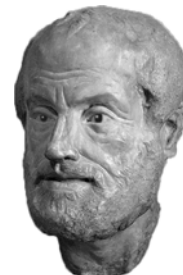


3. Логические основы компьютеров

3.1 Логика и компьютер

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

В естественном языке рассуждения всегда связаны с конкретными предметами и утверждениями, и поэтому исследовать все это многообразие достаточно сложно. Древнегреческий философ Аристотель стал основоположником *формальной логики*, которая отвлекается от конкретного содержания и изучает общие правила построения правильных выводов из известной информации, которая считается истинной. Формальная логика изучает *высказывания*.



Аристотель
(384-322 до н.э.)

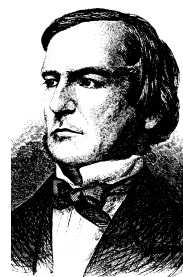
Высказывание — это повествовательное предложение, про которое можно однозначно сказать, что оно истинно или ложно.

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

- 1) *Сейчас идет дождь.*
- 2) *Вчера жирафы улетели на север.*
- 3) *Красиво!*
- 4) *Который час?*
- 5) *В городе N живут более 2 миллионов человек.*
- 6) *Посмотрите на улицу.*
- 7) *У квадрата 10 сторон, и все разные.*
- 8) *История — интересный предмет.*

Здесь высказываниями являются только предложения 1, 2 и 7, остальные не подходят под определение. Утверждения 3 и 4 — это не повествовательные предложения. Предложение 5 станет высказыванием только в том случае, если «N» заменить на название конкретного города. Предложение 6 — это призыв к действию, а не утверждение. Утверждение 8 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.

Какая же связь между логикой и компьютерами? В классической формальной логике высказывание может быть истинно или ложно, третий вариант исключается¹. Если обозначить истинное значение единицей, а ложное — нулем, то получится, что формальная логика представляет собой правила выполнения операций с нулями и единицами, то есть с двоичными кодами. Как вы помните, именно такой способ используется в компьютерах для кодирования всех видов информации. Поэтому обработку информации оказалось возможным свести к выполнению логических операций. Важный шаг в этом направлении сделал английский математик Джордж Буль. Он предложил применить для исследования логических высказываний математические методы. Позже этот раздел математики получил название *алгебра логики* или *булева алгебра*.



Дж. Буль
(1815-1864)

Алгебра логики — это математический аппарат, с помощью которого записывают, вычисляют, упрощают и преобразовывают логические высказывания.

Алгебра логики определяет правила выполнения операций с *логическими* величинами, которые могут быть равны только 0 или 1, то есть с двоичными данными. Используя эти правила, можно

¹ Существуют неклассические логические системы, например, трехзначная логика, где кроме «истинно» и «ложно» есть еще состояние «не определено».

строить элементы памяти и выполнять арифметические действия. О том, как это сделать, вы узнаете в этой главе.



Контрольные вопросы

- Объясните значения слов «логика», «формальная логика», «алгебра логики».
- Чем отличается формальная логика от «обычной», бытовой?
- Что такое высказывание?
- Можно ли считать высказываниями эти предложения:
 - Не плачь, девчонка!
 - Почему я водовоз?
 - Купите слоника!
 - Клубника очень вкусная.
 - Сумма X и Y равна 36.
- Как вы думаете, зачем в курсе информатики изучается логика?

3.2 Логические операции

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

В булевой алгебре 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 идет дождь», а буквой **B** – высказывание «Форточка открыта». Их них строятся сложные высказывания:

не **A**: «Сейчас нет дождя».

не **B**: «Форточка закрыта».

A и **B**: «Сейчас идет дождь и открыта форточка».

A или **B**: «Сейчас идет дождь или открыта форточка».

если **A**, то **B**: «Если сейчас идет дождь, то форточка открыта».

Кроме этих есть еще и другие высказывания, которые можно получить из двух исходных. С некоторыми из них мы также познакомимся.

Операции «НЕ», «И» и «ИЛИ» используются чаще других. Оказывается, с их помощью можно выразить любую логическую операцию, поэтому эти три операции можно считать основными, базовыми.

3.2.1 Операция «НЕ»

Операция «НЕ» часто называется отрицанием или *инверсией*. В алгебре логики всего два знака, 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.

Для обозначения операции «НЕ» используется несколько способов. Выражение «не **A**» в алгебре логики записывается как $\bar{A}$ или $\neg A$, в языках программирования Паскаль и Бейсик – как `not A`, в языке Си – как `!A`.

Операцию «НЕ» можно задать в виде таблицы:

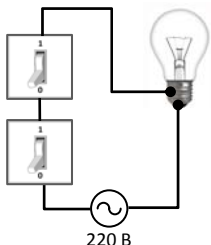
A	$\bar{A}$
0	1
1	0

Эта таблица состоит из двух частей: слева перечисляются все возможные значения исходного сигнала (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 задает **логическую функцию**, то есть правила преобразования входных логических значений в выходные.

3.2.2 Операция «И»

Пусть есть два высказывания, **A** – «Сейчас идет дождь», **B** – «Форточка открыта». Сложное высказывание «**A** и **B**» выглядит так: «Сейчас идет дождь и форточка открыта». Оно будет истинным (верным) 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, соединенных последовательно (см. рисунок). Чтобы лампочка загорелась, нужно обязательно включить оба выключателя. С другой стороны, чтобы выключить лампочку, достаточно выключить любой из них.

Операция «И» (в отличие от «НЕ») выполняется с двумя логическими значениями, которые мы обозначим как **A** и **B**. Результат этой операции в алгебре логики записывают как **A·B**, **A ∧ B** или **A & B**. В языках программирования используют обозначения «**A and B**» (Паскаль, Бейсик) или «**A && B**» (Си).

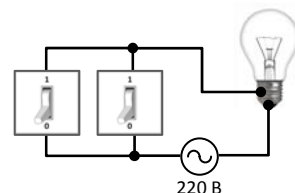
В таблице истинности будет уже не один столбец с исходными данными, а два. Число строк также выросло с 2 до 4, поскольку для 2 бит мы получаем 4 разных комбинации: 00, 01, 10 и 11. Эти строчки расположены в определенном порядке: двоичные числа, полученные соединением битов **A** и **B**, идут в порядке возрастания (слева от таблицы они переведены в десятичную систему). Как следует из определения, в последнем столбце будет всего одна единица, для варианта **A = B = 1**.

	A	B	A·B
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

Легко проверить, что этот результат можно получить «обычным» умножением **A** на **B**, поэтому операцию «И» называют *логическим умножением*. Существует и другое название этой операции – *конъюнкция* (от латинского *conjunctio* – союз, связь).

3.2.3 Операция «ИЛИ»

Высказывание «Сейчас идет дождь или форточка открыта» истинно тогда, когда истинно хотя бы одно из входящих в него высказываний, или оба одновременно. В алгебре логики операция «ИЛИ» обозначается как **A+B** или **A ∨ B**, в языках программирования – «**A or B**» (Паскаль, Бейсик) или «**A | B**» (Си).



Можно представить себе схему с двумя выключателями, соединенными параллельно (см. рисунок). 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**.

	A	B	A+B
	0	0	0
	0	1	1
	1	0	1
	1	1	1

Операцию «ИЛИ» называют *логическим сложением*, потому что она похожа на обычное математическое сложение. Единственное отличие – в последней строке таблицы истинности: в математике 1+1 равно 2, а в алгебре логики – 1. Другое название операции «ИЛИ» – *дизъюнкция* (от латинского *disjunctio* – разделение).

В учебнике для обозначения операций «И» и «ИЛИ» мы будем использовать знаки умножения и сложения (например, **A·B** и **A+B**). Это очень удобно потому, что они привычны для нас и позволяют легко увидеть аналогию с обычной математикой.

Доказано, что операций «НЕ», «И» и «ИЛИ» достаточно для того, чтобы записать с их помощью любую логическую операцию, которую только можно придумать. Например, для двух переменных существует всего $2^4 = 16$ логических операций: их таблицы истинности отличаются только

последним столбцом, в котором 4 двоичных значения (4 бита). Далее мы рассмотрим еще три известных операции и покажем, как их можно представить через операции «НЕ», «И» и «ИЛИ».

3.2.4 Операция «исключающее ИЛИ»

Операция «исключающее ИЛИ» отличается от обычного «ИЛИ» только тем, что результат равен 0, если оба значения равны 1 (последняя строчка в таблице истинности). То есть ее результат – истина в том и только в том случае, когда два значения *не равны*.

«Исключающее ИЛИ» в алгебре логики обозначается знаком $\oplus$, в языке Паскаль как **xor** (например, «**A xor B**»), а в языке Си – знаком $\wedge$ («**A $\wedge$ B**»). Эту операцию можно представить через базовые операции («НЕ», «И», «ИЛИ») следующим образом:

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

Пока мы не можем вывести эту формулу, но можем доказать ее (или опровергнуть – доказать, что она неправильная). Для этого достаточно для всех возможных комбинация **A** и **B** вычислить значения выражения, стоящего в правой части равенства и сравнить его со значением $A \oplus B$ для тех же исходных данных. Поскольку провести такие вычисления в уме достаточно сложно, сначала вычислим значения $\bar{A}$, $\bar{B}$, $\bar{A} \cdot B$ и $A \cdot \bar{B}$, а потом уже $\bar{A} \cdot B + A \cdot \bar{B}$. В таблице истинности появятся дополнительные столбцы для промежуточных результатов:

A	B	$\bar{A}$	$\bar{B}$	$\bar{A} \cdot B$	$A \cdot \bar{B}$	$\bar{A} \cdot B + A \cdot \bar{B}$	$A \oplus B$
0	0	1	1	0	0	0	0
0	1	1	0	1	0	1	1
1	0	0	1	0	1	1	1
1	1	0	0	0	0	0	0

Легко видеть, что выражение $\bar{A} \cdot B + A \cdot \bar{B}$ совпадает с $A \oplus B$ для всех возможных вариантов. Это значит, что формула доказана.

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 ее результат равен остатку от деления «обычной» суммы **A + B** на 2:

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

Здесь **mod** обозначает операцию взятия остатка от деления.

Операция «исключающее ИЛИ» обладает интересными свойствами. По таблице истинности несложно проверить, что

$$A \oplus 0 = A, \quad A \oplus 1 = \bar{A}, \quad A \oplus A = 0.$$

Для доказательства этих равенств можно просто подставить в них **A = 0** и **A = 1**. Теперь докажем, что

$$(A \oplus B) \oplus B = A. \quad (*)$$

Подставляя в левую часть **B = 0**, получим $(A \oplus 0) \oplus 0 = A \oplus 0 = A$. Аналогично для **B = 1** имеем $(A \oplus 1) \oplus 1 = \bar{A} \oplus 1 = A$. Это означает, что формула (*) справедлива для любых значений **B**. Отсюда следует важный вывод: если два раза применить операцию «исключающее ИЛИ» с одним и тем же **B**, мы восстановим исходное значение. В этом смысле «исключающее ИЛИ» – *обратимая* операция (кроме нее обратима также операция «НЕ» – если применить ее дважды, мы вернемся к исходному значению).

Формула (*) верна не только для высказываний, но и для чисел, состоящих из нескольких двоичных разрядов. Чтобы зашифровать данные, надо применить операцию «исключающее ИЛИ» с некоторым числом (кодом) отдельно для каждого разряда. Для расшифровки еще раз применяется «исключающее ИЛИ» с тем же ключом. Нужно отметить, что такой метод шифрования очень нестойкий: для больших текстов его легко раскрыть частотным анализом.

3.2.5 Импликация

Мы часто используем логическую связку «если..., то», например: «Если пойдет дождь, то я надену плащ» или «Если все стороны прямоугольника равны, то это квадрат». В логике эта связка называется *импликацией*² (следованием) и обозначается стрелкой: $A \rightarrow B$ («если A, то B», «из A следует B»).

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

Разобраться с импликацией будет легче, если мы рассмотрим конкретное высказывание, например, такое: «Если хорошо работаешь, то получаешь большую зарплату». Обозначим буквами два простых высказывания: **A** – «хорошо работаешь» и **B** – «получаешь большую зарплату». Понятно, что если высказывание $A \rightarrow B$ истинно, то все, кто хорошо работают ($A = 1$) должны получать большую зарплату ($B = 1$). Если же кто-то работает хорошо ($A = 1$), а получает мало ($B = 0$), то высказывание $A \rightarrow B$ ложно.

