (y \leq -x).$$

Поскольку нам нужно «сложить» две области, эти выражения нужно объединить с помощью операции логического сложения (ИЛИ):

$$(x^2 + y^2 \leq 1) \text{ и } (x \leq 0) \text{ или } (x^2 + y^2 \leq 1) \text{ и } (x \geq 0) \text{ и } (y \leq -x).$$

Общее условие  $x^2 + y^2 \leq 1$  можно вынести за скобки:

$$(x^2 + y^2 \leq 1) \text{ и } ((x \leq 0) \text{ или } (x \geq 0) \text{ и } (y \leq -x)).$$



Попробуйте ещё упростить полученное условие.



Запишите условие, которое определяет области на рисунках (рис. 2.28).

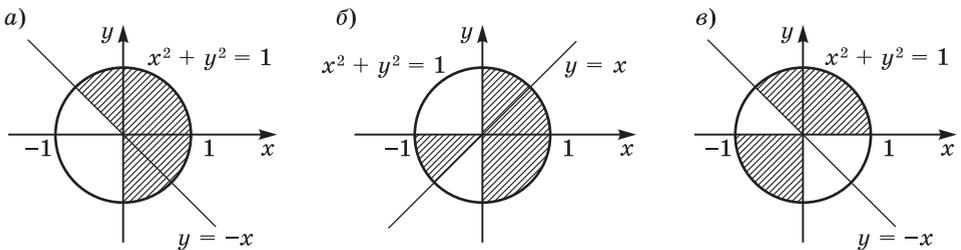


Рис. 2.28



## Логические схемы

Вернёмся снова к примеру с системой аварийной сигнализации самолёта. Предположим, что на каждом двигателе установлен логический датчик, который выдаёт условный сигнал 1 (например, высокий уровень напряжения), если двигатель неисправен, и условный сигнал 0, если двигатель исправен. Требуется построить **логическую схему** — схему логического устройства, — которая при аварии выдаёт условный сигнал 1, а в режиме нормальной работы — сигнал 0.

Сигналы от трёх двигателей назовём  $A$ ,  $B$  и  $C$ . В начале параграфа мы уже составили логическое выражение для запуска аварийной сигнализации:

$$X = A \cdot B + A \cdot C + B \cdot C.$$

Здесь три логических умножения и два логических сложения. Сначала выполняются все операции умножения (слева направо), а затем — все операции сложения (тоже слева направо). Расставим номера операций:

$$X = A \cdot B + A \cdot C + B \cdot C.$$

Последней выполняется вторая операция сложения. Поэтому последний элемент в схеме — это элемент логического сложения ИЛИ (рис. 2.29).

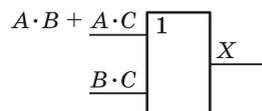


Рис. 2.29

На первый вход этого элемента ИЛИ подаётся сигнал  $A \cdot B + A \cdot C$ , в этом выражении последняя операция — логическое сложение, добавляем ещё один элемент ИЛИ (рис. 2.30).

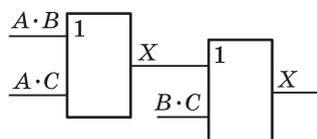


Рис. 2.30

Три операции логического умножения добавляю в схему три элемента И (рис. 2.31).

