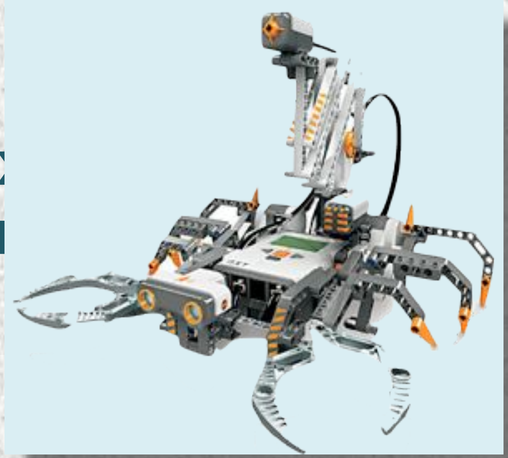
ВВЕДЕНИЕ В ТЕОРИЮ АЛГОРИТМОВ

Алгоритм – предписание, точный набор инструкций, описывающих порядок действий исполнителя для достижения результата - решения задачи за конечное число шагов.

Алгоритм описывает **процесс преобразования исходных данных в результаты.**



Алгоритм описывается в командах исполнителя, который это алгорит будет выполнять.

Исходные данные и результаты любого алгоритма всегда принадлежат среде исполнителя, для которого предназначен алгоритм.

Алгоритм всегда рассчитан на выполнение «неразмышляющим» исполнителем.

Алгоритм не содержит ошибок, если он дает правильные результаты для любых допустимых исходных данных.

Алгоритм содержит ошибки, если

- он приводит к получению неправильных результатов;
- он завершает работу ранее запланированного шага (аварийный останов), не получив ожидаемых результатов;
- завершения работы алгоритма не происходит – исполнитель переходит от шага к шагу бесконечное число раз.

СВОЙСТВА АЛГОРИТМОВ

Обязательные свойства алгоритма:

1. Дискретность
2. Элементарность шагов
3. Детерминированность
4. Понятность
5. Результативность
6. Конечность (финитность)
7. Массовость

Свойства рецепта, **процесса**, **метода**, **способа**

Полный набор **обязательных свойства алгоритма** обеспечивает получение результата неразмышляющим исполнителем, в расчете на которого создан алгоритм. При условии, что он будет однозначно и точно следовать командам, определенным на каждом этапе алгоритма.

СРАВНЕНИЕ АЛГОРИТМОВ

- Получить решение поставленной задачи нередко можно разными способами, привлекая ***разные алгоритмы***.
- Как выбрать ***наиболее эффект****ивный* из конкурирующих алгоритмов?
- Сравнение алгоритмов правомерно только для ***одного и того же*** исполнителя и актуально лишь для ***массового*** применения.

Задание «неразмышляющему» исполнителю: вычислить 85 × 85.

Алгоритм 1. Угадывать последовательным перебором чисел из [101, 10 000] до «победного конца».

Алгоритм 2. Умножение «в столбик». Требуется оперативная память тетрадочного листа.

Алгоритм 3. По формуле (8 × (8 + 1)) × 100 + 5 × 5. Вычисления – устный счет.

Алгоритм 4. По вычисленной ранее таблице умножения 100×100, имеющейся у исполнителя.

http://rain.ifmo.ru/cat/view.php/theory/school

КРИТЕРИИ СРАВНЕНИЯ АЛГОРИТМОВ

Алгоритм 1
Угадывать последователь-
ным перебором чисел из [101, 10 000] до «победного конца».

Алгоритм 2
Умножение «в столбик». Требуется оперативная память для записи промежуточных результатов.

Алгоритм 3
По формуле
(8 × (8 + 1)) × 100 + 5 × 5.
Вычисления – устный счет.

Алгоритм 4
По вычисленной ранее таблице умножения 100×100, имеющейся у исполнителя.

- ***Ёмкостная сложность.*** Оценивается объем используемой оперативной памяти.
  Алгоритм 1 – лучший.
- ***Объём внешней памяти*.** Оцениваются привлекаемые ресурсы внешней памяти, например, при сравнении алгоритмов сортировки массива.
  Алгоритм 4 - самый затратный, расширенная таблица умножения хранится во ***внешней памяти.***
- ***Оценка временной сложности в среднем*** — оценивается время исполнения алгоритма.
  Алгоритм 3 - лучший.
- **По времени исполнения алгоритма в *худшем случае*.**
  Алгоритм 1 – аутсайдер.

http://rain.ifmo.ru/cat/view.php/theory/school

КРИТЕРИИ СРАВНЕНИЯ АЛГОРИТМОВ

- Оценивать ***эффективность компьютерного алгоритма*** следует ***до написания и отладки*** компьютерной программы.

