Алгебра высказываний

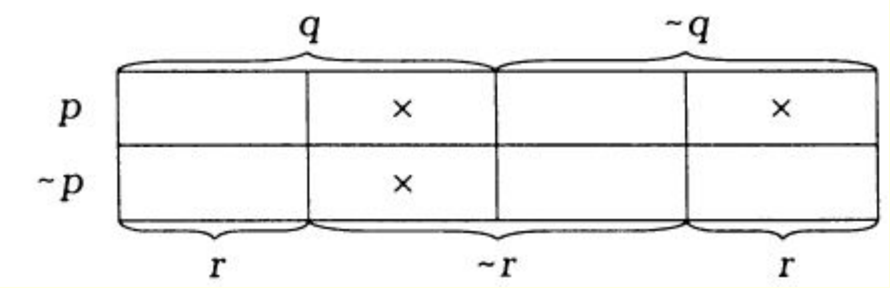
Упрощение СДНФ при помощи Карты Карно

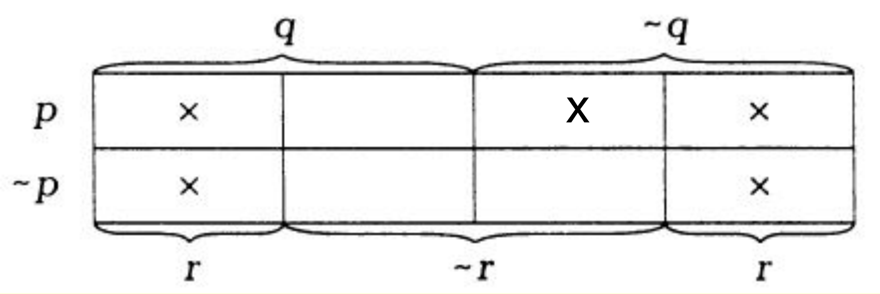
- **Карта Карно** – это таблица каждый элемент которой является элементарной конъюнкцией
- **Для 2 переменных p, q** возможны 4 элементарные конъюнкции pq, p]q,]pq,]p]q, которые являются элементами следующей таблицы
- Для представления картой Карно, высказывания в виде СДНФ с двумя переменными, необходимо **отметить клетки соответствующие элементарным конъюнкциям**. Например высказыванию pq ∨ p]q, соответствуют следующие отметки.
- Если высказыванию соответствуют **две соседние** (вертикальные или горизонтальные) отметки, то выражение можно упростить **оставив их общий элемент**. Так в приведенном примере выражение соответствует общему элементу p.
- Это же можно получить используя **эквивалентные соотношения** pq ∨ p]q = p(q ∨]q)=p&1=p

	q	$\sim q$
p	×	×
$\sim p$		

Карта Карно для СДНФ с 3 переменными

- Процедура упрощения **для СДНФ с 3 переменными p, q, r по Карте Карно** остается такой же.
- **В левой таблице** представлена СДНФ: pq]r ∨ p]qr ∨]pq]r = q]r ∨ p]qr
- Если **4** отмеченные клетки **образуют прямоугольник или выстроены в линию**, то после сокращения **остается общий элемент**. Например: на правом рисунке представлена СДНФ: pqr ∨]pqr ∨ p]qr ∨]p]qr ∨ p]q]r = r ∨ p]q.
- Заметим, что **r появляется на обоих концах карты**. Поэтому мы **можем «скрутить» карту Карно** и считать, что отмеченные клетки образуют квадрат. После сокращения квадрата получим r.
- **Одну и ту же клетку при сокращении можно использовать дважды**. Две правые верхние клетки сократятся до p]q.





