МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГУВПО «БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «АСУ»

Конспект лекций
по дисциплине
«МАТЕМАТИЧЕСКИЕ МОДЕЛИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И УПРАВЛЕНИЯ»

студент гр. АСОИ-081
Дегтярев А.Н.
преподаватель
Якимов А.И.

Могилев, 2011

Элементы теории множеств и отношений

1. **Основные определения**
2. **Конечные и бесконечные множества**
3. **Способы задания множеств**
4. **Счетные и несчетные множества**
5. **Операции над множествами**

Способы задания:

1. Перечисление $A=\{a_1, a_2,..., a_n\}$, $|A|$- мощность
2. С помощью характеристического уравнения $A=\{x|p(x)=1\}$
3. С помощью процедуры $A=\{x|x:=f\}$

Операции над множествами

1. Объединение $\cup$

$$A \cup B = \{x|x \in A \lor x \in B\}$$

2. Перечисление

$$A \cap B = \{x|x \in A \,\&\, x \in B\}$$

3. Разность \

$$A \backslash B = \{xx \in A \,\&\, x \notin B\}$$

4. Дополнение

$$\bar{A} = \{x|x \notin A\}$$

5. Симметричная разность $\oplus \div$

$$A \oplus B = (A \cup B)(A \cap B)$$

Следует провести аналогию между операциями над множествами и логическими операциями:

1. $\vee$ - дизъюнкция
2. $\wedge$ - конъюнкция
3. $\backslash$ - импликация
4. - отрицание

Соотношения между множествами и составными высказываниями, если рассматривается несколько высказываний, то сопоставив каждое из этих высказываний некоторое множество можно вполне логичным путём:

Сначала мы образуем множество всех логических возможностей для рассмотрения высказываний и назовём его универсальным множеством

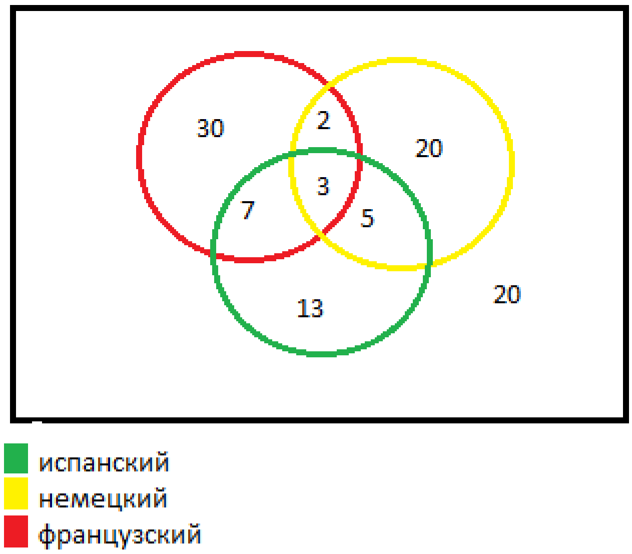
Затем к каждому высказыванию подставим в соответствии подмножество тех логических возможных универсальных, для которых это высказывание истинно.

Если X и Y высказывания, то $X \vee Y$, $X \wedge Y$ – так же высказывания и следовательно они должны иметь множество истинности. Что бы найти множество истинности всех $X \vee Y$, заметим, что высказывание истинно, когда истинно X, истинно Y или оба истинны. Таким образом высказывания $X \vee Y$ поставить в те логические возможности, которые лежат во множествах A или B, или в них обоих. Иным образом говоря множеству $X \vee Y$ следует поставить в соответствие $A \cap B$.

С другой стороны следует высказывание $X \wedge Y$ – истинно только тогда, когда истинно X и истинно Y. Тогда высказывание $X \wedge Y$: $A \cap B$.

Всё проиллюстрировано на диаграмме Венна:

Связь между высказыванием и его множеством истинности создаёт возможность перевода любой задачи, относящейся к соответствующему высказыванию, в задачу теории множеств.
Возможно также и обратное т.е. если поставленная задача в теории множеств, то универсальное множество можно себе представить как некоторое множество логических возможных подмножеств которые являются множеством истинности некоторого высказывания. Следовательно задачу, относящуюся к множественным, можем перевести к составу высказываний.
После рассмотрения множеств истинности состав высказываний образуется связующими дизъюнктами, конъюнктами, отрицанием; все остальные связи выводятся из $\wedge$, $\vee$, ⍰.
Например:
Известно, что $X \to Y = \overline{X} \vee Y$, поэтому множество истинно для $X \to Y$ будет тоже самое $X \to Y : A \cap B$
$A \setminus B = A \cap \overline{B}$.
Опрос 100 студентов показал следующие результаты, по количеству студентов изучающих иностранные языки: испанский - 28, немецкий - 30, испанский и немецкий - 8, испанский и французский - 10, немецкий и французский - 5, все 3 языка - 3. Сколько студентов не изучают ни одного языка.



Пусть мы рассмотрим высказывание X - истинное во всех случаях. Какую область занимает множество истинности – универсум.

Кортежи и операции над ними.
Пусть дано множество $X_1, X_2,\dots, X_n$, кортежем длинны n, составим из элементов этих множеств называется конечная последовательность $\alpha=\langle x_1,x_2,\dots,x_n\rangle$, где $x_1 \in X_1$, $x_2 \in X_2,\dots, x_n \in X_n$
Элемент x_i – называется координатой, или i-тая компонента кортежа α.
Два кортежа равны только в том случае, когда они имеют одинаковые длину, причём их координаты, стоящие на позициях с одинаковыми номерами, равны, т.е. кортеж $\alpha=\langle x_1,x_2,\dots,x_n\rangle$ и $\beta=\langle y_1,y_2,\dots,y_m\rangle$: α=β, если n=m и $x_i=y_i$ (для всех i=1,n).
Кортежи длинной 2-упорядоченные пары, 3-упорядоченные тройками, длинны n-упорядоченными n-компонентами; кортеж длинны 0-пустой.
Основные отличия понятия кортежа множества:
1. В множестве порядок элементов не играет роли, а кортежи отличают порядком элементов, различны, даже в случае когда они имеют одинаковый состав. Пример:
$A_1=\{1;2;3\}$ $A_2=\{3;2;1\}$ => $A_1=A_2$
$\alpha=\langle 1,2,3\rangle$ $\beta=\langle 3,2,1\rangle$ => $\alpha\neq\beta$
2. Во множестве все элементы различны, а в кортеже координаты могут повторяться.
Декартово произведение:
$A_1*A_2=\{(1,3),(1,2),(1,1),(2,3),(2,2),(2,1),(3,3),(3,2),(3,1)\}$

Основные понятия и правила комбинаторики.

На практике часто встречаются задачи, где необходимо подсчитать число всех возможных способов размещения некоторых предметов конечного множества или число всех возможных способов выполнения определенного действия из конечного множества таких действий. Задачи такого типа называются ***комбинаторными,*** а методы их решения – ***методами комбинаторного анализа*.** Поскольку комбинаторика имеет дело с конечными множествами, то ее часто называют ***теорией конечных множеств*.**

Все разнообразие комбинаторных формул может быть выведено из двух основных утверждений, касающихся конечных множеств – правило произведения и правило суммы.

Правило произведения

Пусть необходимо выполнить последовательно k действий. Если

1-ое действие можно выполнить n_1 способами,

2-ое – n_2 способами,

…

k-ое - n_k способами,

то все k действий можно выполнить $n_1 \cdot n_2 \cdot \ldots \cdot n_k$ способами.

Правило суммы

Если элемент x может быть выбран m способами, а элемент y другими n способами, то выбор либо x, либо y может быть выбран m+n способами.

Будем рассматривать задачи, связанные с нахождением числа *способов построения кортежей* из элементов конечного множества. Простейшими такими кортежами являются размещения, перестановки, сочетания. Эти задачи образуют часть комбинаторики, называемой перечислительной комбинаторикой или теорией перечислений.

Пусть A – конечное множество, состоящее из элементов |A|=n.

1)Размещения без повторений.

Кортежи длины k ($1 \leq k \leq n$), состоящие из различных элементов n-элементного множества A, называются размещениями из n элементов множества A по k. Обозначение A_n^k. Схема выбора состоит в выборе k элементов из n-элементного множества без возвращения.

Тогда необходимо совершить k действий,причем первое действие можно совершить n способами, второе (n-1) способами и т.д., k-е действие n-(k-1) способами. Согласно комбинаторному правилу умножения, получим формулу: $A_n^k = n \cdot (n-1) \cdot \ldots \cdot (n-k+1)$.

Если умножить и разделить полученное выражение на (n-k)!, получим:

$$A_n^k = \frac{n!}{(n-k)!},$$

Где $n! = n \cdot (n-1) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$ и называется факториалом n (читается n-факториал).

Пример:

Дано множество A {1, 3, 5}. Выпишем все размещения из трех элементов по два: <1, 3>, <1, 5>, <3, 5>, <3, 1>, <5, 3>, <5, 1>.

Число этих размещений можно найти по формуле:

$$A_3^2 = \frac{3!}{(3-2)!} = \frac{1 \cdot 2 \cdot 3}{1!} = 6.$$

2)Перестановки без повторений

Пусть есть n-элементное множество A, будем строить из этого множества размещения в виде кортежей длины n. Эти размещения будут отличаться друг от друга только порядком, поскольку в каждом из них встречаются по одному разу все элементы множества A. Такие размещения называются перестановками и обозначаются P_n(буква P от англ. Слова permutation – перестановка). Поскольку $P_n = A_n^n$, то число перестановок вычисляется по формуле $P_n = n!$.

Пример:

Сколько трехзначных чисел можно составить из цифр 1, 2, 3, если каждая входит в число только один раз.

Составим такие числа: <1, 2, 3>, <1, 3, 2>, <2, 1, 3>, <2, 3, 1>, <3, 1, 2>, <3, 2, 1> - то есть шесть чисел.

С помощью введенной формулы можно сразу определить число перестановок, не выписывая их:

$P_3 = 3! = 1 \cdot 2 \cdot 3 = 6.$

2) Сочетания без повторений

Из n-элементного множества A будем строить упорядоченные множества длины k ($1 \leq k \leq n$), не учитывая порядок элементов, т. е. размещения с одними и теми же элементами, расположенными в разном порядке, будем считать равными.

Такие размещения называются сочетаниями и обозначаются C_n^k(буква C от англ. Слова combination - комбинация).

Число сочетаний из n элементов по k меньше числа размещений из n элементов по k в P_k раз, т. е.

$$C_n^k = \frac{A_n^k}{P_k}.$$

Используя это утверждение, выведем формулу для вычисления числа сочетаний:

$$C_n^k = \frac{A_n^k}{P_k} = \frac{n!}{(n-k)! \cdot k!} = \frac{n(n-1)\dots(n-k+1)}{k!}.$$

Из этой формулы непосредственно вытекает, что $C_0^0 = C_n^0 = C_n^n = 1$;

$C_n^1 = n$; $C_n^k = C_n^{n-k}$.

Пример:

Какие парные сочетания можно составить из цифр 1, 3, 5 и сколько их?

Выпишем эти сочетания: <1, 3>, <1, 5>, <3, 5>,

т. е. $C_3^2 = \frac{3!}{1! \cdot 2!} = 3$ или $C_3^2 = C_3^{3-2} = C_3^1 = 3$.

Можно доказать следующие тождества:

а) $C_n^k \cdot C_k^r = C_n^r \cdot C_{n-r}^{k-r}, где\ 0 \leq r \leq k \leq n.$

$$C_n^k \cdot C_k^r = \frac{n! \cdot k!}{(n-k)! \cdot k! \cdot (k-r)! \cdot r!} = \frac{n!}{(n-r)! \cdot r!} \cdot \frac{(n-r)!}{(n-k)! \cdot (k-r)!} = C_n^r \cdot C_{n-r}^{k-r}.$$

б) $C_{n-1}^k + C_{n-1}^{k-1} = C_n^k$;

$$C_{n-1}^k + C_{n-1}^{k-1} = \frac{(n-1)!}{(n-k-1)! \cdot k!} + \frac{(n-1)!}{(n-k)! \cdot (k-1)!} = \frac{(n-1)! \cdot (n-k+k)}{(n-k)! \cdot k!} = \frac{n!}{(n-k)! \cdot k!} = C_n^k.$$

Комбинации элементов с повторениями

Все приведенные формулы справедливы в том случае, когда *n* элементов множества *A* различны. Если же некоторые элементы повторяются, то в этом случае рассматриваются комбинации **с повторениями,** число которых вычисляется по другим формулам.

3) Размещения с повторениями

Размещениями с повторениями из n элементов по k называются кортежи длины k, составленные из n-элементного множества A. Число этих кортежей обозначают $\bar{A}_n^k$. Черта указывает на возможность повторения элементов $\bar{A}_n^k = n^k$.

Пример:

Сколько пятизначных номеров можно составить из элементов множества {1, 2, 3, 4, 5, 6, 7, 8, 9}?

Такими номерами являются кортежи длины 5, составленные из девятиэлементного множества, где схема выбора состоит в выборе 5 элементов из девятиэлементного множества с возвращением, т. е. для каждого из пяти элементов есть девять способов выбора, т. е. $\bar{A}_9^5 = 9^5 = 59049$.

4) Перестановки с повторениями

Перестановкой с повторениями состава (n_1 ,…, n_k) из букв (a_1,…,a_k) называют любой кортеж длины $n=n_1+n_2+\ldots+n_k$, в который a_1 входит n_1 раз, a_2 входит n_2 раз, …, a_k - n_k раз. Число таких перестановок обозначают

$$P(n_1, n_2, \dots, n_k) = \frac{n!}{n_1! \cdot n_2! \cdot \ldots \cdot n_k!} = \frac{(n_1 + n_2 + \cdots + n_k)!}{n_1! \cdot n_2! \cdot \ldots \cdot n_k!}.$$

5) Сочетания с повторениями

Пусть имеются предметы n видов, и из них составляется набор, содержащий k элементов, т. е. различными исходами будут всевозможные наборы длины k, отличающиеся составом, и при этом отдельные наборы могут содержать повторяющиеся элементы. Такие наборы называются сочетаниями с повторениями, а их общее число определяется формулой:

$$\bar{C}_n^k = C_{n+k-1}^k.$$

Пример:

Сколько наборов из 7 пирожных можно составить, если в продаже имеются 4 сорта?

Искомое число равно $\bar{C}_4^7 = C_{4+7-1}^7 = C_{10}^3 = \frac{10\cdot 9\cdot 8}{1\cdot 2\cdot 3} = 120.$

Отображения и функции. Обратная функция. Понятие функционала

Говорят, что между множествами E, F определено соответствие Г, если задано некоторое произвольное подмножество некоторого декартового произведения $E\times F$.

Множество E называют областью определения, множество F- областью значений соответствия F.

Соответствие, обратное Г, обозначим как Γ^{-1}, где F- область определения, E-область значения Γ^{-1}.

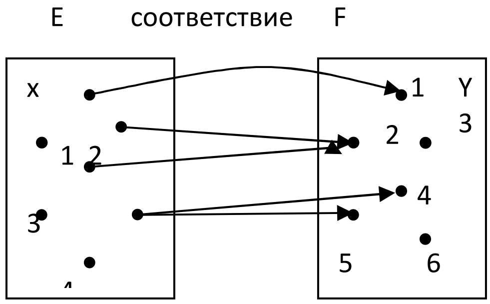
$\Gamma^{-1} \subset F \times E.$

Отображением множества E на множество F называется такое соответствие, которое каждому элементу $\forall x \in E$ сопоставляет по крайней мере один элемент из множества F $\exists y \in F$.

Тогда элемент y называется образом элемента x, а x- прообразом элемента y.

Отображение E в F обозначим $f: E \to F \mid f -$ имя отображения.

Пример 1

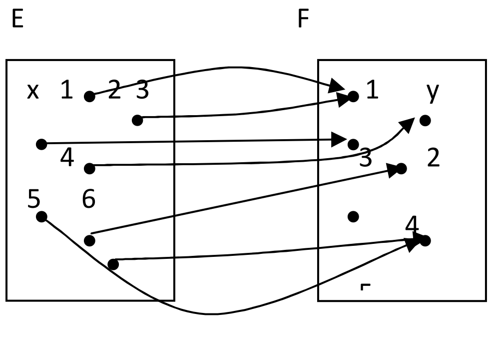


$\Gamma \subset E \times F.$

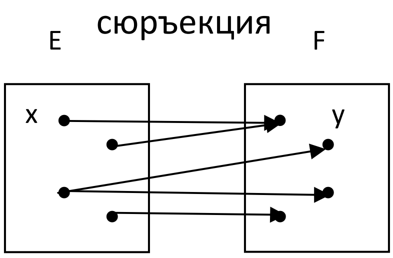
Г={(1,1),(2,2),(4,2),(6,4),(6,5)}.

/ $E \times F$ /=42

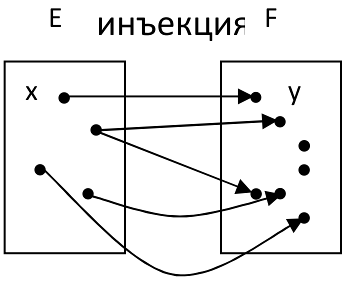
Пример 2



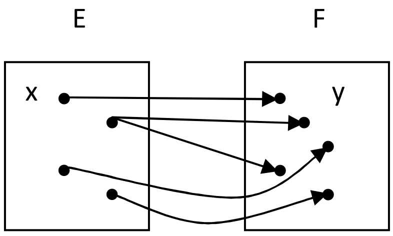
Отображение E на F называется сюръективным или сюръекцией (наложением), если каждый элемент $y \in F$ есть образ по крайней мере одного элемента $x \in E$, т.е.
$\forall y \in F \ \exists x \in E$ y=Г(x).



Отображение E в F называется инъективным или инъекцией (вложением), если каждый элемент $y \in F$ есть образ только одного элемента $x \in E$,либо вообще не имеет прообраза.

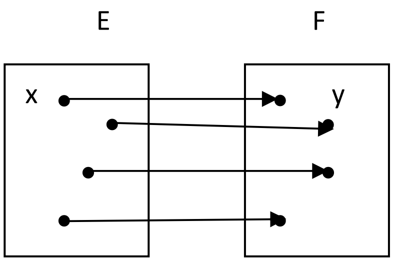


Если отображение является одновременно и сюръекцией, и инъекцией, то оно называется биективным или биекцией.
В этом случае каждый элемент F является образом некоторого и причем единственного элемента из E.

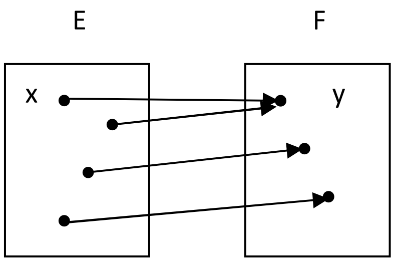


Соответствие, при котором каждому x из множества E сопоставляется один и только один элемент y из множества F, называется функциональным соответствием или функцией.
Если функциональное соответствие не является отображением, т.е. в E существуют элементы, не имеющие образов в F, то оно называется частично определенной функцией.
Функциональное отображение является полностью определенной функцией или просто функцией.
Функция может быть инъективной, биективной и сюръективной.

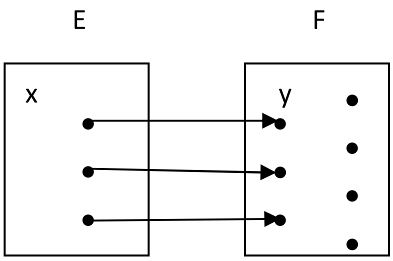
Функциональная биекция



Функциональная сюръекция



Функциональная инъекция



Отображение множества E во множество E, определенное равенством f(x)=x, называется тождественным отображением.

Пример 1. Если *E* - множество ограниченных кривых на плоскости, то функция вычисления длины кривой есть сюръекция $f: E \rightarrow R^+$

Пример 2. Отображение $f: R \rightarrow R$, такое, что *f(x) = 2x+3*, т.е. прямая, есть биекция.

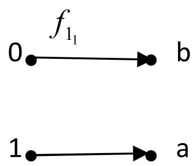
Пример 3. Отображение $g: N \rightarrow R$, такое, что $g(x) = \pm\sqrt{x}$, является биекцией, но не является функциональным отображением.

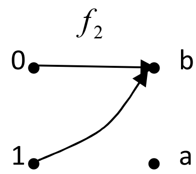
Пример 4. Соответствие $f: R \rightarrow R$, такое, что *f(x)=x/sin(x)*, является частично определенной функцией: для sin(x)= 0 значение функции не определено.

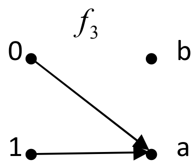
Пример 5. Индексирование элементов некоторого множества есть однозначное отображение этого множества в подмножество натуральных чисел N.

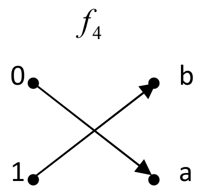
Если E и F- два множества, то можно говорить о некотором новом множестве- множестве функциональных отображений E в F.

Пусть E={0,1}; F={b,a}.

1) 

2) 

3) 

4) 

Множество отображений E в F можно задать в виде таблицы:

x	f_1	f_2	f_3	f_4
0	b	b	a	a
1	a	b	a	b

Множество всех функциональных отображений $f: E \to F$ называется степенью множеств и обозначается F^E.

Свойства функциональных отображений

Пусть f- функциональное отображение E в F и A- подмножество E. Обозначим через f(A) подмножество F, образованное из всех элементов $f(x) \mid x \in A$.

Подмножество f(A) называется образом подмножества A при отображении $f : A \to f(A)$ или сужением функции $E \to f(E)$.

Например, $f(\emptyset) = \emptyset$.

Исходя из отображения f определим некоторое отображение $A \to f(A)$

1.Если $A \subset B$, то $f(A) \subset f(B)$

$2. f(A \cup B) = f(A) \cup f(B)$

$3. f(A \setminus B) = f(A) \setminus f(B)$

$4. f(A \cap B) = f(A) \cap f(B)$

5.Если $A \subset B$, то $f^{-1}(A) \subset f^{-1}(B)$

$6. f^{-1}(A \cup B) = f^{-1}(A) \cup f^{-1}(B)$

$7. f^{-1}(A \setminus B) = f^{-1}(A) \setminus f^{-1}(B)$

$8. f^{-1}(A \cap B) = f^{-1}(A) \cap f^{-1}(B)$

Пусть даны три множества: E, F, G и заданы отображения $f : E \to F$; $g : F \to G$. Тогда композицией отображений $g \circ f : E \to G$ называется отображение E в G, которое определяется формулой $g \circ f = g(f(x))$.

Запись $g \circ f$ производится в порядке, обратном тому, в котором производятся операции f и g.

Т.о. принято правило в математике, согласно которому композицию $g \circ f$ надо начинать с выполнения операции f, которая расположена справа.

Композиция отображений ассоциативна, т.е. если $f : E \to F$, $g : F \to G$, $h : G \to H$, то $h \circ (g \circ f) = (h \circ g) \circ f = h \circ g \circ f$.