- Нередко ***оценка временной эффективности*** опытным путем, в реальном времени, принципиально невозможна. Например, при неоправданно больших затратах машинного времени.

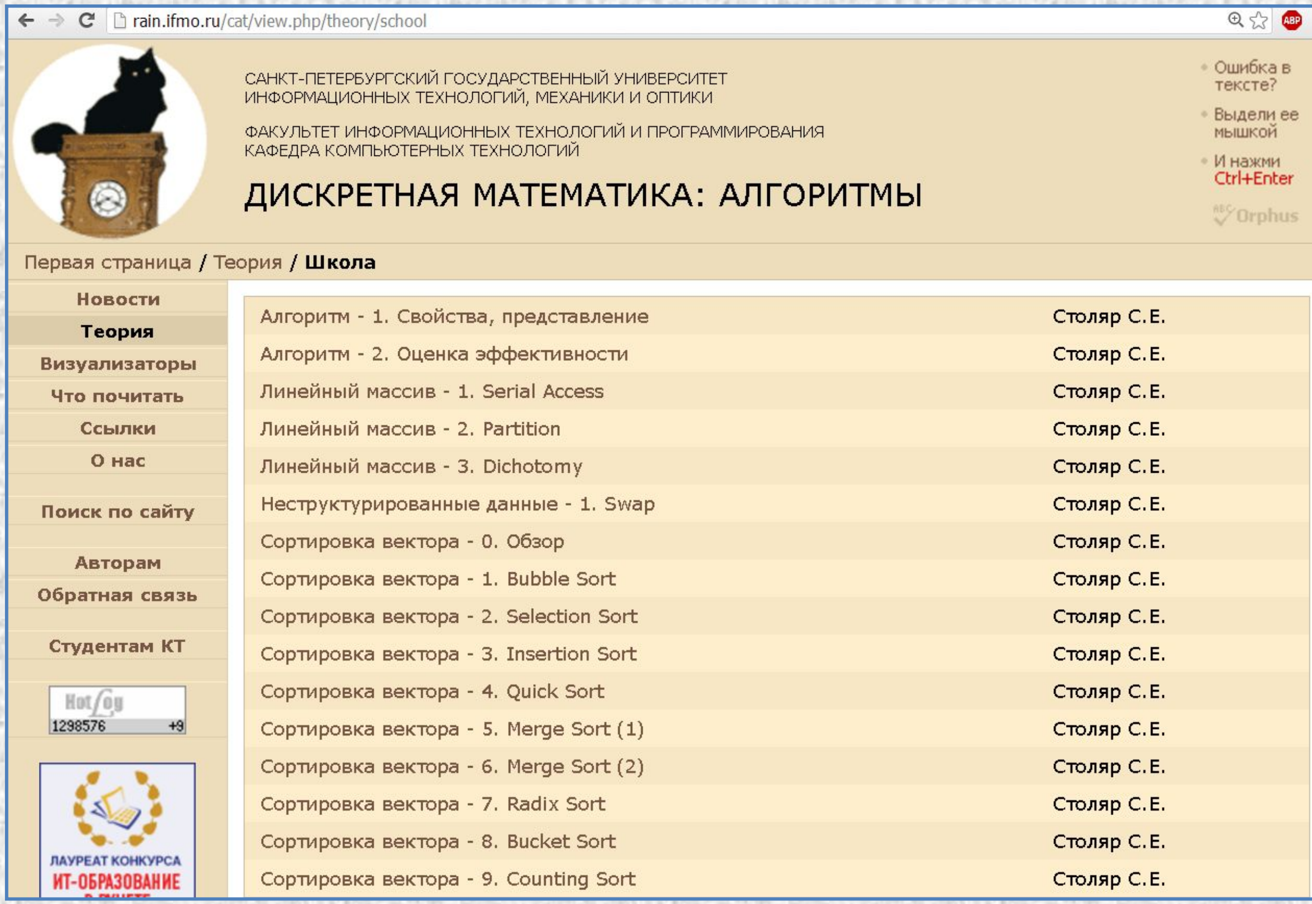
- При оценке временной сложности принято ориентироваться либо на ***число шагов алгоритма*** либо на ***машинную операцию*** (инструкцию программного кода). Шаг алгоритма – это инструкция ***абстрактного исполнителя***, не требующая более подробного алгоритмического измельчения.

- Алгоритмическое время выполнения одного шага не должно зависеть от параметров задачи. В противном случае стоимость укрупненного шага известна и будет учитываться в общей оценке.

http://rain.ifmo.ru/cat/view.php/theory/school

1. Могилев А.В. Информатика / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. — М.: Издательский центр «Академия». Изд. 8, - 2012.