Карта Карно для СДНФ с 4 переменными

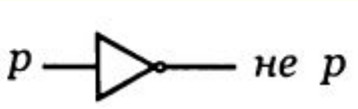
- Процедура упрощения **для СДНФ с 4 переменными p, q, r, s по Карте Карно** остается такой же (от выделенного блока остается общая часть)
- **Пример: левая карта Карно** приводит к следующим сокращениям: **8-кратный** блок имеет общий элемент **q, 4-кратный** горизонтальный блок имеет общий элемент **ps, 2-кратный** вертикальный блок имеет общий элемент **p]qr**. Таким образом получена минимальная ДНФ **q ∨ ps ∨ p]qr**
- Так как s находится в **верхней и нижней строке**, то их можно **считать прилегающими,** по аналогии с правым и левым краем.
- **Пример: правая карта Карно** приводит к сокращению **rs ∨ ps**

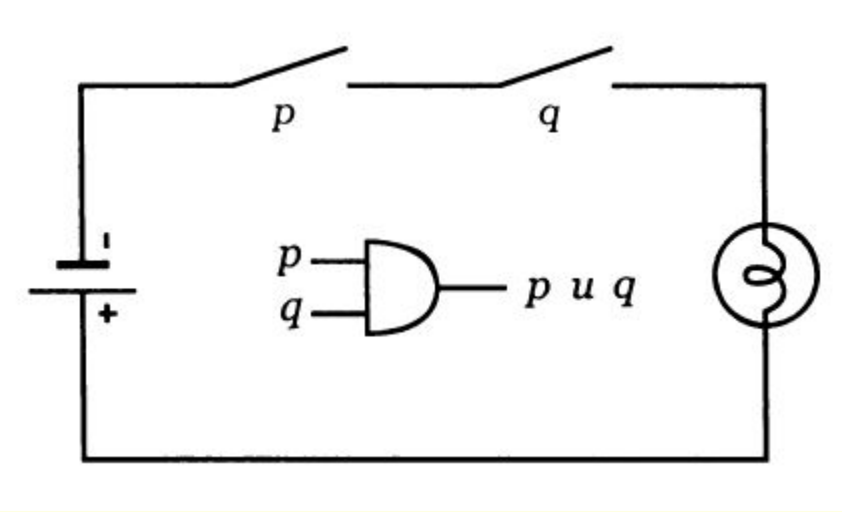
	q	q	~q	~q	
p	×	×	×	×	s
p	×	×		×	~s
~p	×	×			~s
~p	×	×			s
	r	~r	~r	r	

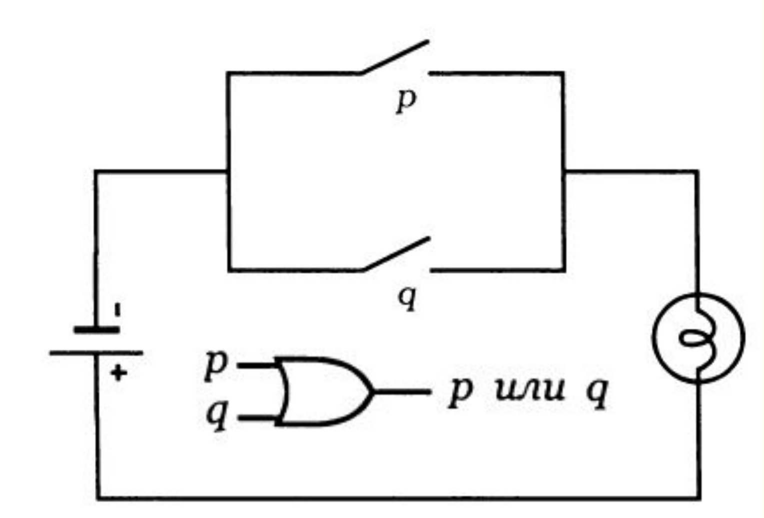
	q	q	~q	~q	
p	×			×	s
p					~s
~p					~s
~p	×			×	s
	r	~r	~r	r	

Булева алгебра и коммутационные схемы

- В 1938 г. Клод Шеннон заметил **связь между таблицами истинности и электрическими цепями**.
- В схеме представленной на **левом рисунке** лампочка загорается (имеет значение 1) если оба переключателя замкнуты (значения 1), что соответствует высказыванию **pq**.
- В схеме на **правом рисунке** лампочка загорается (1) если хотя бы один из переключателей замкнут (т.е. хотя бы один имеет значение 1), что соответствует высказыванию **p ∨ q**
- Предполагается, что имеется схема в которой лампочка загорается, если выключатель разомкнут.