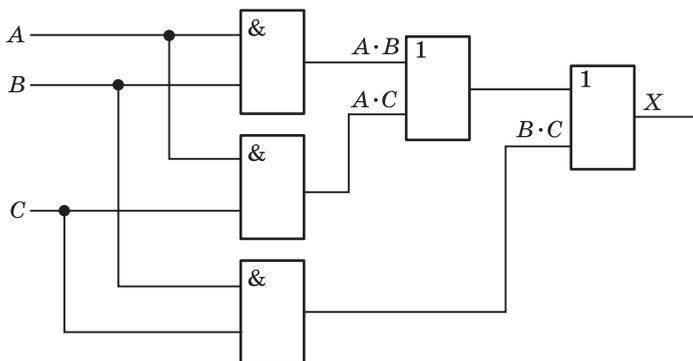


Рис. 2.31



## Выводы

- Формализация — это переход от конкретного содержания высказываний к записи с помощью формального языка.
- В логических выражениях операции выполняются в следующем порядке:
  - 1) действия в скобках;
  - 2) отрицание (НЕ);
  - 3) логическое умножение (И), слева направо;
  - 4) логическое сложение (ИЛИ), слева направо.Для изменения порядка действий используются скобки.
- Таблица истинности логического выражения показывает, чему равно значение выражения при всех возможных комбинациях значений исходных переменных.
- Логические выражения, истинность которых зависит от значений исходных переменных, называют вычислимыми.
- Логическое выражение, которое всегда истинно, называется тождественно истинным или тавтологией. Выражение, которое всегда ложно, называют тождественно ложным или противоречием. Пример тождественно истинного выражения:  $A + \bar{A}$ , пример тождественно ложного:  $A \cdot \bar{A}$ .
- Два выражения, принимающие одинаковые значения при всех значениях переменных, называются равносильными или тождественно равными. Равносильные выражения определяют одну и ту же логическую функцию.

Нарисуйте в тетради интеллект-карту этого параграфа.



## Вопросы и задания

1. Что можно сделать для того, чтобы изменить порядок выполнения действий в логических выражениях?
2. Поясните разницу между терминами «логическое выражение» и «логическая функция».
3. Как можно доказать (или опровергнуть) логическую формулу?
4. Можно ли сказать, что таблица истинности однозначно определяет:
  - а) логическое выражение;
  - б) логическую функцию?
5. Что такое вычисляемое логическое выражение?
6. Что такое равносильные выражения?
7. Выполните по указанию учителя задания в рабочей тетради.



## § 12

### Множества и логика

**Ключевые слова:**

- множество
- дополнение
- пересечение
- объединение
- диаграмма Эйлера–Венна
- поисковый запрос

### Множества

**Множество** — некоторый набор элементов, каждый из которых отличается от остальных. Множество может состоять из конечного числа элементов (например, множество букв русского алфавита), бесконечного числа элементов (например, множество натуральных чисел) или вообще быть пустым (например, множество слонов, живущих на Северном полюсе). Пустое множество обозначается символом  $\emptyset$ . Множества, с которыми работает компьютер, не могут быть бесконечными, потому что его память конечна.

Чтобы определить множество, мы можем перечислить все его элементы. Например, множество, состоящее из Васи, Пети и Коли, можно записать так: {Вася, Петя, Коля}.



Запишите в виде перечисления элементов:

- а) множество натуральных чисел на отрезке  $[-5; 5]$ ;
- б) множество чётных однозначных чисел;
- в) множество целых чисел, делящихся на 4, на отрезке  $[0; 22]$ ;
- г) множество простых чисел на отрезке  $[5; 20]$ .

Можно задать множество иначе: определить условие (*логическое выражение*), которое должно быть истинным для всех элементов множества и ложным для всех элементов, не входящих во множество. Например, можно ввести множество драконов с пятью зелёными хвостами или множество чисел, делящихся на 11.



Запишите (словами или в символьном виде) условие, которое определяет множество:

- а)  $\{1, 3, 5, 7, 9\}$ ;
- б)  $\{5, 6, 7\}$ ;
- в)  $\{а, е, ё, и, о, у, ы, э, ю, я\}$ ;
- г)  $\{17, 34, 51, 68, 85\}$ ;
- д)  $\{00, 01, 10, 11\}$ ;
- е)  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, А, В, С\}$ ;
- ж) отрезок  $[0; 1]$ .