Композиция отображений некоммутативна, т.е. $g \circ f \neq f \circ g$.

Отображение f имеет обратное отображение тогда и только тогда, когда f- биекция.

$f : E \to F \ \exists \ f^{-1} : F \to E$.

Если f и g –функциональные отображения: либо сюръекции, либо инъекции, то можно доказать ряд убеждений о свойствах композиций этих отображений.

$g \circ f$	О	С	И	Б
О	О	О	О	О
С	О	С	О	С
И	О	О	И	И
Б	О	С	И	Б

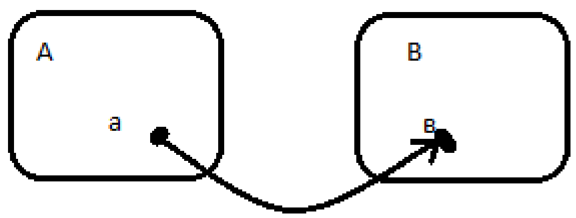
Эти свойства отображены в таблице, где О-отображение, С-сюръекция, Б-биекция.

Отношения. Свойства отношений.

Пусть А и В множества. Бинарным отношением R из множества А в В называется подмножество прямого произведения множеств А и В:RcA+B.

Для бинарных отношений обычно используют симплексную форму записи:

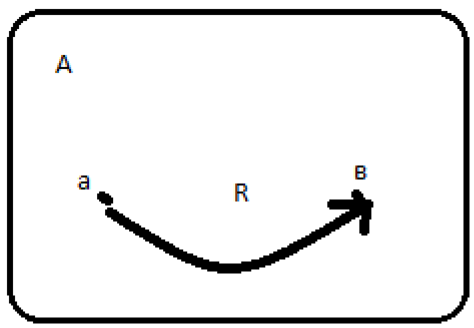
aRb|a $\in$A, b $\in$B|



Запись aRb формально описывается следующим образом:

$(a,b) \in R \subset AxB \mid a \in A, b \in B \mid$

Если множества А и В совпадают, то говорят, А=В | А$\subset$ВΔ В$\subset$ А| что R есть R$\subset$AxA



Другая запись –R $\subset$H^2

Пусть А и В множества, тогда введём следующие понятия:

1. Обратное отношение (R^{-1}={(a,b)|(b,a) $\in$R})
2. Дополнительные отношения: R={(a,b)|(a,b) $\notin$R}
3. Тождественное отношение I={(a,a)|a$\in$A& a$\in$B}
4. Универсальное отношение $U = \{(a,b) \mid a \in A \,\&\, b \in B\}$

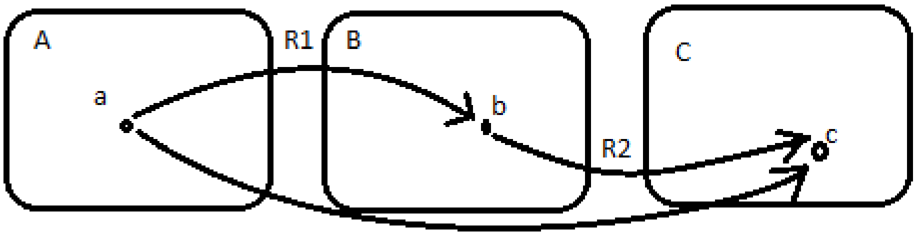
Обобщающим понятием отношений является n-местное отношение. Оно определяется следующим образом: R={(a_1, a_2, a_2,…, a_n)($a_1 \in A_1$,$a_2 \in A_2$,…, $a_n \in A_n$)}

множества A_1,A_2,…,$A_т$ не обязательно различны.

Пример бинарного отношения где n=4:

Фамилия	Год рождения	Место жительства	Образование
Петров …	1991 …	Киев …	Высшее …

Пусть R_1 отношение на множествах A и B : $R_1 \subset AxB$ и $R_2 \subset BxC$, тогда $R = R_1 * R_2 = \{(a,c) \mid a \in A \,\&\, c \in C \,\&\, b \in B, aR_1b\partial bR_2C\}$



R=R1*R2

Свойства отношений:

Пусть R – отношение на множестве A, тогда:

1. R называется рефлективностью, если для любых $a \in A$,aRa
2. R называется антирефлективно, если для любых $a \in A$ не существует aRa
3. R называется симметричным, если $\forall a,b \in A$, aRb=>bRa
4. R называется антисимметричным, если $\forall a,b \in A$, aRb & bRa =>a=b
5. R антисимметрично, если $\forall a,b \in A$, aRb=> не существует
6. R транзитивно, если $\forall a,b,c \in A$, aRb&bRc =>aRc
7. R называется полным, если $\forall a,b \in A$, a≠b aRb V bRa

Пусть R отношение на множестве A:RcA^2 тогда

1. R рефлексивно тогда и только тогда, когда IcR
2. R симметрично $\leftrightarrow$ R=R^{-1}
3. R транзитивно $\leftrightarrow$ R⍰RcR
4. R антирефликтивно $\leftrightarrow$R⍰I=$\varnothing$
5. R антисимметрично $\leftrightarrow$ R⍰R^{-1}
6. R полно $\leftrightarrow$ RUR^{-1}UI=V

Отношения эквивалентности, порядка и доминирования. Алгебра отношений.

На практике встречаются отношения, которые обладают устойчивым набором тех или иных свойств. Таким отношениям даны специальные названия и они требуют дальнейшего изучения.

1. Отношения которые обладают свойствами рефлективности, симметрии и транзитивности называются отношениями эквивалентности. Обозначают такое отношение: R⍰i⍰ Пример: отношение «равенство чисел и множеств» является отношением эквивалентности.

Пример: отношение «быть студентом группы» - является отношением эквивалентности и разбивает множество всех студентов.

Пример: M={x|P(x)=1} |P(x) – быть студентом группы АСОИ -081;

$M=\{a_1,a_2,a_3,a_4,a_5\}$, |M|=5

$|2^M|=32$

R:=быть одного возраста

$M_{1/1990}=\{a_1,a_3\}$; $M_{2/1991}=\{a_2,a_4\}$; $M_{3/1992}=\{a_5\}$;

$M_{/R}=\{M_1,M_2,M_3\}$

2. Отношение параметра позволяет сравнить между собой элементы одного множества.

Определение: отношением на множестве M, удовлетворяющее свойствам рефлективности (P_1),P_2: антисимметричности, P_3 транзитивности, называется отношением порядка.

Свойства которые удовлетворяют это отношение приводят к понятию упорядоченности множества.

Непустое множество M, на котором задано бинарное отношение порядка удовлетворяющее свойствам P_1 P_2 P_3 называется частично упорядоченное множество.

Отношение порядка со свойствами P_1 P_2 P_3 называют относительностью нестрого порядка.

Если отношение порядка обладает свойствами P_2 P_3 и свойствами антирефлексивности, то его называют относительным строгого порядка.

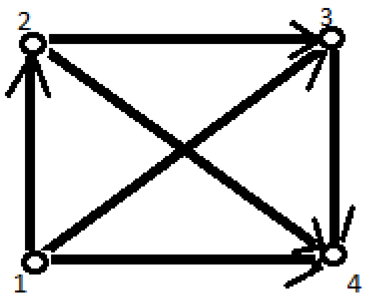
Обычно отношения порядка называют: ≤ (не строго), а строгого порядка - <.

Множество M, на котором задоно отношение порядка, обозначают: < M; ≤ > (не строгого порядка).

Упорядоченное множество (< M; ≤ >) с отношением порядка, удовлетворяет свойству линейности, называется линейно упорядоченным множеством (или цепью). В цепи каждые 2 произвольно взятые элементы сравнимы и нет несравнимых элементов. $a \le b$ (сравнимые елементы), $a||b$ (несравнимые элементы). Если в упорядоченном множестве M существует единственный, такой элемент a, такой что для любого $\forall a \in M$, $a \le x$, тогда элемент a называется наименьшим элементом упорядоченного множества.

Если в упорядоченном множестве M существует единственный элемент, что $\forall a \in M$ выполняется $x \le b$, то элемент b – наибольший элемент упорядоченного множества.

Упорядоченные множества с небольшим числом элементов наглядно иллюстрируют диаграммы, при этом элементы на диаграмме соответствуют точкам M={1,2,3,4}. Рассмотрим пример, где задано множество M и относительно – быть меньше.



Быть меньше

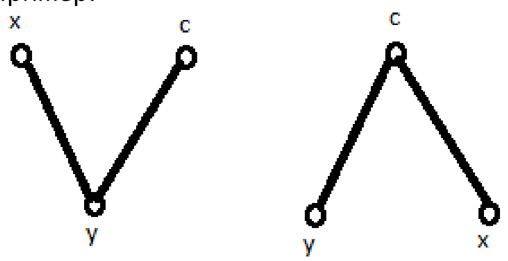
можно ещё матрицу отношения –быть меньше:

≤	1	2	3	4
1	0	1	1	1
2	0	0	1	1
3	0	0	0	1
4	0	0	0	0

Где 1 должно быть равно количеству дуг на диаграмме.

С относительным … непосредственно связано отношение доминирования (покрываем). Определение: пусть упорядоченное множество M соответствующего порядка, … элементов x и y, будем говорить, что x доминирует над y, если $y \le x$ и $x \ne y$и не существует такого элемента z где $z \in M$, $z \le x$ & $y \le z$.

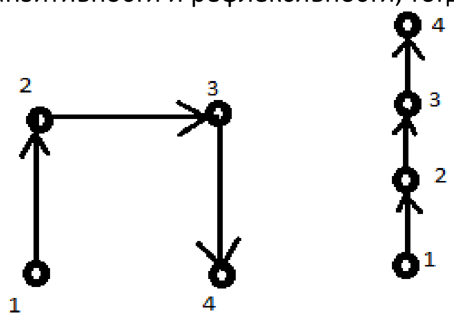
Пример:



С учётом отношения длин, упорядоченное множество можно отобразить в виде диаграмм.

Диаграммы отношения доминирования не содержат транзитивных дуг и петель, которые соответствуют отношению рефлексивности.

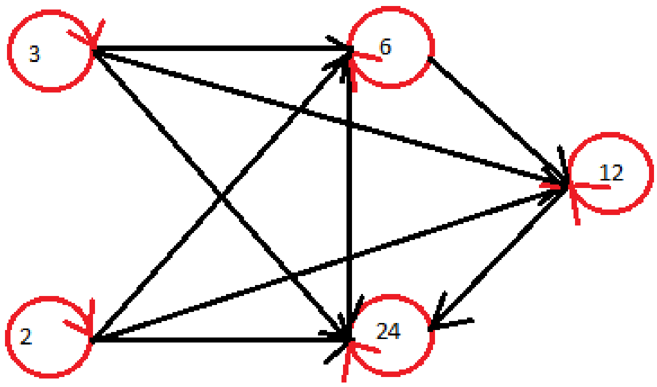
Построим диаграмму Хассе из множества «отношение – быть истинным», убрав отношение транзитивности и рефлексивности, тогда диаграмма Хассе имеет следующий вид: M={1,2,3,4}



Пример: задано множество с элементами A={2,3,6,12,24}, с определённым по нём соотношением порядка $x \le y$:"x делит y".

Построить диаграмму отношений и диаграмму Хассе:

≤	2	3	6	12	24
2	0	1	1	1	1
3	0	0	1	1	1
6	0	0	0	1	1
12	0	0	0	0	1
24	0	0	0	0	0



Так как отношения на множестве М задаются подмножеством $R \subseteq MxM$, то для них определены те же операции что и подмножествами. Возьмём 2 отношения α и β.

1. Перечисление отношений $\alpha \cap \beta$ называется отношением, определяемое перечислением соответствующих подмножеств. $\alpha \cap \beta = \{(x, y) \mid (x, y) \in \alpha \,\&\, (x, y) \in \beta\}$

$x(\alpha \cap \beta)y$ выполняется только тогда и только тогда, когда одновременно выполняются соотношения $x\alpha y \,\&\, x\beta y$

2. Объединением отношений $\alpha \cup \beta$ называется отношение, определённое объединением соответствующих множеств: $\alpha \cup \beta = \{(x, y) \mid (x, y) \in \alpha V (x, y) \in \beta\}$

$x(\alpha \cup \beta)y$ выполняется тогда и только тогда, когда $x\alpha y V x\beta y$

3. Разностью называют отношение, удовлетворяющее условию: $\alpha \mid \beta = \{(x, y)(x, y) \in \alpha \,\&\, (x, y) \notin \beta\}$

4. Дополнением называют отношение, удовлетворяющее следующему условию: $\overline{\alpha} = \nu \mid \alpha = \{(x, y) \mid (x, y) \notin \alpha \,\&\, (x, y) \in \nu\}$, если все отношения α и β рефлективны, то рефлективны следующие выражения: $\alpha \cap \beta$, $\alpha \cup \beta$, $\alpha o \beta$, α^{-1}

Если α и β симметричны, то симметричны и: $\alpha \cap \beta$, $\alpha \cup \beta$, α^{-1}

Если α и β антисимметричны, то антисимметричны и: $\alpha \cap \beta$, α^{-1}

Если α и β транзитивны, то транзитивны и: $\alpha \cap \beta$, α^{-1}

Тема 2. Элементы булевской логики и логико-алгебраические модели.

Модели алгебры и исчисления высказываний

Логика-это наука о правильных рассуждениях, о приемах и методах познания с помощью рассуждений.

Методологические основы формальной логики заключаются в выполнении следующих принципов:

1) принцип тождества. Истинность фактов, лежащих в основе высказываний и рассуждений, устанавливается на основе действительности, известных законов, наблюдений. Если истинность какого-то факта установлена, то она не подвергается сомнению и не изменяется.

2) принцип непротиворечивости означает, что, утверждая что-либо, нельзя это отрицать. Один и тот же факт не может быть и истиной, и ложью.

3) принцип исключенного третьего . Нельзя одновременно отвергать высказывание и его отрицание. Любое высказывание может быть либо истинным, либо ложным- третьего не дано.

4) Закон достаточного основания. Всякое высказывание должно быть обоснованно, т.е. истинность утверждения нельзя принимать на веру. Если утверждение выводится из каких-либо суждений, данных, фактов, т.е. оснований, то их должно быть достаточно для установления истинности.

Понятие высказывания. Виды высказываний.

Основным (неопределяемым) понятием математической логики является понятие «простого высказывания».

Под высказыванием обычно понимают всякое повествовательное предложение, утверждающее что-либо о чем-либо, и при этом мы можем сказать, истинно оно или ложно. Логическими значениями высказываний являются «истина» и «ложь».

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

Каждое простое высказывание обозначают символами латинского алфавита с индексами или без, которые называют пропозициональными символами.

Сложные высказывания составляют из простых с помощью союзов не,и,или,если…,то..,тогда и только тогда. Этим союзам соответствуют логические операции:

$\neg$ не (отрицание)

& и (конъюнкция)

$\vee$ или(дизъюнкция)

$\rightarrow$импликация

$\equiv$ операция эквивалентности.

Символы операций называют пропозициональными связками.

Истинность или ложность сложного высказывания зависит от истинности или ложности входящих в него простых высказываний.

Каждая логическая связка определяется своей таблицей истинности.

A	B	$A \& B$	$A \vee B$	$A \rightarrow B$	$A \equiv B$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Для операции НЕ:

A	$\neg A$
0	1
1	0

Пример: Пусть высказывание А – воробей- птица. $\neg A$ -не верно, что воробей –птица.

Конъюнкция истинна тогда и только тогда, когда истинны оба входящих в нее высказывания.

Для конъюнкции выполняется закон противоречия: $A \& \neg A \equiv F$.

Дизъюнкция истинна в любом случае, когда хоть одно высказывание истинно. Например, $2*2=4 \vee 2*2=5$ истинно.

Для дизъюнкции выполняется закон исключения третьего:

$A \vee \neg A \equiv T$.

Импликация формализует высказывание, в котором из посылки А (анцедент) следует заключение (консеквент) В.

Импликация истинна в случае, если из истинной посылки следует истинное заключение, и ложно, если из истинной посылки следует ложное заключение.

Если посылка ложна, то из нее может следовать как истинное, так и ложное заключение, т.е. высказывание истинно в обоих случаях.

$F \rightarrow T = T$

$F \rightarrow F = T$, т.е. из лжи следует все, что угодно.

Эквивалентность утверждает равнозначность двух высказываний А и В. Она истинна тогда, когда истинностные значения А и В совпадают.

С помощью логических связок логические высказывания можно записать в виде формулы, которую называют пропозициональной формулой.

Каждая пропозициональная буква есть формула.

Если А и В- формулы, то формулами являются $\neg A$, $A \& B$, $A \vee B$. Других формул нет.

При записи формул приняты следующие соглашения:
1. внешние скобки можно опускать;
2. установлен приоритет операций $\neg \wedge \vee \rightarrow \equiv$.
Логические связки $\equiv$ и $\rightarrow$ вводятся для сокращения записи формул.
$A \rightarrow B = \neg A \vee B = \neg A(A \& \neg B)$.
$A \equiv B = (A \rightarrow B) \& (B \rightarrow A)$.

Приписывание пропозициональным буквам их истинностных значений называется *интерпретацией* формулы.
Множество всех интерпретаций формулы образует ее *таблицу истинности*.
Если выполнить отображения $0 \leftrightarrow F$ и $1 \leftrightarrow T$, то каждой пропозициональной связке будет соответствовать булева операция, а каждой формуле логики высказываний- булева формула, следовательно, логика высказываний является интерпретацией булевой алгебры.
В связи с этим в ней сохраняются все аксиомы и теоремы булевой алгебры.
Тождественно истинная формула называется **тавтологией**,
Тождественно ложная формула называется **противоречием**.
Формула, которая принимает истинное значение хотя бы на одной из своих интерпретаций, называется *выполнимой*.
Две формулы называют эквивалентными, если их таблицы истинности совпадают.
/=А означает, что формула А- тавтология.
Если /=А-тавтология, то $\neg A$ - противоречие.
Тавтологии являются выделенными формулами логики высказываний, т.к. формализуют правильные схемы рассуждений.
Например, больной либо умрет, либо выживет - тавтология.
Законы де Моргана также являются тавтологиями.
/=$\neg(A \vee B) \equiv (\neg A \& \neg B)$.
Неверно, что А или В совершили преступление.
Проблема разрешимости в алгебре высказываний заключается в том, чтобы отыскать эффективный алгоритм, с помощью которого для каждой формулы логики высказываний можно установить, является она тавтологией или нет.
Такая процедура для формул логики высказываний существует и является построением таблиц истинности.

Основные законы, определяющие свойства логических связок

1. законы идемпотентности конъюнкции и дизъюнкции.
$x \vee x \equiv x$

$x \wedge x \equiv x$

2. коммутативность дизъюнкции или конъюнкции.
$x \vee y \equiv y \vee x$
$x \wedge y \equiv y \wedge x$

3. дистрибутивность конъюнкции относительно дизъюнкции.
$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$
дистрибутивность дизъюнкции относительно конъюнкции
$x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$.

4. ассоциативность конъюнкции
$x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$.
ассоциативность дизъюнкции
$x \vee (y \vee z) \equiv (x \vee y) \vee z$.

5.Двойное отрицание
$\bar{\bar{x}} \equiv x$

6. Законы де Моргана

$\overline{x \vee y} \equiv \overline{x} \wedge \overline{y}$.

$\overline{x \wedge y} \equiv \overline{x} \vee \overline{y}$.

7. Закон склеивания (расщепления).

$(x \vee y) \wedge (x \vee \overline{y}) \equiv x$

$(x \wedge y) \vee (x \wedge \overline{y}) \equiv x$

8. законы поглощения

$x \vee (x \wedge y) \equiv x$

$x \wedge (x \vee y) \equiv x$

9.Действие с логическими константами

$x \vee 0 \equiv x$

$x \vee 1 \equiv 1$

$x \wedge \overline{x} \equiv 0$

$x \wedge 0 \equiv 0$

$x \wedge 1 \equiv x$

10.Закон исключения третьего

$x \vee \overline{x} \equiv 1$

11.Тождество

$x \equiv x$

12. Отрицание противоречия

$\overline{\overline{x} \wedge x} \equiv 1$

13.Контропозиция

$(x \to y) \equiv (\overline{y} \to \overline{x})$

14.Цепное исключение

$(x \to y) \wedge (y \to z) \equiv (x \to z)$

15.Противоположность

$(x \equiv y) \equiv (\overline{x} \equiv \overline{y})$

16. modus ponens

$(x \wedge (x \to y)) \to y \equiv 1$

Основные законы исчисления высказываний

Если A и B – формулы, то говорят, что B логике следует из A. Если на всех интерпретациях по которым A – истинно , B – также истинно A|B; A≥B.

В этом случае говорят, что логическое следование сохранит истинность.

Теорема: логичное следование (A|=B) выполнено тогда и только тогда, когда формула A→B – тавтология.

Определение: формула B логически следует из формул A_1 A_2 … A_n принимает истинное значение формула B так же принимает истинное значение A_1 A_2 … A_n |=B.

Теорема: логичное следование A_1 A_2 … A_n |=B тогда и только тогда, когда

A_1 &A_2 &… &A_n имплицирует (→) B.

Определение: Если A|=B&B|=A, то формула A логически эквивалентна формуле B (A<⍰>B,A⍰B). Если формула A<⍰>B, то A⍰B- тавтология. Теоремы следующего о тавтологии позволяет получить новые тавтологии из доказанных ранее.

Теорема: если A – тавтология и(A – имплицирует B) – тавтология, то B – тавтология: |=A&|=A→B, то |=B.

Доказательство: предположим, что на некоторой интерпретации |B|=f (ложь), тогда $|A \to B|=|A \to f|=T$ на той же интерпретации следовательно формула |A|=F, что невозможно, так как А – тавтология.
Это правило ещё называют правилом отделения и обозначают MP.
Правило MP говорит о том, что следствие В поступает при выполнении условия А, то есть при нечёткости посылки.
Правило подстановки: если А – тавтология, содержащая пропорциональные переменные $a_1,a_2,\dots,a_n$, то формула В, полученная из А подстановкой формул A_1 A_2 … A_n, вместо каждого вхождения $a_1,a_2,\dots,a_n$ соответственно так же будет тавтологией.
Правило эквивалентной записи: если В получается из А подстановкой формулы B_1 вместо одного или нескольких вхождений подформ A_1 в А, то $((A_1 \square B_1) \to (A \square B))$ тавтология. Следовательно, если A_1 и B_1 логически эквивалентны, то А и В так же логически эквивалентны. Иными словами, или есть тавтология А и в ней есть подформула A_1 , если заменить A_1 на эквивалентную ей B_1 , то полученная формула В будет эквивалентна А.