2. http://rain.ifmo.ru/cat/view.php/theory/school



Основные принципы построения архитектуры ЭВМ

- Использование двоичной системы представления данных
- Принцип программного управления
- Принцип однородности памяти
- Принцип хранимой программы
- Принцип адресности

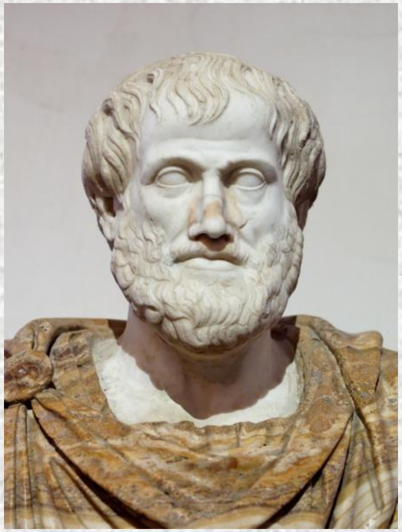
Логические основы построения и работы ЭВМ

Базовые логические элементы ЭВМ

Логические элементы компьютера, реализующие элементарные логические функции (И,ИЛИ, НЕ, ИЛИ-НЕ, И-НЕ).
Электронные схемы (сумматор, триггер)

Основы алгебры логики

- Таблицы истинности
- Логические функции
- Аксиомы алгебры логики
- Элементарные логические операции



ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

Логика – наука о формах и способах мышления

Основы формальной логики заложил Аристотель. Он впервые отделил логические формы мышления от его содержания.

Законы логики отражают в сознании человека свойства, связи и отношения объектов окружающего мира.

Логика позволяет строить формальные модели окружающего мира, отвлекаясь от содержательной стороны.

Формы мышления

Понятие

Понятие – это форма мышления, фиксирующая основные, существенные признаки предмета

⇨ Компьюте
р

⇨ Набор электронных устройств

Формы мышления

Понятие

Содержание

Содержание понятия составляет совокупность существенных признаков объекта

⇨ Компьютер ⇨ Универсальное устройство для автоматической обработки информации

Формы мышления

Понятие

Содержани
е

Объем

Объем понятия определяется совокупностью предметов, на которое оно распространяется

**Компьюте
р**

Формы мышления

Понятие

Высказывание

Высказывание – это форма мышления, в которой что-либо утверждается или отрицается о реальных предметах, их свойствах и отношениях между ними.

2 x 2 =4 - математический язык -

Дважды два равно пять – естественный язык - Ложно

Высказывание может быть либо истинным, либо ложным

Алгебра высказываний определяет истинность или ложность составных высказываний

Формы мышления

Понятие

Высказывание

Умозаключение

Умозаключение – это форма мышления, с помощью которой из одного или нескольких суждений (посылок) может быть получено новое суждение (вывод)

Все углы треугольника равны $\Rightarrow$ Треугольник равносторонний

__Алгебра высказываний__ **служит для определения истинности или ложности составных высказываний,**
не вникая в их содержание

__Простое высказывание__ **состоит из одного высказывания и не содержит логической операции.**

Высказывание принимает одно из двух значений:
(1) истина, (0) – ложь

Пример.
Простые высказывания:
«процессор является устройством обработки информации»
«принтер является устройством печати»

Составное высказывание **содержит высказывания, объединенные логическими операциями.**

Логические операции:

- **И - логическое умножение, конъюнкция**
- **ИЛИ - логическое сложение, дизъюнкция**
- **НЕ - логическое отрицание, инверсия**
- **«ЕСЛИ - ТО» - логическое следование, импликация**
- **«тогда и только тогда, когда» - эквивалентность, равнозначность**

Пример.

Составное высказывание, состоящее из двух простых, соединённых союзом операцией «И»:

«процессор является устройством обработки информации И принтер является устройством печати»

Логическое умножение (конъюнкция) -
объединение двух или более высказываний в одно при помощи операции «И».

Составное высказывание, образованное в результате операции «конъюнкция», истинно только тогда, когда истинны входящие в него простые высказывания.

Конъюнкция обозначается: &, ^, *

Пример 1.