Хотя множество — это математическое понятие, с множествами мы имеем дело каждый раз, когда обращаемся к поисковой системе в Интернете: ведь нас интересует множество страниц, на которых есть нужная нам информация. Задать такое множество перечислением элементов невозможно: во-первых, мы не знаем адресов этих страниц; во-вторых, их очень много. Поэтому для того, чтобы задать нужное нам множество, требуется написать **поисковый запрос** — логическое выражение, которое его определяет.

## Диаграммы Эйлера–Венна

Множества удобно изображать графически, в виде диаграмм. Их называют **диаграммами Эйлера–Венна** в честь авторов этой идеи — математика Леонарда Эйлера и логика Джона Венна. На такой диаграмме каждому множеству соответствует какая-то область (круг, прямоугольник и др.) — рис. 2.34. Все элементы внутри этой области принадлежат множеству, все элементы вне области — не принадлежат.

Вы уже знаете, что множество можно задать условием (логическим выражением), которое выполняется для всех элементов множества и не выполняется для всех элементов, не входящих в него. Дальше для сокращения записи мы будем вместо слов «множество, для которого выполняется условие  $A$ » писать просто «множество  $A$ ». Тогда множество «НЕ  $A$ » на диаграмме — это все точки за границами круга (рис. 2.35).

Такое множество называется **дополнением** множества  $A$  до **универсального множества**  $U$ , включающего все элементы некоторого класса. Например, если мы рассматриваем только целые числа и  $A$  — это множество чётных целых чисел, то  $\bar{A}$  — множество нечётных целых чисел.

Можно считать, что дополнение  $\bar{A}$  — это «разность» между универсальным множеством  $U$  и множеством  $A$ , т. е. все элементы из  $U$ , которые не входят в  $A$ .

Для каждого из следующих множеств выберите универсальное множество и запишите дополнение  $\bar{A}$ :

- $A = \{1, 3, 5, 7, 9\}$ ;
- $A = \{a, e, \text{ё}, и, o, y, ы, э, ю, я\}$ ;
- $A = \{17, 34, 51, 68, 85\}$ ;
- $A = \{00, 10\}$ ;
- $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C\}$ ;
- $A =$  отрезок  $[0; 1]$ .



Рис. 2.34

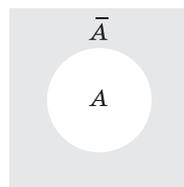


Рис. 2.35



На диаграмме можно изображать несколько множеств, каждому из них соответствует своя область (круг). Круги на диаграмме могут пересекаться. Элементы, расположенные в общей части кругов  $A$  и  $B$ , — это **пересечение** множеств  $A$  и  $B$ . Для этих элементов выполняется как условие  $A$ , так и условие  $B$ , т. е. выполняется условие  $A$  и  $B$  ( $A \cdot B$ ) — рис. 2.36.

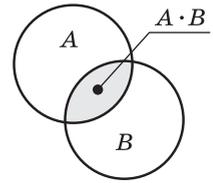


Рис. 2.36

Если круги не пересекаются (множества не содержат общих элементов), их пересечение — это пустое множество  $\emptyset$ .

Для пары множеств определите пересечение  $A \cdot B$ .

- а)  $A = \{1, 3, 5, 7, 9\}$ ,  $B = \{1, 5, 6, 9, 12\}$ ;  
 б)  $A = \{а, б, в, г, д, е, ё, ж\}$ ,  $B = \{а, е, ё, и, о, у, ы, э, ю, я\}$ ;  
 в)  $A = \{17, 34, 51, 68, 85\}$ ,  $B = \{17, 34, 51, 68, 85\}$ ;  
 г)  $A = \{00, 10\}$ ,  $B = \{01, 11\}$ ;  
 д)  $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C\}$ ,  $B = \{A, B, C, D, E, F, G, H\}$ ;  
 е)  $A = [5; 15]$ ,  $B = [10; 20]$ ;  
 ж)  $A = [5; 15]$ ,  $B = [0; 20]$ ;  
 з)  $A = [5; 15]$ ,  $B = [10; 12]$ ;  
 и)  $A = [5; 15]$ ,  $B = [20; 30]$ .