Булева алгебра. Основные операции булевой алгебры.
Простые высказывания можно рассматривать как некоторые переменные, которые принимают значения из множества $E=\{0;1\}$ такие переменные называют Булевыми переменными.
Тогда сложные высказывания можно рассматривать как функции, определённые на множестве $E^n=E*E*\dots*E$ (n – раз) булевых переменных принимающих значения из множества Е такие функции – булевы функции. Другими словами булева функция f рассматривается как отображение: $f: E^n \to E$
Определение 1. Всякую переменную, которая может принимать одно из двух возможных значений, обозначаемых 0 и 1, назовем булевой переменной.
Определение 2. Функция $F(x_1,\dots,x_{n)}$, принимающая значения на множестве $\{0,1\}$ вместе со своими переменными, называется булевой функцией.
Всякая булева функция $f: E^n \to E$.
Булева функция $f \in P_n$ существенно зависит от переменной x_i, если существует такой набор значений $a_1,\dots,a_{i-1},a_{i+1},\dots a_n$, что
$f(a_1,\dots,a_{i-1},0,a_{i+1},\dots a_n) \neq f(a_1,\dots,a_{i-1},1,a_{i+1},\dots a_n)$.
В этом случае x_i называют существенной переменной, в противном случае x_i называют несущественной (фиктивной) переменной.
Совокупность значений переменных булевой функции будем называть набором.
Пример
$E=\{0,1\}$.
$E^2 = E \times E = \{(0,0),(0,1),(1,0),(1,1)\}$.
Наборы значений, на которых функция принимает значение, равное 1, называются единичными наборами.(6)
Утверждение 1. Количество наборов булевой функции $F(x_1,\dots,x_{m)}$ от n переменных равно 2^n.
Количество булевых функций от n переменных равно 2 в степени 2^n .
Доказательство этого утверждения непосредственно следует из основных соотношений кардинальной математики. Действительно, множество всех наборов булевой функции от n переменных образовано декартовым произведением $\{0,1\}^n$, мощность которого равна 2^n. Множество всех булевых функций от n переменных есть множество отображений $\{0,1\}^n \to \{0,1\}$, мощность которого равна 2^{2^n} .
Существует 4 булевых функций от одной переменной и 16 булевых функций от двух переменных.
К функциям нуля переменных относятся константы 0 и 1. (6)

В таблице 2 приведены булевы функции от одной переменной:

f_1 - константа 0,

f_2 - тождественная функция $f(x) = x$,

f_3 - отрицание $f(x) = -x$,

f_4 - константа 1.

Таблица 2.

x	f_1	f_2	f_3	f_4
0	0	0	1	1
1	0	1	0	1

В таблице 3 приведены основные функции от двух переменных. Следует отметить, что любая булева функция может быть задана таблицей, в которых перечисляются всевозможные значения переменных и для каждого набора значений переменных указывается соответствующее ему значение булевой функции. Эта таблица – аналог таблицы истинности для высказываний.

Таблица 3.

x_1	x_2	F_0	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}
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

Здесь к функциям двух переменных относятся и те, которые зависят от нуля или одной переменной.

F_0 и F_{15} - соответственно, «константа 0» и «константа 1.

Функции $F_3 = x_1$ и $F_5 = x_2$ - функции повторения, соответственно аргументов x_1 и x_2. Функции $F_{12} = \neg x_1$ и $F_{10} = \neg x_2$ - функции инверсии соответственно аргументов x_1 и x_2. Эти функции считают функциями одного аргумента.

$F_1 = x_1 \,\&\, x_2 = x_1 \wedge x_2 = x_1 \cdot x_2 = x_1 x_2$ - конъюнкция. Допустимыми являются все виды приведенных обозначений, но поскольку эта функция называется логическое умножение, функция «И», то, как и в алгебре логики, знак умножения часто опускается.

$F_7 = x_1 \vee x_2 = x_1 + x_2$. Эта функция называется логическое сложение, функция «ИЛИ», но знак сложения «+» практически не используется. (3, с.5)

Для приведенных функций в таблице имеются инверсии. Так $F_{14} = \overline{F_1} = x_1 \mid x_2$ (штрих Шеффера),

$F_8 = \overline{F_7} = x_1 \downarrow x_2$ (стрелка Пирса).

$F_9 = x_1 \equiv x_2$ - эквивалентность.

$F_6 = x_1 \oplus x_2$ - сумма по модулю 2, сумма Жегалкина.

$F_{13} = x_1 \to x_2$ - импликация.

$F_{11} = x_2 \to x_1$ - конверсия.

Функции F_2 и F_4 логически несовместимы с импликацией и конверсией и называются функциями запрета. (6)

Утверждение 2. Множество, состоящее из двух значений 0 и 1, на котором определены унарная операция отрицания $\neg$ согласно табл.2, и бинарные операции дизъюнкции $\vee$ и конъюнкции $\wedge$ согласно табл.3, является булевой алгеброй. (1, с.151)

В изображении формул приняты следующие допущения: внешние скобки опускают, устанавливают приоритеты выполнения операций в следующем порядке:

$\neg$ - отрицание (наивысший приоритет),
$\wedge$ - конъюнкция,
$\vee$ - дизъюнкция,
$\rightarrow, \equiv$ - импликация и эквивалентность (имеют одинаковый приоритет).

С учетом этих приоритетов избыточные скобки также опускаются.

Рассмотрим теперь основные свойства булевых операций.

Ассоциативность:

а) $x_1(x_2x_3) = (x_1x_2)x_3$;

б) $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$.

Коммутативность:

а) $x_1x_2 = x_2x_1$;

б) $x_1 \vee x_2 = x_2 \vee x_1$.

Дистрибутивность конъюнкции относительно дизъюнкции:

$x_1(x_2 \vee x_3) = x_1x_2 \vee x_1x_3$.

Дистрибутивность дизъюнкции относительно конъюнкции:

$x_1 \vee (x_2x_3) = (x_1 \vee x_2)(x_1 \vee x_3)$.

Идемпотентность:

а) $xx = x$;

б) $x \vee x = x$.

Двойное отрицание:

$\overline{\overline{x}} = x$.

Свойства констант:

а) $x \& 1 = x$;

б) $x \& 0 = 0$;

в) $x \vee 1 = 1$;

г) $x \vee 0 = x$;

д) $\overline{0} = 1$;

е) $\overline{1} = 0$.

Правила де Моргана:

а) $\overline{x_1x_2} = \overline{x_1} \vee \overline{x_2}$;

б) $\overline{x_1 \vee x_2} = \overline{x_1}\,\overline{x_2}$.

Закон противоречия:

$x\overline{x} = 0$;

Закон исключения третьего:

$x \vee \overline{x} = 1$.

Выражение дизъюнкции через конъюнкцию и сумму по модулю 2:

$x_1 \vee x_2 = x_1 \wedge x_2 \oplus x_2 \oplus x_1$;

Выражение дизъюнкции через импликацию:

$x_1 \vee x_2 = (x_1 \rightarrow x_2) \rightarrow x_2$;

Выражение отрицания через штрих Шеффера, стрелку Пирса, сумму по модулю 2 и эквиваленцию:

$\overline{x} = x \mid x = x \downarrow x = x \oplus 1 = x \leftrightarrow 0$;

Выражение конъюнкции через штрих Шеффера:

$x_1 \wedge x_2 = (x_1 \mid x_2) \mid (x_1 \mid x_2)$;

Выражение дизъюнкции через стрелку Пирса:

$x_1 \vee x_2 = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2)$;

Закон поглощения:

$x_1 \wedge x_2 \vee x_1 = x_1$;

$x_1 \wedge (x_2 \vee x_1) = x_1$;

Закон склеивания:

$\overline{x} \oplus x = 1$;

Некоторые свойства сложения по модулю 2:

$x \oplus x = 0$;

$\overline{x} \oplus x = 1$;

$x \oplus 0 = x$;

$x \oplus 1 = \overline{x}$.

Для дизъюнкции и конъюнкции справедливы следующие свойства:

$x_1 \vee \overline{x_1} \wedge x_2 = x_1 \vee x_2$;

$x_1 \wedge (\overline{x_2} \vee x_1) = x_1 \wedge x_2$.

Понятие логического базиса. Понятие о полноте системы булевских функций. Полиномы Жегалкина.

Общее число булевых функций : $|P| = 2^{2^n}$, где n – число переменных (число элементов булевой функции). $f(x_1, x_2, ..., x_n) \in P_2$ - обозначение булевой функции.

Из множества этих функций некоторые подмножества играют важную роль в теории булевых функций. Пусть имеется булева функция $f(x_1, x_2, ..., x_n)$ и каждая переменная принимает некоторое значение $x_1 = \sigma_1, x_2 = \sigma_2, ..., x_n = \sigma_n$, причём $\sigma_i \in \{0,1\} \mid i = \overline{1,n}$. Тогда $f(\sigma_1, \sigma_2, ..., \sigma_n)$ есть значения функции при заданном значении окружности.

1. Говорят что булева функция f сохраняет количество 0, если на наборе $f(0,0,0,...,0) = 0$

Множество всех булевых функций, сохраняет константу 0, обрабатывает класс k_0с P_0.

Число всех булевых функций сохраняющих константу 0, $|K_0| = 2^{2^{n+1}-1}$

2. Говорят что булева функция сохраняет константу 1, если на единичном наборе $f(1,1,1,...,1) = 1$

Множество всех булевых функций, сохраняет константу 0, обрабатывает класс k_1с P_2.

Число всех булевых функций сохраняющих константу 1, $|K_0| = 2^{2^n-1}$

3. Функция $g(x_1, x_2, ..., x_n) = \overline{f(\overline{x_1}, \overline{x_2}, ..., \overline{x_n})}$ называется двойственной функции f и обозначается f^*

Булева функция – самодвойственна, если она совпадает с двойственной с ней функцией, то есть имеет место равенство $f = f^*$.

План самодвойственных функций: SCP_2

Пример: проверить является ли $f = \bar{x}$ - самодвойственной

x	$\bar{x}$
0	1
1	0

$= f$

x	$\bar{x}$
1	1
0	0

⍰ f →

x	$\bar{x}$
1	0
0	1

x	$\bar{x}$
0	1
1	0

$= f^*$

feS

4. Булева функция f называется линейной , если она может быть записана в следующем виде: $f(x_1, x_2, ..., x_n) = C_0 \oplus C_1 x_1 \oplus C_2 x_2 \oplus ... \oplus C_n x_n$, (2) где $C_k \in \{0,1\}$ где $k = \overline{0,n}$.

Множество всех линейных функций: LcP$_2$

Число всех линейных булевых функций: $|L| = 2^{n+1}$

Полином вида (2) показывает полином Жегалкина. Если в полиноме отсутствует сомножество переменных, то полином называют линейным.

Пример: $X \wedge Y$

1. Построим таблицу истинности:

x y	$X \wedge Y$
0 0	0
0 1	0
1 0	0
1 1	1

2. По таблице истинности строим СДНФ:

$f_{cdnf} = x_1 x_2 \Rightarrow (x \wedge y) \notin L$

А) $f_{cdnf} = \bar{x}_1 \bar{x}_2 \vee x_1 x_2$

Б) $V \to \oplus$, $f_{cdnf} = \bar{x}_1 \bar{x}_2 \oplus x_1 x_2$

В) Избавляемся от инверсии переменных: $\bar{x} = x \oplus 1$

$f = (x_1 \oplus 1)(x_2 \oplus 1) \oplus x_1 x_2 = x_1 x_2 \oplus x_2 \oplus x_1 \oplus 1 \oplus x_1 x_2 = [x \oplus x = 0] = x_2 \oplus x_1 \oplus 1$ полученный полином является линейным => ⍰(x⍰y) $\in$ L

Будем говорить, что набор значений переменных (δ_1,…,δ_n) не меньше набора переменных (δ'_1,…,δ'_n), если ($\delta_j \ge \delta'_j$) $\forall$ j=1,n. В этом случае запишем: (3) (δ_1,…,δ_n) ≥(δ'_1,…,δ'_n) в противном случае наборы не сравнимые.

Пример: n=3

(1,1,0)≥(0,0,0)

(0,1,0)≤(1,1,0)

(0,1,1)≠(1,0,1)

Определение: булева функция называется монотонной, если для любых 2-х наборов таких, что выполняется (3), имеет неравенство.

f (δ_1,…,δ_n) ≥ f (δ'_1,…,δ'_n) (4)

Множество всех монотонных булевых функций: McP$_2$.В отличии от предыдущих функций, число монотонных булевых функций оцениваются.

$$2^{c_n^{|n\backslash 2|}} \le card(M) \le 2^{AC_n^{|n\backslash 2|}}$$

Система булевых функций называется функционально полной на множестве P$_2$ всех булевых функций, если в результате суперпозиции этой системы можно получить любую булеву функцию.

Теорема поста: для того что бы система булевых функций была полной, необходимо и достаточно, что бы она создавала хотя бы одну функцию:

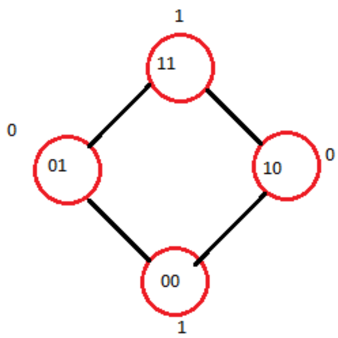
1. Не сохраняющая константу 1
2. Не сохраняющая константу 0
3. Не самодвойственную
4. Не линейную
5. Не монотонную

При определении полноты системы булевой функции пользуются таблицами Поста:
Принадлежность булевой функции разным планам отличается знаком +
Пример:

Функция	K_0	K_1	L	M	S
Отрицание	-	-	+	-	+
Дизъюнкция	+	+	-	+	-
Конъюнкция	+	+	-	+	-
Импликация	-	+	-	-	-
Сложение по 2	+	-	+	-	-
эквивалентность	-	+	-	-	-

Из таблицы нетрудно определить, что система булевых функций является полной.
Пример: проверить принадлежит ли эквивалентность к классу М.
При упорядочивании наборов воспользуемся диаграммой Хоссе:

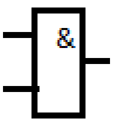
x y	$X \equiv Y$
0 0	1
0 1	0
1 0	0
1 1	1

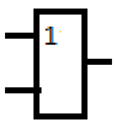


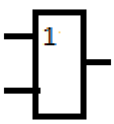
$(x \equiv y) \notin M$, так как $f(0,0) \geq f(0,1) \mid (0,1) \geq (0,0)$

Понятие логической схемы. Реализация логических схем

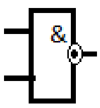
Задача синтеза логической схемы.
По заданной функции f требуется построить некоторое устройство, реализовав данную функцию.
Устройство будим представлять в виде логической схемы, а схему состоящую из элементов, которые будим называть логическими элементами. Каждый из таких элементов реализует некоторый элемент булевой функции. Когда построение логической схемы по известной булевой функции – задача синтеза логической схемы. Такая задача решается не однозначно. Можно поставить в соответствие заданной булевой функции. Для построения логической схемы требуются элементы для реализации логических операций, указанных в заданной функции f .
Основные операторы которые реализуют булеву функцию:

$y = x_1 \wedge x_2$ 

$y = \overline{x}$ 

$y = x_1 \vee x_2$ 

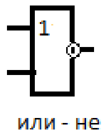
< ∧ ∨¬ > функционально полный базис
Существуют и другие функционально полные базисы:

$y = x_1 \mid x_2 = \overline{x_1 \wedge x_2}$ 

и - не

< | > формально полный базис

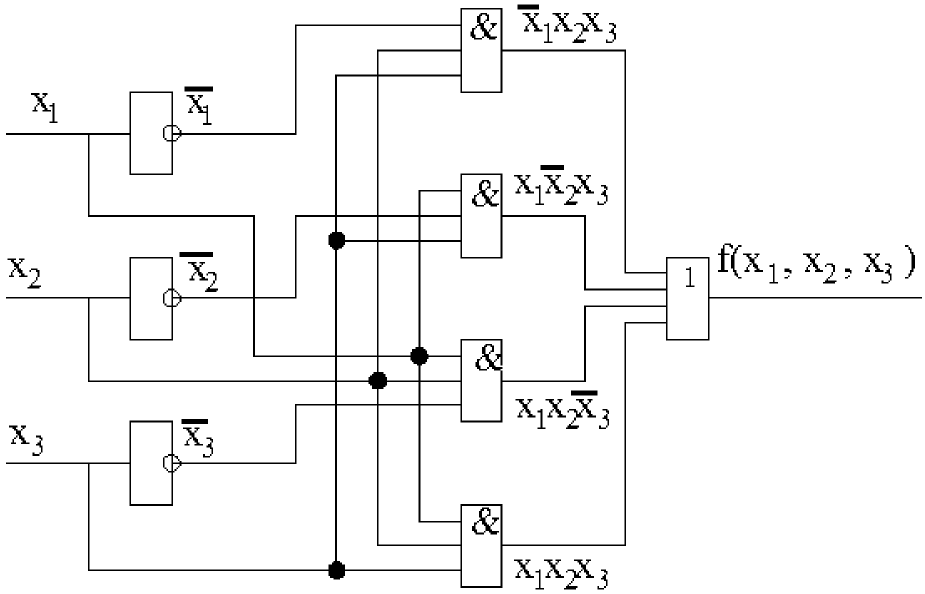
$$y = x_1 \downarrow x_2 = \overline{x_1 \vee x_2}$$



Рассмотрим систему на примере:

Построить логическую схему устройства, реализующего следующую функцию: $f(x_1, x_2, x_3) = \overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3 \vee x_1x_2\overline{x_3} \vee x_1x_2x_3$

Логическая схема: < и, или, не >



Если задан < и – не >, то путём двойного инвертирования 1-го выражения (части) и применения теоремы Де Моргана, логическая функция приводится к виду, содержащему только операции логического умножения и инвентирования.

Если же задан < или – не >, то находим логическую функцию теми же приёмами приводят к виду, содержащему только операции логического соложения и инвертирования. Далее логическое выражение записывается через условное выражение выбранное оператором.

Пример: задана функция f перевести в заданные варианты: < и, не > < или, не>

$f(x_1, x_2, x_3, x_4) = x_2x_4 \vee \overline{x_1}x_3\overline{x_4} \vee x_1\overline{x_2}\,\overline{x_3}$ = и – не

1. <u>Двойное инвертирование</u>

$$= \overline{\overline{x_2x_4 \vee \overline{x_1}x_3\overline{x_4} \vee x_1\overline{x_2}\,\overline{x_3}}} = [\overline{\overline{x}} = x] =$$

2. Законы Де Моргана $\overline{x_1 \vee x_2} = \overline{x_1} \wedge \overline{x_2}$

$$= \overline{\overline{x_2x_4} \wedge \overline{\overline{\overline{x_1}}x_3\overline{\overline{x_4}}} \vee \overline{x_1\overline{\overline{x_2}}\,\overline{\overline{x_3}}}} = (x_2 \mid x_4) \mid (\overline{x_1} \mid x_3 \mid \overline{x_4}) \mid (\overline{x_1} \mid \overline{x_2} \mid \overline{x_3})$$

Пусть задана функция в КНФ: или – не

$$f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2)(\overline{x_1} \vee x_2 \vee x_3)(\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4) =$$

$$= \overline{\overline{(x_1 \vee x_2)(\overline{x_1} \vee x_2 \vee x_3)(\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4)}} =$$

$$= \overline{(\overline{x_1 \vee x_2})(\overline{\overline{x_1} \vee x_2 \vee x_3})(\overline{\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4})} =$$

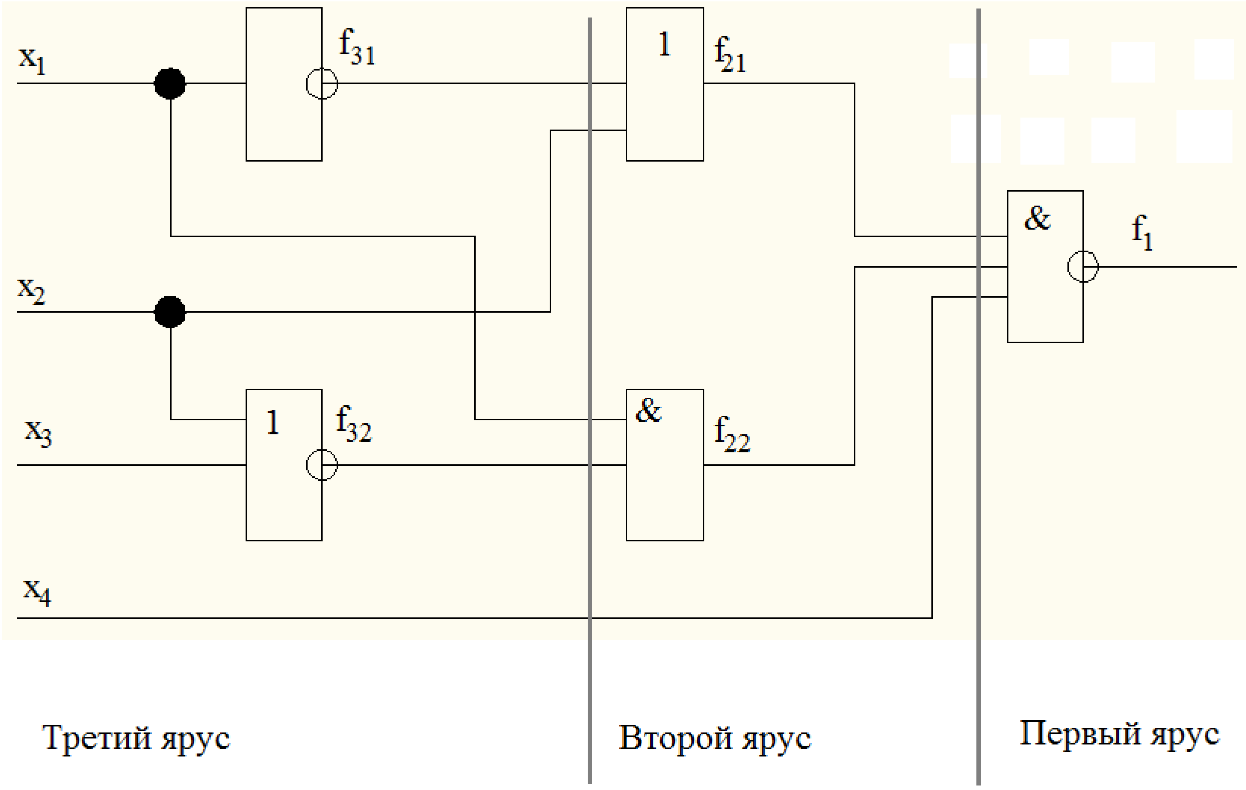
$$= (x_1 \downarrow x_2) \downarrow (\overline{x_1} \downarrow x_2 \downarrow x_3)(\overline{x}_2 \downarrow \overline{x}_3 \downarrow \overline{x}_4)$$

Задача анализа логической схемы.

Формулировка задачи: по заданной логической схеме требуется определить функцию f, реализованную данной схемой. При решении задачи анализа следует придерживаться следующей последовательность действий:

1. Заданная схема разбирается по ярусам
2. Начиная с последнего, выходы каждого элемента обозначаются функциями в зависимости от яруса, которым относятся элементы.
3. Записываются выходные функции каждого элемента в виде формул в соответствии с введёнными обозначениями.
4. Производится подстановка одинаковых выходных функций через другие, используются входящие переменные.
5. Записывается получившееся булева функция через входные переменные.