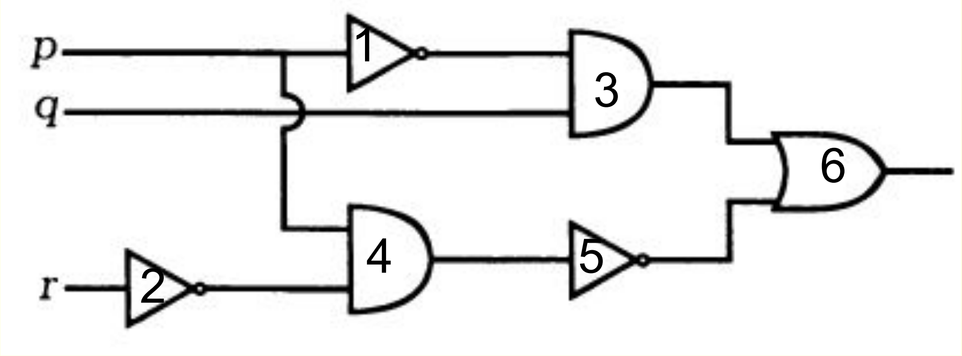




Анализ коммутационных схем

- **Анализ** коммутационных схем заключается в **определении булевой формулы** соответствующей рассматриваемой схемы
- Для этого **составляют таблицу**, в которой для каждого функционального элемента определяют значения входов и выхода.
- **Выход последнего элемента** определяет итоговую булевую функцию.

Эл	Вх1	Вх2	Вых
1	p		**]p**
2	r		**]r**
3	**]p**	q	**]pq**
4	p	**]r**	**p]r**
5	**p]r**		**] (p]r)**
6	**]pq**	**] (p]r)**	**]pq ∨] (p]r)**

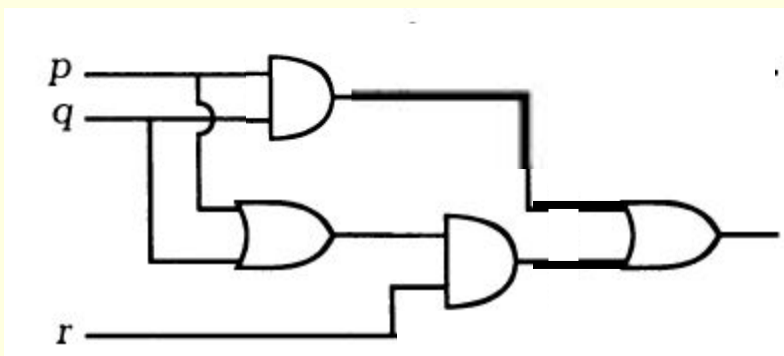


Синтез коммутационных схем

- **Синтез** коммутационных схем заключается в **построении схемы по заданной формуле**
- **Пример:** Муниципальный совет состоит из 3 человек. Каждый член совета имеет для голосования кнопку «за» и «против». Решение принимается если за него проголосует большинство. Построить коммутационную схему устройства, сигнализирующего о том, что решение принято.
- Решение будет принято, если голосование пройдет по любому из вариантов **pqr,]pqr, p]qr, pq]r.** Поэтому итоговая булевая функция: **pqr ∨]pqr ∨ p]qr ∨ pq]r**
- В дальнейшем, если **элементарная конъюнкция** состоит **из n переменных,** то ее функциональный элемент **имеет n входов.** Аналогичное применяется для **элементарной дизъюнкции.**

Синтез коммутационных схем
Пример

- Прежде чем строить коммутационную схему формулу следует максимально упростить, используя карту Карно или закону булевой алгебры **$pqr \vee \neg pqr \vee p\neg qr \vee pq\neg r = pq \vee qr \vee pr = pq \vee r\,(q \vee p)$**



Проектирование полубитного сумматора

- Полубитный сумматор складывает два одноразрядных числа (p, q), представленных в двоичном виде. На выходе получают двухразрядное число (d_1d_0), где **d0 -** первый разряд **d1-** второй разряд (разряд переноса)