Элементы, входящие хотя бы в одно из множеств: в  $A$  или в  $B$ , образуют новое множество, которое называется **объединением** множеств  $A$  и  $B$ . Для всех элементов этого множества выполняется условие  $A$  **или**  $B$  ( $A + B$ ) — рис. 2.37.

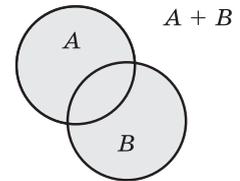


Рис. 2.37

Для пары множеств определите объединение  $A + B$ :

- а)  $A = \{1, 3, 5, 7, 9\}$ ,  $B = \{1, 5, 6, 9, 12\}$ ;  
 б)  $A = \{а, б, в, г, д, е, ё, ж\}$ ,  $B = \{а, е, ё, и, о, у, ы, э, ю, я\}$ ;  
 в)  $A = \{17, 34, 51, 68, 85\}$ ,  $B = \{17, 34, 51, 68, 85\}$ ;  
 г)  $A = \{00, 10\}$ ,  $B = \{01, 11\}$ ;  
 д)  $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C\}$ ,  $B = \{A, B, C, D, E, F, G, H\}$ ;  
 е)  $A = [5; 15]$ ,  $B = [10; 20]$ ;  
 ж)  $A = [5; 15]$ ,  $B = [0; 20]$ ;  
 з)  $A = [5; 15]$ ,  $B = [10; 12]$ ;  
 и)  $A = [5; 15]$ ,  $B = [20; 30]$ .

Подобные диаграммы можно нарисовать для любого логического выражения, ведь каждое из них определяет некоторое множество. Например, возьмём выражение  $A + B$ . Оно равно 0 только при  $A = 1$  и  $B = 0$ , поэтому на диаграмме незакрашенной

останется только область, которая входит в круг  $A$  и не входит в круг  $B$  (рис. 2.38).

В тетради постройте диаграммы для логических выражений:

- а)  $A + \bar{B}$ ;
- б)  $A \cdot \bar{B} + \bar{A} \cdot B$ ;
- в)  $A \cdot B + \bar{A} \cdot \bar{B}$ .

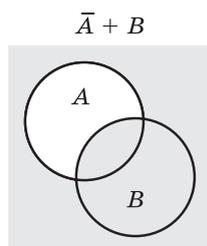


Рис. 2.38

Диаграмма для трёх переменных содержит три круга, каждый из которых (в общем случае) пересекается с двумя другими (рис. 2.39).

Для удобства на рис. 2.39 области пронумерованы. Запишем, для примера, логическое выражение для области 3. Эта область находится внутри кругов  $A$  и  $B$  (следовательно, выражения  $A$  и  $B$  истинны), но вне круга  $C$ , поэтому выражение  $C$  ложно. Получается условие  $A$  и  $B$  и (не  $C$ ), или, в других обозначениях,  $A \cdot B \cdot \bar{C}$ .

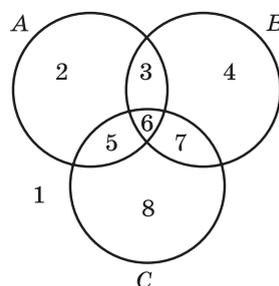


Рис. 2.39

Запишите в тетради логические выражения для остальных областей на рис. 2.39.

Для того чтобы найти выражение для объединения двух или нескольких областей, надо сложить (используя логическое сложение — операцию ИЛИ) выражения для всех составляющих. Например, выражение для объединения областей 3 и 4 на рис. 2.39 имеет вид:

$$3 + 4: A \cdot B \cdot \bar{C} + \bar{A} \cdot B \cdot \bar{C}.$$

Вместе с тем точки в этих областях отличаются от других тем, что они входят в область  $B$  и не входят в область  $C$ . Поэтому справедлива более простая формула:

$$3 + 4: B \cdot \bar{C}.$$

Это означает, что логические выражения в некоторых случаях можно упростить.