Пример:



Согласно приведённой выше последовательности действий, производим разбиение схемы на ярусы, пронумеруем ярусы и введём для каждой выходной функции:

1. $f_1 = f_{21} \cdot f_{22} \cdot x_4$
2. $f_{21} = f_{31} \vee x_2 \qquad f_{22} = f_{32} \vee x_1$
3. $f_{31} = \overline{x_1} \qquad f_{32} = \overline{x_2 \vee x_3}$

Запишем все функции, подставив входные переменные:

x_1, x_2, x_3, x_4

$f_{22} = x_1(\overline{x_2 \vee x_3}) \qquad f_{21} = \overline{x_1} \vee x_2$

$f_1 = \overline{(\overline{x_1} \vee x_2) \cdot (x_1(\overline{x_2 \vee x_3})) \cdot x_4} = \overline{(\overline{x_1} \vee x_2)(\overline{x_2 \vee x_3})x_1 \cdot x_4}$

Минимизация булевых функций. Задачи минимизация булевых функций.

Задача поиска наибольшей простой записи булевой функции задачи минимизации.

Решение такой задачи основывается на понятии несущественности переменных.

Переменными называются несущественные на наборе, если при изменении её значения на противоположное булева функция сохраняет свой значение:

$f(a_1, a_2, ..., a_{i-1}, 0, a_{i+1}, ..., a_n) = f(a_1, a_2, ..., a_{i-1}, 1, a_{i+1}, ..., a_n)$

Две конъюнкции содержащих несущественные переменные, заменяются данной в которой несущественные переменные отсутствуют.

Например: $x_1x_2x_3 \vee x_1\overline{x_2}x_3 = x_1x_3$

$f(1,1,1) = 1; f(1,0,1) = 1$

ДНФ булевой функции от $f(x_1, x_2, .., x_n)$ называется кратчайшей, если она содержит наименьшее число элементов конъюнкции по сравнению с другими ДНФ этой же функции.

ДНФ булевой функции $f(x_1, x_2, .., x_n)$ называется минимальной, если она имеет наименьшее число аргументов среди всех эквивалентных ей ДНФ.

Импликантой функции $f(x_1, x_2, .., x_n)$ называют такую элементарную конъюнкцию К над множеством переменных $x_1, x_2, .., x_n$, что $f(x_1, x_2, .., x_n) \vee k = (x_1, x_2, .., x_n)$.

Импликанта называется простой, если при отбрасывании любого аргумента из конъюнкции К получится элемент конъюнкции, не являющийся импликантой функции f.

Дизъюнкция всех простых импликант функции f, называется сокращённая ДНФ функции f.

ДНФ булевой функции $f(x_1, x_2, .., x_n)$ называют тупиковой если отбрасывание любого слагаемого или аргумента приводит к не эквивалентной ДНФ.

Тупиковая ДНФ функции $f(x_1, x_2, .., x_n)$ получается из сокращённой ДНФ этой функции путём отбрасывания некоторых элементов конъюнкции среди тупиковых. ДНФ считается минимальной и кратчайшей ДНФ функции.

1. Метод неопределённых коэффициентов

Данный метод может быть применен для минимизации функций алгебры логики от любого числа аргументов. Однако для простоты будим рассматривать методы на примере минимизации функции 3-х переменных. $f(x_1, x_2, x_3)$ - представим в виде ДНФ общем случае:

$$\begin{aligned} f(x_1,x_2,x_3) = {} & k_1^0\overline{x_1} \vee k_1^1x_1 \vee k_2^0\overline{x_2} \vee k_2^1x_2 \vee k_3^0\overline{x_3} \vee k_3^1x_3 \vee k_{12}^{00}\overline{x_1}\,\overline{x_2} \vee k_{12}^{01}\overline{x_1}x_2 \vee k_{12}^{10}x_1\overline{x_2} \vee k_{13}^{11}x_1x_2 \\ & \vee k_{13}^{00}\overline{x_1}\,\overline{x_3} \vee k_{13}^{01}\overline{x_1}x_3 \vee k_{13}^{10}x_1\overline{x_3} \vee k_{13}^{11}x_1x_3 \vee k_{23}^{00}\overline{x_2}\,\overline{x_3} \vee k_{23}^{01}\overline{x_2}x_3 \vee k_{23}^{10}x_2\overline{x_3} \\ & \vee k_{23}^{11}x_2x_3 \vee k_{123}^{000}\overline{x_1}\,\overline{x_2}\,\overline{x_3} \vee k_{123}^{001}\overline{x_1}\,\overline{x_2}x_3 \vee k_{123}^{010}\overline{x_1}\,x_2\overline{x_3} \vee k_{123}^{011}\overline{x_1}x_2x_3 \vee k_{123}^{100}x_1\overline{x_2}\,\overline{x_3} \\ & \vee k_{123}^{101}x_1\overline{x_2}x_3 \vee k_{123}^{110}x_1x_2\overline{x_3} \vee k_{123}^{111}x_1x_2x_3 \end{aligned}$$

В выражении представлены все возможные конъюнкции члены которых могут входить в дизъюнкцию функции 3-х переменных. Коэффициент К с различными индексами является неопределённым и подбирается так, что бы получилась дизъюнктивная формула ДНФ была минимальна: $k \in \{0; 1\}$

Если теперь задавать все возможные наборы аргументов x_1, x_2, x_3 и приравнять полученные параметры этого выражения значению функции на выбранных наборах, отбрасывая при этом нулевые элементы конъюнкции, то получим систему из 2^3 уравнений для определения коэффициента К:

Система (2):

$$\begin{gathered} k_1^0 \vee k_2^0 \vee k_3^0 \vee k_{12}^{00} \vee k_{13}^{00} \vee k_{23}^{00} \vee k_{123}^{000} = f(0,0,0) \\ k_1^0 \vee k_2^0 \vee k_3^1 \vee k_{12}^{00} \vee k_{13}^{01} \vee k_{23}^{01} \vee k_{123}^{001} = f(0,0,1) \\ k_1^0 \vee k_2^1 \vee k_3^0 \vee k_{12}^{01} \vee k_{13}^{00} \vee k_{23}^{10} \vee k_{123}^{010} = f(0,1,0) \\ k_1^0 \vee k_2^1 \vee k_3^1 \vee k_{12}^{01} \vee k_{13}^{01} \vee k_{23}^{11} \vee k_{123}^{011} = f(0,1,1) \\ k_1^1 \vee k_2^0 \vee k_3^0 \vee k_{12}^{10} \vee k_{13}^{10} \vee k_{23}^{00} \vee k_{123}^{100} = f(1,0,0) \\ k_1^1 \vee k_2^0 \vee k_3^1 \vee k_{12}^{10} \vee k_{13}^{11} \vee k_{23}^{01} \vee k_{123}^{101} = f(1,0,1) \\ k_1^1 \vee k_2^1 \vee k_3^0 \vee k_{12}^{11} \vee k_{13}^{10} \vee k_{23}^{10} \vee k_{123}^{110} = f(1,1,0) \\ k_1^1 \vee k_2^1 \vee k_3^1 \vee k_{12}^{11} \vee k_{13}^{11} \vee k_{23}^{11} \vee k_{123}^{111} = f(1,1,1) \end{gathered}$$

Пусть таблично задана функция f от 3-х переменных. Определить значение правой части системы, принимает 0 или 1.

Если функция на определённом наборе системы (2) =0, то все коэффициенты входящие в это уравнение также равны 0.

Рассмотрев все наборы, на которых данная функция обращается в 0, получим нулевую.

В уравнениях в которых справа стоят 1-цы, вычеркнем все нулевые коэффициенты. Из оставшихся коэффициентов, приравняем коэффициенты к единице, определим конъюнкцию наименьшую возможного ранга.

Ранг – число переменных в конъюнкции. Остальные коэффициенты в левой части данных уравнений примем = 0. Это можно сделать так как n – обращается в единицу, если хотя бы один её член равен 1.
Единичный коэффициент К определяет из выражения (1) соответствующей ДНФ.
Описанный метод эффективен для минимизации функции, число аргументов которого не более 5,6 для ручного расчёта.

Правило минимизации с использованием диаграмм Вейча (метод карт Карно)

В диаграмме Вейча таблица истинности булевой функции представлена в виде координатной карты, которые содержат 2^n – клеток. по числу входящих наборов булевой функции n-переменных.
Переменные функции разбиваются на 2 группы так, что одна группа определяет координаты столбца а другая координаты строки.
При таком способе построения каждая клетка определяется значением переменных соответственно определённых двоичному набору.
Внутри каждой клетки карты Карно ставится значение функции на данном наборе.
Переменные в строках и столбцах располагаются так, что бы соседние клетки карты Карно различались только в одном размере переменной, то есть были соседними.
Поэтому значения переменных в строках и столбцах карты образуют соседний код Грея.
Такой способ представления удобен для наглядного, при не оптимизированной булевой функции.
При этом используют следующие правила:
1. В карте Карно группе единиц (для получения ДНФ) и группе нулей (для получения КНФ) необходимо обвести четырёх угольник контурами. Внутри контура должны находится только одноименные значения функции. Этот процесс соответствует операции нахождения импликант данной функции, или по нему несущественных переменных.
2. Количество клеток внутри контура должно быть целой степенью двойки (1,2,4,8,16)
3. При проведении контуров крайние строки карты (верхние или нижние, левые или правые), а так же углов клетки считается соседними (для карт до 4-х переменных)
4. Каждая карта должна включать максимальное возможное количество клеток. В этом случае она будет соответствовать простой импликанте.
5. Все единицы (нули) в карте, даже одиночные, должны быть охвачены контуром. Любая единица (0) может входить в контур произвольное количество раз.
6. Множество контуров, покрывающих все единицы (нули) функции образуют тупиковую ДНФ(КНФ).

Целью минимизации является нахождение минимального из множества тупиковых форм.
7. В элементах конъюнкции(дизъюнкции) которая соответствует одному контуру, остаются только те переменные, значения которых не изменяется внутри обведённого контура. Переменные булевой функции входят в элементы конъюнкции (для значений функции 1) без инверсии, если их значения не соответствуют координатам равны 1, и с инверсией, если равны 0.

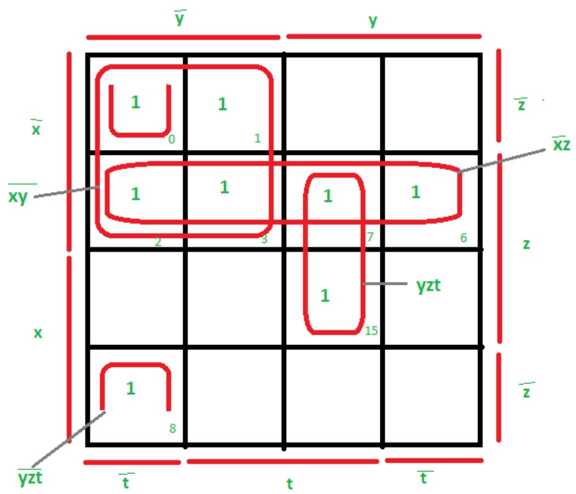
Для значений булевой функции =0 записываем элементарную дизъюнкцию, куда переменные входят без инверсии, или их значения по соответствующим координатам равны 0, и с инверсией, если равны 1.

Пример: $f(x,y,z,t) = 1|0,1,2,3,6,7,8,15$

	x	y	z	t
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
15	1	1	1	1

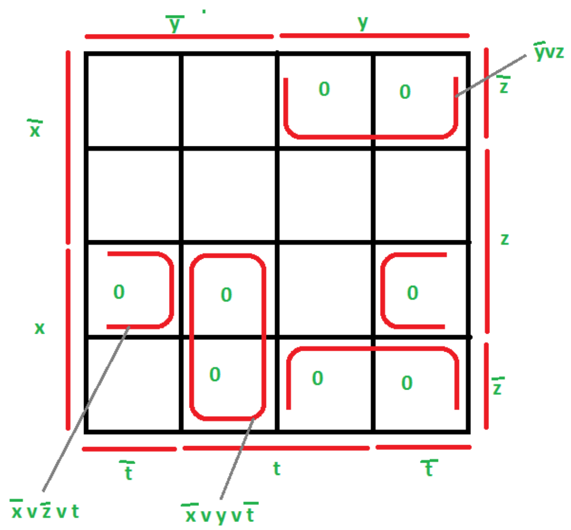
$f(x,y,z,t) = \bar{x}\,\bar{y} \vee \bar{x}\,z \vee yzt \vee \bar{y}\,\bar{z}\,\bar{t}$ - min ДНФ

$f(x,y,z,t) = (\bar{y} \vee z)(\bar{x} \vee \bar{z} \vee t)(\bar{x} \vee y \vee \bar{t})$



При минимизации булевой функции на диаграмме Вейча сначала находят покрытие, содержащее максимальное число единицу (8,7,2), а затем покрытие, покрывающее остальные единицы таким образом что бы они так же были максимальны по величине и при удалении этого покрытия, хотя бы одна единичная функция осталась непокрытой. При этом некоторые 1 могут быть покрыты неоднократно.

Минимальная КНФ строится двойственно по диаграмме Вейча:



$$(\bar{y} \vee z)(\bar{x} \vee \bar{z} \vee t)(\bar{x} \vee y \vee \bar{t})$$

Элементы логики предикатов.

Логика предикатов – новая логическая система, представляющая развитие логики высказываний.

Исторически понятие о предикатах являлось вследствие логического анализа высказываний, т.е. выполнение их логической структуры (выяснение того, какой логический смысл может быть выражен этими высказываниями).

Пример: x – простое число.

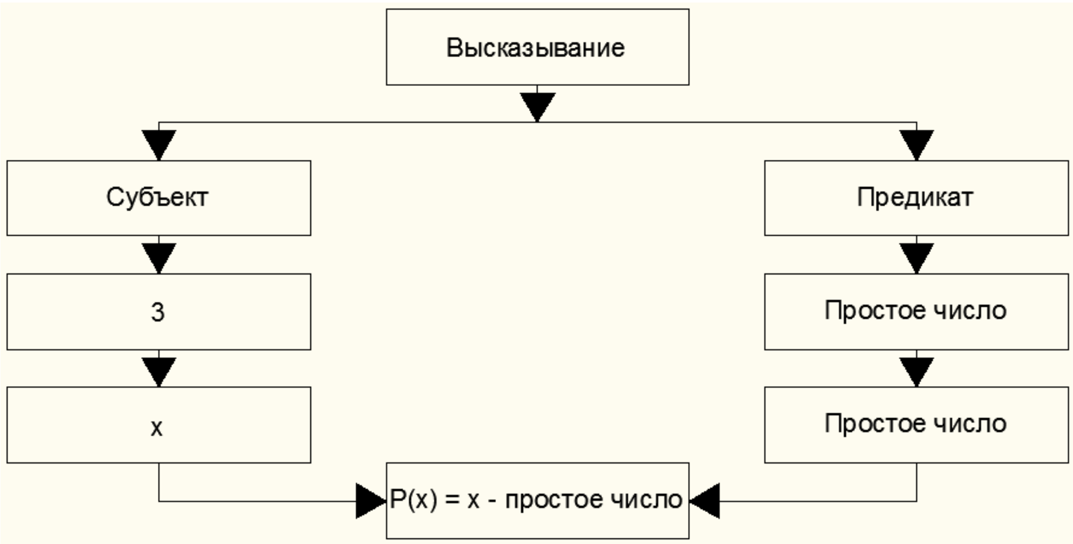
Если вместо x подставить 3 – простое число – то высказывание истинно, или же подставить число 8 – высказывание ложно. Таким образом, x – простое число можно рассматривать как функцию P(x), зависящую от x. Область определения – множество чисел, а область значений – высказывания.

Определение. Одноместным предикатом P(x), определенном на множестве M, называют выражение, которое после подстановки в него вместо x – предмета из области M обращается в высказывание.

Область определения предиката называется предметной областью.

Элементы из области определения называются предметными постоянными (предметами).

Переменная, от которой зависит предикат, называется предметной переменной.



Одноместные предикаты традиционно служат для формализации понятий.

Понятие представляет собой единицу мышления. Абстрактное мышление основывается на понятиях, отображающих действительность, поэтому абстрактное мышление называют понятийным.

Понятия возникают как результат обобщения множества предметов по системе признаков.

Признак - это наличие или отсутствие свойства у предмета.

Понятие характеризуется своим содержанием и объемом.

Содержание понятия – система признаков, на основе которых множество предметов обобщается в понятие.

Объем понятия – множество предметов, обобщаемых и выделяемых в понятии, т.е. множество предметов, которое характеризуется системой признаков, составляющих содержание понятия.

Чтобы задать n-местный предикат $P(x_1, \dots, x_n)$ следует указать множество $x_1, x_2, \dots, x_n$, т.е. область изменения переменных $x_1, x_2, \dots, x_n$, причем чаще всего рассматривается случай, когда: $x_1=x_2=\dots=x_n$. С точки зрения теории множеств в общем случае предмет определяется заданием подмножеством M в декартовом произведении: $McX_1 \times X_2 \times \dots \times X_n$.

Переменные $x_1, x_2, \dots, x_n$ называются предметными переменными. Элементы множеств $X_1, X_2, \dots, X_n$ называют предметами.

M – множество кортежей длины n:

$$M = \{< x_1, x_2, \dots, x_n > | x_1 \in X_1, x_2 \in X_2, \dots, x_n \in X_n\}$$

Предметные переменные называют (обозначают) малыми буквами конца латинского алфавита: x, y, z. Иногда снабжают индексами.

Предметы из множества $X_1, X_2, \dots, X_n$ обозначают малыми буквами начала латинского алфавита a,b, c.

Предметы обозначают большими буквами латинского алфавита: A, B, C предписывая имя переменной ($A(x)$, $P(x_1, x_2, x_3)$) или без них (A, B, P).
Число переменных предиката указывает нам верхний индекс предиката: $P^k(x_1, x_2, \ldots, x_k)$ – как k-местный предикат.
$Q^2(x,y)$ – двуместный предикат.
k-местный предикат – есть функция предметной переменной, которая принимает значение из некоторого множества M_k, а сама функция принимает значение истинна или ложь:
$P^k(x_1, x_2, \ldots, x_k)$: $M_k \rightarrow \{0, 1\}$.
Предикат называют разрешимым, если существуют такие кортежи, компоненты которых обращают предикат в истинное высказывание.
Если предикат при подстановке любых конкретных элементов из соответствующих множеств обращается в ложное высказывание, то он называется тождественно ложным.
Если предикат обращается при подстановке любых конкретных элементов в истинное высказывание, то он называется тождественно истинным.
К предикатам, определенным на одном и том же множестве можно применять операции алгебры высказываний: $\cup, \cap, \equiv, \rightarrow, -$ и тем самым получать новые предикаты.
Например, предикаты P(x,y), Q(x,y).
К этим предикатам применима операция конъюнкции: $P(x,y) \wedge Q(x,y)$
Т.к. к предикатам можно применять логические операции, то для них справедливы основные законы булевой алгебры:
1. $\overline{\overline{P}} = P$
2. $P \vee Q = Q \vee P$ (свойство коммутативности)
3. $P \wedge Q = Q \wedge P$ (свойство коммутативности)
4. $P \vee (Q \vee R) = (P \vee Q) \vee R$ (ассоциативность)
5. $P \wedge (Q \wedge R) = (P \wedge Q) \wedge R$ (ассоциативность)
6. $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$ (свойство дистрибутивности)
7. $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$
8. Законы де Моргана:
$\overline{P \vee Q} = \overline{P} \wedge \overline{Q}$
$\overline{P \wedge Q} = \overline{P} \vee \overline{Q}$
9. $P \vee P = P;\ \ P \wedge P = P$
10. $P \vee \overline{P} = 1; P \wedge \overline{P} = 0$
11. $P \vee 1 = 1; P \wedge 0 = 0; P \vee 0 = P; P \wedge 1 = P$
12. $P \vee (P \wedge Q) = P$
13. $P \wedge (P \vee Q) = P$
Кроме операторов алгебры высказываний в логике предикатов определены 2 новые операции (операции навешивания кванторов).
Квантор всеобщности - ∀.
Квантор существования - ∃.
Пусть дан предикат P(x), зависящий от одной переменной и определенный на поле M.
1. $(\forall x)P(x)$, тогда выражение с квантором всеобщности означает высказывание, истинное тогда, когда предикат P(x) истинен для всех значений поля M.
$(\forall x)P(x)$ для всякого x, P(x).
2. $(\exists x)P(x)$ существует x, что P(x).
Это выражение означает высказывание, которое истинно тогда, когда предикат P(x) истинен хотя бы для одного предмета из поля M.
Квантор общности – это оператор, приводящий в соответствие любому заданному предикату такую 2х значную логическую переменную z, которая принимает значение 1 тогда и только тогда, когда y=1 при всех значения x. (y=P(x), z, y=1, x).
Квантор существования – оператор, приводящий в соответствие любому одноместному предикату, такую двузначную логическую переменную z, которая принимает значение 0, тогда, когда y=0 при всех значениях x. (y=P(x), z, y=0, x).
Если $M=\{Q_1, Q_2, \ldots, Q_n\}$ конечная область определения предиката P(X), то формулы с квантором могут быть выражены через коньюнкцию или дизъюнкцию:

$\forall x P(x) = P(Q_1) \wedge P(Q_2) \wedge \ldots \wedge P(Q_n)$
$\exists x P(x) = P(Q_1) \vee P(Q_2) \vee \ldots \vee P(Q_n)$
Кванторы общности и существования связаны по принципу двойственности по закону де Моргана:
$\overline{\forall x P(x)} = \exists x \overline{P(x)}$
$\overline{\exists x P(x)} = \forall x \overline{P(x)}$
Рассмотрим некоторые общие свойства введенных операторов.
В соответствии с определением кванторов, логическая переменная z в выражениях $z = \forall x P(x)$ и $z = \exists x P(x)$ не является функцией предметной переменной x.
Для того, чтобы отметить отсутствие функциональной зависимости z от x, предметную переменную x называют связанной.
Не связанные переменные называют свободными.
Например: $\forall x A(x, y) \vee \exists z B(z, v)$ - переменные x и z – связанные, а y и v свободные.
Если квантор общности или существования применяется не к одноместному предикату, а к k-местному, получается предикат (k-1)-местный.
Основными символами логики предикатов являются пропозициональные символы:
1. □, →
2. ∀, ∃
3. Вспомогательные символы: , ()
4. Предметные переменные: $x_1, x_2, \ldots, x_n$
5. Предметные постоянные: $a_1, a_2, \ldots, a_n$
6. Функциональные символы: $f_1^1, f_1^2, \ldots, f_k^j$
7. $P_1^1, P_1^2, \ldots, P_k^j$
Нижний символ – номер, который служит для различения одноименных символов с одинаковым числом аргументов. Верхний индекс указывает число аргументов.
Определение терма:
1. Каждая предметная переменная есть терм.
2. Каждая предметная постоянная есть терм.
3. Если f_k^j - функциональный символ, а $t_1, \ldots, t_n$ – термы, то $f_k^j(t_1, t_2, \ldots, t_n)$ - терм.
4. Других термов нет.
Определение формулы:
1. $P_i^n(t_1, t_2, \ldots, t_n)$, где P_i^n - предикатный символ, $t_1, \ldots, t_n$ – термы, то $P_i^n(t_1, t_2, \ldots, t_n)$ называется атомарной или элементарной формулой.
2. Если A и B формулы и x – предметная переменная, то формулами является:
□A; A→B, $\forall x A$; $\exists x A$.
3. Других формул нет.
Формула, на которую распространяется действие квантора, называется областью действия квантора.
Переменная, по которой навешивается квантор и попадающая в него область действий, называется связанной переменной.
Переменная, лежащая вне области действия квантора называется свободной переменной.
Формула, не содержащая свободных переменных, называется замкнутой. Замкнутые формулы называются высказываниями.
Область действия квантора ограничивается скобками, если она содержит более одного предиката.
Пример: $\forall x(P(x) \rightarrow Q(x, y))$.
Формулы имеют смысл тогда, когда имеется какая-либо интерпретация входящих в него символов.
Определение. Под интерпретацией будем понимать систему, состоящую из непустого множества Д (называется областью интерпретации), а также соответствие, ставящее каждой предметной букве P_i^n некоторое отношение на некоторой области Д, каждой предметной постоянной a_i некоторый элемент из области Д, каждой функционной букве f_i^n некоторую n-местную операцию на Д ($Д^n$→Д).
Интерпретация называется моделью для данного множества формул Г, если каждая формула из Г истинна в данной интерпретации.