+	0	1
0	0	1
1	1	10

+	0	1
0	00	01
1	01	10

p	q	d_0
1	1	0
1	0	1
0	1	1
0	0	0

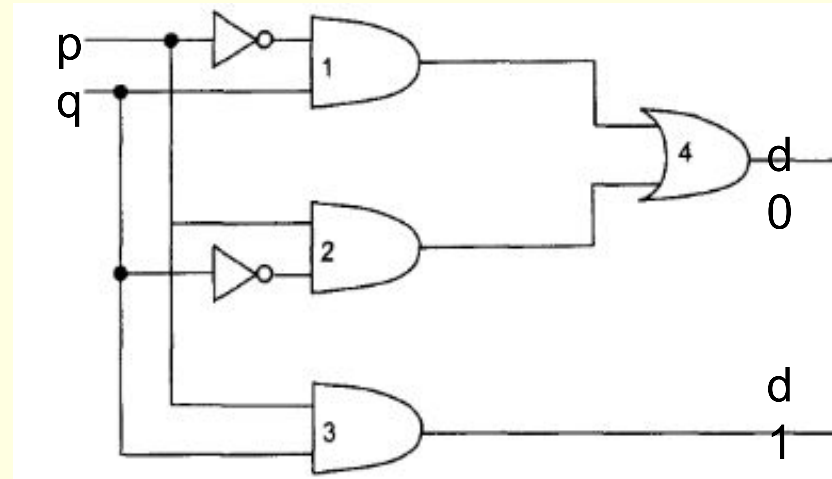
p	q	d_1
1	1	1
1	0	0
0	1	0
0	0	0

Построим СДНФ для $d0 = p\neg q \vee \neg pq$, $d1=pq$

Проектирование полубитного сумматора

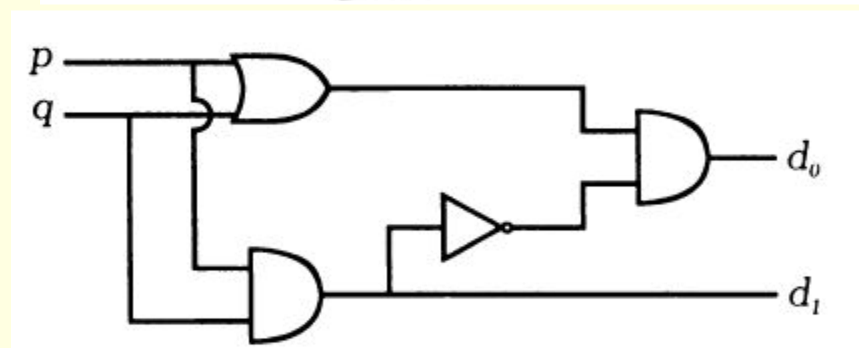
Построим схему

$d0 = p\neg q \vee \neg pq, \quad d1 = pq.$

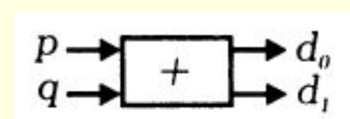


Можно построить эквивалентную схему, содержащую меньшее число функциональных элементов.
Для этого используя булеву алгебру проведем упрощение d0

$$d0 = p\neg q \vee \neg pq = \neg\neg(p\neg q \vee \neg pq) =$$
$$= \neg(\neg(p\neg q)\ \&\ \neg(\neg pq)) =$$
$$= \neg((\neg p \vee q)\ \&\ (p \vee \neg q)) =$$
$$= \neg(p\neg p \vee qp \vee \neg p\neg q \vee q\neg q) =$$
$$= \neg(qp \vee \neg p\neg q) = \neg(qp)\ \&\ \neg(\neg p\neg q) =$$
$$= (\neg p \vee \neg q)\ \&\ (q \vee p) = \neg(pq)\ \&\ (q \vee p)$$