### Количество элементов во множестве

Предположим, что множество  $A$  содержит 10 элементов, множество  $B$  — 15 элементов, а их пересечение (множество  $A \cdot B$ ) —

два элемента. Как определить, сколько элементов содержится во множестве  $A + B$ ?

Попробуем рассмотреть задачу в общем виде и вывести формулу для её решения. Обозначим через  $N_X$  число элементов в области  $X$ . Далее операцию И будет обозначать символом  $\&$ , а операцию ИЛИ — символом  $|$  (именно эти символы используются в поисковых запросах в Интернете).

Построим диаграмму с двумя областями —  $A$  и  $B$ . Эти области могут быть разделены (рис. 2.40, а) или пересекаться (рис. 2.40, б).

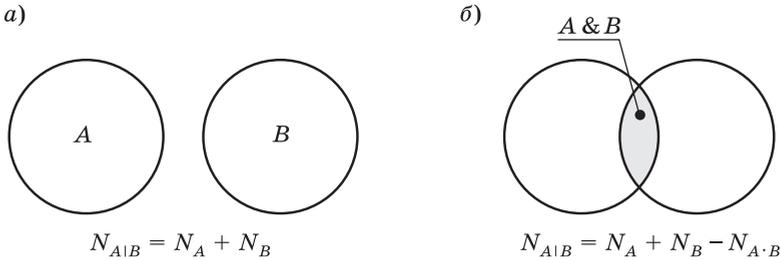


Рис. 2.40

В первом случае (рис. 2.40, а), когда области не пересекаются, получаем очевидную формулу:

$$N_{A|B} = N_A + N_B.$$

Во втором случае (рис. 2.40, б) в сумму  $N_A + N_B$  общие элементы (элементы множества  $N_{A\&B}$ ) входят дважды. Поэтому, чтобы получить количество элементов в объединении множеств, нужно из этой суммы вычесть число общих элементов:

$$N_{A|B} = N_A + N_B - N_{A\&B}. \quad (*)$$

Эта формула, которую называют формулой включений и исключений, справедлива и для рис. 2.40, а, где  $N_{A\&B} = 0$ .



Используя формулу (\*), постройте выражения для вычисления  $N_A$  и  $N_{A\&B}$ .



Рыбаки в посёлке ловят только лещей и судаков. 25 рыбаков ловят лещей, 12 рыбаков — судаков, причём 5 рыбаков ловят и лещей, и судаков. Сколько всего рыбаков в посёлке? Выполните формализацию задачи и решите её.



У дяди Вани живёт 30 животных: овцы и кролики. Все кролики белые, а у овец разный цвет шерсти. Известно, что у дяди Вани живёт 18 овец и 25 животных с белой шерстью. Сколько белых овец у дяди Вани? Выполните формализацию задачи и решите её.

В физико-математическом классе 27 учеников. Среди них нет таких, которые не программируют и не ходят в турпоходы. Известно, что 20 человек ходят в турпоходы, среди них 5 программистов. Сколько в классе программистов? Выполните формализацию задачи и решите её.



## Сложные запросы в поисковых системах

Для решения задач, в которых используются множества, например множества страниц, полученных от поисковой системы в ответ на какой-то запрос, удобно применять диаграммы Эйлера–Венна.

**Задача 1.** Известно количество страниц, которые находит поисковый сервер по следующим запросам (здесь символ «&» обозначает операцию И, а «|» — операцию ИЛИ):

собаки   кошки	770
кошки	550
собаки & кошки	100

Сколько страниц будет найдено по запросу собаки?