Определение. Формула называется выполнимой, если существует хотя бы одна интерпретация, на которой формула истинна.

Формула называется логически общезначимой, если она истинна на любой интерпретации при любых значениях переменных.

Формула, которая ложна на любой интерпретации называется противоречием.

Логически общезначимые формулы являются выделенными формулами логических предикатов.

Существуют способы, которые в частных случаях позволяют определить логическую общезначимость, эквивалентность формул, выполнимость.

Например. Возьмем область интерпретаций состоящую из двух элементов. Построим таблицу истинности двух форм $Д = \{a, b\};\ E_1 = \exists x P(x);\ E_2 = \forall x P(x)$.

Одноместный предикат на области определения из двух элементов может принимать одно из четырех значений, которые определяются таблицами истинности.

Таблица 1.

x	$P_1()$	$P_2()$	$P_3()$	$P_4()$
a	F	F	T	T
b	F	T	F	T

Формулы E_1 и E_2 будут принимать на этих интерпретациях следующие значения:

Таблица 2.

P()	$\exists x P(x)$	$\forall x P(x)$
P_1	F	F
P_2	T	F
P_3	T	F
P_4	T	T

Пусть формулы F и G имеют одно и тоже множество свободных переменных. Формулы F и G равносильны в данной интерпретации, если они принимают одинаковое значение на любом наборе свободных переменных, т.е. выражают в данной интерпретации один и тот же предикат.

F и G тождественны (равносильны) на множестве M, если они принимают одинаковое значение во всех интерпретациях, заданных на множестве M.

Рассмотрим правила перехода от одних формул к другим, им равносильным.

1. Перенос квантора через отрицание:

Пусть W(x) – формула, содержащая переменную x, тогда равносильны следующие формулы:

$\overline{\forall x W(x)} = \exists x \overline{W(x)}$

$\overline{\forall x \overline{W(x)}} = \exists x W(x)$

$\overline{\exists x W(x)} = \forall x \overline{W(x)}$

$\overline{\exists x \overline{W(x)}} = \forall x W(x)$

2. Вынос квантора за скобки:

Пусть W(x) – формула, содержащая переменную x, а формула B не содержит переменную x. Эти формулы соответствуют третьему правилу создания формул. Тогда:

$\exists x(W(x) \wedge B) \equiv \forall x W(x) \wedge B$

$\forall x(W(x) \wedge B) \equiv \forall x W(x) \wedge B$

$\exists x(W(x) \vee B) \equiv \exists x W(x) \vee B$

$\forall x(W(x) \vee B) \equiv \forall x W(x) \vee B$

3. Перестановка одноименных кванторов.

$\exists x \exists y W(x, y) \equiv \exists y \exists x W(x, y)$ табл. 2

$\forall x \forall y W(x, y) \equiv \forall y \forall x W(x, y)$

4. Переименование связанных переменных.

Запишем связанную переменную формулы W другой переменной, не входящей в эту формулу, в кванторе и всюду в области действия квантора, получим формулу равносильную W.

Приведенные и нормальные формы в логике предикатов. Рассмотрим способ упрощения формул, опирающийся на известные равносильности.

Формулы, в которых из логических символов имеются только конъюнкция, дизъюнкция и отрицание, причем символ отрицания встречается над символами предикатов, будем называть приведенными.

Для любой формулы существует равносильная ей приведенная формула, причем множество свободных и связанных переменных этих формул совпадают.
В логике высказываний были введены 2 нормальные формы: КНФ и ДНФ. В логике предикатов также имеется нормальная форма, цель которой упрощение процедуры доказательств.
Приведенная формула называется нормальной, если она не содержит символов кванторов или все символы кванторов стоят впереди, т.е. логические символы и символы предикатов стоят в области действия каждого квантора.
Для любой приведенной формулы существует равносильная ей нормальная формула той же длины. Под длиной формулы будем понимать общее число, входящих в нее символов предикатов, логических символов и символов кванторов.
Нормальная формула называется нормальной формой данной формулы.
1. Неприведенная формула:
$\overline{(\forall x_2)}A_1^1(x_2) \to A_2^1(x_1)$
2. Приведенная формула:
$(\forall x_1)A_1^1(x_1) \vee (\exists x_2)\overline{A_2^2(x_2, x_3)}$
3. Нормальные формы:
$-\forall x \forall y(P(x) \wedge Q(y))$
$-\forall x \forall y(\overline{P(x, y)} \vee Q(y))$
$-\forall x \forall y \exists z(Q(x, y) \vee P(z))$
Алгоритм преобразования формул в нормальную форму.
1. Исключить логические связки ($\leftrightarrow, \to$) с помощью формул:
$F \leftrightarrow G = (F \to G) \wedge (G \to F)$
$F \to G = \bar{F} \vee G$
2. Использовать законы двойного отрицания и законы де Моргана:
$\bar{\bar{F}} = F$
$\overline{F \vee G} = \bar{F} \wedge \bar{G}$
$\overline{F \wedge G} = \bar{F} \vee \bar{G}$
а также следующие законы (законы де Моргана для логики предикатов):
$\overline{\forall x P(x)} = \exists x \overline{P(x)}$
$\overline{\exists x P(x)} = \forall x \overline{P(x)}$
3. Переименовать связанные переменные, если это необходимо.
4. Использовать равносильные формулы логики предикатов, чтобы вынести кванторы в самое начало формулы для приведения ее к нормальной форме.
Пример:
$\forall x P(x) \to \exists x Q(x) = \overline{\forall x P(x)} \vee \exists x Q(x) = \exists x \overline{P(x)} \vee \exists x Q(x) = \exists x(\overline{P(x)} \vee Q(x))$

Правило вынесения кванторов и скалеризация.
Предваренная нормальная форма, содержащая только кванторы всеобщности, называются скумелевской нормальной формой (стандартная форма) – ССФ.
Процедура приведения приведенной нормальной формы заключается в элиминации (удалении) кванторов существования.
Формула A находится в предваренной нормальной форме, если она имеет следующий вид:
$Q(x_1) \ldots Q(x_n)M$, где $Q(x_i) = \forall(x_i) \vee \exists(x_i) |\ i = 1, \ldots, n$.
M – КНФ (формула в КНФ, не содержащая кванторов).
$Q(x_1) \ldots Q(x_n)$ – называется префиксом.
M – матрица формулы A.
Существует эффективная процедура приведения любой формулы к ПНФ (предваренной нормальной форме).
Для получения предваренной нормально формы может быть использован приведенный выше алгоритм, в котором, если формула A образована из двух формул, используются законы замены связанных переменных:
$\forall x P(x) = \forall y P(y)$
$\exists x P(x) = \exists y P(y)$

Кроме этих законов используют законы коммутативности для конъюнкции и дизъюнкции, и законы перенесения кванторов:

$\forall x(P(x) \land B) = \forall x P(x) \land B$

$\forall x(P(x) \lor B) = \forall x P(x) \lor B$

$\exists x(P(x) \land B) = \exists x P(x) \land B$

$\exists x(P(x) \lor B) = \exists x P(x) \lor B$

Пусть формула А находится ПНФ: $\exists(x_1)Q(x_2) \dots Q(x_n)$. Если квантор существования (1-й слева квантор в префиксе), то его можно удалить. Для этого нужно выбрать константу с, отличную от других констант входящих в М. Записать все вхождения x_1, встречающиеся в М на с, и вычеркнуть квантор существования из префикса. Если же перед квантором существования стоит квантор всеобщности ($\forall x \exists y M$), то переменная у попадает в область действия квантора всеобщности и выражение $\forall x \exists y$ - означает наличие некоторой функциональной зависимости.

Если квантору существования предшествует … квантор всеобщности, то функция зависит от всех переменных, по которым навешены все эти кванторы.

В общем случае, если $Q_{s1} \dots Q_{sm} \exists x_r$ – список всех кванторов всеобщности, встречающихся левее квантора существования $1<s_1<s_2<\dots<s_m<r$, то выберем m-местный функциональный символ f, отличающийся от друх функциональных символов. Заменим все x_r в М на $f(x_{s1}, x_{s2}, \dots, x_{sm})$ и вычеркнем квантор существования x_r из префикса. Затем весь этот процесс применим для всех кванторов существования в префиксе.

Последняя из полученных формул - есть скумелевская стандартная форма (ССФ).

Функции, используемые для замены переменных, кванторов существования, называются скумелевскими функциями.

Константы есть 0-местные функции.

Пример: получим ССФ:

$A = \exists x \forall y \forall z \exists u \forall v \exists w P(x, y, z, u, v, w)$

В этой формуле левее $\exists x$ нет никаких кванторов всеобщности.

Левее $\exists u$ стоят $\forall y$ и $\forall z$, а левее $\exists w - \forall v, \forall y, \forall z$.

Следовательно, заменим переменную x на константу a, переменную u – на двухместную функцию f(y,z), а переменную w – на трехместную функцию g(y,z,v).

Т.о. получим следующую стандартную форму формулы:

$$S = \forall y \forall z \forall u P(a, y, z, f(y, z), u, g(y, z, u))$$

Если предваренная нормальная форма эквивалентна исходной формуле, то ССФ формулы А – вообще говоря не эквивалентна ей.

Например, пусть формула $A = \exists x P(x)$; $S = P(a)$. Пусть I – есть следующая интерпретация.

I: $Д = \{a, b\}; P(a) = F; P(b) = T$. Тогда А истинно в I, но S – ложно в I.

Таким образом, А не эквивалентно S.

Однако, если P(a)=F и P(b)=F, тогда и А и S принимают значение F для любого x.

Таким образом, $A \equiv S$ в том и только в том случае, если А – противоречиво.

Понятие выводимости в логике предикатов.

Для определения понятия рассмотрим исчисление предикатов (теория первого порядка).

В исчислении предикатов также как и в исчислении высказываний на первом месте стоит проблема разрешимости. Замечание: интерпретируемость, разрешаемость, выполнимость, общезначимость.

Интерпретация – сопоставление каждому элементу высказывания некоторых значений истинности.

Интерпретация, при которой значение формулы истинно, называют моделью формулы.

Формула называется выполнимой, если имеет ее модель. В противном случае формула – невыполнимая.

Формулы, которые всегда истинны или тождественно истинны, называются общезначимыми (или тавтологиями). Поиск таких формул – одна из основных задач логики.

Алгоритмы проверки общезначимости для логики высказываний существуют, поэтому говорят, что логика высказываний разрешима.

В исчислении высказываний проблема разрешимости состоит в решении вопроса – является ли данная функция тождественно истинной, выполнимой, тождественно ложной.
В логике предикатов этот вопрос – принимает ли данная функция значение истинна, при:
1. Любых предметных переменных и любых предикатах.
2. На некотором множестве предметных переменных и любых предикатах.
3. При некотором значении предметных переменных и при некоторых предикатах.
4. Является ли она тождественно ложной, т.е. невыполнимой.
Таким образом, в логике предикатов в отличии от логики высказываний нет эффективного способа для распознавания общезначимости функций.
Поэтому в исчислении предикатов указывается используемая совокупность формул, которая называется аксиомами и составляет аксиоматическую теорию.
Также указывается конечное множество отношений между формулами, составляющая правила вывода.
Аксиоматическая теория и правила вывода составляют исчисление предикатов.
Символьным исчислением предикатов или алфавитом являются символы предметных переменных, символы предикатов, логические символы (□ и →), символы кванторов, а также скобки и запятая.
Сформулируем аксиомы исчисления предикатов и правила вывода исчисления предикатов:
Пусть A, B, C – любые формулы:
A1. $A \to (B \to A)$ (истина «из чего-угодно»)
A2. $\left(A \to (B \to C)\right) \to ((A \to B) \to (A \to C))$
A3. $(\bar{B} \to \bar{A}) \to ((B \to A) \to B)$
A4. $\forall x A(x) \to A(y)$, если y свободно для x в формуле A(x) (формула A(x) не содержит переменной y).
A5. $\forall x\left(A \to B(x)\right) \to (A \to \forall x B(x))$, если A не содержит свободных вхождений x.
Эти формулы являются логическими аксиомами в теории K.
Собственные аксиомы формулируются отдельно для каждой конкретной содержательной предметной области. Во всякой теории первого порядка:
1. modus ponens (MP): из A и A→B следует B.
2. Правило обощения (Gen): если $F \vDash A(x)$, то $\Gamma \vDash \forall x A(x)$, если x не входит свободно ни в одну из формул Г.
Теория K, не содержащая собственных аксиом, называется исчислением предикатов первого порядка. Моделью теории первого порядка K называется всякая интерпретация, в которой истинны все аксиомы теории K.
Если все правила вывода MP и Gen применяются к истинным на данной интерпретации формулам, то результатом являются формулы также истинные в той же интерпретации.
Следовательно, вся теория K – истинная во всякой ее модели.
Множество формул, выводимых по правилу вывода из аксиом теории K, являются теоремами теории K.
Кроме теории K существуют и другие теории, например аксиомы A1, A2, A3 и правило MP определены в теории L.
Следовательно все теоремы теории L включены в множество теорем теории K.
Говорят, что формула B – логически следует из A, если в любой интерпретации, в которой A принимает истинное значение, B также принимает истинное значение. $A \vDash B$
В общем случае формула B является логическим следствием множества формул Г, если она истинна на всех тех интерпретациях, на которых выполнима, т.е. истинны одновременно все формулы из множества Г: $\Gamma \vDash B$.
Говорят, что A равносильна (логически эквивалентна) формуле B, если каждая из них логически влечет другую: $A \vDash B \& B \vDash A \ => A <=> B$.
Из определений следуют следующие утверждения:
1. $A \vDash B$ тогда и только тогда, когда $\vDash A \to B$
2. $A_1, A_2, \dots, A_n \vDash B$ тогда и только тогда, когда $\vDash A_1 \& A_2 \& \dots \& A_n \to B$
3. $A <=> B$ только тогда, когда $\vDash A \equiv B$
4. Если $A \vDash B$ и $|A| = T$, то $|B| = T$ - в некоторой интерпретации

5. Если $\Gamma \vDash B$ и $\forall i|\Gamma_i| = T$, то $|B| = T$

Для рассмотрения вопроса выводимости исчисления предикатов, рассмотрим подробно следующие аксиомы:

$\forall x(G(x) \rightarrow G(y))$ *(1)*

$H(y) \rightarrow \exists x H(x)$ *(2)*

Смысл аксиомы (1): если предикат G(x) – истинен ($\forall x$), то он истинен и для любого y.

Смысл аксиомы (2): если предикат H(y) истинен для какого-нибудь y, то существует такой x, что H(x) – истинен.

1. Правило для квантора всеобщности (∀):

если $\varphi_1 \rightarrow \varphi_2$ выводимо и φ_1 не имеет *x* в качестве свободной переменной, а φ_2 – переменная x содержится в виде свободной переменной, тогда $\varphi_1 \rightarrow \forall x \varphi_2$ также выводимо.

2. Правило для квантора существования (∃):

если $\varphi_1 \rightarrow \varphi_2$ выводимо и x содержится в качестве свободной переменной в φ_1 и не содержится в качестве свободной переменной в φ_2, то формула $\exists x \varphi_1 \rightarrow \varphi_2$ также выводима.

3. Правило переименования связанных переменных:

если φ_1 выводимая формула и в φ_1 имеется квантор всеобщности или квантор существования, то одна связанная переменная φ_1 может быть заменена другой связанной переменной одновременно во всех областях действия квантора и в самом кванторе. Полученная формалв также выводима.

Правила логического вывода. Вывод на основе принципа резолюции.

Правило логического вывода:

Вывод заключенный из множества посылок записывается также как и в исчислении высказываний:

$$F_1, F_2, \dots, F_n \vdash B$$

где слева от знака вывода записывается множество формул посылок и необходимые аксиомы, а справа формулу заключение B.

Знак ⊢ означает что верно, что B выводится из F_1, F_2, …, F_n.

Отношение логического вывода эквивалентно теореме: $\vdash F_1, F_2, \dots, F_n \rightarrow B$.

Другая формула записи логического вывода: $\frac{F1, F2, \dots, Fn}{B}$.

Поскольку для исчисления высказываний известно, что теоремами являются общезначимые формулы можно воспользоваться простым методом проверки общезначимости формулы с помощью таблиц истинности, а именно достаточно вычислить истинностное значение формулы при всех возможных интерпретациях. Если во всех случаях получится значение истина, то проверяемая формула тавтология. Следовательно является теоремой теории L. Если хотя бы в одном случае получится значение ложь, то проверяемая формула не является тавтологией и следовательно не является теоремой теории L. Приведенный пример является алгоритмом автоматического доказательства теорем теории L. Хотя и не является алгоритмом автоматического поиска вывода из аксиом теории L.

Наиболее известный алгоритм автоматического доказательства теорем называется методом резолюций. Для любого прикладного исчисления предикатов первого порядка любой формулы *S* и множества формул *Г* теории τ метод резолюции выдает ответ ДА если верно, что Г⍰$_{\tau}$S, или не выдает никакого ответа (зацикливается) или выдает ответ НЕТ, если неверно Г⍰$_{\tau}$S.

В основе метода резолюций лежит идея доказательства от противного. Которая формулируется в виде теоремы следующим образом:

Если из множества формул Г, $S \vdash F$, где F – любое противоречие, то есть тождественно ложная формула, то верно $\Gamma \vdash S$.

Метод резолюций работает с особой стандартной формой формул, которые называются предложениями. Предложение – бескванторная дизъюнкция литералов. Любая формула исчислений может быть преобразована в множество предложений и последовательность преобразований формулы исчислений:

1. Иллюминация импликаций.

$A \rightarrow B| \rightarrow \bar{A} \vee B$

После первого этапа формула содержит: ⍰, ∧,∨, ∀, ∃
2. Протаскивание отрицаний.
$\overline{\forall x A} \mid \rightarrow \ \exists x \bar{A}$
$\overline{\exists x A} \mid \rightarrow \ \forall x \bar{A}$
$\bar{\bar{A}} \mid \rightarrow A$
$(\overline{A \vee B}) \mid \rightarrow \bar{A} \wedge \bar{B}$
$(\overline{A \wedge B}) \mid \rightarrow \bar{A} \vee \bar{B}$
После второго этапа формула содержит отрицание только перед атомами.
3. Разделение связанных переменных.

$$Q_1 x A\big(\dots Q_2 x B(\dots x \dots)\big) \mid \rightarrow \ Q_1 x A(\dots \ Q_2 y B(\ \dots y \dots)\dots)$$

Где Q_1 и Q_2 – любые кванторы.
После третьего этапа формула не содержит случайно совпадающих связанных переменных
4. Приведение к предваренной форме:
$QxA \vee B \mid \rightarrow Qx(A \vee B); QxA \wedge B \mid \rightarrow Qx(A \wedge B)$
После четвертого этапа формула находится в предваренной форме.
5. Иллюминация кванторов существования (сколенизация).

$$\exists x_1 Q_2 x_2 \dots Q_n x_n A(x_1, x_2, \dots, x_n) \mid \rightarrow Q_2 x_2 \dots Q_n x_n A(a, x_2, \dots, x_n)$$

$$\forall x_1 \dots \forall x_{i-1} \exists x_i Q_{i+1} \dots Q_n x_n A(x_1, \dots, x_i, \dots, x_n) \mid \rightarrow$$

$$\mid \rightarrow \forall x_1 \dots \forall x_{i+1} Q_{i+1} x_{i+1} \dots Q_n x_n A(x_1, \dots, f(x_1, \dots, x_{i-1}), \dots, x_n)$$

a – новая предметная констакта, f – новый функциональные символы, Q_1, .., Q_n – любые кванторы.
После пятого этапа, формула содержит только кванторы всеобщности.
6. Импликация кванторов всеобщности:

$$\forall x A(x) \mid \rightarrow A(x)$$

формула не содержит кванторов.
7. Приведение к КНФ:
$A \vee (B \wedge C) \mid \rightarrow (A \vee B) \wedge (A \vee C)$
$(A \wedge B) \vee C \mid \rightarrow (A \vee C) \wedge (B \vee C)$
Формула находится в КНФ.
8. Импликация конъюнкций.
$A \wedge B \mid \rightarrow A, B$
Формула распадается на множество предложений.
Не все преобразования на этапах 1-8 являются логически эквивалентными.
Если Г-множество предложений, полученное из формулы S, то S – является противоречием тогда и только тогда, когда множество Г невыполнимо.
Множество формул Г невыполнимом, это означает, что множество Г не имеет модели, т.е. не существует интерпретации, на которой все формулы Г имели бы значение истина.

Правило резолюций для исчисления высказываний.
Метод резолюций основан на использовании специального правила вывода:
Пусть C_1, C_2 – 2 предложения в исчислении высказываний. Пусть $C_1 = P \vee C_1'$; $C_2 = \bar{P} \vee C_2'$, где P – пропозиционная переменная, C'_1, C'_2 – любые предложения, в частности могут быть пустые и состоящие из одного литерала.
Правило вывода называют правилом резолюций: $\frac{C_1, C_2}{C_1' \vee C_2'} R$.
Предложения C_1, C_2 называют родительскими (резольвирующими).
$C_1' \vee C_2'$ - резольвента.
Формулы P и $\bar{P}$ – контрарные литералы.
Резольвента является предложением, т.е. множество предложений замкнуты вокруг правила резолюций.
Многие ранее рассмотренные правила являются частными случаями правила резолюций.
Например:
1. $\frac{A, A \rightarrow B}{B}$ - modus poners
$\frac{A, \bar{A} \vee B}{B} R$ - частный случай правила резолюций

2. $\frac{A\to B, B\to C}{A\to C}$ - транзитивность $\frac{\bar{A}\vee B, \bar{B}\vee C}{\bar{A}\vee C}R$
3. $\frac{A\vee B, A\to B}{B}$ - слияние $\frac{A\vee B, \bar{A}\vee B}{B}R$
Правило резолюций полно, т.е. всякая тавтология может быть выведена, только с использование правила резолюций.