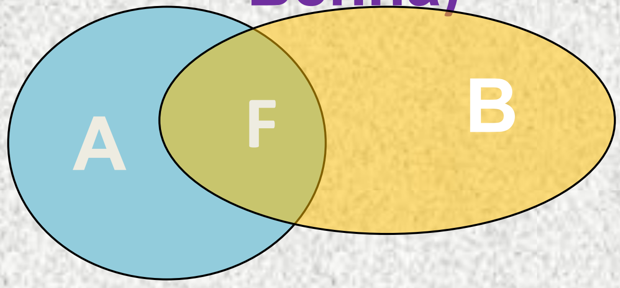
A. На улице идет дождь
B. На улице светит солнце
C. Стоит теплая погода
D. Стоит холодная погода

E. На улице идет дождь и стоит холодная погода E = A & D
F. На улице светит солнце и стоит теплая погода F = B & C
G. На улице идет дождь и стоит теплая погода G = A & C
H. На улице светит солнце и стоит холодная погода H = B & D

Таблица истинности операции «конъюнкция»

A	B	F=A&B
0	0	0
0	1	0
1	0	0
1	1	1

Пересечение множеств (диаграмма Эйлера – Венна)



Логическое сложение (дизъюнкция)- **объединение двух или более высказываний в одно при помощи союза «ИЛИ»**

Составное высказывание, образованное в результате операции дизъюнкции, истинно тогда, когда истинно хотя бы одно из входящих в него простых высказываний

Дизъюнкция обозначается: ∨, +

Пример 2.

A. 2 x 2 = 4
B. 3 x 3 = 9
C. 2 x 2 = 5
D. 4 x 4 = 4
E. 3 x 3 = 6

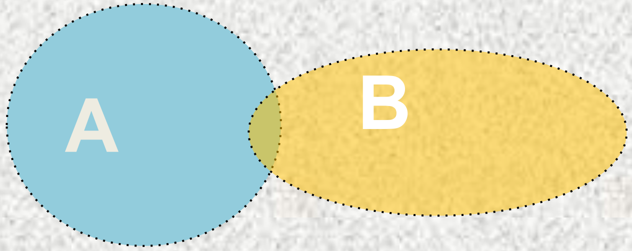
F. 2 x 2 = 4 или 4 x 4 = 4
G. 3 x 3 = 9 или 2 x 2 = 5
H. 2 x 2 = 4 или 2 x 2 = 5
I. 2 x 2 = 5 или 3 x 3 = 6

$F = A \vee D$
$G = B \vee C$
$H = A \vee C$
$I = C \vee E$

Таблица истинности операции «дизъюнкция»

A	B	$F=A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Объединение множеств (диаграмма Эйлера – Венна)



Логическое отрицание (инверсия) – **присоединение частицы «не» к высказыванию**

Логическое отрицание (инверсия) делает истинное высказывание ложным и, наоборот, ложное - истинным

Инверсия обозначается: ¯, ¬

Пример 3.

1) A: 2 x 2 = 4
A - истина
¬A - ложь

2) B: 2 x 2 = 5
B – ложь
¬B - истина

Дополнение до универсального множества (диаграмма Эйлера – Венна)

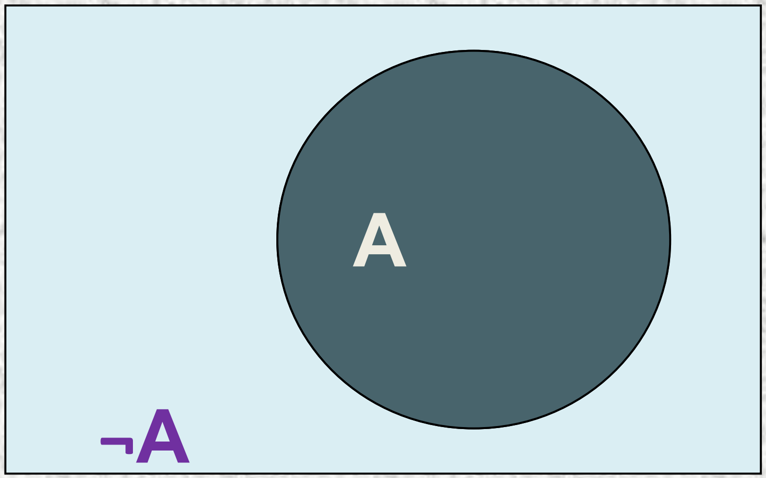


Таблица истинности операции «инверсия»

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

Импликация **двух высказываний А и В - такое высказывание, которое ложно тогда и только тогда, когда А - истинно, а В - ложно.**

Таблица истинности операции «импликация»

А	В	А → В
0	0	1
0	1	1
1	0	0
1	1	1

Импликация обозначается: ®, →

Логическое выражение «А → В» в устной интерпретации «звучит»: «если А, то В» или «А имплицирует В».

<u>Эквиваленция</u> двух высказываний А и В - такое высказывание, которое истинно тогда и только тогда, когда оба высказывания либо истинны, либо ложны.

Таблица истинности операции «эквиваленция»

А	В	А ↔ В
0	0	1
0	1	0
1	0	0
1	1	1

Эквиваленция обозначается: ↔
Логическое выражение «А ↔ В» в устной интерпретации «звучит»: «А тогда и только тогда, когда В».