Лодыри и бездельники ($A = 0$) могут получать как маленькую ($B = 0$), так и большую зарплату ($B = 1$), это не нарушает справедливость высказывания $A \rightarrow B$. Иногда, определяя импликацию, говорят так: из истины следует истина, а из лжи – что угодно. Это значит, что при ложном высказывании **A** высказывание **B** может быть как ложно, так и истинно.

Нужно обратить внимание на разницу между высказываниями вида «если **A**, то **B**» в обычной жизни и в алгебре логики. В быту мы чаще всего имеем в виду, что существует причинно-следственная связь между **A** и **B**, то есть именно **A** вызывает **B**. Алгебра логики не устанавливает взаимосвязь явлений; истинность высказывания $A \rightarrow B$ говорит только о *возможности* такой связи. Например, с точки зрения алгебры логики может быть истинным высказывание «если Вася – студент, то Петя – лыжник».

Импликация чаще всего используется при решении логических задач. Например, условие «если **A**, то **B**» можно записать в виде $A \rightarrow B = 1$.

Для импликации (в отличие от других изученных операций с двумя переменными) не действует переместительный закон: если в записи $A \rightarrow B$ поменять местами **A** и **B**, то результат изменится: $A \rightarrow B \neq B \rightarrow A$. Внешне это видно по стрелке, которая указывает «направление».

Импликацию можно заменить на выражение, использующее только базовые операции (здесь – только «НЕ» и «ИЛИ»):

$$A \rightarrow B = \overline{A} + B.$$

Доказать это равенство вы уже можете самостоятельно.

3.2.6 Эквивалентность

Эквивалентность (также эквиваленция, равносильность) – это логическая операция, которая соответствует связке «тогда и только тогда». Высказывание $A \leftrightarrow B$ истинно в том и только в том случае, когда $A = B$ (см. таблицу истинности).

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

Возможно, вы заметили, что эквивалентность – это обратная операция для «исключающего ИЛИ» (проверьте по таблицам истинности), то есть

$$A \leftrightarrow B = \overline{A \oplus B}.$$

Здесь черта сверху, охватывающая все выражение в правой части равенства, означает отрицание (инверсию), которое применяется к результату вычисления выражения $A \oplus B$, а не к отдельным высказываниям.

Можно заменить эквивалентность выражением, которое включает только базовые логические операции:

² От латинского *implicatio* – сплетение, тесная связь.

$$A \leftrightarrow B = \overline{A} \cdot \overline{B} + A \cdot B.$$

Эту формулу вы можете доказать (или опровергнуть) самостоятельно.

3.2.7 Другие логические операции

Мы уже говорили, что существуют и другие логические операции. Таблицы истинности операций с двумя переменными содержат 4 строки и отличаются только значением последнего столбца. Поэтому любая новая комбинация нулей и единиц в этом столбце дает новую логическую операцию (логическую функцию). Всего их, очевидно, столько, сколько существует четырехразрядных двоичных чисел, то есть $16 = 2^4$. Из тех, что мы еще не рассматривали, наиболее интересны две – *штрих Шеффера* («И-НЕ», англ. **nand** = «not and»)

$$A \mid B = \overline{A \cdot B}$$

и стрелка Пирса («ИЛИ-НЕ», англ. **nor** = «not or»).

$$A \downarrow B = \overline{A + B}.$$

Особенность этих операций состоит в том, что с помощью любой

одной из них можно записать произвольную логическую операцию. Например, операции «НЕ», «И» и «ИЛИ» (базовый набор) выражаются через штрих Шеффера так:

$$\overline{A} = A \mid A, \quad A \cdot B = \overline{A \mid B} = (A \mid B) \mid (A \mid B), \quad A + B = \overline{\overline{A} \mid \overline{B}} = (A \mid A) \mid (B \mid B).$$

Эти формулы можно доказать через таблицы истинности.

Штрих Шеффера		
A	B	A B
0	0	1
0	1	1
1	0	1
1	1	0

Стрелка Пирса		
A	B	A ↓ B
0	0	1
0	1	0
1	0	0
1	1	0

3.2.8 Логические выражения

Обозначив простые высказывания буквами (переменными) и используя логические операции, можно записать любое высказывание в виде логического выражения. Например, пусть система сигнализации должна дать аварийный сигнал, если вышли из строя два из трех двигателей самолета. 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 формулы

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

Таким образом, мы выполнили *формализацию*.

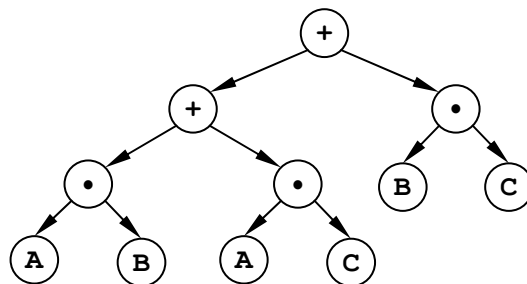
Формализация — это переход от конкретного содержания к формальной записи с помощью некоторого языка.

В логических выражениях операции выполняются в следующем порядке

- 1) действия в скобках;
- 2) отрицание («НЕ»);
- 3) логическое умножение («И»);
- 4) логическое сложение («ИЛИ») и «исключающее ИЛИ»;
- 5) импликация;
- 6) эквивалентность.

Такой порядок означает, что все скобки в выражении (*) для X можно убрать. Порядок вычисления выражения можно, так же, как и для арифметических выражений, определить с помощью дерева (см. рисунок). Вычисление начинается с листьев, корень — это самая последняя операция.

Здесь каждая операция выполняется с двумя значениями. Такие операции называются *бинарными* (лат. *bis* — дважды) или *двуместными*.



Операции, которые выполняются над одной величиной, называют *унарными* (лат. *uno* – один) или *одноместными*. Пример унарной логической операции – это отрицание (операция «НЕ»).

Любую формулу можно задать с помощью *таблицы истинности*, которая показывает, чему равно значение логического выражения при всех возможных комбинациях значений исходных переменных. Сложные выражения удобно разбить на несколько более простых, сначала вычислить значения этих промежуточных величин, а затем – окончательный результат.

Рассмотрим формулу (*). Выражение в правой части зависит от трех переменных, поэтому существует $2^3 = 8$ комбинаций их значений. Таблица истинности выглядит так, как показано на рисунке. По ней видно, что при некоторых значениях переменных значение **X** истинно, а при некоторых – ложно. Такие выражения называют *вычислимыми*.

A	B	C	A · B	A · C	B · C	X
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	0	0	0
1	0	1	0	1	0	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

Высказывание «Вася – школьник, или он не учится в школе» всегда истинно (для любого Васи). Оно может быть записано в виде логического выражения $A + \bar{A}$. 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 $A \cdot \bar{A}$ всегда ложно, оно называется *тождественно ложным* или *противоречием*.

Если два выражения принимают одинаковые значения при всех значениях переменных, они называются *равносильными* или *тождественно равными*. Например, равносильны выражения $A \rightarrow B$ и $\bar{A} + B$. Равносильные выражения определяют одну и ту же логическую функцию, то есть, при одинаковых исходных данных приводят к одинаковым результатам.

3.2.9 Некоторые задачи

Рассмотрим ряд задач, в которых требуется исследовать логическое выражение.

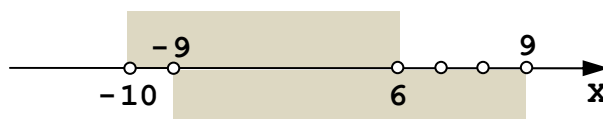
Задача 1. Каково наибольшее целое число **X**, при котором истинно высказывание

$$A = (90 < X^2) \rightarrow (80 > (X + 2)^2)$$

Сначала удобно заменить импликацию по формуле $A \rightarrow B = \bar{A} + B$. Отрицание для высказывания $90 < X^2$ запишется как $90 \geq X^2$, поэтому

$$A = (90 \geq X^2) \text{ или } (80 > (X + 2)^2).$$

В этой задаче нас интересуют только целые числа. Поэтому условие $90 \geq X^2$ можно заменить на $|X| \leq 9$ или $-9 \leq X \leq 9$, а условие $80 > (X + 2)^2$ – на $|X + 2| \leq 8$ или $-10 \leq X \leq 6$. Таким образом, требуется выбрать наибольшее целое число, которое входит в один или в другой промежуток:



Это число – 9.

Задача 2. **A**, **B** и **C** – целые числа, для которых истинно высказывание

$$X = (\bar{A} = B) \cdot ((A > B) \rightarrow (B > C)) \cdot ((B > A) \rightarrow (C > B))$$

Чему равно **B**, если **A** = 27 и **C** = 25?

Это сложное высказывание состоит из трех простых:

$$(\bar{A} = B), (A > B) \rightarrow (B > C), (B > A) \rightarrow (C > B)$$

Они связаны операцией «И», то есть должны выполняться одновременно.

Из $(A = B) = 1$ сразу следует, что $A \neq B$. Предположим, что $A > B$, тогда из второго условия получаем $1 \rightarrow (B > C)$ это выражение может быть истинно тогда и только тогда, когда $(B > C) = 1$, поэтому имеем $A > B > C$ этому условию соответствует только число 26.

На всякий случай проверим и вариант $A < B$, тогда из второго условия получаем $0 \rightarrow (B > C)$, это выражение истинно при любом B . Теперь проверяем третье условие: получаем $1 \rightarrow (C > B) = 1$; это выражение может быть истинно тогда и только тогда, когда $C > B$, и тут мы получили противоречие, потому что нет такого числа B , для которого $C > B > A$. Таким образом, правильный ответ – 26.



Контрольные вопросы

1. Даны два высказывания: A – «В Африке водятся жирафы» и B – «В Мурманске идет снег». Постройте из них различные сложные высказывания.
2. Дано высказывание «Винни-Пух любит мед, и дверь в дом открыта». Как бы вы сформулировали отрицание этого высказывания?
3. Что такое таблица истинности?
4. Почему в таблице истинности для операции «НЕ» две строки, а для других изученных операций – четыре? Сколько строчек в таблице истинности выражения с тремя переменными? с четырьмя? с пятью?
5. В каком порядке обычно записываются значения переменных в таблице истинности?
6. Когда истинно высказывание « A и B »? « A или B »?
7. Какие электрические схемы можно использовать для иллюстрации операций «И» и «ИЛИ»?
8. Какие знаки применяют для обозначения операций «НЕ», «И», «ИЛИ»?
9. Почему операция «И» называется логическим умножением, а «ИЛИ» – логическим сложением?
10. В чем отличие «обычного» и логического сложения?
11. Сколько существует различных логических операций с двумя переменными? С тремя переменными?
12. Чем отличается операция «исключающее ИЛИ» от «ИЛИ»?
13. Почему операция «исключающее ИЛИ» называется сложением по модулю 2?
14. Как записать выражение $A \oplus B$ с помощью базового набора операций («НЕ», «И», «ИЛИ»)?
15. Как можно доказать или опровергнуть логическую формулу?
16. Какими интересными свойствами обладает операция «исключающее ИЛИ»?
17. Что значит выражение «обратимая операция»? Какие изученные логические операции являются обратимыми?
18. Какое свойство операции «исключающее ИЛИ» позволяет использовать ее для простейшего шифрования?
19. Чем отличается смысл высказывания «если A , то B » в обычной речи и в математической логике?
20. Запишите в виде формулы высказывание «если утюг горячий, то лоб холодный».
21. Запишите в виде формулы высказывание «неверно, что если утюг горячий, то лоб холодный». Можно ли в этом случае сразу сказать, какой утюг и какой лоб?
22. Как выразить импликацию через операции «НЕ» и «ИЛИ»? Докажите эту формулу.
23. Как выразить эквивалентность через операции «НЕ», «И» и «ИЛИ»? Докажите эту формулу.
24. Чем интересны операции «штрих Шеффера» и «стрелка Пирса»?
25. Докажите формулы, позволяющие представить базовые логические операции через штрих Шеффера. Попробуйте построить и доказать аналогичные формулы для операции «стрелка Пирса».
26. Что такое формализация?
27. В каком порядке выполняются действия в логических выражениях?
28. Что можно сделать для того, чтобы изменить «естественный» порядок действия?