Правило резолюций для исчисления предикатов.

Для применения правила резолюции нужны контрарные литералы в резольвируемом предложении. В случае исчисления высказываний, контрарные литералы вычисляются очень просто – это пропозиционная переменная и ее отрицание. Для исчисления предикатов определение несколько сложнее.

Пусть C_1 и C_2 – предложения в исчислении предикатов.

Правило вывода:

$\frac{C_1, C_2}{(C_1' \vee C_2')_\sigma}R$ - правило резолюций в исчислении предикатов, если в предложениях C_1, C_2 существуют унифицируемые контрарные литералы P_1 и P_2. Т.е. $C_1 = P_1 \vee C_1';\ C_2 = \overline{P_2} \vee C_2'$.

Причем атомарные формулы P_1 и P_2 являются унифицированными наиболее общим унификатором δ.

В этом случае, резольвентой предложений C_1, C_2 является $(C_1' \vee C_2')_\sigma$, полученная из предложения $C_1' \vee C_2'$ с применением унификатора δ.

Опровержение методов резолюций – это алгоритм автоматического доказательства теорем в прикладном исчислении предикатов, который сводится к следующему:

Пусть нужно установить выводимость: $S \vdash G$. Каждая формула множества S и формула отрицание G независимо преобразуются в множество предложений. В полученной совокупности множества предложений C отыскивается резольвированные предложения. К ним применяется правило резолюций и резольвента добавляется в множество до тех пор, пока не будет получено пустое предложение. При этом возможны 3 случая:

1. Среди текущего множества предложений нет резольвируюемых (или все резольвенты уже присутствуют в текущем множестве предложений) – это означает, что теорема опровергается, т.е. формула G не выводится из множества формул S.
2. В результате очередного применения правила резолюций получено пустое предложение, это означает, что доказана теорема: $S \vdash G$.
3. Процесс не заканчивается, т.е. множество предложений пополняется все новыми резольвентами, среди которых нет пустых. Это ничего не означает.

Пример: докажем методом резолюций теорему в исчислении высказываний:

$$\vdash_L (((A \to B) \to A) \to A)$$

Сначала нужно преобразовать в предложение отрицание целевой формулы:

$$\overline{(((A \to B) \to A) \to A)}$$

1. $\overline{(\overline{\overline{(\overline{(\bar{A} \vee B)} \vee A)}} \vee A)}$
2. $(((A \wedge \bar{B}) \vee A) \wedge \bar{A}$
3-6. Формулы без изменений.
7. $(A \vee A) \wedge (\bar{B} \vee A) \wedge \bar{A}$
8. $A \vee A, \bar{B} \vee A, \bar{A}$

После этого, проводится резольвирование предложений 1, 2, 3.

1. $A \vee B$
2. $\bar{B} \vee A$
3. $\bar{A} \vee$
4. A (из 1 и 3 по правилу резолюций)
5. (из 3 и 4 по правилу резолюций)

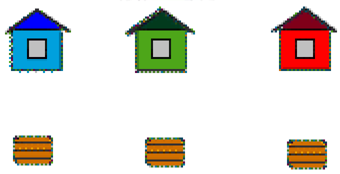
Таким образом, теорема доказана.

Графы. Сети. Автоматы.

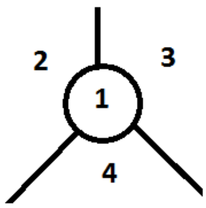
Основные понятия и определения теории графов. Способы задания графов. Эйлеровы графы. Деревья. Сети.

Основные задачи, которые привели к созданию теории графов.

Задача о Кенингсберских мостах.
Обойти все четыре части суши, пройдя по каждому мосту один раз и вернуться в исходную точку.



Задача о трех домах и трех колодцах.
Имеется три дома и три колодца. Провести от каждого дома по тропинке так, чтобы они не пересекались.



Задача о четырех красках. Любую карту на плоскости раскрасить четырьмя красками так, чтобы никакие две соседние области не были закрашены одним цветом.

Графом G(V,E) называется совокупность двух множеств: непустого множества вершин V и множество E неупорядоченных пар различных элементов множества V. Множество E называется множеством ребер $G(V,E) \coloneqq < V, E > \mid |V| \neq \emptyset, E = V * V, E = E^{-1}$

Число вершин графа $p(G) = |V|$.

Число ребер графа $q(G) = |E|$.

Пусть v_1, v_2 – вершины графа G(V,E) с множеством вершин V. Тогда e=(v_1, v_2) – ребро, соединяющее эти вершины. Тогда говорят, что вершины v_1 и v_2 – смежные, поскольку имеют общее ребро. То, что вершина v_1 является концом ребра e, говорят v_1 и e – инцидентны.

Два ребра инциндентны одной вершине - смежные. Две вершины инциндентные одному ребру – называются смежными.

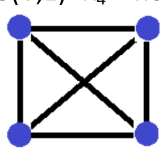
Множество вершин, смежными с вершиной ω называют множеством смежности множества ω. Обозначают: $\Gamma(\omega)=\{u \mid u \in V, (u, \omega) \in E\}$

Способы задания графов.

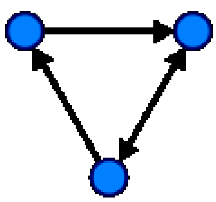
Обычно граф изображают диаграммой в которой вершины представлены окружностями, а ребра – линиями.

Полным является граф, у которого все вершины смежны между собой.

G(V,E)=K_4 – полный граф с четырьмя вершинами.

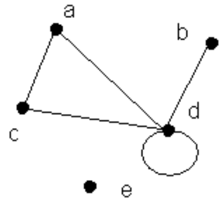


Если элементами множества E являются упорядоченные пары, то граф называется ориентированным или орграфом.

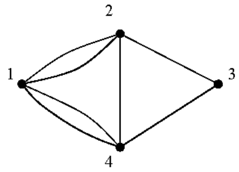


Ребра в таком орграфе называют дугами.

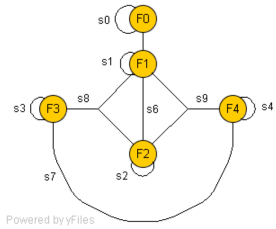
Если элементом множества E может быть пара одинаковых элементов множества V, то такой элемент множества E называют петлей. А граф называют – графом с петлями или псевдографом.



Если E не является множеством, а набором (кортежем), содержащим несколько одинаковых элементов, то эти элементы называют кратными ребрами, а граф – мультиграфом.



Если элементами множества E являются необязательно двух элементные, то такие элементы E называют гипердугами, а граф – гиперграфом.



Если задана функция F: V→M v E→M, то M – множество пометок, а граф помеченный или нагруженный.

Основной способ задания – в виде матрицы смежности вершин. Элемент M_{ij} равен 1 если, (v_I, v_j)⍰E и равен 0 если, (v_I, v_j)E

M=

	1	2	3	4
1	0	1	1	0
2	1	0	1	1
3	1	1	0	1
4	0	1	1	0

Матрица инцидентности. Представление графа G(V,E) с помощью матрицы H, отражающей инцидентность вершин и ребер.
Для неориентированного графа: H_{ij} равен 1, если v_i инцидентна ребру e_j, иначе 0.

H_g=

	1	2	3	4	5
1	1	1	0	0	0
2	0	1	1	0	1
3	1	0	0	1	1
4	0	0	1	1	0

Граф задается также списками смежности. Это списочная структура отражающая смежность вершин и состоит из массива указателей на списки смежных вершин.

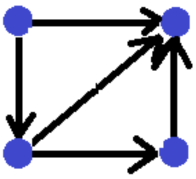
Валентность – количество ребер инцидентных вершине ω. Обозначают d(ω).
Для любой ω: 0 ≤ d(ω) ≤ p-1.

Минимальная степень вершины графа G обозначается δ(G): min(d(ω)).
Максимальная степень вершины графа G обозначается Δ (G) man(d(ω))
Если степени от всех вершин равны, то k=d(ω)= δ(G)= Δ(G) – граф регулярный.
r(G) – степень регулярности графа. Для нерегулярных графов r(G) неопределенно.

Для орграфа число дуг исходящих из вершины ω называют полустепенью исхода и обозначают $d^-(\omega)$, а число дуг входнящих в вершину ω называют полустепенью захода и обозначают $d^+(\omega)$.

$H_g = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 1 & 0 & 1 \\ 3 & 1 & 0 & 0 & 1 & 1 \\ 4 & 0 & 0 & 1 & 1 & 0 \end{array}$ $d(1)=2$; $d(2)=3$; $d(3)=3$; $d(4)=2$.

Определим полустепень захода всех вершин графа.

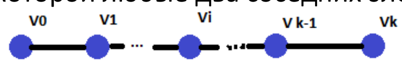


$d^-(1)=2$; $d^-(2)=2$; $d^-(3)=1$; $d^-(4)=0$;
$d^+(1)=0$; $d^+(2)=1$; $d^+(3)=1$; $d^+(4)=3$.

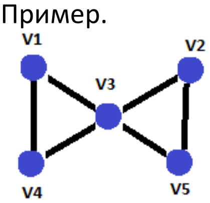
Ориентированный граф имеющий две вершины, в одной (сток) из которых полустепень захода равна нулю, а у другой (исток) полустепень исхода равна нулю называют сетью.

Теорема Эйлера. Сумма степеней вершин графа равна удвоенному количеству ребер.
Теорема справедлива как и для неориентированных графов, так и для ориентированных графов.
Для неориентированных графов: $\sum_{\omega\in V} d(\omega) = 2q$
Для ориетированных графов: $\sum_{\omega\in V} d^-(\omega) + \sum_{\omega\epsilon V} d^+(\omega) = 2q$
Доказательство:
При подсчете суммы степеней вершин графа – каждое ребро учитывают два раза.

Маршруты. Цепи. Циклы.
Маршрутом в графе G(V,E) называют чередующуюся последовательность вершин и ребер, в которой любые два соседних элемента инцидентны.



Если $v_0=v_k$, то маршрут называют замкнутым.
Если все ребра различны, то маршрут называют цепью.
Если вершины, а значит и все ребра, различны – то маршрут называют простой цепью.
Обозначают маршрут $<v_0,v_k>!$. Восклицательный знак – обозначение простой цепи.
$|<v_0,v_k>|$ - длина маршрута $<v_0,v_k>$.
В маршруте v_0 и v_k называют концами. Замкнутая цепь называется циклом. Замкнутая простая цепь называется простым циклом. Число циклов в графе обознают Z(G).
Граф без циклов Z(G)=0 называют ациклическим.
Пример.



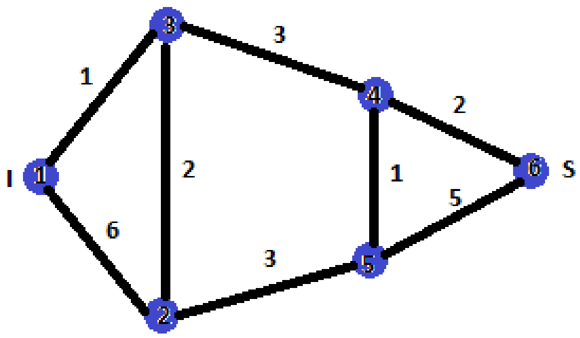
v_1-v_3-v_4-v_3 – маршут, но не цепь
v_1-v_3-v_5-v_2-v_3-v_4 – цепь, но не простая цепь
v_1-v_4-v_3-v_5 – простая цепь
v_1-v_3-v_5-v_2-v_3-v_4-v_1 – цикл, но не простой цикл
v_1-v_3-v_4-v_1 – простой цикл

Длиной маршрута называют количество ребер в нем (учитывая повторения).

Расстоянием между вершинами u и v, обозначается d(u,v) называется длина кратчайшей цепи d(u,v) = min |<u,v>|, а сама кратчайшая цепь называется геодезической цепью.
Множество вершин находящихся на одинаковом расстоянии n от вершины v называют ярусом.

Задачи оптимизации решаемые на графах: задача о максимальном потоке в сети, задача о кратчайшем пути.

Пример практической задачи о максимальном потоке. Пусть имеется сеть трубопроводов соединяющих пункт А с пунктом Б. Трубопроводы могут соединяться и разветвляться в промежуточных пунктах. Количество ресурса, которое может быть перекачено по каждому отрезку трубопровода в единицу времени не безгранично, и определяется диаметром трубы, мощностью насоса. Вопрос: сколько ресурса можно прокачать через эту сеть трубопровода в единицу времени?



$d^+(I)=0$
$d^-(S)=0$
Пусть G(V,E) – сеть с двумя особыми вершинами исток и сток. До 10 нагруженный неотрицательными вещественными числами.
Множество вещественных чисел будем формулировать следующим образом. $C:E \rightarrow R^+$.
Числа С будем называть пропускными способностями дуги. Пропускная способность дуги – максимальное количество ресурса, которое может быть передано по дуге в единицу времени. По дугам сети можно пропустить некоторый поток по пути из истока в сток.
Будем рассматривать поток из истока в сток.
Рассмотрим поток по пути μ_1=<1-3-4-6>. Введем понятие мощности потока по пути и будем обозначать его $|\mu_1|$. Мощность потока по пути – количество ресурса, пропускаемого по заданному пути в единицу времени.
Мощность потока будет определяться выражением $|\mu_1|=\min C_{ij}=\min(1,3,2)=1$
Рассмотрим поток по пути μ_2=<1-2-5-6>
Определим мощность потока $|\mu_2|=\min C_{ij}=\min(6,3,5)=3$
Обратим внимание на дуги:
(1,3): $x_{13}=C_{13}$
(2,5): $x_{25}=C_{25}$
Дуга, по которой пропущено количество ресурса в единицу времени, совпадающей с пропускной способностью этой дуги называется насыщенной.
Построим матрицу пропускных способностей транспортной системы.

C=

	1	2	3	4	5	6
1		6	1			
2	6		2		3	
3	1	2		3		
4			3		1	2
5		3		1		5
6				2	5	

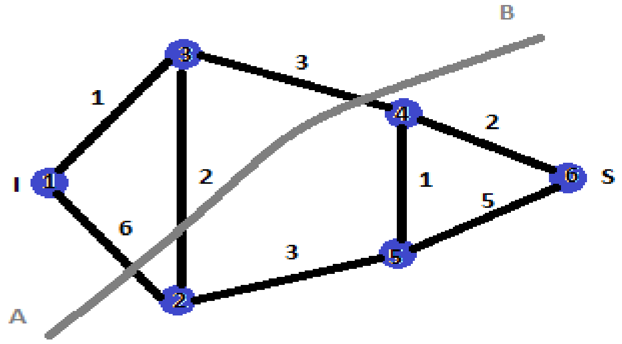
Матрицу потока по сети

$X^0=$

	1	2	3	4	5	6
1		3	1			
2	-3				3	
3	-1			1		
4			-1			1
5		-3				3
6				-1	-3	

Разобъем множество вершин сети на два непересекающихся подмножества. Чтобы множество вершин сети $V = A \cup B; \emptyset = A \cap B, I \in A \& S \in B$

В этом случае говорят что на сети произведен разрез, отделяющий исток от стока. Разрез это совокупность ребер.



$A \backslash B = \{(i,j) | i \in A, j \in B, i,j \in V\}$

A\B={(3,4),(3,2),(1,2)}

Для разреза вводят понятие пропускной способности. Пропускная способность разреза $C_{A\backslash B}$ – сумма пропускных способностей ребер этого разреза.

$$C_{A\backslash B} = \sum_{(i,j)\in A\backslash B} C_{ij} = C_{34} + C_{32} + C_{12} = 3 + 2 + 6 = 11$$

Теорема Форда-Фалкерсона.

На любой сети максимальная величина потока из источника *I* в сток *S* равна минимальной пропускной способности разреза, отделяющего источник от стока.

Алгоритм нахождения максимального потока.

1. Построить некоторый начальный поток по сети X_0.
2. Организовать процедуру построения подмножества *A* вершин, достижимых из источника *I* по ненасыщенным ребрам. Если в этом процессе сток *S* не попадет в подмножество *A* вершин, то построенный поток максимален и задача решена. Если *S* попадает в подмножество *A*, то перейти к шагу 3.
3. Выделить путь из источника в сток по ненасыщенным ребрам и увеличить поток x_{ij} на величину $\Delta=\min_{i,j\in\mu k}(C_{ij}-X_{ij})$. Тем самым будет построен новый поток по сети. Перейти к шагу 2.

Пример (продолжение)

3.

	1	2	3	4	5	6
1		3	0			
2	9		2		0	
3	2	2		2		
4			4		1	1
5		6		1		2
6				3	8	

2.

1||2,

2||3,

3||4

4||5,6

μ_3=<1-2-3-4-6>

$|\mu_3|$=min C_{ij}=min(3,2,2,1)=1

μ_3=

	1	2	3	4	5	6
1		1				
2	-1		1			
3		-1		1		
4			-1			1
5						
6				-1		

$(C-X_0)-\mu_3$=

	1	2	3	4	5	6
1		2	0			
2	10		1		0	
3	2	3		1		
4			5		1	0
5		6		1		2
6				4	8	

1||2

2||3

3||4

4||5

5||6

μ_4=<1-2-3-4-5-6>

$|\mu_4|$=min C_{ij}=min(1,1,1,1,2)=1

μ_4=

	1	2	3	4	5	6
1		1				
2	-1		1			
3		-1		1		
4			-1		1	
5				-1		1
6					-1	

$(C-X_0)-\mu_3-\mu_4$=

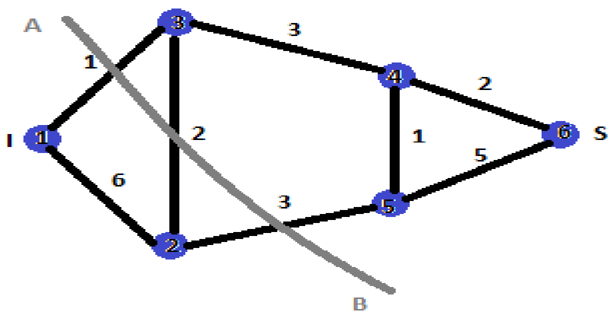
	1	2	3	4	5	6
1		1	0			
2	11		0		0	
3	2	4		0		
4			6		0	0
5		6		2		1
6				4	9	

1||2

2|| -

Строим разрез A={1,2} , B = {все оставшиеся}

Пропускная способность полученного разреза: 1+2+3=6



min $C_{A\setminus B}$=max X.

Т.о. по теореме Форда-Фалкерсона получено максимальный поток равен 6.

Другой задачей оптимизации на графе является задача нахождения кратчайшего пути в графе. Известно несколько алгоритмов:

1. Флойда – находит кратчайшие пути между всеми парами вершин в орграфе. В этом алгоритме для хранения информации о путях используется матрица H(1..p, 1..p), где H_{ij} равен k, если k – первая вершина достигаемая на кратчайшем пути из i в j, 0 – если из i вершины в j нет вершин.
2. Дейкстры.

<u>Задачи коммивояжера. Задача о минимальной раскраске графа.</u>

Деревья. Свободные деревья. Основные свойства деревьев.

Граф без циклов называют **ациклическим** или **лесом**. Связный ациклический граф называют **деревом** или **свободным деревом**.

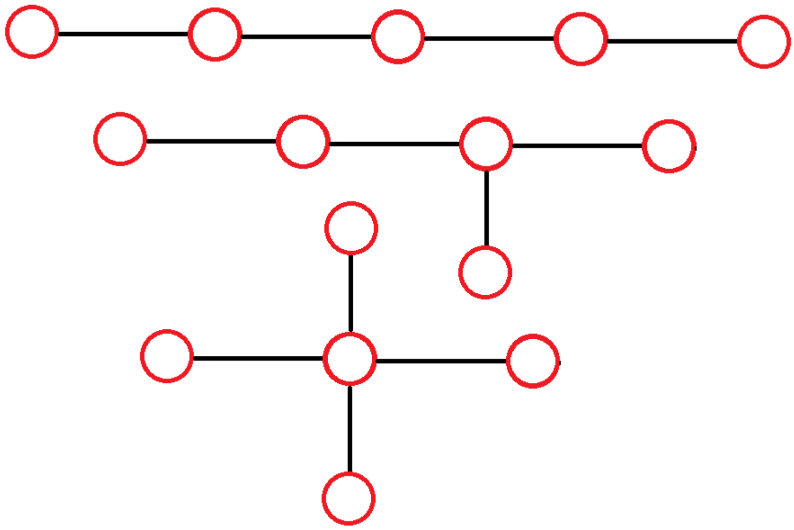
В связном графе G выполняется неравенство q(G)≤P(G)-1.

Граф G в котором q(G)=P(G)-1 называют **древовидным**.

В ациклическом графе число циклов z(G)=0.

Пусть u и v две несмежные вершины графа G. Пусть x – ребро соединяющее вершины u, v. x=(u,v) | x$(\text{не} \in)E$

Тогда z(G+x) имеет только один простой цикл, тограф называют **субциклическим.** Приведем пример графов-деревьев в которых P(G)=5



Свойства деревьев.

Теорема о свойствах деревьев устанавливает, что два из четырех свойств деревьев – связность, ацикличность, древовидность, субцикличность.

Теорема. Пусть G(V,E) граф с p вершинами, q – ребрами, k – компонентами связности и z – простыми циклами. Пусть x ребро соединяющее любую пару несмежных вершин в графе G. Тогда следующие убеждения эквивалентны:

1. G: – дерево, т.е. связные граф без циклов

$$\mathrm{G}\colon \mathrm{k}(\mathrm{G}) = 1;\ \mathrm{z}(\mathrm{G}) = 0$$

2. Любые две вершины в графе G соединены единственной простой цепью.

$$\forall u, v \exists! < u, v >$$

3. G связный граф и любое ребро есть мост.

$$k(G) = 1, \forall e \in E\ k(G - e) > 1$$

4. Граф G связный и древовидный.

$$k(G) = 1, q(G) = p(G) - 1$$

5. Граф G ациклический и древовидный.

$$z(G) = 0, q(G) = p(G) - 1$$

6. Граф G ациклический и субциклический

$$z(G) = 0, z(G + x) = 1$$

7. Граф G связный, субциклический и неполный.

$$k(G) = 1, z(G + x) = 0, G \neq k_p, p \geq 3$$

8. Граф G древовидный и субциклический за двумя исключениями.

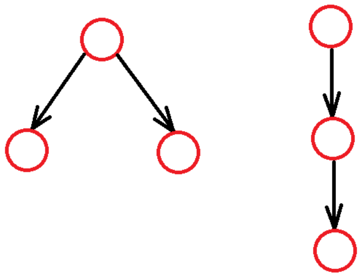
$$q(G) = p(G) - 1, G \neq K_1 \cup K_3, G \neq K_2 \cup K_3, z(G + x) = 1$$

Ориентированные деревья.