Введём два множества:  $A$  — множество страниц, где есть слово «собаки»,  $B$  — множество страниц со словом «кошки». По формуле, которая получена в предыдущем пункте, получаем:

$$N_A = N_{A|B} - N_B + N_{A\&B} = 770 - 550 + 100 = 320.$$

Известно количество страниц, которые находит поисковый сервер по следующим запросам:

незабудка	220
лилия & незабудка	100
лилия   незабудка	450

Сколько страниц найдёт этот сервер по запросу лилия?

Известно количество страниц, которые находит поисковый сервер по следующим запросам:

енот	200
кашалот	300
кашалот   енот	450

Сколько страниц найдет этот сервер по запросу

кашалот & енот?





Известно количество страниц, которые находит поисковый сервер по следующим запросам:

Италия	320
Франция	450
Франция & Италия	80

Сколько страниц найдёт этот сервер по запросу  
Франция | Италия?

Рассмотрим теперь более сложную задачу с тремя областями.

**Задача 2.** Известно количество страниц, которые находит поисковый сервер по следующим запросам:

собаки & лемуры	320
кошки & лемуры	280
(кошки   собаки) & лемуры	430

Сколько страниц будет найдено по запросу  
кошки & собаки & лемуры?

Заметим, что во всех запросах есть часть & лемуры. Это означает, что область поиска во всех случаях ограничена страницами, на которых встречается слово «лемуры».

Обозначим буквами  $C$ ,  $K$  и  $L$  области (группы страниц), содержащие ключевые слова «собаки», «кошки» и «лемуры» соответственно. Нас интересует только область, выделенная фоном на рис. 2.41, а.

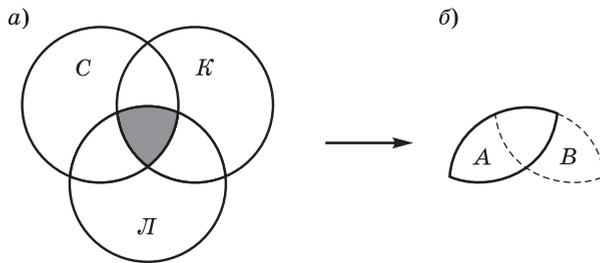


Рис. 2.41

Эта область образована в результате пересечения двух областей (рис. 2.41, б):

$$A = \text{собаки \& лемуры}$$

$$B = \text{кошки \& лемуры}$$

Поэтому задачу можно свести к задаче с двумя областями.

Известно количество страниц, которые находит поисковый сервер по следующим запросам:

A	320
B	280
A   B	430

Сколько страниц будет выдано по запросу A & B?

Используя формулу включений и исключений, полученную в предыдущем пункте, находим:

$$N_{A \& B} = N_A + N_B - N_{A|B} = 320 + 280 - 430 = 170.$$

Известно количество страниц, которые находит поисковый сервер по следующим запросам:

берёза & сирень	220
берёза & сирень & арбуз	30
сирень & (берёза   арбуз)	340

Сколько страниц найдёт этот сервер по запросу арбуз & сирень?

Известно количество страниц, которые находит поисковый сервер по следующим запросам:

яхта & диван	270
диван & пирог	350
яхта & диван & пирог	80

Сколько страниц найдёт этот сервер по запросу (пирог | яхта) & диван?

Задачу с тремя областями не всегда удаётся свести к более простой задаче с двумя областями. Серьёзным упрощением может стать то, что какие-то два множества не имеют общих элементов.

Если два множества не имеют общих элементов, что можно сказать об их изображении на диаграмме Эйлера–Венна?

**Задача 3.** Известно количество страниц, которые находит поисковый сервер по следующим запросам:

собаки	200
кошки	250
лемуры	450
кошки   собаки	450
кошки & лемуры	40
собаки & лемуры	50

Сколько страниц найдёт этот сервер по запросу  
(кошки | собаки) & лемуры?