Логической переменной **называется переменная, значением которой может быть любое высказывание, например:** x, y, x_1, y_1, x_k, y_n

Логической формулой **является:**

1) **любая логическая переменная, а также каждая из двух логических констант — 0 (ложь) и 1 (истина);**
2) **если *A* и *B* — формулы, то B и *A*B* — тоже формулы, где знак «*» означает любую из логических бинарных операций.**

Пример 4:

$A=(x \,\&\, y) \rightarrow z$

Формула принимает одно из двух значений — 0 или 1.

Формулы А и В, зависящие от одного и того же набора переменных $x_1, x_2, x_3, \dots x_n$, называют <u>***равносильными***</u> или <u>***эквивалентными***</u>, если на любом наборе значений переменных $x_1, x_2, x_3, \dots x_n$ они имеют одинаковые значения, т.е. А = В

Любую формулу можно преобразовать к равносильной ей, в которой используются только операции: &, v и ¬.

ПРИОРИТЕТ ЛОГИЧЕСКИХ ОПЕРАЦИЙ

В формуле логические переменные, обозначающие высказывания, объединяются знаками логических операций и скобками.

Пример: F = A или B и не A или не B = A + B & ¬A + ¬B

- действия в скобках
- инверсия
- конъюнкция
- дизъюнкция
- импликация
- эквивалентность

При необходимости скобки задают требуемый порядок выполнения.

Пример.

$U \vee (B \Rightarrow C) \,\&\, D \Leftrightarrow \bar{U}$

Порядок вычисления:

1) $(B \Rightarrow C)$
2) $\bar{U}$
3) $(B \Rightarrow C) \,\&\, D$
4) $U \vee (B \Rightarrow C) \,\&\, D$
5) $U \vee (B \Rightarrow C) \,\&\, D \Leftrightarrow \bar{U}$

Пример 5.

Даны простые высказывания:

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

A: Процессор – устройство для обработки информации

B: Сканер – устройство вывода информации

C: Монитор – устройство ввода информации

D: Клавиатура – устройство вывода информации

Определите истинность логических выражений:

(AVB) <=> (C&D)

(A&B) -> (CVD)

(AVB) -> (C&D)

(A&B) <=> (CVD)

($\bar{A}$ -> B)&(CVD)

(C <=> $\bar{A}$)&B&D

(A&B)VC <=> (A&C)V(A&B)

(AVB)VC -> (A&C&D)&(BVD)

Ответы: A=1, B=0, C=0, D=0

(AVB) <=> (C&D) = 0

(A&B) -> (CVD) = 1

(AVB) -> (C&D) = 0

(A&B) <=> (CVD) = 1

($\bar{A}$ -> B)&(CVD) = 0

(C <=> $\bar{A}$)&B&D = 0

(A&B)VC <=> (A&C)V(A&B) = 1

(AVB)VC -> (A&C&D)&(BVD) = 0

Логические выражения и таблицы истинности

Таблица истинности определяет истинность или ложность высказывания (логического выражения) при всех возможных комбинациях исходных значений простых высказываний (логических переменных).

Количество строк в таблице истинности логического выражения определяется количеством логических переменных (N), равно 2^N.

Логические выражения, у которых таблицы истинности совпадают, называются *равносильными* или *эквивалентными.*

ЛОГИЧЕСКИЕ ФУНКЦИИ

Любое составное высказывание можно рассматривать как логическую функцию $F(X_1, X_2, \ldots, X_N)$, аргументами которой являются логические переменные $X_1, X_2, \ldots, X_N$ - простые высказывания.

Функция и аргументы могут принимать только два различных значения: «истина» (1) и «ложь» (0).

БУЛЕВЫ ФУНКЦИИ ДВУХ АРГУМЕНТОВ

Количество строк в таблице: $N_1=2^2 = 4$.

Количество столбцов в таблице истинности: $N_2=2^{N1} =2^4 = 16$.

F(A,B)=0

F(A,B)=A&B

F(A,B)=A V B

F(A,B)=¬(A V B)

F(A,B)=A↔B

F(A,B)=A→B

F(A,B)=¬(A&B)

F(A,B)=1

Аргу-менты		Таблица истинности логических функций двух аргументов															
A	B	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Инверсия дизъюнкции («стрелка Пирса», «ИЛИ-НЕ»): $F(A,B)= A\downarrow B = \neg (A \lor B)$

A	B	$F(A,B)= A\downarrow B = \neg (A \lor B)$
0	0	1
0	1	0
1	0	0
1	1	0

Инверсия конъюнкции («штрих Шеффера», «И-НЕ»): $F(A,B)= A\,|\,B = \neg (A \,\&\, B)$