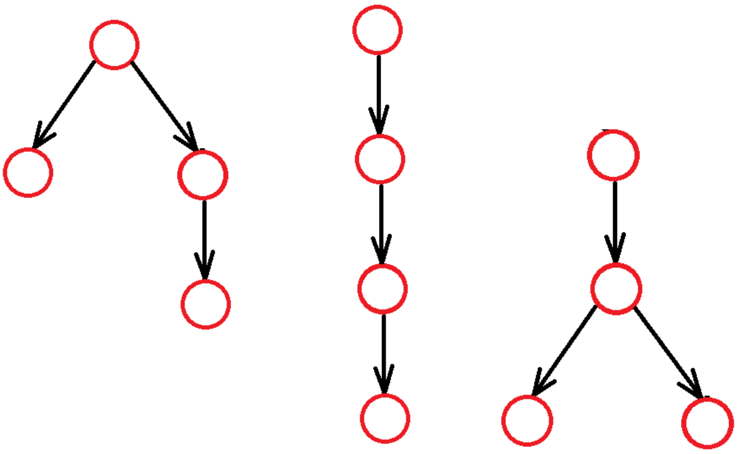
Ориентированное дерево или ордерево – это орграф со следующими свойствами:

1. Существует единственный узел полстепень захода которого равна 0. Он называется **корнем** ордерева.
2. Полустепень захода всех остальных узлов равна единице.
3. Каждый узел достижим из корня.

Пример диаграммы ордерева с 3 узлами:



Пример диаграммы ордерева с 4 узлами:



Свойства ордерева:

1. Число ребер на единицу меньше числа узлов. q=p-1.
2. Если в ордереве отменить ориентацию ребер то получится свободное дерево.
3. В ордереве нет контуров.
4. Для каждого узла существует единственный путь ведущий в этот узел из корня.
5. Подграф определяемый множеством узлов достижимых из некоторого узла v является ордеревом с корнем v.
6. Если в свободном дереве любую вершину назначить корнем, то получится ордерево.

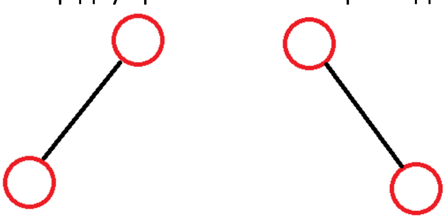
Эквивалентное определение ордерева. Ордерево T это конечное множество узлов таких, что:

1. Имеется один узел r называемый *корнем дерева.*
2. Остальные узлы, исключая корень r, содержатся в k попарно непересекающихся множествах T_1, …, T_k, k>0, каждое из которых является ордеревом.

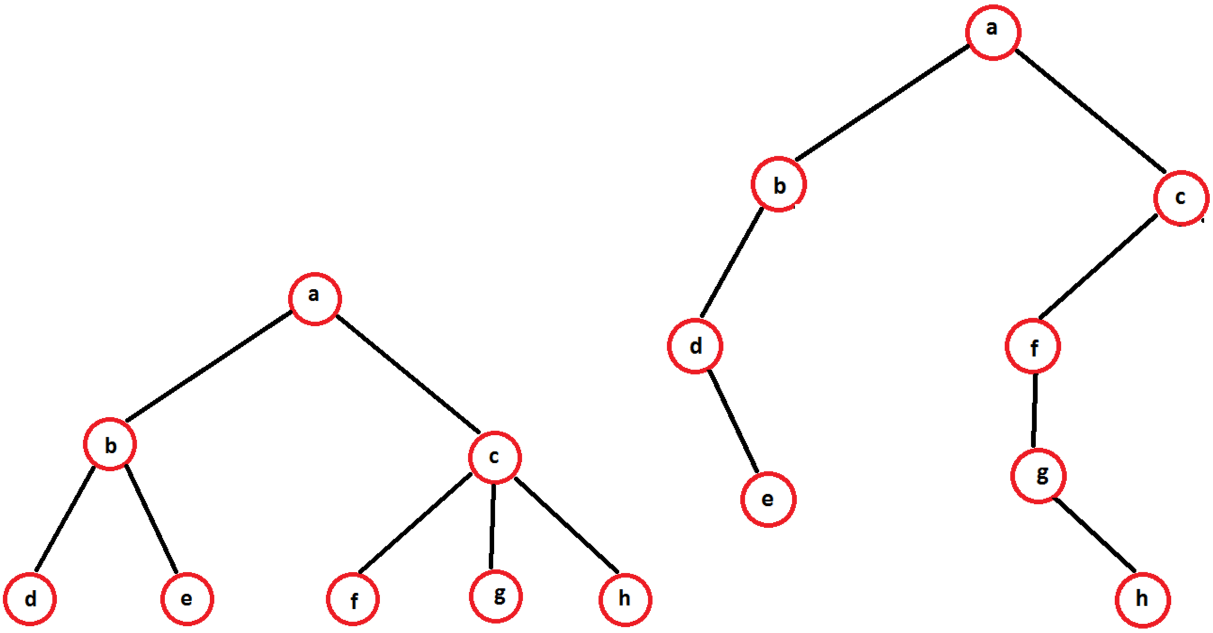
$T=\{\{r\}, T_1, …, T_k\}$. Если относительный порядок поддеревьев T_1, …, T_k фиксирован, то ордерево называется упорядоченным.

Бинарное дерево - это конечное множество узлов, которое либо пусто, либо состоит из корня и двух непересекающихся бинарных деревьев. Бинарное дерево не является упорядоченным поддеревом.

Пример двух различных бинарных деревьев.



Всякое свободное дерево можно ориентировать, назначив один из узлов корнем. Всякое ордерево можно произвольно упорядочить. Всякое упорядоченное дерево можно представить бинарным деревом, проведя правую связь к старшему брату, а левую к младшему сыну. Приведем диаграмму упорядоченного дерева и соответствующего ему бинарного дерева.



Если граф имеет цикл, необязательно простой, содержащий все ребра графа, то такой цикл называют эйлеровым циклом, а граф называют эйлеровым графом.

Если граф имеет цепь, необязательно простую, содержащую все вершины по одному разу, то такая цепь называется эйлеровой цепью, а граф называют полуэйлеровым графом.

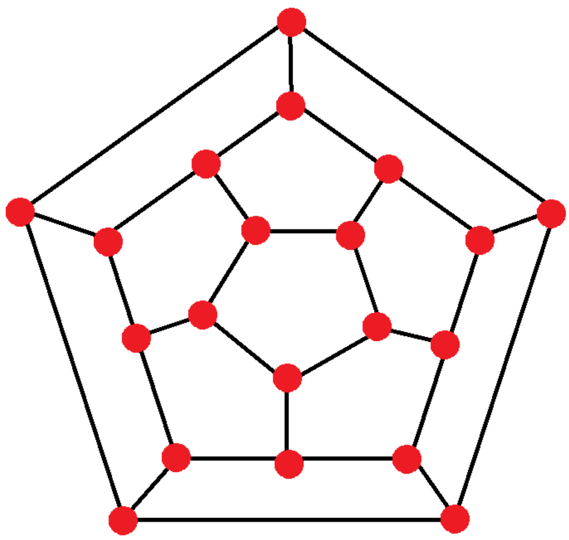
Теорема. Если граф G связен и не тривиален то следующие утверждения эквивалентны.

1. Граф G эйлеров граф.
2. Каждая вершина графа G имеет четную степень.
3. Множество ребер графа G можно разбить на простые циклы.

Теорема дает решение задачи о кенигсбергских мостах. Т.е. для того чтобы обойти все ребра по одному разу – степени все вершин должны быть четны.

Гамильтоновы циклы.

Название гамильтонов цикл произошло от задачи о кругосветном путешествии придуманной Гамильтоном (1805-1856). С точки зрения теории графов она формулировалась следующим образом: нужно обойти все вершины графа по одному разу и вернуться в исходную точку. Граф представлял собой уклад додекаэдра.



Если граф имеет простой цикл содержащий все вершины графа по одному разу, то такой цикл называют **гамильтоновым циклом,** а граф называют **гамильтоновым графом.** Гамильтонов цикл не обязательно содержит все ребра графа.

Теорема Дирака. Если в графе G(V,E) для любой $v \in V$ степень вершины $d(v) \geq \frac{p}{2}$, то граф G называется гамильтоновым.

Задача комивояжера. Имеется p городов, расстояния между которыми известны. Комивояжер должен посетить все p городов по одному разу, вернувшись в тот, с которого начал. Требуется найти такой маршрут движения при котором суммарное пройденное расстояние будет минимальным.

Задача коммивояжера – задача отыскания кратчайшего гамильтонова цикла в полном графе.

Раскраска графа. Раскраской графа называется такое предписывание цветов – натуральных чисел - его вершинам, что никакие две смежные вершины не получают одинаковых цветов. Наименьшее возможное число цветов в раскраске называют хроматическим числом и обозначают χ(G). Очевидно, что существует некоторая m-раскраска графа, причем $\chi(\mathrm{G}) \leq \mathrm{m} \leq \mathrm{p}$. Множество вершин, покрашенных в один цвет, называют **одноцветным классом**. Одноцветные классы образуют **независимые множества вершин,** т.е. никакие две вершины в одноцветном классе не смежны.

Пример хроматических чисел для различных графов.

$\chi(K_p)=p$

$\chi(\overline{K_p})=1$

$\chi(T)=2$

Для хроматического числа доказана **теорема**.

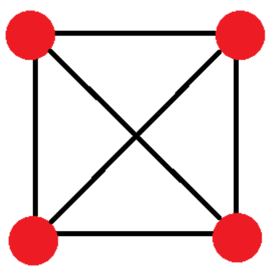
$$\chi(\mathrm{G}) \leq 1 + \Delta(G)$$, где $\Delta(G)$ - максимальная степень свободных вершин графа.

Говорят, что граф **<u>укладывается</u>** на некоторой поверхности, если его можно нарисовать на этой поверхности так, чтобы ребра графа при этом не пересекались.

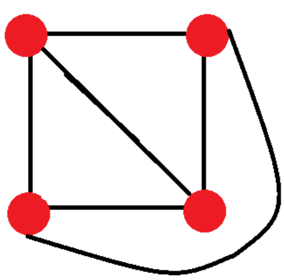
Граф называется **планарным** если его можно уложить на плоскости. Плоский граф – граф G уложенный на плоскости.

Область, ограниченная ребрами в плоском графе и не содержащая внутри себя вершин и ребер называется гранью. Число граней плоского графа обозначают r(G).

Пример. Является ли граф K_4 планарным?



Является.



$r(K_4)=4$

Для графов справедливо соотношение, которое называется **эйлеровой характериской поверхности** или **формулой эйлера.**

$$P - q + r = 2$$

Проверим для нашего примера K_4: 4-6+4=2.

Если граф G – связный планарный граф, где p>3, то $q \leq 3p - 6$

Следствие из формулы эйлера для связного планарного графа: K_5 и $K_{3,3}$ не планарны.

Всякий планарный граф можно раскрасить пятью красками.

Алгоритм раскрашивания.

Рассмотрим следующую схему рекурсивной процедуры P.

1. Выбрать в графе G некоторое максимальное независимое множество вершин S.
2. Покрасить вершины множества S в очередной цвет.
3. Применить процедуру P к графу G-S.

Понятие автомата. Автомат Мили и Мура. Их описание с помощью графа состояния. Построение автомата по блок-схеме алгоритма.

Теория автоматов представляет собой раздел дискретной математики изучающий модели, преобразователи дискретной информации. Такими преобразователями являются как реальные устройства: компьютеры, живые организмы; так и воображаемые устройства: аксиоматические теории, математические машины. Конечный автомат можно охарактеризовать как устройство имеющее входной и выходной каналы, при этом в каждый дискретный момент времени – тактные моменты времени – оно находится в одном из конечных состояний.

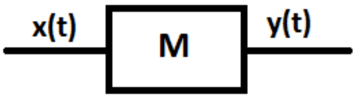


Рисунок – Конечный автомат.

Конечный автомат является математической моделью реальных дискретных устройств по переработке информации. Конечным автоматом называется структура $M =< X, Q, Y, \lambda, \delta >$

Множество X – входной алфавит,

Множество Q – множество состояний, его элементы – состояния.

Множество Y – выходной алфавит, его элементы – выходные символы, последовательность выходных символов – выходные слова.

Функция δ: $X * Q \rightarrow Q$ - функция переходов

Функция λ: $X * Q \rightarrow Y$ - функция выходов

$$\delta(x, q) \epsilon Q$$
$$\lambda(x, q) \epsilon Y$$

Под воздействием символа x(t) автомат перейдет в новое состояние q(t+1), и выдаст выходной сигнал y(t). Величины x(t), q(t+1), q(t), y(t) связаны между собой уравнениями:

q(t+1)= $\delta\big(x(t), q(t)\big)$

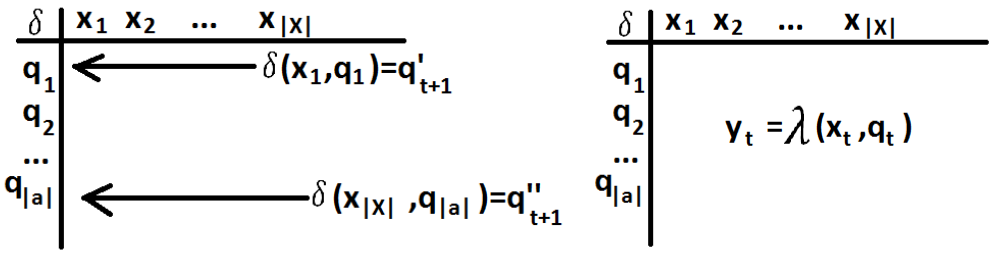
y(t)= $\lambda(x(t), q(t)$

Эти уравнения называют каноническими уравнениями.

Существует несколько эквивалентных способов задания абстрактных конечных автоматов.

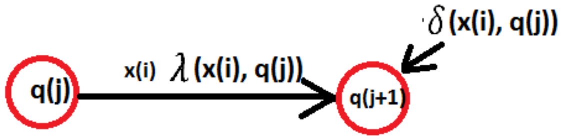
1. Табличный.
2. Графический.
3. Матричный.
4. Функциональный.

1. Табличный



2. Графический

При этом способе состояние автомата q(j) изображают кружками, в которые вписывают состояния q(j). Из каждого кружка проводятся стрелки – ориентированные ребра – причем количество выходящих дуг из каждой вершины определяется мощностью множества X. Дуге соответствующей символу $x(i)\epsilon X$



Автоматы Мили и Мура.

Автомат Мили задан системой канонических уравнений:

q(t+1)= $\delta\big(x(t), q(t)\big)$

y(t)= $\lambda(x(t), q(t))$

Автомат Мура задан системой канонических уравнений:

q(t+1)= $\delta\big(x(t), q(t)\big)$

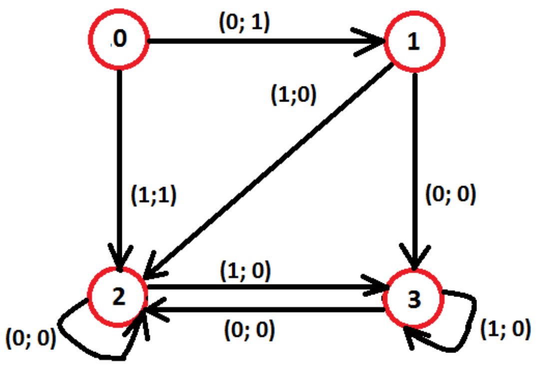
y(t)= $\lambda(q(t))$

Конечное состояние автомата Мура не зависит от входного символа.

Пример. Автомат T задан таблицей. Для этого автомата построить диаграмму Мура.

x	q			
	0	1	2	3
0	(1; 1)	(3; 0)	(2; 0)	(2; 0)
1	(2; 1)	(2; 0)	(3; 0)	(3; 0)

X={0, 1}, Q={0, 1, 2, 3}, Y={0, 1}.

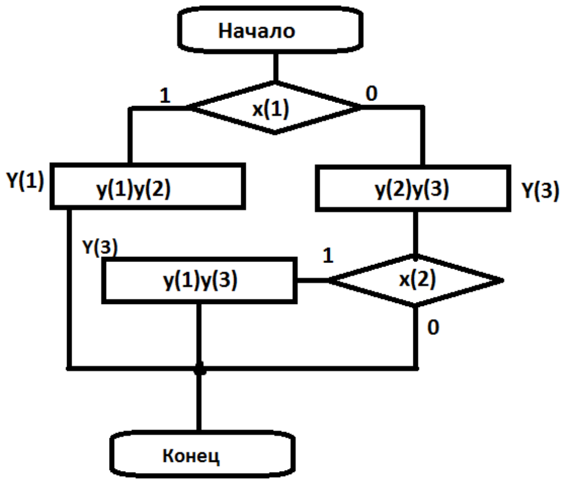


Построение автомата по блок-схеме алгоритма

Существует класс конечных автоматов, который называют микропрограммными автоматами. На вход такого автомата поступают некоторые логические условия из множества X. Автомат вырабатывает микрокоманды состоящие из микроопераций $y(i)\epsilon Y$. Микрокоманда – это набор микроопераций. Последовательность выполнения микрокоманд в зависимости от логических условий задается блок-схемой алгоритма. Блок-схема алгоритма фактически определяет функцию перехода. Совокупность микрокоманд и функций переходов образуют микропрограмму.

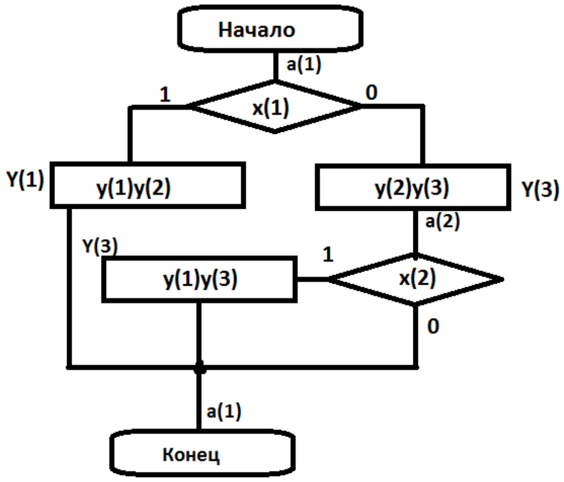
Блок-схема алгоритма или граф-схема алгоритма(ГСА) состоит из четырех типов вершин.

1-я начальная, 2-я конечная, 3-я операторная, 4-я условная. Приведем пример некоторой микропрограммы или микропрограммного автомата.



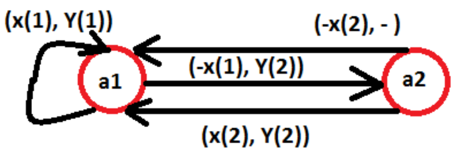
Построение автомата Мили.
1. Получение схемы ГСА
2. Построение графа переходов (диаграммы Мура)

1. Отметка ГСА проводится по алгоритму ф1. Метками a_1, …, a_m. Символом a_1 отмечается вход в вершины. Входы всех вершин, кроме начальной и конечной отмечают a_2, …, a_m. Входы различных вершин, за исключением конечной отмечают различными символами.



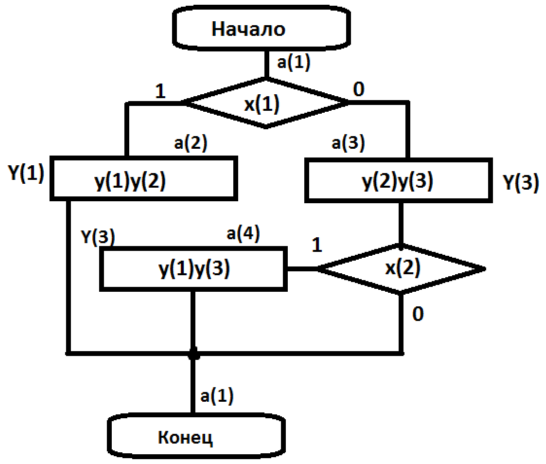
2. Если рассмотреть пути от одной метки к другой в направлении огранизации дуг ГСА, то в графе можно выделить несколько путей переходов.
$a_m X Y a_s$
$a_m X a_s$

$a_1 x_1 Y_1 a_1$
$a_1 \text{-} x_1 Y_2 a_2$
$a_2 x_2 Y_3 a_1$
$a_2 \text{-} x_2 a_1$
Получили четыре перехода. Для построения графа переходов пользуются следующим правилом: число вершин графа равно числу отметок.



Построение микропрограммного автомата Мура по блок-схеме алгоритма. Состоит из двух этапов.
1. Построение отмеченной ГСА.
2. Построение графа переходов автомата.
Блок-схема автомата Мура отмечается по алгоритму ф2.

1. Начальные и конечные вершины отмечаются символами a_1.
2. Все операторные вершины отмечаются различными символами.



Для построения графа переходов по блок-схеме алгоритма находят пути переходов.

$a_m X a_s$

Найдем все пути переходов.

$a_1 x_1 a_2$

$a_2 \text{-} x_1 a_3$

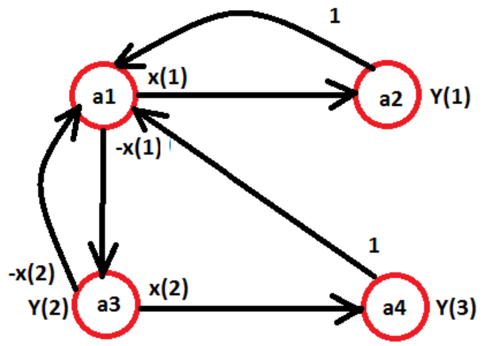
$a_2 1 a_1$

$a_3 x_2 a_4$

$a_3 \text{-} x_2 a_1$

$a_4 1 a_1$

Число отметок определяет число вершин графа.



X={x_1, -x_1, x_2, -x_2}

Q={a_1, a_2, a_3, a_4}

Y={y_1, y_2, y_3}

	-	$y_1 y_2$	$y_2 y_3$	$y_1 y_3$
x	q			
	a_1	a_2	a_3	a_4
x_1	a_2	a_1	-	a_1
$-x_1$	a_3	a_1	-	a_1
x_2	-	a_1	a_4	a_1
$-x_2$	-	a_1	a_1	a_1

Реализация автомата логической схемой. Программируемые логические матрицы.

Построение логической схемы микропрограммного автомата Мили. Создание таблицы переходов микропрограммного автомата. При большом числе состояний наглядность графа теряется, синтез микропрограммного автомата усложняется и тогда используется таблица переходов.

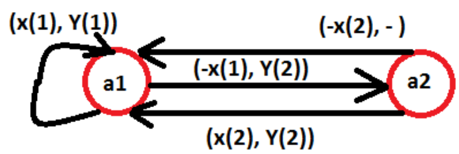


Таблица 1 - Таблица переходов содержит 5 основных столбцов: столбец a_m, a_s, X, Y, h

a_m	a_s	X	Y	h
a_1	a_1	x_1	y_1	1
	a_2	$-x_1$	y_2	2
a_2	a_1	x_2	y_3	3
	a_1	$-x_2$	-	4

$X(a_m, a_s)$ – коньюкция переменных из множества X. Коньюкция принимает логическое значение 1 на переходе из a_m в a_s.