В дальнейшем используют следующее обозначение полубитного сумматора



Проектирование полного сумматора

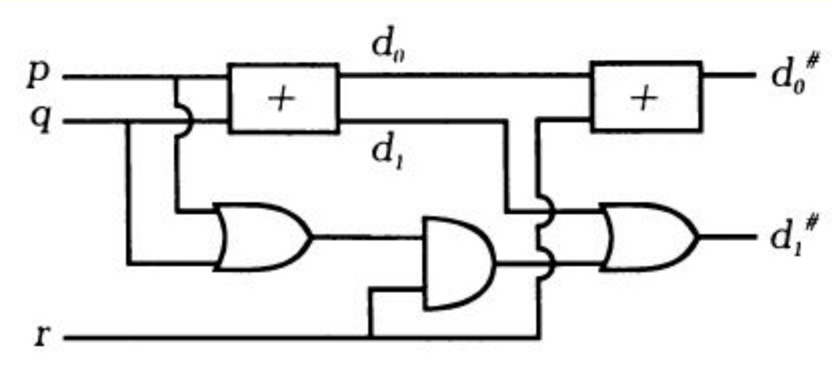
p	q	r	$d_1^\#$	$d_0^\#$
1	1	1	1	1
1	1	0	1	0
1	0	1	1	0
1	0	0	0	1
0	1	1	1	0
0	1	0	0	1
0	0	1	0	1
0	0	0	0	0

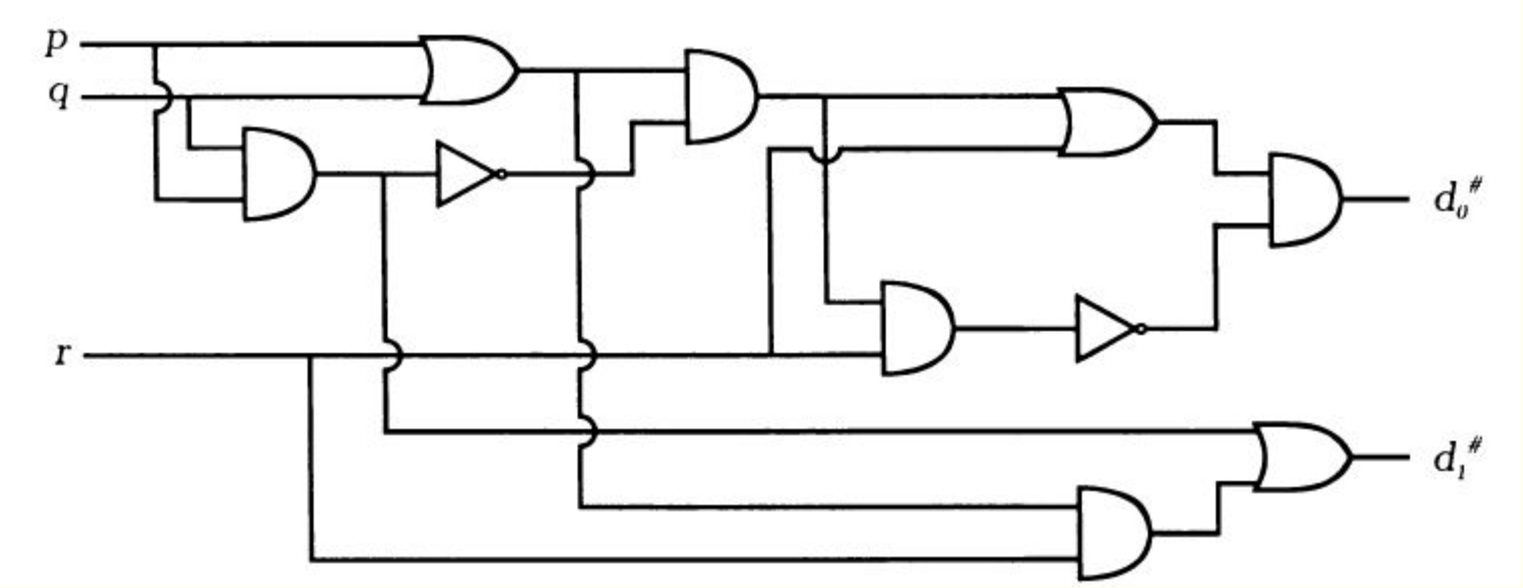
- Полный сумматор складывает три одноразрядных двоичных числа.
- Следовательно его можно использовать для сложения двух одноразрядных чисел **p,q** с тем числом которое переносится **r**. Такое действие лежит в сложении любых чисел.
- Представим результат сложения в таблице, где где **d0# -** первый разряд **d1#-** второй разряд (разряд переноса)