A	B	$F(A,B)= A\,|\,B = \neg (A \,\&\, B)$
0	0	1
0	1	1
1	0	1
1	1	0

Основные законы и тождества булевой алгебры

	Для дизъюнкции	Для конъюнкции
Ассоциативность	AV(BVC)=(AVB)VC	A&(B&C)=(A&B)&C
Коммутативность	A VB=BVA	A&B=B&A
Дистрибутивность	A&(BVC)=(A&B)V(A&C)	AV(B&C)=(AVB)&(AVC)
Идемпотентность	AVA=A	A&A=A
Операции с константами	AV1=1 AV0=A	A&1=A A&0=0
Правила де Моргана	¬ (AVB)=¬A & ¬B	¬ (A&B)=¬A V ¬B
Операции переменной с её инверсией	$AV\bar{A}=1$	$A\&\bar{A}=0$
Правила поглощения	AV(A&B)=A	A&(AVB)=A
Правила склеивания	(A&B)V(¬A & B)= B	(AVB)&(¬A V B)= B

Правило замены операции импликации: A → B = ¬ A V B

Правило замены операции эквивалентности: A ↔ B = (¬ A V B) V (A V ¬ B)

Правило двойной инверсии: ¬ ¬ A =A

Любой из основных законов и тождеств булевой алгебры может быть доказан с помощью таблиц истинности.

Пример 6.
Правило де Моргана: ¬(x & y) = ¬x V ¬y

x	y	x & y	¬(x & y)	¬x	¬y	¬x V ¬y
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Законы алгебры логики можно доказать **путем логических рассуждений.**

Пример 7. Доказательство первого закона поглощения:

x V (x & y)= x

- Пусть истинна правая часть, т. е. *x = 1*, тогда в левой части дизъюнкция x v (x & y) истинна.
- Пусть истинна левая часть.
  Тогда по определению дизъюнкции истинна или формула x, или формула (x & y), или обе эти формулы одновременно.
- Если x ложна, тогда (x & y) ложна, следовательно, x может быть только истинной.

Законы алгебры логики можно доказать путем тождественных преобразований.

Пример 8.

Доказательство первого закона поглощения

$x \vee (x \& y) = x$

$$x \vee (x \& y) = (x \& 1) \vee (x \& y) = x \& (1 \vee y) = x$$

Формула *A* называется тавтологией (или тождественно истинной), если она истинна при любых значениях своих переменных.

Пример 9.

x V ¬x =1

(операция переменной с её инверсией)

Формула А называется *тождественно ложной*,

если она равна 0 при любых значениях своих переменных.

Пример 10.

$x \,\&\, \neg x = 0$

Пример 11.
Определить *x*, если:

$$\neg(x \lor a) \lor \neg(x \lor \neg a) = b$$

Решение

$$\neg(x \lor a) \lor \neg(x \lor \neg a) =$$

$$= (\neg x \,\&\, \neg a) \lor (\neg x \,\&\, \neg\neg a) =$$

$$= (\neg x \,\&\, \neg a) \lor (\neg x \,\&\, a) =$$

$$= (\neg x \,\&\, \neg x) \lor (\neg a \,\&\, a) =$$

$$= \neg x \,\&\, 1 = \neg x$$

$$\neg x = b$$

$$x = \neg b$$

Пример 12.

Какие формулы являются тавтологиями?

1) $\neg(a \,\&\, \neg a)$
2) $a \rightarrow (b \rightarrow a)$
3) $(a \,\&\, b) \rightarrow a$

Таблицы истинности логических операций (для справки):

A	B	F=A⋁B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	F=A&B
0	0	0
0	1	0
1	0	0
1	1	1

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

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

1) ¬(a & ¬a)

a	$\overline{a}$	$a \,\&\, \overline{a}$	$\overline{a \,\&\, \overline{a}}$
0	1	0	1
1	0	0	1

2) $a \rightarrow (b \rightarrow a)$

a	b	$b \rightarrow a$	$a \rightarrow (b \rightarrow a)$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	1	1

3) (a & b) → a

a	b	b & a	(a & b) → a
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Пример 13.
Является ли формула тождественно ложной?

$$a \mathbin{\&} (a \to b) \mathbin{\&} (a \to \neg b)$$

a	b	$a \to b$	$\neg b$	$a \to \neg b$	$a \mathbin{\&} (a \to b) \mathbin{\&} (a \to \neg b)$
0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	1	0	0	0

Любую формулу можно преобразовать к равносильной ей, в которой используются только операции НЕ, И, ИЛИ.

Пример 14.

Упростить: $A \& B \vee A \& \overline{B}$

По закону дистрибутивности вынесем А за скобки:

$$A \& B \vee A \& \overline{B} = A \& (B \vee \overline{B}) = A \& 1 = A$$

Пример 15.

Способ 1. Применим закон дистрибутивности:

Способ 2. Перемножим скобки на основании закона дистрибутивности:

Пример 16.