Здесь часть & лемуры встречается не во всех запросах, поэтому свести задачу к задаче с двумя областями не удаётся. Используя те же обозначения, что и в задаче 2, построим диаграмму с тремя переменными и выделим интересующую область, которая соответствует запросу

(кошки | собаки) & лемуры.

На рисунке 2.42 эта область выделена фоном.

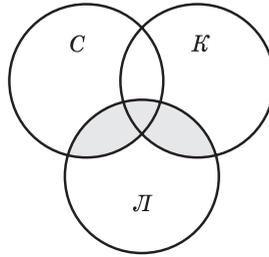


Рис. 2.42

В общем виде задача с тремя областями очень сложна. Попробуем найти какое-нибудь упрощающее условие. Например, выделим три условия:

собаки	200
кошки	250
кошки   собаки	450

Это означает, что область **кошки | собаки** равна сумме областей **кошки** и **собаки**, т. е. эти области *не пересекаются*! Таким образом, в нашем случае диаграмма выглядит так (рис. 2.43).

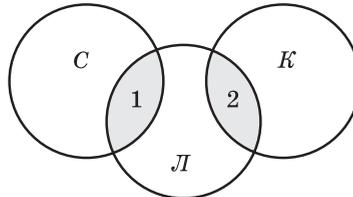


Рис. 2.43

Размеры областей 1 (собаки & лемуры) и 2 (кошки & лемуры) нам известны, они составляют соответственно 40 и 50 страниц, поэтому по запросу

(кошки | собаки) & лемуры

поисковый сервер найдёт  $40 + 50 = 90$  страниц.

Известно количество страниц, которые находит поисковый сервер по следующим запросам:

солнце		230	
крабы		220	
лето		100	
крабы		солнце	450
крабы	&	лето	60
солнце	&	лето	20

Сколько страниц найдёт этот сервер по запросу

крабы | солнце | лето?

## Выводы

- Множество — это набор неповторяющихся элементов.
- Множество может состоять из конечного числа элементов, бесконечного числа элементов или быть пустым. Множества, с которыми работает компьютер, не могут быть бесконечными, потому что его память конечна.
- Чтобы определить множество, можно перечислить все его элементы или задать условие, которое определяет элементы множества. Для всех элементов множества это условие должно быть истинным, для элементов, не входящих во множество, — ложным.
- Дополнение множества  $A$  до универсального множества  $U$ , включающего все элементы некоторого класса, — это все элементы из  $U$ , которые не входят в  $A$ .
- Пересечение двух множеств — это множество, составленное из элементов, входящих в оба исходных множества.
- Объединение двух множеств — это множество, составленное из элементов, которые входят хотя бы в одно из этих множеств.
- Для наглядного изображения множеств используют диаграммы Эйлера–Венна, на которых каждое множество обозначается кругом или другой фигурой.
- На диаграмме Эйлера–Венна дополнение множества  $A$  — это все точки за пределами области  $A$ ; пересечение множеств  $A$  и  $B$  — это общая часть областей  $A$  и  $B$ , а объединение множеств  $A$  и  $B$  — это все точки, входящие в область  $A$  или в область  $B$ .

- Количество элементов в объединении двух множеств вычисляется по формуле включений и исключений:

$$N_{A|B} = N_A + N_B - N_{A\&B},$$

где  $N_A$  и  $N_B$  — число элементов соответственно в множествах  $A$  и  $B$ , а  $N_{A\&B}$  — число их общих элементов.



Нарисуйте в тетради интеллект-карту этого параграфа.

### Вопросы и задания



Выполните по указанию учителя задания в рабочей тетради.



### ЭОР к главе 2 из Единой коллекции цифровых образовательных ресурсов ([school-collection.edu.ru](http://school-collection.edu.ru))

Элементарные логические операции

С чего начинается логика

Беседа 5а. Логические связи

Беседа 5б. Логические формулы

Логические задачи и алгебра высказываний

Схема множеств-1

Схема множеств-2

Схема множеств-3