- Заметим, что первый разряд **d0#** представляет собой результат сложения первого разряда сложения чисел p,q с числом r.
  Для **d1# построим СДНФ** $pqr \vee pq\neg r \vee p\neg qr \vee \neg pqr$.
  Используя законы булевой алгебры получим **d1#** = $pq \vee (p \vee q)r = d1 \vee (p \vee q)r$

Проектирование полного сумматора

- **$d0^{\#}$** представляет собой результат сложения первого разряда d0 сложения чисел p,q с числом r.
- **$d1^{\#}$ = d1 ∨** (p ∨ q) r





Конечнозначные логики

- Функция , отображающая n-мерный k-значный кортеж $(\sigma_1, \sigma_2, \ldots, \sigma_n),\ \sigma_i \in \{0, 1, \ldots, k-1\},\ i = 1, 2, \ldots, n$ в множество $\{0, 1, ..., k - 1\}$, называется **функцией k-значной логики.**

- Будем задавать функцию k-значной логики с помощью таблицы истинности (одномерной таблицы), число строк которой равно k^n, или двумерной таблицы, число клеток которой равно k^2.

Конечнозначная *алгебра Вебба*

Конечнозначная **функция Вебба**

$$y = x_a \circ x_b = \max(x_a, x_b) + 1 \pmod{k}$$

Пример: k=3

x_a	x_b	y
0	0	1
0	1	2
0	2	0
1	0	2
1	1	2
1	2	0
2	0	0
2	1	0
2	2	0

x_a	x_b		
	0	1	2
0	1	2	0
1	2	2	0
2	0	0	0

Конечнозначная *алгебра Вебба*

$$A_B = \langle M, o \rangle, M = \{0,1,2,\ldots,k-1\},$$

Конечнозначная алгебра Поста

$$A_{\Pi} = \langle M, \vee, \sim \rangle, M = \{0,1,2,\ldots,k-1\},$$

- где дизъюнкция $x_a \vee x_b = \max(x_a, x_b)$
- Цикл $\tilde{x} = x + 1 \pmod{k}$
- В случае k=2, цикл и дизъюнкция соответствуют булевым операциям.

Алгебра Россера—Тьюкетта

$$A_{PT} = \langle M, \vee, \&, j_i, i \rangle, \;\; M = \{0, 1, 2, \dots, k-1\}, 0 \le i \le k-1,$$

- конъюнкция

$$x_a \& x_b = \min(x_a, x_b)$$

- характеристические функции

$$j_i(x) = \begin{cases} k-1, & x = i \\ 0\quad, & x \ne i \end{cases}$$

Исчисление высказываний

- **Математическая логика** состоит из двух разделов: логики высказываний и логики предикатов.
- **Логика высказываний** может быть представлена двумя подходами: алгеброй высказываний и исчислением высказываний.
- **Исчисление высказываний** является аксиоматической (формальной) теорией.

Аксиоматическая (формальная) теория Т

считается определенной, если выполнены следующие условия:

- Задано некоторое ***счетное множество*** символов — **алфавит теории *Т***.
- **Определено** подмножество правильно построенных последовательностей символов теории *Т*, называемых ***формулами теории Т*** *(язык теории)*.
- **Выделено** некоторое множество формул, называемых ***аксиомами теории Т***;
- **Задано** конечное множество R_1, R_2, ..., R_n отношений между формулами, называемых ***правилами вывода***.

Задание Исчисления Высказываний алфавит и формулы

- **Алфавит** исчисления высказываний состоит из переменных высказываний (пропозициональных букв): A, B, C …, знаков логических связок $\vee$, &, ¬, → и скобок (,).
- **Формулы:**

а) переменное высказывание (пропозициональная буква) есть формула;

б) если A и B — формулы, то $(A \vee B)$, $(A \& B)$, $(A \rightarrow B)$ и $(\neg A)$ также формулы;

в) выражение является *формулой* тогда и только тогда, когда это может быть установлено с помощью пп. а) и б)

- Существует **несколько систем аксиом** исчисления высказываний.