29. Какие операции называются бинарными и унарными? Приведите примеры унарных и бинарных операций в математике.
30. Поясните разницу между терминами «логическое выражение» и «логическая функция».
31. Можно ли сказать, что таблица истинности однозначно определяет
- логическое выражение;
 - логическую функцию.
32. Что такое вычисляемое логическое выражение?
33. Что тавтология? противоречие? Приведите примеры.
34. Что такое равносильные выражения?



Задачи

1. Составьте деревья для вычисления логических выражений и их таблицы истинности:

- | | |
|---|--|
| а) $\overline{A \cdot B} + A \cdot B$ | ж) $\overline{A \cdot C} + \overline{B \cdot C}$ |
| б) $A \cdot B + \overline{A} \cdot \overline{B} + A \cdot \overline{B}$ | з) $\overline{(A + C)} + \overline{(B + C)}$ |
| в) $(A + B) \cdot (\overline{A} + \overline{B}) \cdot (A + \overline{B})$ | и) $\overline{(\overline{A \cdot C}) \cdot (\overline{B \cdot C})}$ |
| г) $A \cdot \overline{B} + B \cdot \overline{C} + C \cdot \overline{A}$ | к) $A \cdot (C + B \cdot \overline{C}) + C \cdot (\overline{A + B})$ |
| д) $A \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} + B \cdot C$ | л) $A \cdot (C + \overline{(B + C)}) + B \cdot (\overline{A \cdot C})$ |
| е) $A \cdot (\overline{B \cdot C} + \overline{A}) \cdot (\overline{C} + B)$ | |

2. Составьте деревья для вычисления логических выражений и их таблицы истинности:

- | | |
|--|--|
| а) $(A \rightarrow B) + (\overline{A} \rightarrow \overline{B})$ | е) $(\overline{A} \rightarrow \overline{B}) \rightarrow (A \rightarrow \overline{C})$ |
| б) $(\overline{A} \rightarrow B) \cdot (A \rightarrow \overline{B})$ | ж) $A \cdot B \rightarrow (B + \overline{C})$ |
| в) $(A \cdot B) \rightarrow (\overline{A} + \overline{B})$ | з) $(\overline{A} \rightarrow B) \rightarrow \overline{(\overline{A} \rightarrow C)}$ |
| г) $(A + \overline{B}) \rightarrow (A \cdot B)$ | и) $(A \leftrightarrow B) + (\overline{A} \leftrightarrow B)$ |
| д) $(A \rightarrow \overline{B}) \cdot (A + C) \cdot (\overline{A} \rightarrow C)$ | к) $(A \leftrightarrow \overline{B}) + (A \leftrightarrow C) + (\overline{B} \leftrightarrow C)$ |

3. Символом **F** обозначено одно из указанных ниже логических выражений от трех аргументов: **X**, **Y**, **Z**. Дан фрагмент таблицы истинности выражения **F**. Какие из этих выражений могут соответствовать **F**? (Ответ: а, б)

X	Y	Z	F
1	1	1	1
1	1	0	1
1	0	0	1

- м) $X + \overline{Y} + \overline{Z}$
 н) $X + Y + Z$
 о) $\overline{X} + \overline{Y} + \overline{Z}$
 п) $\overline{X} + \overline{Y} + Z$

4. Для предыдущего задания определите, сколько различных логических функций соответствует заданной частичной таблице истинности? (Ответ: 32)
5. Задано 5 строк таблицы истинности некоторого логического выражения с тремя переменными. Сколько различных логических функций ей соответствуют? (Ответ: 8)
6. Символом **F** обозначено одно из указанных ниже логических выражений от трех аргументов: **X**, **Y**, **Z**. Дан фрагмент таблицы истинности выражения **F**. Какие из этих выражений могут соответствовать **F**? (Ответ: б, г)

X	Y	Z	F
0	1	0	0
1	1	0	1
0	1	1	0

- а) $\overline{X} + Y + \overline{Z}$
 б) $X \cdot Y \cdot \overline{Z}$
 в) $\overline{X} \cdot \overline{Y} \cdot Z$
 г) $X + \overline{Y} \cdot Z$

7. Символом **F** обозначено одно из указанных ниже логических выражений от трех аргументов: **X**, **Y**, **Z**. Дан фрагмент таблицы истинности выражения **F**. Какие из этих выражений могут соответствовать **F**? (Ответ: а, в)

X	Y	Z	F
1	0	0	1
0	0	0	1

- а) $X \rightarrow (\overline{Y} + \overline{Z})$
 б) $\overline{X} \cdot \overline{Y} \cdot \overline{Z}$
 в) $\overline{X} + \overline{Y} + \overline{Z}$

1	1	1	0
---	---	---	---

г) $X + Y + Z$

8. Символом **F** обозначено одно из указанных ниже логических выражений от трех аргументов: **X**, **Y**, **Z**. Дан фрагмент таблицы истинности выражения **F**. Какие из этих выражений могут соответствовать **F**? (Ответ: а)

X	Y	Z	F
1	0	0	1
0	0	0	0
1	1	1	0

- а) $X \cdot \bar{Y} \cdot \bar{Z}$
 б) $X \rightarrow (\bar{Y} + \bar{Z})$
 в) $X + Y + Z$
 г) $Y \rightarrow (X \cdot Z)$

9. Символом **F** обозначено одно из указанных ниже логических выражений от трех аргументов: **X**, **Y**, **Z**. Дан фрагмент таблицы истинности выражения **F**. Какие из этих выражений могут соответствовать **F**? (Ответ: б, г)

X	Y	Z	F
0	0	0	0
0	1	1	1
1	0	0	1

- а) $(X + \bar{Y}) \rightarrow Z$
 б) $(\bar{X} + Y) \rightarrow Z$
 в) $X + (Y \rightarrow Z)$
 г) $X + Y \cdot Z$

10. Определите значение логического выражения $(X > 2) \rightarrow (X > 3)$ для $X = 1, 2, 3, 4$. (Ответ: 1, 1, 0, 1)

11. Определите значение логического выражения

$$((X < 5) \rightarrow (X < 3)) \cdot ((X < 2) \rightarrow (X < 1)) \text{ для } X = 1, 2, 3, 4.$$

(Ответ: 0, 1, 0, 0)

12. Определите значение логического выражения

$$((X > 3) + (X < 3)) \rightarrow (X < 1) \text{ для } X = 1, 2, 3, 4.$$

(Ответ: 0, 0, 1, 0)

13. Определите значение логического выражения

$$((X < 4) \rightarrow (X < 3)) \cdot ((X < 3) \rightarrow (X < 1)) \text{ для } X = 1, 2, 3, 4.$$

(Ответ: 0, 0, 0, 1)

14. Определите значение логического выражения

$$(X \cdot (X - 8) > 2 \cdot X - 25) \rightarrow (X > 7) \text{ для } X = 4, 5, 6, 7.$$

(Ответ: 0, 1, 0, 0)

15. Найдите все целые значения X , при которых логическое выражение

$$(X > 2) \rightarrow (X > 5) \text{ ложно.}$$

(Ответ: 3, 4, 5)

16. Найдите все целые значения X , при которых логическое выражение

$$((X > 0) + (X > 4)) \rightarrow (X > 4) \text{ ложно.}$$

(Ответ: 1, 2, 3, 4)

17. Автопилот может работать, если исправен главный бортовой компьютер или два вспомогательных. Выполните формализацию и запишите логические формулы для высказываний «автопилот работоспособен» и «автопилот неработоспособен».

18. Каково наибольшее целое положительное число X , при котором истинно высказывание:

$$(X(X + 3) > X^2 + 9) \rightarrow (X(X + 2) \leq X^2 + 11) ?$$

(Ответ: 5)

19. Каково наибольшее целое положительное число X , при котором истинно высказывание:

$$(121 < X^2) \rightarrow (X < X + 5) ?$$

(Ответ: 11)

20. Каково наибольшее целое положительное число X , при котором ложно высказывание:

$$(X(X + 6) + 9 > 0) \rightarrow (X^2 > 45) ?$$

(Ответ: 6)

21. Каково наибольшее целое положительное число X , при котором истинно высказывание:

$$(X^2 - 1 > 100) \rightarrow (X(X - 1) < 100) ?$$

(Ответ: 10)

22. Каково наибольшее целое положительное число X , при котором ложно высказывание:

$$(7x - 3 < 75) \rightarrow (x(x - 1) > 65)?$$

(Ответ: 8)

23. Известно, что для чисел А, В и С истинно высказывание

$$((C < A) + (C < B)) \cdot ((C + 1) < A) \cdot ((C + 1) < B).$$

а) Чему равно С, если $A = 25$ и $B = 48$?б) Чему равно С, если $A = 45$ и $B = 18$?

(Ответ: 47, 44)

24. Известно, что для чисел А, В и С истинно высказывание

$$(A = B) \cdot ((B < A) \rightarrow (2C > A)) \cdot ((A < B) \rightarrow (A > 2C)).$$

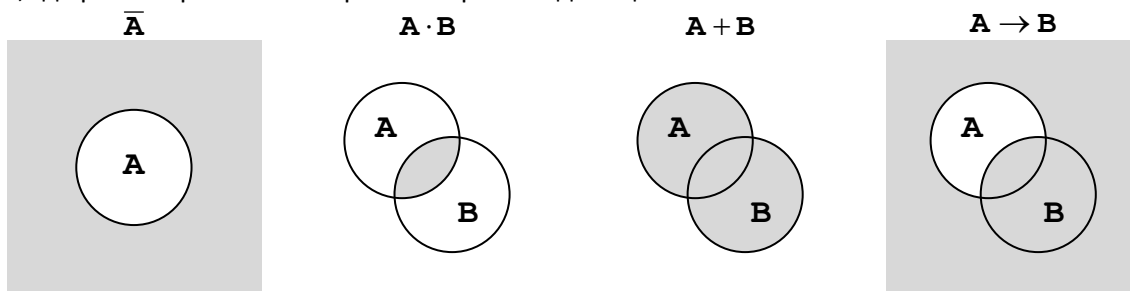
Чему равно А, если $C = 10$ и $B = 22$?

(Ответ: 21)

3.3 Диаграммы

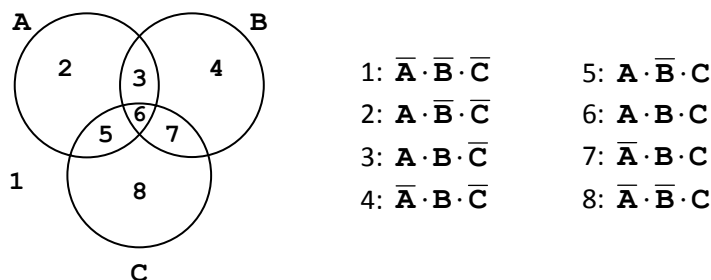
Выражения, зависящие от небольшого количества переменных (обычно не более 4-х), удобно изображать в виде диаграмм, которые называют *диаграммами Венна* или *кругами Эйлера*.

На такой диаграмме каждой переменной соответствует круг, внутри которого она равна единице, а вне его – нулю. Круги пересекаются, каждый с каждым. Области, в которых рассматриваемое логическое выражение истинно, закрашиваются каким-либо цветом. Ниже приведены диаграммы для простейших операций с одной и двумя переменными. Серым цветом залиты области, где рассматриваемое выражение равно единице.



Такие диаграммы часто используются при работе с множествами: операция «И» соответствует пересечению двух множеств, а «ИЛИ» – объединению.

Для трех переменных диаграмма будет немного сложнее. Для каждой из областей показанной ниже диаграммы запишем логические выражения:



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 имеет вид

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

С другой стороны, можно заметить, что справедлива формула

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

Это означает, что логические выражения в некоторых случаях можно упростить. Как это делается, вы узнаете в следующем параграфе.

Диаграммы удобно применять для решения задач, в которых используются множества, например, множества ссылок, полученных от поисковой системы в ответ на какой-то запрос. Рассмотрим такую задачу:

Известно количество ссылок, которые находит поисковый сервер по следующим запросам (здесь символ «&» обозначает операцию «И», а «|» – операцию «ИЛИ»):

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

Сколько ссылок найдет этот сервер по запросу
(кошки | собаки) & лемуры?