F_1 = {если одно слагаемое делится на 3 и сумма делится на 3, то и другое слагаемое делится на 3};

F_2 = {если одно слагаемое делится на 3, а другое не делится на 3, то сумма не делится на 3}.

Формализуйте эти высказывания, постройте таблицы истинности для каждой из полученных формул и убедитесь, что результирующие столбцы совпадают.

x = <одно слагаемое делится на 3>

y = <сумма делится на 3>

z = <другое слагаемое делится на 3>

$F_1 = x \& y \rightarrow z$

$F_2 = x \& \neg z \rightarrow \neg y$

$F_1 = x \& y \rightarrow z$

$F_2 = x \& \neg z \rightarrow \neg y$

x	y	z	x & y	F_1	x & ¬z	¬y	F_2
0	0	0	0	1	0	1	1
0	0	1	0	1	0	1	1
0	1	0	0	1	0	0	1
0	1	1	0	1	0	0	1
1	0	0	0	1	1	1	1
1	0	1	0	1	0	1	1
1	1	0	1	0	1	0	0
1	1	1	1	1	0	0	1

Решение логических задач

1. Выделить из условия задачи элементарные высказывания и обозначить их буквами.
2. Записать условие задачи с помощью логических операций.
3. Составить единое логическое выражение для всех требований задачи.
4. Используя законы алгебры логики, упростить выражение и вычислить его значения либо построить для него таблицу истинности.
5. Выбрать решение — набор значений простых высказываний, при котором построенное логическое выражение является истинным.
6. Проверить, удовлетворяет ли полученное решение условию задачи.

Пример 17.

На вопрос «Кто из трех студентов изучал логику?», был получен ответ:

«Если изучал первый, то изучал и второй, но неверно, что если изучал третий, то изучал и второй». Кто из учащихся изучал логику?

На вопрос «Кто из трех студентов изучал логику?», был получен ответ:

«Если изучал первый, то изучал и второй, но неверно, что если изучал третий, то изучал и второй». Кто из учащихся изучал логику?

Обозначим:

P_1 – <логику изучал первый учащийся>,

P_2 – <логику изучал второй учащийся>,

P_3 – <логику изучал третий учащийся>.

Выражение $(P_1 \rightarrow P_2) \& \neg(P_3 \rightarrow P_2) = 1$.

Упростим выражение

$(P_1 \rightarrow P_2) \& \neg(P_3 \rightarrow P_2) = (\neg P_1 \vee P_2) \& \neg(\neg P_3 \vee P_2) =$

$= (\neg P_1 \vee P_2) \& P_3 \& \neg P_2 = \neg P_1 \& P_3 \& \neg P_2 \vee P_2 \& P_3 \& \neg P_2$

Высказывание P2 & ¬P2 =0 (правило операции переменной с ее инверсией), значит: P2 & P3 & ¬P2=0.

Поэтому высказывание: ¬P1 & P3 & ¬P2=1.

Логику изучал третий учащийся, а первый и второй не изучали.

Пример 18.

Три подразделения A, B, C фирмы стремились получить максимальную прибыль.

1. Если A получит максимальную прибыль, то B и C получат максимальную прибыль.
2. Либо A и C получат максимальную прибыль одновременно, либо одновременно не получат.
3. Для того чтобы подразделение C получило максимальную прибыль, необходимо, чтобы и B получило максимальную прибыль.

Одно из трех предположений оказалось ложно, а остальные два истинны.

Какие подразделения получили максимальную прибыль?

A = {A получит максимальную прибыль},

B = {B получит максимальную прибыль},

C = {C получит максимальную прибыль}.

Если A получит максимальную прибыль, то B и C получат максимальную прибыль.

Либо A и C получат максимальную прибыль одновременно, либо одновременно не получат.

Для того чтобы подразделение С получило максимальную прибыль, необходимо, чтобы и В получило максимальную прибыль.

1) $F_1 = A \rightarrow B \& C$;

2) $F_2 = A \& C \vee \neg A \& \neg C$;

3) $F_3 = C \rightarrow B$.