Аксиомы исчисления высказываний (система аксиом 1)

- I1. $A \to (B \to A)$;
- I2. $(A \to B) \to ((A \to (B \to C)) \to (A \to C))$;
- I3. $(A \mathbin{\&} B) \to A$;
- I4. $(A \mathbin{\&} B) \to B$;
- I5. $A \to (B \to (A \mathbin{\&} B))$;
- I6. $A \to (A \vee B)$;
- I7. $B \to (A \vee B)$;
- I8. $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$;
- I9. $(A \to B) \to ((A \to \neg B) \to \neg A)$;
- I10. $\neg\neg A \to A$

Аксиомы исчисления высказываний (система аксиом II)

- II1. $A \to (B \to A)$;
- II2. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$;
- II3. $(\neg A \to \neg B) \to ((\neg A \to B) \to A)$

$A \vee B$ заменяет $\neg A \to B$,

$A \;\&\; B$ заменяет $\neg(A \to \neg B)$

Система аксиом III (дизъюнктивный базис Буля)

- III1. $A \vee A \to A$
- III2. $A \vee B \to B \vee A$;
- III3. $A \to A \vee B$;
- III4. $(B \to C) \to (A \vee B \to A \vee C)$,

где запись $\alpha \to \beta$ эквивалентна записи $\neg\alpha \vee \beta$.

Свойства систем аксиом

- В каждой системе аксиом, аксиомы не могут быть получены друг из друга по правилам вывода
- Все системы аксиом эквивалентны так как любая система аксиом может быть выведена из другой системы аксиом по правилам вывода

Правила вывода исчисления высказываний

- **правило подстановки:** если α — выводимая формула, содержащая букву *A* (обозначив это как α(*A*)), то выводима формула α(β), получающаяся из α заменой всех вхождений *A* на произвольную формулу b

$$\beta : \frac{\alpha(A)}{\alpha(\beta)}$$

- **правило заключения** (modus ponens): (если α → β и α — выводимые формулы, то β также выводимая формула)

$$\frac{\alpha, \alpha \rightarrow \beta}{\beta}$$

Выводимость формул

- Если формулы F_1, …, F_n, G находятся в отношении R, то формула G называется **непосредственно выводимой** из F_1, …, F_n по правилу R. формулы F_1, …, F_n, называются *посылками* правила R, а G — его следствием или *заключением*.
- ***Выводом в T*** формулы G из формул A_1, …, A_n, называется всякая последовательность F_1, F_2, ..., F_m формул такая, что F_m = G, а для любого i формула F_i есть либо аксиома теории T, либо одна из исходных формул A_1, …, A_n, либо выводима из формул F_1, …, F_{i-1} по одному из правил вывода.
- Если существует вывод G из A_1, …, A_n, то говорят, что **G *выводима из A_1, …, A_n*** т.е. является теоремой формальной теории T. Этот факт обозначается $A_1, \dots, A_n \vdash G$.
- Формулы A_1, …, A_n называются **гипотезами** или *посылками вывода*. Переход в выводе от F_{i-1} к F_i называется ***i-м шагом вывода.***

Разрешимые и неразрешимые формулы

- В общем случае может не существовать эффективной процедуры, с помощью которой можно определить по данной формуле, существует ли ее вывод в теории *T*.

- Формула, для которой такая процедура существует, называется **разрешимой** в этой теории, в противном случае — **неразрешимой**.

- Иначе говоря, для неразрешимых формул нельзя построить алгоритм, который ответит на вопрос: является ли формула теоремой.

Свойства Исчисления Высказывания

Полнота. Имеют место следующие метатеоремы.

Теорема 1: Всякая теорема исчисления высказываний (Т) есть тавтология (т.е. тождественно истинная формула)

Теорема 2: Всякая тавтология является теоремой исчисления высказываний.

Таким образом, теоремами теории Т являются тождественно истинные формулы и только они.

Формальная теория Т (исчисление высказываний) является полной теорией

Свойства Исчисления Высказывания

Непротиворечивость: **в теории Т нет одновременной выводимости теоремы и ее отрицания.**

Из теоремы 1 следует, что теория Т непротиворечива.