Обозначим буквами С, К и Л высказывания «ключевое слово – собаки», «ключевое слово – кошки» и «ключевое слово – лемуры». Построим диаграмму с тремя переменными и выделим интересующую область, которая соответствует запросу

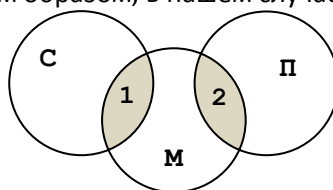
(кошки | собаки) & лемуры

На рисунке эта область закрашена серым цветом.

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

собаки	200
кошки	250
кошки собаки	450

Это означает, что область «кошки ИЛИ собаки» равна сумме областей «кошки» и «собаки», то есть эти области *не пересекаются*! Таким образом, в нашем случае диаграмма выглядит так:



Области 1 (собаки & лемуры) и 2 (кошки & лемуры) нам известны, они составляют соответственно 40 и 50 ссылок, поэтому по запросу (кошки | собаки) & лемуры поисковый сервер выдаст $40 + 50 = 90$ ссылок.



Задачи

- Используя диаграмму с тремя переменными, запишите логические выражения для объединения областей $2 + 5$, $3 + 6$, $4 + 7$, $6 + 7$, $5 + 6$, $5 + 8$, $7 + 8$. Для каждой сложной области найдите два эквивалентных выражения.
- Известно количество ссылок, которые находит поисковый сервер по следующим запросам:

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

Сколько ссылок найдет этот сервер по запросу

кошки | собаки | лемуры?

(Ответ: 810)

- Известно количество ссылок, которые находит поисковый сервер по следующим запросам:

собаки	250
кошки	200
лемуры	500
собаки & лемуры	0

собаки & кошки 20

кошки & лемуры 10

Сколько ссылок найдет этот сервер по запросу

кошки | собаки | лемуры?

(Ответ: 920)

4. Известно количество ссылок, которые находит поисковый сервер по следующим запросам:

собаки 120

кошки 270

лемуры 100

кошки | собаки 390

кошки & лемуры 20

собаки & лемуры 10

Сколько ссылок найдет этот сервер по запросу

кошки | собаки | лемуры?

(Ответ: 460)

3.4 Упрощение логических выражений

3.4.1 Законы алгебры логики

Для упрощения логических выражений используют законы алгебры логики. Они формулируются для базовых логических операций – «НЕ», «И» и «ИЛИ».

Закон	для «И»	для «ИЛИ»
двойного отрицания	$\overline{\overline{A}} = A$	
исключения третьего	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
операции с константами	$A \cdot 1 = A, A \cdot 0 = 0$	$A + 1 = 1, A + 0 = A$
повторения	$A \cdot A = A$	$A + A = A$
переместительный	$A \cdot B = B \cdot A$	$A + B = B + A$
сочетательный	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
распределительный	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
поглощения	$A + A \cdot B = A$	$A \cdot (A + B) = A$
законы де Моргана	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

Закон двойного отрицания означает, что операция «НЕ» обратима: если применить ее два раза, логическое значение не изменится. Закон исключения третьего основан на том, что любое логическое выражение либо истинно, либо ложно («третьего не дано»). Поэтому если $A = 1$, то $\overline{A} = 0$ (и наоборот), так что произведение этих величин всегда равно нулю, а сумма – единице.

Операции с константами и закон повторения легко проверяются по таблицам истинности операций «И» и «ИЛИ». Переместительный и сочетательный законы выглядят вполне привычно, так же, как и в математике. Почти везде «работает» аналогия с обычной алгеброй, нужно только помнить, что в логике $1 + 1 = 1$, а не 2.

Распределительный закон для «ИЛИ» – это обычное раскрытие скобок. А вот для операции «И» мы видим незнакомое выражение, в математике это равенство неверно. Доказательство можно начать с правой части, раскрыв скобки:

$$(A + B) \cdot (A + C) = A \cdot A + A \cdot C + B \cdot A + B \cdot C$$

Дальше используем закон повторения ($A \cdot A = A$) и заметим, что

$$A + A \cdot C = A \cdot (1 + C) = A \cdot 1 = A.$$

Аналогично доказываем, что $A + B \cdot A = A \cdot (1 + B) = A$, таким образом,

$$(A + B) \cdot (A + C) = A + B \cdot C.$$

Равенство доказано. Попутно мы доказали также и закон поглощения для операции «И» (для «ИЛИ» вы можете сделать это самостоятельно). Отметим, что из распределительного закона следует полезная формула

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



А. де Морган
(1806-1871)

Правила, позволяющие раскрывать отрицание сложных выражений, названы в честь шотландского математика и логика де Моргана. Обратите внимание, что при этом не просто «общее» отрицание переходит на отдельные выражения, но и операция «И» заменяется на «ИЛИ» (и наоборот). Доказать законы де Моргана можно с помощью таблиц истинности.

Теперь с помощью приведенных законов алгебры логики упростим полученное ранее логическое выражение для объединения областей 3 и 4 на диаграмме с тремя переменными:

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

Здесь мы сначала вынесли общий множитель двух слагаемых за скобки, а затем применили закон исключения третьего.

В общем случае можно рекомендовать такую последовательность действий:

1. Заменить все «небазовые» операции (исключающее ИЛИ, импликацию, эквивалентность и др.) на их выражения через базовые операции «НЕ», «И» и «ИЛИ».
2. Раскрыть отрицания сложных выражений по законам де Моргана так, чтобы операции отрицания остались только у отдельных переменных.
3. Используя вынесение общих множителей за скобки, раскрытие скобок и другие законы алгебры логики, упростить выражение.

Пример.

$$\begin{aligned} \overline{(A + B) \cdot (A + B) \cdot (\bar{A} + C)} &= (A + B) \cdot \bar{A} \cdot \bar{B} \cdot (\bar{A} + C) = (A \cdot \bar{A} + B \cdot \bar{A}) \cdot \bar{B} \cdot (\bar{A} + C) = \\ &= \bar{B} \cdot \bar{A} \cdot \bar{B} \cdot (\bar{A} + C) = \bar{A} \cdot \bar{B} \cdot \bar{B} \cdot (\bar{A} + C) = \bar{A} \cdot \bar{B} \cdot (\bar{A} + C) = \bar{B} \cdot \bar{A} \cdot (\bar{A} + C) = \bar{B} \cdot \bar{A}. \end{aligned}$$

Здесь последовательно использованы закон де Моргана, распределительный закон, закон исключения третьего, переместительный закон, закон повторения, снова переместительный закон и закон поглощения.

3.4.2 Логические уравнения

Если приравнять два логических выражения, мы получим уравнение. Его решением будут значения переменных, при которых уравнение превращается в тождество, то есть значения левой и правой частей совпадают. Например, уравнение $\mathbf{A \cdot B = 1}$ имеет единственное решение: $\mathbf{A = B = 1}$, для остальных комбинаций значений переменных левая часть равна нулю. В то же время уравнение $\mathbf{A + B = 1}$ имеет три решения: $(\mathbf{A = 0, B = 1})$, $(\mathbf{A = 1, B = 0})$ и $\mathbf{A = B = 1}$.

Пример 1. Требуется найти все решения уравнения

$$\overline{(B + C) \cdot A} \rightarrow (\bar{A} \cdot \bar{C} + D) = 0.$$

Вспоминаем, что импликация равна нулю только тогда, когда первое выражение равно 1, а второе – 0. Поэтому исходное уравнение сразу разбивается на два

$$\overline{(B + C) \cdot A} = 1, \quad \bar{A} \cdot \bar{C} + D = 0.$$

Первое уравнение с помощью закона де Моргана можно преобразовать к виду $\mathbf{\bar{B} \cdot \bar{C} \cdot A = 1}$, откуда сразу следует, что все три сомножителя должны быть равны 1. Это значит, что $\mathbf{A = 1, B = 0}$ и $\mathbf{C = 0}$. Кроме того, из второго уравнения следует, что $\mathbf{D = 0}$. Решение найдено, причем оно единственное.

Возможен другой вариант – упростить выражение. Заменяя импликацию по формуле $\mathbf{A \rightarrow B = \bar{A} + B}$, получаем

$$\overline{(B + C) \cdot A} + \bar{A} \cdot \bar{C} + D = 0.$$

Используем закон де Моргана:

$$\mathbf{B + C + \bar{A} + \bar{A} \cdot \bar{C} + D = 0}$$

и закон поглощения

$$B + C + \bar{A} + D = 0.$$

Для того, чтобы логическая сумма была равна нулю, каждое слагаемое должно быть равно нулю, поэтому $A = 1, B = C = D = 0$.

Есть и третий вариант – построить таблицу истинности выражения в левой части и найти все варианты, при которых оно равно 0. Однако таблица истинности выражения с четырьмя переменными содержит $2^4 = 16$ строк, поэтому такой подход достаточно трудоемок.

Пример 2. Требуется найти все решения уравнения

$$(A + \bar{B}) \rightarrow (B \cdot C \cdot D) = 1.$$

Преобразуем выражение, раскрыв импликацию через «НЕ» и «ИЛИ» и применив закон де Моргана:

$$\overline{A + \bar{B}} + B \cdot C \cdot D = \bar{A} \cdot B + B \cdot C \cdot D = 1.$$

Если логическая сумма равна 1, то хотя бы одно слагаемое равно 1 (или оба одновременно).

Равенство $\bar{A} \cdot B = 1$ верно при $A = 0, B = 1$ и любых C и D . Поскольку есть всего 4 комбинации значений C и D , уравнение $\bar{A} \cdot B = 1$ имеет 4 решения:

A	B	C	D
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1

Второе уравнение, $B \cdot C \cdot D = 1$, дает $B = C = D = 1$ при любом A , то есть оно имеет два решения:

A	B	C	D
0	1	1	1
1	1	1	1

повторяется

Видим, что первое из этих решений уже было получено раньше, поэтому уравнение имеет всего пять разных решений. Заметим, что определить все повторяющиеся решения можно из уравнения $(\bar{A} \cdot B) \cdot (B \cdot C \cdot D) = 1$, которое имеет единственное решение $A = 0, B = C = D = 1$.

Пример 3. Требуется найти число решений уравнения

$$A \cdot B \cdot C + \bar{B} \cdot \bar{C} \cdot D = 0.$$

Здесь, в отличие от предыдущих задач, не нужно находить сами решения, интересует только их количество. Уравнение распадается на два

$$A \cdot B \cdot C = 0 \text{ и } \bar{B} \cdot \bar{C} \cdot D = 0.$$

Каждое из них имеет достаточно много решений. Можно поступить следующим образом: сначала найти количество решений «обратного» уравнения, с единицей в правой части:

$$A \cdot B \cdot C + \bar{B} \cdot \bar{C} \cdot D = 1,$$

и затем вычесть его из 16 (общего количества комбинаций четырех переменных). Уравнение $A \cdot B \cdot C = 1$ имеет два решения: $A = B = C = 1$ и любое D (0 или 1). Второе уравнение, $\bar{B} \cdot \bar{C} \cdot D = 1$, тоже имеет два решения: A – любое, $B = C = 0, D = 1$. Среди этих четырех решений нет повторяющихся, поэтому исходное уравнение имеет $16 - 4 = 12$ решений.

Обратите внимание, что число решений логических уравнений, в отличие от «обычных уравнений», всегда конечно. Это связано с тем, что каждая переменная может принимать только два значения (0 и 1), и число разных комбинаций значений переменных конечно, оно равно 2^n , где n – это количество переменных. Поэтому уравнение с n переменными имеет не более 2^n решений.