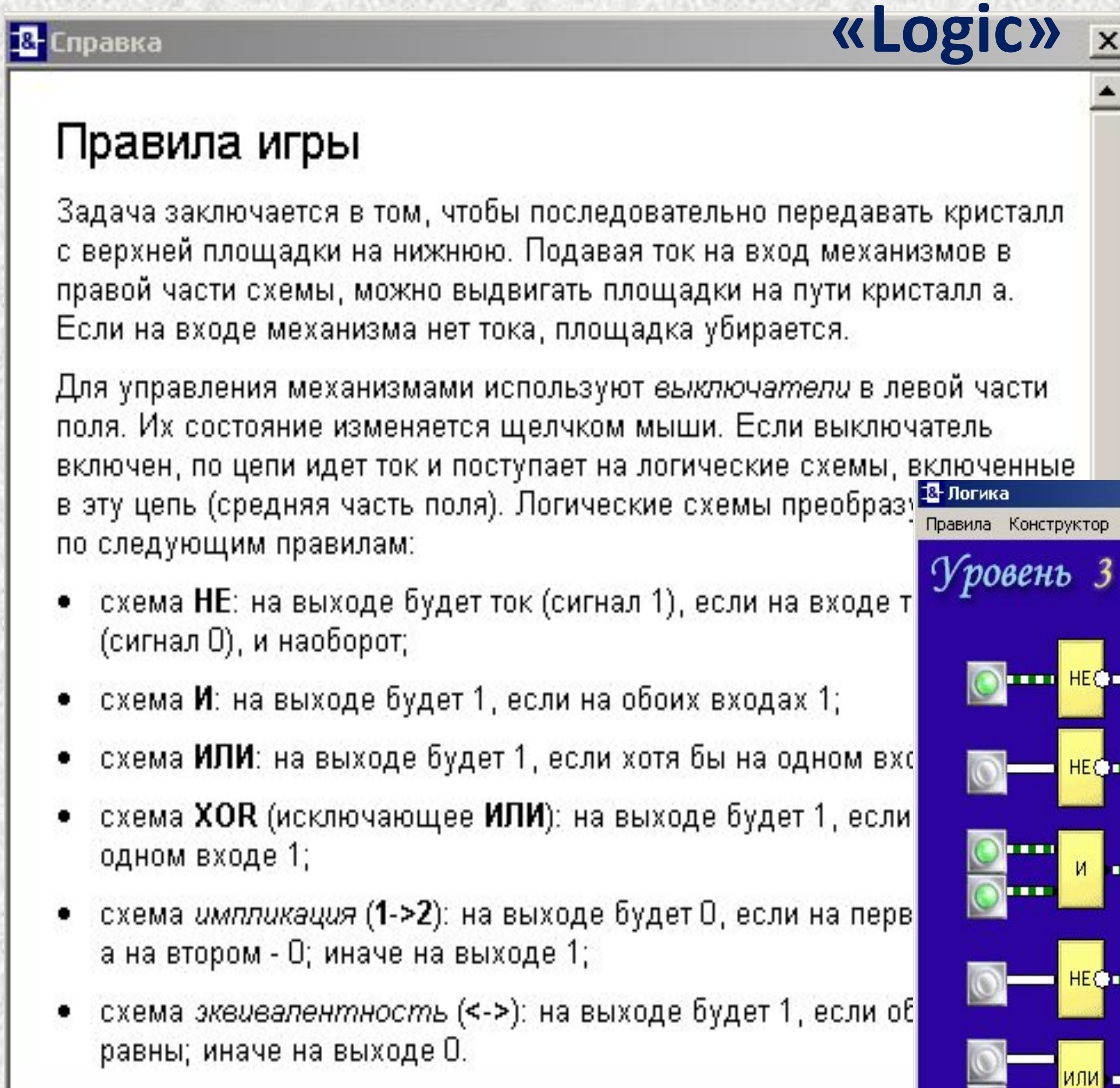
Одно из трех предположений оказалось ложно, а остальные два истинны.

Таблица истинности для F_1, F_2, F_3

A	B	C	F_1	F_2	F_3
0	0	0	1	1	1
0	0	1	1	0	0
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Ответ: В и С получат максимальную прибыль

Таблицы истинности. Обучающая программа «Logic»


Справка

Правила игры

Задача заключается в том, чтобы последовательно передавать кристалл с верхней площадки на нижнюю. Подавая ток на вход механизмов в правой части схемы, можно выдвигать площадки на пути кристалл а. Если на входе механизма нет тока, площадка убирается.

Для управления механизмами используют *выключатели* в левой части поля. Их состояние изменяется щелчком мыши. Если выключатель включен, по цепи идет ток и поступает на логические схемы, включенные в эту цепь (средняя часть поля). Логические схемы преобразу
по следующим правилам:

- схема **НЕ**: на выходе будет ток (сигнал 1), если на входе т
(сигнал 0), и наоборот;
- схема **И**: на выходе будет 1, если на обоих входах 1;
- схема **ИЛИ**: на выходе будет 1, если хотя бы на одном вхо
- схема **XOR** (исключающее **ИЛИ**): на выходе будет 1, если
одном входе 1;
- схема *импликация* (**1->2**): на выходе будет 0, если на перв
а на втором - 0; иначе на выходе 1;
- схема *эквивалентность* (**<->**): на выходе будет 1, если об
равны; иначе на выходе 0.