Действительно, любая теорема в Т есть тождественно истинная формула (тавтология). Ее отрицание не есть тавтология.

Свойства Исчисления Высказывания

Разрешимость

Теория Т разрешима как формальная теория.

Алгоритм, который определяет для любой формулы теории, является ли эта формула теоремой теории, может состоять в вычислении истинностных значений формулы в каждой интерпретации. Принципиально это выполнимо за конечное время в силу конечности числа интерпретаций и числа операций, присутствующих в формуле.

Логика предикатов

Логика предикатов

- **Логика предикатов** представляет собой развитие логики высказываний.
- С помощью логики высказываний можно описать и исследовать структуру сложных высказываний. Установить их истинность или ложность в зависимости от истинностных значений входящих в нее простых высказываний
- **Предикат используется** для описания внутренней логической структуры простых высказываний (т.е. высказываний не содержащих связок)

Определение предиката

- **Предикат** – повествовательное предложение, содержащее предметные переменные, определенные на соответствующих множествах

- **Например:** P(x,y)=«Студент **x** на экзамене получил отметку **y»**. Переменные **x, y** являются предметными и принадлежат следующим множествам x={Иванов, Петров, Сидоров}, y={2,3,4,5}.

- При замене переменных конкретными значениями (элементами множеств) **предикат обращается в высказывание** т.е. принимает значение «истинно» или «ложно». Например: «Студент Иванов на экзамене получил отметку 5»

- **Предикатом** $P(x_1, \ldots, x_n)$ (от позднелат. *praedicatum* — сказанное) зависящим от n переменных называется функция P: $M_1 \times M_2 \times \ldots \times M_n \to B$, где Mi — произвольные множества, а B — двоичное множество. При этом предполагается, что $xi \in Mi$.
- *$M_1, M_2, \ldots, M_n$ – предметные области предиката, $x_1, \ldots, x_n$ – предметные переменные предиката.*

Предикатные формулы

- Под 0-местным предикатом понимается произвольное высказывание.
- Выражения $P(a_1, \ldots, a_n)$, где $a_1, \ldots, a_n \in M$, будем понимать как высказывание «$P(a_1, \ldots, a_n) = 1$» или как «$P(a_1, \ldots, a_n)$ истинно», а выражение $P(x_1, \ldots, x_n)$, где $x_1, \ldots, x_n$ — переменные, как переменное высказывание
- !!! $P(x_1, \ldots, x_n)$ — это логическая переменная, а $x_1, \ldots, x_n$ — нелогические переменные.
- Поскольку предикаты принимают два значения и интерпретируются как высказывания, из них можно образовывать выражения логики высказываний, т.е. формулы вида

$$P_1(x_1, x_2) \vee \overline{(P_2(x_3, x_4) \,\&\, P_1(x_2, x_4))}$$

- В приведенном примере предикатная формула может рассматриваться и как составная булева формула зависящая от 3 переменных (P1(x1,x2), P2(x3,x4), P1(x2,x4)) , и как составной 4-местный предикат зависящий от переменных x1, x2, x3, x4

Логика предикатов – общие характеристики

- Исследование предикатных формул и способов установления их истинности является **предметом изучения** логики предикатов
- Логика предикатов вместе с входящей в нее логикой высказываний является **основой логического языка математики**. С ее помощью удается формализовать и точно исследовать основные методы построения математических теорий.
- Логика предикатов, как и логика высказываний, может быть построена в виде **алгебры предикатов и высказывания предикатов**.
- Исследование формул алгебры предикатов, а так же выполнение их преобразований значительно проще, чем в исчислении предикатов.
- **Ограничения в использовании алгебры предикатов** обусловлено тем, что предметные области (множества на которых определены предметные переменные) могут быть бесконечными. В таких случаях, стандартный метод проверки истинностных значений предикатов, требующий подстановки всех значений предметных переменных, не может быть применен.
- Однако, **в практических ситуациях** при описании реальных систем, в качестве предметных областей, как правило, используются конечные множества. Кроме того, в некоторых случаях бесконечных предметных областей (например, бесконечные множества чисел) проверка истинности предикатных формул основывается на знаниях полученных из других дисциплин (например, арифметики)