Задачи

1. Упростите логические выражения:

а) $A \cdot B \cdot \bar{A} \cdot B + B$

г) $A + \bar{A} \cdot B + \bar{A} \cdot C$

ж) $(\bar{A} + B) \cdot \bar{C} \cdot (C + A \cdot \bar{B})$

б) $(A + B) \cdot (\bar{A} + \bar{B})$

д) $A \cdot (A + B + C)$

з) $\bar{A} \cdot \bar{C} + A \cdot B + \bar{A} \cdot C + A \cdot \bar{B}$

$$\text{в) } \mathbf{A + A \cdot B + A \cdot C} \quad \text{е) } \mathbf{A \cdot B + \overline{B} + \overline{A} \cdot B} \quad \text{и) } \mathbf{A \cdot (\overline{B} \cdot \overline{C} + B \cdot C) + A \cdot (B \cdot \overline{C} + \overline{B} \cdot C)}$$

(Ответ: а – $\overline{B}$, б – $\mathbf{A \cdot \overline{B} + B \cdot \overline{A}}$, в – $\mathbf{A}$, г – $\mathbf{A + B + C}$, д – $\mathbf{A}$, е – 1, ж – 0, з – 1, и – $\mathbf{A}$)

2. Упростите логические выражения:

$$\text{а) } \mathbf{A \cdot (\overline{B} + C)} \quad \text{г) } \mathbf{(\overline{A} + \overline{B} + \overline{C})} \quad \text{ж) } \mathbf{(A + B + C) \cdot (\overline{A} \cdot \overline{B}) + C}$$

$$\text{б) } \mathbf{(\overline{A} + \overline{B}) + (\overline{A} + B) + A \cdot B} \quad \text{д) } \mathbf{(\overline{A} + B) \cdot A \cdot \overline{B}} \quad \text{з) } \mathbf{A \cdot (\overline{C} + \overline{B}) + (\overline{A} + B) \cdot C + A \cdot C}$$

$$\text{в) } \mathbf{A + (\overline{A} + B) + \overline{A} \cdot B} \quad \text{е) } \mathbf{A + B \cdot \overline{C} + (\overline{A} + B + \overline{C})} \quad \text{и) } \mathbf{(A + B) \cdot (\overline{A} + B) \cdot (\overline{A} + \overline{B})}$$

(Ответ: а – $\mathbf{A \cdot B \cdot \overline{C}}$, б – $\mathbf{\overline{A} + B}$, в – 1, г – $\mathbf{\overline{A} \cdot B \cdot C}$, д – 0, е – $\mathbf{A + \overline{B} + C}$, ж – $\mathbf{A + B + C}$, з – $\mathbf{A \cdot C}$, и – $\mathbf{\overline{A} \cdot B}$)

3. Упростите логические выражения:

$$\text{а) } \mathbf{(\overline{A} \rightarrow C) \cdot C} \quad \text{в) } \mathbf{A + (\overline{A} \rightarrow B) + (\overline{A} + \overline{B})} \quad \text{д) } \mathbf{(\overline{A} \rightarrow B) \cdot (\overline{A} \rightarrow B)}$$

$$\text{б) } \mathbf{(\overline{A} \rightarrow \overline{B}) + (\overline{A} \rightarrow B) + A \cdot B} \quad \text{г) } \mathbf{(\overline{A} \rightarrow (B \rightarrow \overline{C}))} \quad \text{е) } \mathbf{A + B \cdot \overline{C} + (\overline{A} \rightarrow \overline{B} \cdot C)}$$

(Ответ: а – 0, б – $\mathbf{\overline{A} + B}$, в – 1, г – $\mathbf{\overline{A} \cdot B \cdot C}$, д – 0, е – $\mathbf{A + \overline{B} + C}$)

4. Решите уравнения

$$\text{а) } \mathbf{A + \overline{B} + (B \rightarrow (C + D)) = 0}$$

$$\text{б) } \mathbf{(A \rightarrow C) + B \cdot A + \overline{D} = 0}$$

$$\text{в) } \mathbf{(\overline{A} + C) \rightarrow (\overline{B} + C + D) = 0}$$

$$\text{г) } \mathbf{(A \rightarrow \overline{C}) + \overline{B} \cdot C \cdot A + D = 0}$$

$$\text{д) } \mathbf{((B + C) \cdot A) \rightarrow ((A + C) + D) = 0}$$

$$\text{е) } \mathbf{(A \rightarrow C) \cdot (A \rightarrow \overline{C}) \cdot (\overline{A} \rightarrow (C \cdot \overline{B} \cdot D)) = 1}$$

(Ответ: а – 0100, б – 1001, в – 0100, г – 1110, д – 1100, е – 0011)

5. Сколько различных решений имеют уравнения

$$\text{а) } \mathbf{A \cdot B + C \cdot D = 1}$$

$$\text{б) } \mathbf{(A + B) \cdot (C + D) = 1}$$

$$\text{в) } \mathbf{(A + B) \rightarrow (B \cdot C \cdot D) = 0}$$

$$\text{г) } \mathbf{A \cdot \overline{B} \cdot C \cdot \overline{D} \cdot (\overline{E} + \overline{E}) = 0}$$

$$\text{д) } \mathbf{(A + B + C) \cdot \overline{B} \cdot \overline{C} \cdot D = 1}$$

$$\text{е) } \mathbf{(A \cdot B \cdot C) \rightarrow (\overline{C} \cdot D) = 1}$$

$$\text{ж) } \mathbf{(A \rightarrow B) \cdot C + \overline{C} \cdot D \cdot C = 1}$$

$$\text{з) } \mathbf{(\overline{A} + \overline{B} + \overline{C}) \cdot (B + \overline{C} + \overline{D}) = 0}$$

(Ответ: а – 3, б – 7, в – 10, г – 30, д – 1, е – 14, ж – 6, з – 4)

3.5 Синтез логических выражений

До этого момента мы считали, что логическое выражение уже задано, и нам надо что-то с ним сделать (построить таблицу истинности, упростить и т.п.). Такие задачи называются задачами *анализа* (от греческого *αναλυσις* – разложение) – мы исследуем имеющееся выражение. При проектировании различных логических устройств, в том числе и узлов компьютеров, приходится решать обратную задачу – строить логическое выражение по готовой таблице истинности, которая описывает нужное правило обработки данных. Эта задача называется задачей *синтеза* (от греческого *συνθεσις* – совмещение).

В качестве простейшего примера построим логическое выражение для операции импликации $\mathbf{X = A \rightarrow B}$.

Способ 1. В таблице истинности мы выделяем все строки, где логическое выражение равно единице. Тогда выражение может быть записано как логическая сумма выражений, каждое из которых истинно только в одном случае.

A	B	X
0	0	1
0	1	1
1	0	0
1	1	1

$$\bullet \overline{A} \cdot \overline{B}$$

$$\bullet \overline{A} \cdot B$$

$$\bullet A \cdot B$$

Например, выражение $\overline{A} \cdot \overline{B}$ истинно только при $\mathbf{A = 0}$ и $\mathbf{B = 0}$, то есть только в первой строке таблицы. Выражение $\overline{A} \cdot B$ истинно только во второй строке, а $\mathbf{A \cdot B}$ – только в последней. Су-

существует простое правило: если в этой строке переменная равна нулю, она входит в произведение с отрицанием, а если равна 1, то без отрицания.

Складывая выражения для всех отмеченных строк (кроме третьей, где функция равна нулю), получаем $X = \bar{A} \cdot \bar{B} + \bar{A} \cdot B + A \cdot B$. Упрощаем это выражение:

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

Таким образом, мы вывели формулу, которая позволяет заменить импликацию через операции «НЕ» и «ИЛИ».

Способ 2. Если в таблице истинности нулей меньше, чем единиц, удобнее сначала найти формулу для обратного выражения, $\bar{X}$, а потом применить операцию «НЕ». В данном случае выражение равно нулю в единственной строчке, при $A = 1$ и $B = 0$, то есть $\bar{X} = A \cdot \bar{B}$. Теперь остается применить операцию «НЕ» и закон де Моргана:

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

Рассмотрим более сложный пример, когда выражение зависит от трех переменных. В этом случае в таблице истинности будет 8 строк.

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

Отметим все строки, где $X = 1$, и для каждой из них построим выражение, истинное только для этой комбинации переменных (см. таблицу). Теперь выполним логическое сложение:

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

Упрощение этого выражения дает

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

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

Используя второй способ, получаем

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

Тогда $\bar{X} = \overline{A \cdot \bar{C}} = \bar{A} + C$. В данном случае второй способ оказался проще, потому что в таблице истинности меньше нулей, чем единиц.

Способ 3. При небольшом количестве нулей можно использовать еще один метод. Попробуем применить операцию «НЕ» к исходному выражению для $\bar{X}$, без предварительного упрощения:

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

Применяя закон де Моргана, получим

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

Используя закон де Моргана еще два раза (для обеих скобок), находим

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

Заметим, что выражение в каждой скобке *ложно* только для одной комбинации исходных данных, при которых $X = 0$. Таким образом, для каждой строчки в таблице истинности, где выражение равно 0, нужно построить логическую сумму, в которую переменные, равные в этой строчке единице, входят с инверсией, а равные нулю – без инверсии. Выражение для X – это произведение полученных сумм.

В нашем примере выражение упрощается с помощью распределительного закона для «И» и закона исключения третьего:

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

Неудивительно, что мы получили тот же ответ, что и раньше.



Задачи

1. Постройте выражения для логических функций, заданных таблицами истинности. Используйте разные методы и сравните их.

а)

A	B	X
0	0	0
0	1	1
1	0	0
1	1	1

б)

A	B	X
0	0	1
0	1	0
1	0	1
1	1	1

в)

A	B	X
0	0	0
0	1	1
1	0	0
1	1	0

г)

A	B	X
0	0	0
0	1	0
1	0	1
1	1	0

(Ответ: а – B , б – $A + \bar{B}$, в – $\bar{A} \cdot B$, г – $A \cdot \bar{B}$)

2. Постройте выражения для логических функций, заданных таблицами истинности. Используйте разные методы и сравните их.

а)

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

б)

A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

в)

A	B	C	X
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	0

(Ответ: а – $\bar{A} \cdot C + B \cdot \bar{C}$, б – $A \cdot B + \bar{B} \cdot \bar{C}$, в – $\bar{A} \cdot \bar{B} + B \cdot \bar{C}$)

3.6 Предикаты и кванторы

В предыдущих разделах мы видели, как алгебра логики позволяет нам записывать высказывания в виде формул и делать выводы. Однако с помощью алгебры логики невозможно доказать некоторые довольно простые утверждения. Рассмотрим такие высказывания:

«Все люди смертны».

«Сократ – человек».

Каждый из нас понимает, что если оба эти высказывания истинны, то Сократ тоже смертен. Однако алгебра высказываний не позволяет это доказать. В таких случаях приходится использовать более сложный математический аппарат, с которым мы познакомимся в этом параграфе.

В начале этой главы было сказано, что утверждение «В городе N живут более 2 миллионов человек» нельзя считать логическим высказыванием, поскольку непонятно, о каком городе идет речь. В этом предложении содержится некоторое утверждение, зависящее от N ; если вместо N подставить название города, можно будет определить, истинно оно или ложно. Такое утверждение, зависящее от переменной, называют логической функцией или *предикатом*.

Предикат (от лат. *praedicatum* — заявленное, упомянутое, сказанное) — это утверждение, содержащее переменные.

Предикаты часто обозначаются буквой P , например,

$P(N) = \text{«В городе } N \text{ живут более 2 миллионов человек»}.$

Если мы задаем конкретные значения переменных, предикат превращается в логическое высказывание. Например, для предиката $P(N)$ полученное высказывание будет истинно для $N = \text{«Москва»}$ и ложно для $N = \text{«Якутск»}$.

Предикат, зависящий от одной переменной – это свойство. Например, только что рассмотренный предикат $P(N)$ характеризует свойство города. Вот еще примеры предикатов-свойств:

Простое (x) = « x – простое число»

Студент (x) = « x – студент»

Спит (x) = « x всегда спит на уроке»

Предикаты могут зависеть от нескольких переменных, например

Больше (x, y) = « x больше y »

Живет (x, y) = « x живет в городе y »

Любит (x, y) = « x любит y »

Это предикаты-отношения, они определяют связь между двумя объектами.

Предикаты нередко используются для того, чтобы задать множество, не перечисляя все его элементы. Так множество положительных чисел может быть задано предикатом, который принимает истинное значение для положительных чисел и ложное для остальных: $P(x) = (x > 0)$. Множество пар чисел, сумма которых равна 1, задается предикатом $P(x, y) = (x + y = 1)$, который зависит от двух переменных.

Существуют предикаты, которые справедливы для всех допустимых значений переменных. Например, $P(x) = (x^2 \geq 0)$, определенный на множестве всех вещественных чисел. В таком случае используют запись $\forall x P(x)$, это означает «при любом x предикат $P(x)$ справедлив». Знак $\forall$ – это буква «А», развернутая вверх ногами (от англ. *all* – все); он обозначает «любой», «всякий», «для любого», «для всех». Символ $\forall$ называют *квантором всеобщности*.

Квантор (от лат. *quantum* – сколько) — это знак или выражение, обозначающее количество.

Выражения «любой», «для всех» и т.п. также можно считать кванторами, они равносильны знаку $\forall$.

Кванторы широко применяются в математике. Например, для натуральных n

$$\forall n: 1 + 2 + \dots + n = \frac{n(n+1)}{2}.$$

Часто используют еще один квантор – *квантор существования* $\exists$ (зеркальное отражение буквы Е, от англ. *exist* – существовать). Знак $\exists$ означает «существует», «хотя бы один». Например, если $P(x) = (x - 5 > 0)$, то можно записать $\exists x P(x)$, что означает «существует x , такой что $x - 5 > 0$ ». Это уже высказывание, а не предикат, потому что можно сразу установить его истинность. Запись $\forall x P(x)$ – это тоже высказывание, но оно ложно, потому что неравенство $x - 5 > 0$ верно не для всех x .

Логическое выражение может включать несколько кванторов. Например, фразу «для любого x существует y , такой что $x + y = 0$ » можно записать как $\forall x \exists y (x + y = 0)$. Это утверждение истинно (на множестве чисел), потому что для любого x существует $(-x)$, число с обратным знаком. Переставлять местами кванторы нельзя, это меняет смысл выражения. Например, высказывание $\exists y \forall x (x + y = 0)$ означает «существует такое значение y , что для любого x выполняется равенство $x + y = 0$ », это ложное высказывание.

Теперь давайте вернемся к Сократу, точнее, к двум высказываниям, приведенным в начале параграфа. Как записать утверждение «Все люди смертны»? Можно сказать иначе: «если x – человек, то x смертен», причем это верно для любого x . Вспоминаем, что связка «если..., то» записывается как импликация, а выражение «для любого x » – в виде квантора $\forall x$. Поэтому получаем

$$\forall x (P(x) \rightarrow Q(x)),$$

где $P(x) = \text{«}x \text{ – человек»}$, $Q(x) = \text{«}x \text{ – смертен»}$. Так как утверждение $P(x) \rightarrow Q(x)$ верно для любого x , оно также верно при подстановке $x = \text{Сократ}$:

$$P(\text{Сократ}) \rightarrow Q(\text{Сократ}) = 1.$$

Поскольку Сократ – человек, $P(\text{Сократ}) = 1$. Поэтому с помощью таблицы истинности для импликации мы находим, что $Q(\text{Сократ}) = 1$, то есть «Сократ смертен».

Если построить отрицание для высказывания с квантором $\forall$ или $\exists$, мы увидим, что один квантор заменяет другой. Например, отрицание высказывания $\forall x P(x)$ («неверно, что для любого x выполняется $P(x)$ ») звучит «существует такой x , для которого не выполняется $P(x)$ » и может быть записано в виде $\exists x \overline{P(x)}$. Здесь, как и раньше, черта сверху обозначает отрицание. Таким образом,

$$\overline{\forall x P(x)} = \exists x \overline{P(x)}.$$

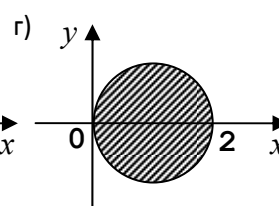
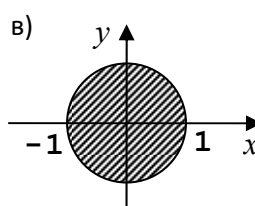
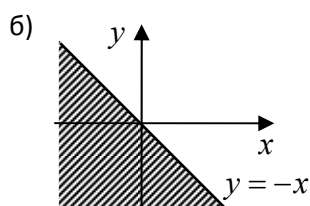
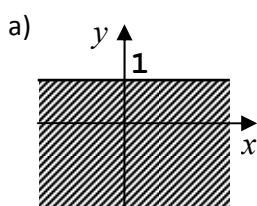
Аналогично можно показать, что $\overline{\exists x P(x)} = \forall x \overline{P(x)}$.

Где можно использовать язык предикатов? Самая подходящая для этого область информатики – системы искусственного интеллекта, в которых моделируется человеческое мышление. Многие из них строятся на языке логического программирования Пролог, в котором программа представляет собой набор данных и правила вывода новых результатов из этих данных.



Задачи

- Какие из следующих предложений являются предикатами (здесь x и y – вещественные числа)?
 - $x + y = 5$
 - $\exists x (x + y = 5)$
 - $\forall y \exists x (x + y = 5)$
 - $\sin^2 x + \cos^2 x = 1$
 - $x^2 + y^2 < 0$
 - « x работает в ВУЗе»
 - $\forall x (\text{«}x \text{ – студент»})$
 - $\exists x (\text{«}x \text{ – учитель } y\text{»})$
- Задайте с помощью предикатов множества точек, соответствующие заштрихованным областям на плоскости:



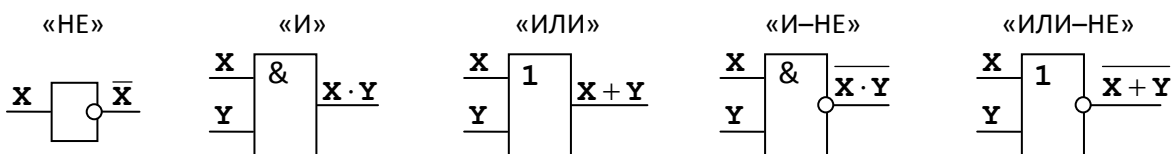
- Поставьте в начале каждого предложения одно из слов: «все» или «не все»:
 - «... окуни – рыбы».
 - «... рыбы умеют плавать».
 - «... реки впадают в моря».
 - «... моря соленые».
 - «... числа четные».
 - «... ломаные состоят из отрезков».
 - «... прямоугольники – квадраты».
 - «... кошки – млекопитающие».
- Запишите с помощью кванторов следующие утверждения:
 - «Существует x , такой что $x > y$ ».
 - «Не существует x , такой что $x > y$ ».
 - «Для любого x имеем $x^2 > 1$ ».
 - «Любая река впадает в Каспийское море».
 - «Существует река, которая впадает в Каспийское море».
 - «Для любой реки существует море, в которое она впадает».
 - «Для любого моря существует река, которая в него впадает».
 - «Существует река, которая впадает во все моря».
 - «Существует море, в которое впадают все реки».
- *Запишите с помощью кванторов следующие утверждения:
 - «Некоторые школьники ходят в театр».
 - «Все кошки серые».
 - «Встречаются злые собаки».
 - «Все люди разные».
 - «Люди ошибаются».

- е) «Никто не обращает на него внимания».
 ж) «Ни одна фирма не обанкротилась».
 з) «Все лебеди – белые или черные».
6. Запишите отрицание для следующих утверждений:
- а) $\exists x (x^2 = 5)$ г) « x работает в ВУЗе»
 б) $\exists x (x + y = 5)$ д) $\forall x (\text{«}x \text{ – студент»})$
 в) $\forall y (x + y = 5)$ е) $\exists x (\text{«}x \text{ – учитель } y\text{»})$
 г) $\forall y \exists x (x + y = 5)$ ж) $\exists x \forall y (\text{«}x \text{ – учитель } y\text{»})$

3.7 Логические элементы компьютера

3.7.1 Простейшие элементы

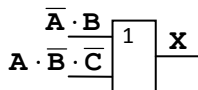
В компьютерах все вычисления выполняются с помощью *логических элементов* – электронных схем, выполняющих логические операции. Обозначения простейших элементов приводятся в таблице (ГОСТ 2.743-91). Заметьте, что небольшой кружок на выходе (или на входе) обозначает операцию «НЕ» (отрицание, инверсию).



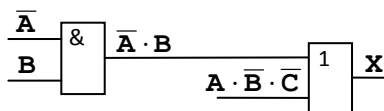
Может показаться, что для реализации сложных логических функций нужно много разных логических элементов. Однако, как мы видели в разд. 3.5, любую логическую функцию можно представить с помощью операций «НЕ», «И» и «ИЛИ» (такой набор элементов называется *полным*). Именно эта классическая «тройка» используется в книгах по логике, а также во всех языках программирования. Тем не менее, инженеры часто предпочитают строить логические схемы на основе элементов «ИЛИ-НЕ». Как показано в разд. 3.2.7, эта функция (штрих Шеффера) позволяет реализовать «НЕ», «И» и «ИЛИ», а значит и любую другую операцию.

Если нужно составить схему по известному логическому выражению, ее начинают строить с конца. Находят операцию, которая будет выполняться последней, и ставят на выходе соответствующий логический элемент. Затем повторяют то же самое для сигналов, поступающих на вход этого элемента. В конце концов, должны остаться только исходные сигналы – переменные в логическом выражении.

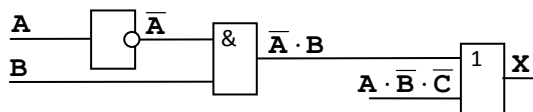
Составим схему, соответствующую выражению $X = \bar{A} \cdot B + A \cdot \bar{B} \cdot \bar{C}$. Последняя операция – это логическое сложение, поэтому на выходе схемы будет стоять элемент «ИЛИ»:



Для того, чтобы получить на первом входе $\bar{A} \cdot B$, нужно умножить $\bar{A}$ на B , поэтому добавляем элемент «И»:



Чтобы получить $\bar{A}$, ставим элемент «НЕ»:



Аналогично разбираем вторую ветку, которая поступает на второй вход элемента «ИЛИ»:

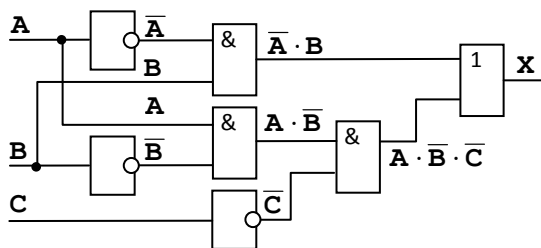


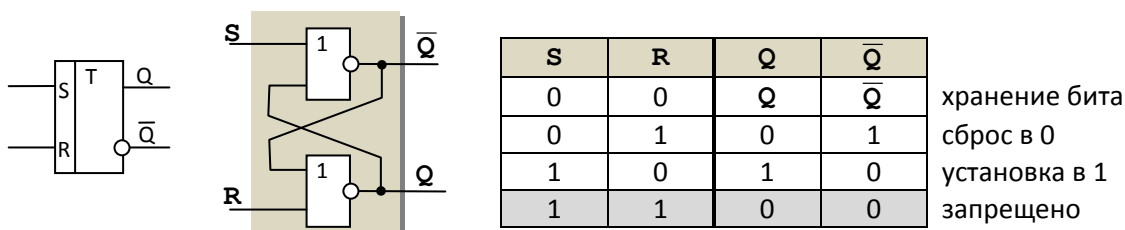
Схема составлена, ее входами являются исходные сигналы **A**, **B** и **C**, а выходом – **X**.

3.7.2 Триггер

Слово *триггер* происходит от английского слова *trigger* – «защёлка» или спусковой крючок³. Так называют электронную схему, которая может находиться только в двух состояниях (их можно обозначить как 0 и 1), и способна почти мгновенно переходить из одного состояния в другое. Триггер изобрели независимо друг от друга М.А. Бонч-Бруевич и англичане У. Икклз и Ф. Джордан в 1918 году.

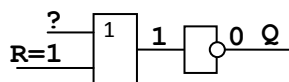
В компьютерах триггер используются для запоминания одного бита информации. Соответственно, для того, чтобы запомнить 1 байт информации требуется 8 триггеров, а для хранения 1 Кб – $8 \cdot 1024 = 8192$ триггера.

Триггеры бывают разных типов. Самый распространенный – это **RS-триггер**. Он имеет два входа, которые обозначаются как **S** (англ. *set* – установить) и **R** (англ. *reset* – сброс), и два выхода – **Q** и $\bar{Q}$, причем выходной сигнал $\bar{Q}$ является логическим отрицанием сигнала **Q** (если **Q** = 1, то $\bar{Q}$ = 0, и наоборот). RS-триггер можно построить на двух элементах «И-НЕ» или на двух элементах «ИЛИ-НЕ». На следующем рисунке показано условное обозначение RS-триггера, внутреннее устройство триггера на элементах «ИЛИ-НЕ» и его таблица истинности.

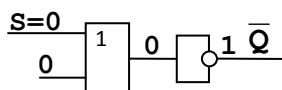


Триггер использует так называемые *обратные связи* – сигналы с выходов схем «ИЛИ-НЕ» поступают на вход соседней схемы. Именно это позволяет хранить информацию.

Рассмотрим таблицу истинности триггера. Начнем с варианта, когда **S** = 0 и **R** = 1. Элемент «ИЛИ-НЕ» в нижней части схемы можно заменить на последовательное соединение элементов «ИЛИ» и «НЕ». Тогда, независимо от второго входа, на выходе «ИЛИ» будет 1, а на выходе «НЕ» – нуль. Это значит, что **Q** = 0.



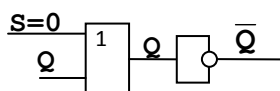
Тогда на входе другого элемента «ИЛИ-НЕ» будут два нуля, а на выходе $\bar{Q}$ – единица.



Поскольку основным выходом считается **Q**, мы записали в триггер значение 0. Схема симметрична, поэтому легко догадаться, что при **S** = 1 и **R** = 0 мы запишем в триггер 1 (**Q** = 1).

Теперь рассмотрим случай, когда **S** = 0 и **R** = 0. На выходе первого элемента «ИЛИ» будет сигнал **Q** + 0 = **Q**, поэтому на выходе $\bar{Q}$ останется его предыдущее значение:

³ В английском языке триггер называется *flip-flop*.



Аналогично легко показать, что на выходе Q тоже остается его предыдущее значение. Это *режим хранения бита*. Для случая $S = 1$ и $R = 1$ мы увидим, что оба выхода становятся равны нулю – в этом нет смысла, поэтому такой вариант запрещен.

Для хранения многоразрядных данных триггеры объединяются в единый блок, который называется *регистром*. Регистры (от 8 до 64 бит) используются во всех процессорах для временного хранения промежуточных результатов.

Над регистром, как над единым целым, можно производить ряд стандартных операций: сбрасывать (обнулять), заносить в него код и т.д. Часто регистры способны не просто хранить информацию, но и обрабатывать ее. Например, существуют регистры-счетчики, которые подсчитывают количество импульсов, поступающих на вход.

Триггеры применяются также в микросхемах быстродействующей оперативной памяти.

3.7.3 Сумматор

Как следует из названия, сумматор предназначен для сложения (суммирования) двоичных чисел. Сначала рассмотрим более простой элемент, который называют *полусумматором*. Он выполняет сложение двух битов с учетом того, что в результате может получиться двухразрядное число (с переносом в следующий разряд).

Обозначим через A и B входы полусумматора, а через P и S – выходы (перенос в следующий разряд и бит, остающийся в текущем разряде). Таблица истинности этого устройства показана на рисунке. Легко увидеть, что столбец P – это результат операции «И», а столбец S – результат «исключающего ИЛИ»:

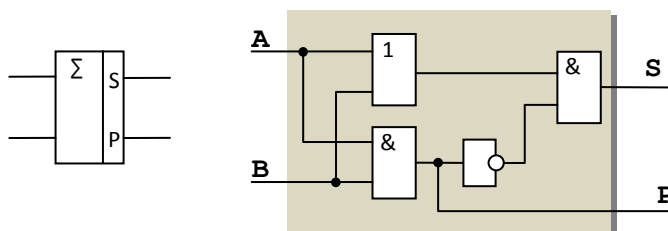
A	B	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$P = A \cdot B, \quad S = A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}.$$

Формулу для S можно также записать в таком виде

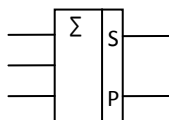
$$S = \overline{A} \cdot B + A \cdot \overline{B} = (A + B) \cdot (\overline{A} + \overline{B}) = (A + B) \cdot \overline{(A \cdot B)} = (A + B) \cdot \overline{P},$$

что позволяет построить полусумматор, используя всего 4 простейших элемента:



Слева показано условное обозначение полусумматора, греческая буква Σ здесь (и в математике) обозначает сумму.

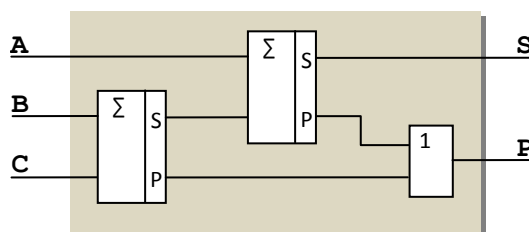
Полный **одноразрядный сумматор** учитывает также и третий бит – перенос из предыдущего разряда C . Сумматор имеет три входа и два выхода. Таблица истинности и обозначение сумматора показаны на рисунках.



A	B	C	P	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

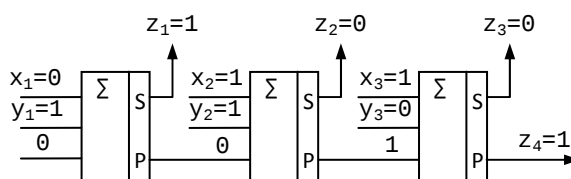
Логические функции для выходов сумматора вы можете найти самостоятельно.

Сумматор можно построить с помощью двух полусумматоров и одного элемента «ИЛИ»:



Сначала складываются биты В и С, а затем к результату добавляется бит А. Перенос на выходе сумматора появляется тогда, когда любое из двух промежуточных сложений дает перенос.

Для сложения многоразрядных чисел сумматоры объединяют в цепочку. При этом выход Р одного сумматора (перенос в следующий разряд) соединяется с входом С следующего. На рисунке показано, как складываются два трехразрядных разрядных числа: $X = 110_2$ и $Y = 011_2$. Сумма $Z = 1001_2$ состоит из четырех бит, поэтому на выходе последнего сумматора бит переноса будет равен 1.



Сложение начинается с самого младшего разряда. На вход первого сумматора подается младшие биты исходных чисел, x_1 и y_1 (см. рисунок), а на третий вход – ноль (нет переноса из предыдущего разряда). Выход S первого сумматора – это младший бит результата, z_1 , а его выход Р (перенос) передается на вход второго сумматора и т.д. Выход Р последнего из сумматоров представляет собой дополнительный разряд суммы, то есть z_4 .

Сумматор играет важную роль не только при сложении чисел, но при выполнении других арифметических действий. Фактически является основой арифметического устройства современного компьютера.

? Контрольные вопросы

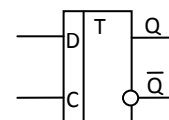
1. Что такое триггер? Объясните его принцип действия.
2. Почему для RS-триггера комбинация входов $S = 1$ и $R = 1$ запрещена?
3. Чем отличается одноразрядный сумматор от полусумматора?
4. Как можно построить сумматор с помощью двух полусумматоров?
5. Постройте логические выражения для выходов сумматора и нарисуйте соответствующие им схемы.
6. Объясните, как работает многоразрядный сумматор.
7. Что такое перенос? Как он используется в многоразрядном сумматоре?

⚙ Задачи

1. Используя логические элементы, постройте схемы, соответствующие логическим выражениям $X_1 = \bar{A} \cdot C + B \cdot \bar{C}$, $X_2 = A \cdot B + \bar{B} \cdot \bar{C}$, $X_3 = \bar{A} \cdot \bar{B} + B \cdot \bar{C}$.
2. Соревнования по поднятию тяжестей судит бригада из трех человек, один из них старший. Лампочка «Вес взят» должна загораться, если проголосовали, по крайней мере, два судьи, причем один из них – старший. Предложите логическую схему, которая решала бы эту задачу.
3. В двухэтажном коттедже есть два выключателя, которые управляют освещением лестницы, один из них – на первом этаже, а второй – на втором. Каждый выключатель имеет два состояния, при нажатии на кнопку состояние изменяется. В исходный момент оба выключателя выключены. Когда человек заходит в неосвещенное здание, он нажимает кнопку выключателя на первом этаже, при этом должна загореться лампочка, освещающая всю лестницу. Поднявшись на второй этаж, он нажимает на кнопку второго выключателя, и

лампочка должна погаснуть. Когда следом идет другой человек, он действует так же (хотя оба выключателя находятся в другом положении). Предложите логическую схему, которая решала бы эту задачу.

4. В самолете есть три бака с горючим. Бортовой компьютер получает сигналы от датчиков уровня в каждом баке: если горючего в баке достаточно, то сигнал равен 0, если горючее кончилось – 1. Когда горючее заканчивается, по крайней мере, в двух баках, должна загореться лампочка «Тревога». Предложите логическую схему, которая решала бы эту задачу.
5. В парламенте некоторой страны выбирают спикера из трех кандидатов. Каждый парламентарий должен нажать одну и только одну из трех кнопок. Если он проголосовал правильно (нажал ровно одну кнопку), на пульте должна загореться зеленая лампочка. Предложите логическую схему, которая решала бы эту задачу.
6. Постройте RS-триггер на элементах «И-НЕ» и составьте его таблицу истинности.
7. *Постройте таблицу истинности и логическую схему D-триггера, который запоминает сигнал на входе D (англ. *data* – данные) при подаче логической единицы на вход C (англ. *clock* – синхронизация). Для этого можно, например, немного изменить входную часть RS-триггера.



3.8 Логические задачи

3.8.1 Метод рассуждений

Задача 1. Среди трех приятелей (их зовут Сеня, Вася и Миша) один всегда говорит правду, второй говорит правду через раз, а третий все время обманывает. Как-то раз они впервые прогуляли урок информатики. Директор школы вызвал их в свой кабинет для разговора. Сеня сказал: «Я всегда прогуливаю информатику. Не верьте тому, что скажет Вася». Вася сказал: «Я раньше не прогуливал этот предмет». Миша сказал: «Все, что говорит Сеня, – правда». Директору стало все понятно. Кто из них правдив, кто лгун, а кто говорит правду через раз?

Для решения используем метод рассуждений. Во-первых, есть «точная» информация, которая не подвергается сомнению: все трое прогуляли урок информатики в первый раз.

Запишем высказывания мальчиков:

Сеня: 1. Я всегда прогуливаю информатику.

2. Вася сейчас соврет.

Вася: 1. Я раньше не прогуливал информатику.

Миша: 1. Сеня говорит правду.

Известно, что один из них говорит правду всегда, второй — через раз, а третий все время лжет. Отметим, что если у нас есть только одно высказывание «полу-лжеца», оно может быть как истинным, так и ложным.

Сопоставив первое высказывание Сени и высказывание Васи с «точной» информацией, сразу определяем, что тут Сеня соврал, а Вася сказал правду. Это значит, что второе высказывание Сени тоже неверно, поэтому мальчик Сеня всегда лжет.

Тогда один из оставшихся, Вася или Миша, правдив, а второй говорит правду через раз. Мишино высказывание неверно, поскольку мы уже определили, что Сеня лжет; это значит, что Миша не всегда говорит правду, он – «полу-лжец». Тогда получается, что Вася правдив.

3.8.2 Табличный метод

Задача 2. Перед началом турнира по шахматам болельщики высказали следующие предположения по поводу результатов:

А) Максим победит, Борис – второй;

Б) Борис – третий, Коля – первый;

В) Максим – последний, а первый – Дима.

Когда соревнования закончились, оказалось, что каждый из болельщиков был прав только в одном из своих прогнозов. Как распределились призовые места?

Запишем высказывания трех болельщиков в форме таблицы (заголовок строки обозначает место в турнирной таблице). Будем считать, что каждое место занял ровно один участник.

Начнем «раскручивать» эту таблицу с той строчки, где больше всего информации, в данном случае – с первой. Предположим, что Максим действительно занял первое место, как и сказал болельщик «А». В этом случае «В» ошибся, поставив на первое место Диму. Тогда получается, что второй прогноз болельщика «В» верен, и Максим – последний.

	А	Б	В
1	Максим?	Коля?	Дима?
2	Борис?		
3		Борис?	
4			Максим?

Так как мы предполагали, что Максим занял первое место, получается противоречие. Следовательно, первый прогноз «А» не сбился. Но тогда должен быть верен его второй прогноз, и Борис занял второе место. При этом он не мог занять еще и третье место, поэтому первый прогноз болельщика «Б» неверный, а верен его второй прогноз: Коля – первый.

В этом случае Дима не может быть первым, поэтому верен первый прогноз «В»: Максим – последний. Диме осталось единственное свободное третье место. В результате места распределились так: I – Коля, II – Борис, III – Дима и IV – Максим.

	А	Б	В
1	Максим?	Коля	Дима?
2	Борис		
3		Борис?	
4			Максим

Задача 3. На одной улице стоят в ряд 4 дома, в каждом из них живет по одному человеку. Их зовут Василий, Семен, Геннадий и Иван. Известно, что все они имеют разные профессии: скрипач, столяр, охотник и врач. Известно, что

- (1) Столяр живет правее охотника.
- (2) Врач живет левее охотника.
- (3) Скрипач живет с краю.
- (4) Скрипач живет рядом с врачом.
- (5) Семен не скрипач и не живет рядом со скрипачом.
- (6) Иван живет рядом с охотником.
- (7) Василий живет правее врача.
- (8) Василий живет через дом от Ивана.

Определите, кто где живет.

Из условий (1) и (2) следует, что охотник живет не с краю, потому что справа от него живет столяр, а слева – врач. Скрипач по условию (3) живет с краю, он может жить как слева, так и справа от остальных:

скрипач?	врач	охотник	столяр	скрипач?
----------	------	---------	--------	----------

Согласно условию (4), скрипач живет рядом с врачом, поэтому он занимает крайний дом слева:

1	2	3	4
скрипач	врач	охотник	столяр

Профессии жильцов определили, остается разобраться с именами. Из условия (5) «Семен не скрипач и не живет рядом со скрипачом» следует, что Семен – охотник или столяр:

1	2	3	4
скрипач	врач	охотник	столяр
		Семен?	Семен?

Из условия (6) «Иван живет рядом с охотником» следует, что он – врач или столяр:

1	2	3	4
скрипач	врач	охотник	столяр
		Семен?	Семен?
	Иван?		Иван?

Из условия (7) «Василий живет правее врача» определяем, что Василий – охотник или столяр:

1	2	3	4
скрипач	врач	охотник	столяр
		Семен?	Семен?
	Иван?		Иван?
		Василий?	Василий?

Согласно условию (8), «Василий живет через дом от Ивана», поэтому Иван – врач, а Василий – столяр:

1	2	3	4
скрипач	врач	охотник	столяр
	Иван	Семен?	Василий

Тогда сразу получается, что Семен – охотник, а Геннадий должен занять оставшееся свободное место, он – скрипач:

1	2	3	4
скрипач	врач	охотник	столяр
Геннадий	Иван	Семен	Василий

Задача 3. Шесть друзей, Саша, Петя, Витя, Дима, Миша и Кирилл, встретившись через 10 лет после окончания школы, выяснили, что двое из них живут в Москве, двое – в Санкт-Петербурге, а двое – в Перми. Известно, что

- (1) Витя ездит в гости к родственникам в Москву и Санкт-Петербург.
- (2) Петя старше Саши.
- (3) Дима и Миша летом были в Перми в командировке.
- (4) Кирилл и Саша закончили университет в Санкт-Петербурге и уехали в другие города.
- (5) Самый молодой из них живет в Москве.
- (6) Кирилл редко приезжает в Москву.
- (7) Витя и Дима часто бывают в Санкт-Петербурге по работе.

Определите, кто где живет.

Составим таблицу, где каждая строка соответствует городу, а столбец – человеку:

	Саша	Петя	Витя	Дима	Миша	Кирилл
Москва						
Санкт-Петербург						
Пермь						

Единица в таблице будет обозначать, что человек живет в данном городе, а ноль – что точно не живет. По условию в каждом городе живут ровно 2 человека, каждый живет только в одном городе. Поэтому в каждой строке должно быть две единицы, а в каждом столбце – одна.

Из условия (1) следует, что Витя живет в Перми:

	Саша	Петя	Витя	Дима	Миша	Кирилл
Москва			0			
Санкт-Петербург			0			
Пермь			1			

Из (2) и (5) находим, что Петя живет не в Москве. Кроме того, как следует из (6), Кирилл – тоже не москвич.

	Саша	Петя	Витя	Дима	Миша	Кирилл
Москва		0	0			0
Санкт-Петербург			0			
Пермь			1			

Согласно (3), Дима и Миша живут не в Перми:

	Саша	Петя	Витя	Дима	Миша	Кирилл
Москва		0	0			0
Санкт-Петербург			0			
Пермь			1	0	0	

Из условия (4) делаем вывод, что Кирилл и Саша живут не в Санкт-Петербурге, отсюда сразу следует, что Кирилл живет в Перми. Двух пермяков мы уже определили, поэтому Саша и Петя живут не в Перми:

	Саша	Петя	Витя	Дима	Миша	Кирилл
Москва		0	0			0
Санкт-Петербург	0		0			0
Пермь	0	0	1	0	0	1

Далее находим, что Саша – москвич, а Петя живет в Санкт-Петербурге.

	Саша	Петя	Витя	Дима	Миша	Кирилл
Москва	1	0	0			0
Санкт-Петербург	0	1	0			0
Пермь	0	0	1	0	0	1

По условию (7) Витя и Дима – не петербуржцы, поэтому в Петербурге живет Миша, а Дима – в Москве:

	Саша	Петя	Витя	Дима	Миша	Кирилл
Москва	1	0	0	1	0	0
Санкт-Петербург	0	1	0	0	1	0
Пермь	0	0	1	0	0	1

Таким образом, Саша и Дима живут в Москве, Петя и Миша – в Санкт-Петербурге, а Витя и Кирилл – в Перми.

3.8.3 Использование алгебры логики

Когда в условии задачи встречаются сложные логические высказывания, удобно использовать методы алгебры логики. Покажем этот подход на примерах.

Задача 4. Следующие два высказывания истинны:

1. Неверно, что если корабль А вышел в море, то корабль С – нет.
2. В море вышел корабль В или корабль С, но не оба вместе.

Определить, какие корабли вышли в море.

Введем три высказывания: **А** – корабль А вышел в море; **В** – корабль В вышел в море; **С** – корабль С вышел в море. Вспомним, что связка «если..., то» в логических выражениях заменяется импликацией, поэтому фразу «если корабль А вышел в море, то корабль С – нет» можно записать как $\overline{A \rightarrow C} = 1$. Но в условии сказано, что это утверждение неверно, поэтому

$$\overline{A \rightarrow C} = 0 \text{ или } A \rightarrow C = 1.$$

Второе условие – это «исключающее ИЛИ», то есть $B \oplus C = 1$. Оба условия истинны одновременно, то есть их логическое произведение («И») тоже истинно:

$$(\overline{A \rightarrow C}) \cdot (B \oplus C) = 1.$$

Нам нужно решить это уравнение и найти неизвестные **A**, **B** и **C**. Для этого выразим импликацию и «исключающее ИЛИ» через базовый набор логических операций («НЕ», «И», «ИЛИ»), а затем раскроем инверсию сложного выражения с помощью закона де Моргана:

$$(\overline{A \rightarrow C}) \cdot (B \oplus C) = (\overline{A + C}) \cdot (B \cdot \overline{C} + \overline{B} \cdot C) = A \cdot C \cdot (B \cdot \overline{C} + \overline{B} \cdot C) = 1.$$

В последнем выражении раскроем скобки и учтем, что $C \cdot \overline{C} = 0$ и $C \cdot C = C$. Получим

$$A \cdot \overline{B} \cdot C = 1.$$

Это уравнение имеет единственное решение: **A** = 1, **B** = 0 и **C** = 1. Это значит, что в море вышли корабли **A** и **C**.

Задача 5. На вопрос «Кто из твоих учеников изучал логику?» учитель ответил: «Если логику изучал Андрей, то изучал и Борис. Однако неверно, что если изучал Семен, то изучал и Борис». Кто же изучал логику?

Обозначим буквами высказывания: **A** – логику изучал Андрей; **B** – логику изучал Борис и **C** – логику изучал Семен. Оба высказывания учителя можно записать в виде импликаций

«Если логику изучал Андрей, то изучал и Борис». $A \rightarrow B = 1$

«Неверно, что если изучал Семен, то изучал и Борис». $C \rightarrow B = 0$

Дальше есть два варианта решения. Во-первых, можно поступить так же, как и в предыдущей задаче: применить операцию «НЕ» ко второму высказыванию и составить уравнение с помощью логического произведения:

$$(A \rightarrow B) \cdot \overline{(C \rightarrow B)} = 1.$$

Теперь представляем импликацию через базовые операции и применяем закон де Моргана

$$(\overline{A} + B) \cdot (\overline{C} + B) = (\overline{A} + B) \cdot C \cdot \overline{B} = \overline{A} \cdot C \cdot \overline{B} = 1.$$

Это уравнение имеет единственное решение: **A** = 0, **B** = 0 и **C** = 1. Значит, логику изучал только Семен.

Можно поступить иначе, вспомнив, что импликация ложна только в том случае, когда первое высказывание истинно, а второе ложно. Поэтому из условия $C \rightarrow B = 0$ сразу следует, что **B** = 0 и **C** = 1. Тогда первое условие, $A \rightarrow B = A \rightarrow 0 = 1$ сразу дает **A** = 0.



Задачи

1. Три школьника, Миша, Коля и Сергей, остававшиеся в классе на перемене, были вызваны к директору по поводу разбитого в это время окна в кабинете. На вопрос директора о том, кто это сделал, мальчики ответили следующее:

Миша: «Я не бил окно, и Коля тоже...»

Коля: «Миша не разбивал окно, это Сергей разбил футбольным мячом!»

Сергей: «Я не делал этого, стекло разбил Миша».

Выяснилось, что один из ребят сказал чистую правду, второй в одной части заявления соврал, а другое его высказывание истинно, а третий оба раза соврал. Кто разбил стекло в классе?

(Ответ: Миша)

2. В финал соревнований по настольному теннису вышли Наташа, Маша, Люда и Рита. Болельщики высказали свои предположения о распределении мест в дальнейших состязаниях. Один считает, что первой будет Наташа, а Маша будет второй. Другой болельщик на второе место прочит Люду, а Рита, по его мнению, займет четвертое место. Третий считает, что Рита займет третье место, а Наташа будет второй. Когда соревнования закончились, оказалось, что каждый из болельщиков был прав только в одном из своих прогнозов. Как распределились места?

(Ответ: I – Наташа, II – Люда, III – Рита, IV – Маша)

3. На одной улице стоят в ряд 4 дома, в каждом из них живет по одному человеку. Их зовут Алексей, Егор, Виктор и Михаил. Известно, что все они имеют разные профессии: рыбак, пчеловод, фермер и ветеринар. Известно, что

(1) Фермер живет правее пчеловода.

(2) Рыбак живет правее фермера.

- (3) Ветеринар живет рядом с рыбаком.
- (4) Рыбак живет через дом от пчеловода.
- (5) Алексей живет правее фермера.
- (6) Виктор – не пчеловод.
- (7) Егор живет рядом с рыбаком.
- (8) Виктор живет правее Алексея.

Определите, кто где живет.

(Ответ: пчеловод Михаил, фермер Егор, рыбак Алексей, ветеринар Виктор)

4. Дочерей Василия Лоханкина зовут Даша, Анфиса и Лариса. У них разные профессии и они живут в разных городах: одна в Ростове, вторая – в Париже и третья – в Москве. Известно, что

- (1) Даша живет не в Париже, а Лариса – не в Ростове.
- (2) Парижанка – не актриса.
- (3) В Ростове живет певица.
- (4) Лариса – не балерина.

Определите, где живет каждая из дочерей и чем занимается.

(Ответ: Даша – певица, Ростов; Анфиса – балерина, Париж; Лариса – актриса, Москва)

5. В состав экспедиции входят Михаил, Сергей и Виктор. На обсуждении распределения обязанностей с руководителем проекта были высказаны предположения, что командиром будет назначен Михаил, Сергей не будет механиком, а Виктор будет утвержден радистом, но командиром не будет. Позже выяснилось, что только одно из этих четырех утверждений оказалось верным. Как распределились должности?

(Ответ: Виктор – командир, Михаил – механик, Сергей – радист)

6. В ходе заседания суда выяснилось, что:

- (1) Если Аськин не виновен или Баськин виновен, то виновен Сенькин.
- (2) Если Аськин не виновен, то Сенькин не виновен.

Виновен ли Аськин?

(Ответ: виновен)

7. Аськин, Баськин и Васькин стали свидетелями ограбления банка. Во время расследования Аськин сказал, что взломщики приехали на синей «Тойоте». Баськин считает, что это был красный «BMW», а Васькин утверждает, что это был «Форд-Фокус», но не синий. Выяснилось, что каждый из них назвал неправильно либо марку, либо цвет машины. На каком автомобиле приехали преступники?

(Ответ: красная «Тойота»)