$Y(a_m, a_s)$ – подмножество выходных переменных, принимающих логическое значение 1 на переходе из a_m в a_s.

Таблица переходов в данном случае построена по графу переходов. Но можно также построить по схеме алгоритма.

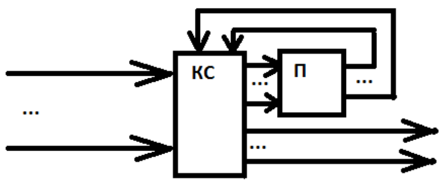
Следующий шаг состоит в построении структурной таблицы микропрограммного автомата Мили.

В отличии от табл. 1 в структурную таблицу дополнительно входят 3 столбца.

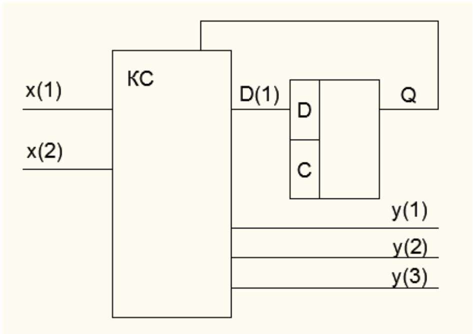
Таблица 2.

a_m	K_{am}	a_s	K_{as}	$X(a_m, a_s)$	$Y(a_m, a_s)$	$D(a_m, a_s)$	h
a_1	0	a_1	0	x_1	y_1	-	1
		a_2	1	$-x_1$	y_2	D_1	2
a_2	1	a_1	0	x_2	y_3	-	3
		a_1	0	$-x_2$	-	-	4

Представим структурную таблицу проектируемого автомата в следующем виде.



Проектируемый микропрограммный автомат построен по структурной схеме, содержащей комбинированную схему и память. Расчитаем для проектируемого автомата Мили.



Тривиальный синтез логической схемы микропрограммного автомата.

По структурной схеме сформируем логические выражения для функции выходов и функции возбуждения.

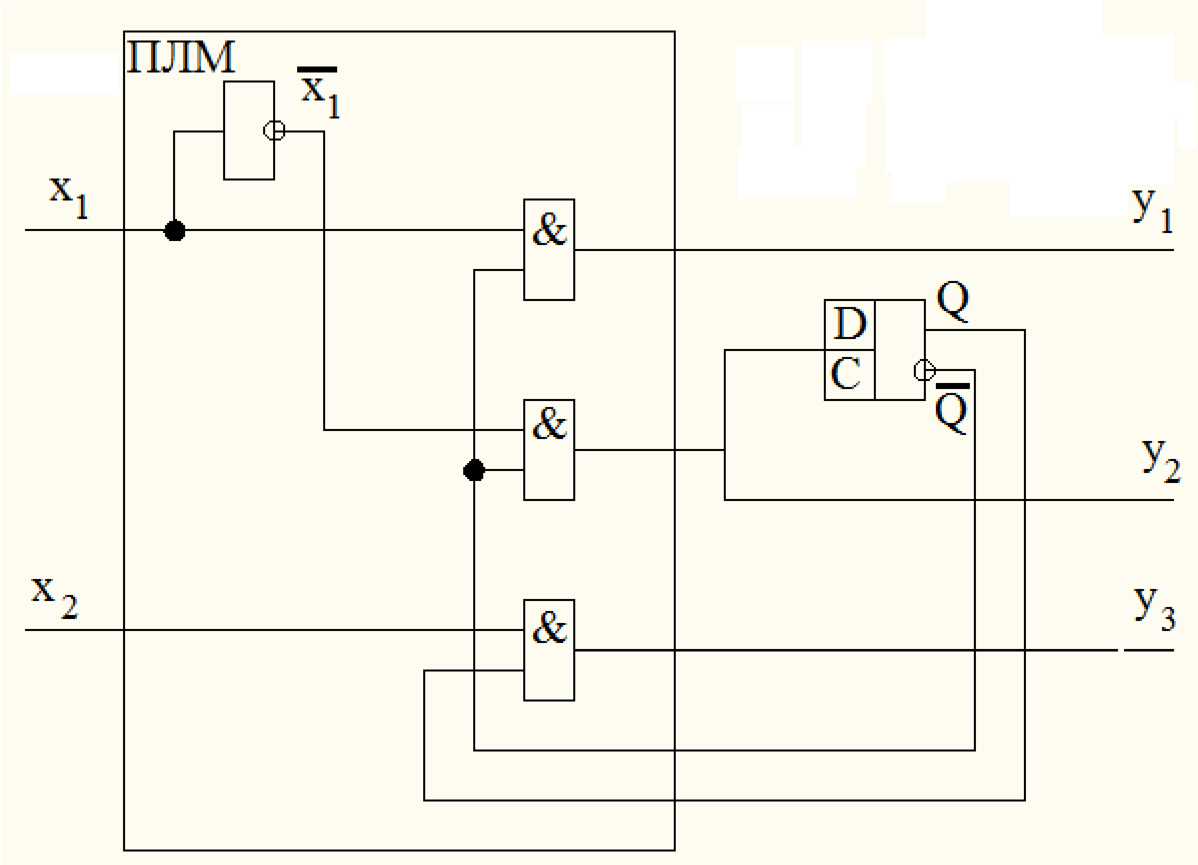
a_1: 0; a_2: 1;

$y_1=x_1\text{-}Q$

$y_2=-x_1\text{-}Q$

$y_3=x_2Q$

$D_1=-x_1\text{-}Q$



Синтез логической схемы микропрограммного автомата Мура.

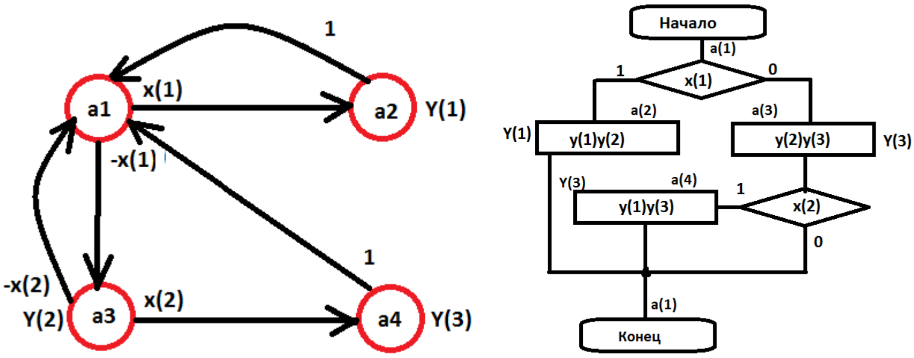


Таблица переходов микропрограммного автомата Мура содержит 3 основных столбца и один вспомогательный.

Таблица 3 – Таблица микропрограммного автомата Мура.

a_m, $Y(a_m)$	a_s	$X(a_m, a_s)$	h
a_1	a_2	x_1	1
	a_3	$-x_1$	2
a_2, y_1y_2	a_1	1	3
a_3, y_2y_3	a_1	$-x_2$	4
	a_4	x_2	5
a_4, y_1y_3	a_1	1	6

При наличии таблицы переходов, строят структурную таблицу переходов. В структурную таблицу переходов вводят дополнительно три столбца.

Таблица 4. – Структурная таблица переходов автомата Мура.

a_m, $Y(a_m)$	K_{am}	a_s	K_{as}	$X(a_m, a_s)$	$D(a_m, a_s)$	h
a_1	00	a_2	01	x_1	D_2	1
		a_3	10	$-x_1$	D_1	2
a_2, y_1y_2	01	a_1	00	1	-	3

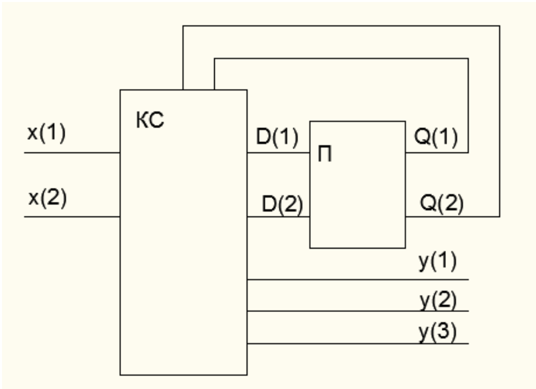
a_3, y_2y_3	10	a_1	00	$-x_2$	-	4
		a_4	11	x_2	D_1D_2	5
a_4, y_1y_3	11	a_1	00	1	-	6

Микропрограммный автомат Мура также строится по обобщенной структурной схеме построения конечных автоматов.

Число кодов x_1, …, x_n определяет число логических переменных.

Число микроопераций определяет число элементов множества Y.

Количество элементов памяти определяется по формуле sup $\log_2 |\{a_i\}| = 2$



a_1: 00

a_2: 01

a_3: 10

a_4: 11

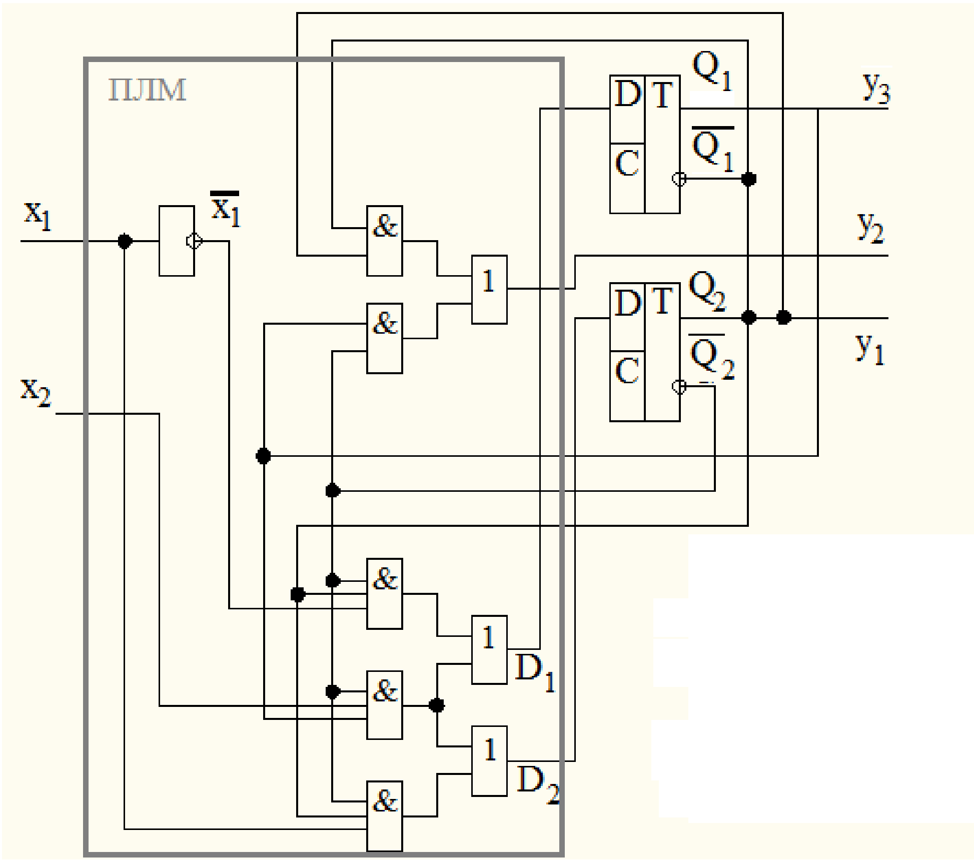
$y_1=-Q_1Q_2vQ_1Q_2=Q_2$

$y_2=-Q_1Q_2vQ_1-Q_2$

$y_3=Q_1-Q_2vQ_1Q_2=Q_1$

$D_1=-x_1-Q_1-Q_2vx_2Q_1-Q_2$

$D_2=x_1-Q_1-Q_2vx_2Q_1-Q_2$



Сети Петри. Построение моделей на основе сетей Петри.

В связи с широким использованием параллельных и распределенных вычислительных систем особую актуальность приобретают дискретные структуры, представляющие параллельные процессы.

Аппаратом описания сложных систем взаимодействия процессов является сети Петри моделирующие динамические свойства системы.

Формализм сетей Петри общего вида основан на понятии комплекта, который является общим понятием множества.

Комплект – набор элементов, но в отличии от множества – всякий элемент может входить более одного раза. Иначе говоря, отношение включения связывающие элементы и множества заменяется на функцию числа экземпляров элементов в комплекте.

$\#(x, b) | x \in B$

Множество – частный случай комплекта. Многие понятия теории множеств распространяются и на комплекты.

$\emptyset$ - пустой комплект.

Мощность комплекта – общее число экземпляров элементов в комплекте.

Комплект AcB | $\forall x\ \#(x, A) \leq \#(x, B)$

С помощью функции числа экземпляров легко определить операции над комплектами:

1. Объединение комплектов

$$\#(x, A \cup B) = \max(\#(x, A); \#(x, B))$$

2. Пересечение комплектов

$$\#(x, A \cap B) = \min(\ \#(x, A); \#(x, B))$$

3. Разность комплектов

$$\#(x, A \backslash B) = \#(x, A) - \#(x, B)$$

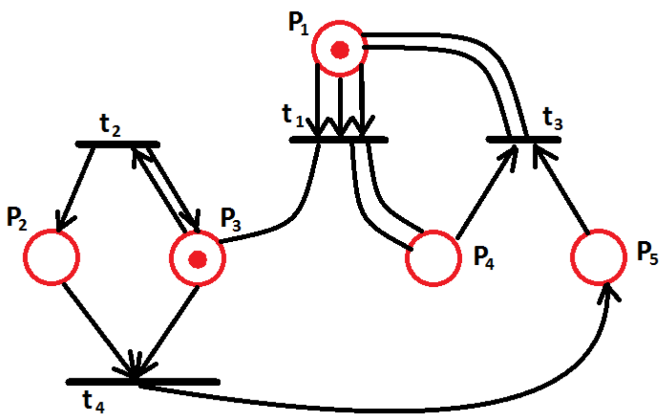
Если N – множество, то M^n множество всех таких комплектов, построенных из элементов N что $M^n \colon \#(x, b) \leq n, B \in M^n$.

M^∞ - множество всех комплектов, построенное из элементов M без ограничения на число экземпляров элементов в комплекте.

Определение. *Сеть Петри это четверка C=<P, T, I, O>, где P – конечное множество позиций; T – конечное множество переходов; I – входная функция, отображающая переходы в комплекты позиций; O – выходная функция, отображающая переходы в комплекты позиций.*

Графически сеть Петри изображают в виде мультиграфа с вершинами двух видов. Кружки соответствуют позициям, планки – переходам. Функции I, O представляются дугами.

Пример.



Позиции, дуги из которых ведут в переход t_j, называются входными для t_j. Аналогично позиции, в которые ведут дуги из пехода t_j, называются выходными для t_j.

Множество входных позиций обозначается $I(t_j)$. Множество выходных позиций $O(t_j)$.

Для приведенного примера:

$I(t_1)=\{p_1, p_1, p_1\}$; $O(t_1)=\{p_3, p_4, p_4\}$.

Функции I и O удобно обобщить и на отображении из позиции в комплекты переходов, что позволяет обозначить множество входных и выходных переходов позиций p_i, определ. аналогично множествами входных и выходных позиций соотв. как $I(p_i)$, $O(p_i)$.

Например, для позиции p_3:

$I(p_3)=\{t_2, t_1\}$, $O(p_3)=\{t_2, t_4\}$.

Введенные понятия относятся к статическим структурам сети Петри. Динамические свойства определяются с помощью понятия маркировки.

Маркировка μ сети Петри – функция, отображающая множество позиций p в множество неотрицательных чисел N: μ: $p \rightarrow N$.

Маркировка изображения с помощью позиций фишек(точек).

Маркировка сети в приведенном примере определяется следующим образом:

$\mu(p_1)= \mu(p_3)=1$

$\mu(p_2)= \mu(p_4)= \mu(p_5)=0$

Удобно представить маркировку как n-вектор.

$\mu(\mu_1, \mu_2, \ldots, \mu_n)$, где $n=|p|$, каждый элемент которого μ⍰ $\mu(p_i)$.

Сеть Петри с определенной на ней маркировкой называется маркированной сетью Петри.

Маркировка сети может изменяться в результате запуска переходов.

Переход t_j маркировки сети Петри с маркировкой μ называется разрешенным, если $I(t_j)= \mu$, т.е. в каждой входной позиции наход. не меньше фишек, чем из этой позиции исходит дуг в t_j. Всякий разрешенный переход может запуститься. В результате запуска перехода t_j маркировка сети μ заменяется на новую μ': $\mu'= \mu-I(t_j)+O(t_j)$.

Из всякой входной позиции p_i удаляется столько фишек, сколько дуг ведет из p_i в t_j, а в каждую выходную позицию p_k помещается столько фишек, сколько дуг ведет из t_j в p_k.

Последовательность запусков переходов называется выполнением сети Петри. Рассмотрим выполнение сети Петри для примера.

В качестве маркировки разрешен только t_2. При его запуске фишка удаляется из p_3, а затем в позиции p_2 и p_3 добавится по фишке, т.е. в результате запуска в новой макировке μ' появляется еще фишка в p_2. Теперь становятся разрешенными переходы t_2 и t_4. Поскольку запуститься может любой разрешенный переход, предположим, что запустился t_4. После его запуска из p_2 и p_3 фишки удалятся, а в позиции p_5 появится одна фишка. В полученной маркировке μ'' не разрешен ни один переход. На этом выполнение сети Петри заканчивается.

Рассмотрим маркировку сети Петри, представленную четверкой <P, T, I, O>. Маркировка μ' называется непосредственно достижимой из μ, если найдется такой переход t_j, разрешенный в μ, что при его запуске получается маркировка μ'.

В этом случае пара (μ, μ') принадлежит отношению непосредственной достижимости.

Транзитивное замыкание этого отношения называется отношением достижимости. Маркировки μ', такие что (μ, μ') принадлежат отношению достижимости, называются достижимыми из μ.

Множество достижимых из μ маркировок сети Петри называется множеством достижимости и обозначается $R(C, \mu)$.

Интерпретация сетей Петри основана на понятиях условия и события. Состояния системы описываются совокупностью условий. Функционирование системы состоит в осуществлении последовательности определенных действий, т.е. событий.

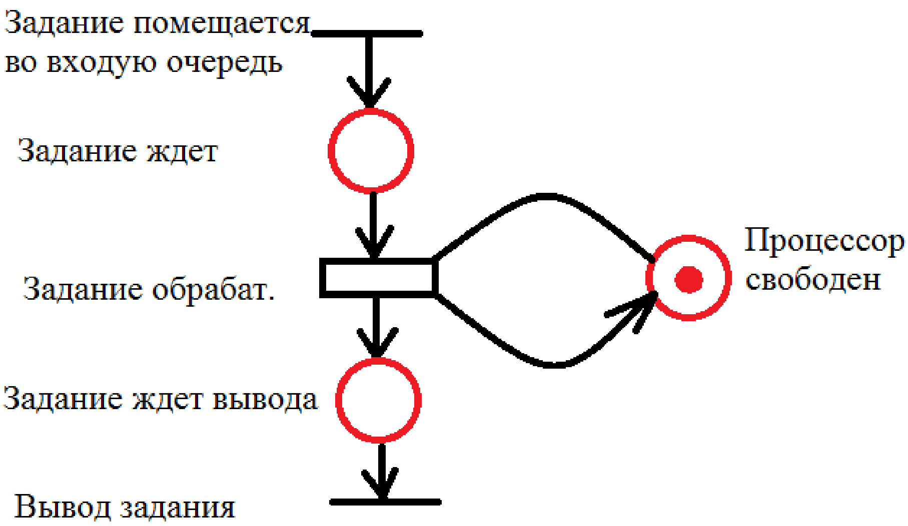
Для возникновения события необходимо выполнение некоторых условий, называемых предусловиями. Возникновение события может привести к нарушению предусловия и выполнению постусловий. Условия моделируются позициями, события – переходами.

Предусловия событий представляются входными позициями соответствующего перехода, постусловия – выходными позициями. Возникновение события моделируется запуском перехода.

Выполнение условий представляет наличие фишек в соответствующих позициях, невыполнение – их отсутствие.

Рассмотрим пример простой вычислительной системы последовательной обработки заданий, которые поступают во входную очередь. Когда процессор свободен и во входной очереди имеется задание, оно обрабатывается процессором, затем выводится.

Система будет представлена сетью Петри:



Установим, какие особенности систем учитывают сети Петри. Это, прежде всего, асинхронность. В сетях Петри отсутствует понятие времени. Время возникновения события не указывается, но структура сети Петри устанавливает частотный порядок возникновения событий.

Поскольку возникновения событий представляется запуском переходов, предполагается, что события происходят мгновенно. Если же моделируемое событие имеет отличную от 0 длительность (например: «задание обрабатывается») и это существенно, то оно представляется в виде двух мгновенных событий типа начало события - конец события и условия «событие происходит».

Считается, что события происходят неодновременно. Другим важным свойством сетей Петри является их способность представлять параллелизм и конфликтные ситуации. Параллелизм представляется 2 разрешимыми переходами, множества которые не пересекаются.

Нечеткие подмножества.

$$M_A(x) = \begin{cases} 1, \text{если } x \in A \\ 0, \text{в противном случае} \end{cases}$$

$M_A(x)$ - характеризует множество, которое принимает значение от 0 до 1.

Математически объект определяется выражением:

$$A = \{ \left(\frac{x_1}{0.2}\right), \left(\frac{x_2}{0}\right), \left(\frac{x_3}{0.3}\right), \left(\frac{x_4}{1}\right), \left(\frac{x_5}{0.8}\right) \}$$

где x_i – объект универсального множества M.

Обозначение нечеткого множества: AcM

А не содержит x_2, полностью содержит x_4, в значительной мере x_5.

Пусть M – множество, x – элементы. Тогда нечеткое A определяется как множество упорядоченных пар:

$\{< x, \mu_A(x) >\} \forall x \in M,$ где $\mu_A(x)$ – характеристическая функция принадлежности, принимающая свои значения на упорядоченном множестве P.

$\mu_A(x)$ указывает степень, или уровень, принадлежности x к подмножеству A.

Множество P называется множеством принадлежностей. Пусть M – множества P – множество принадлежностей, A, B – два нечетких множества. Говорят, что A=B тогда и только тогда, когда $\forall x \in M \; \mu_A(x) = \mu_B(x)$.

Если найдется хотя бы один $x \in M$, такой, что $\mu_A(x) \neq \mu_B(x)$, то $A \neq B$ (такие множества не равны).

Операции над нечеткими множествами:

1. Дополнение

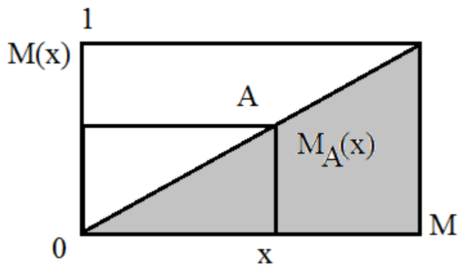
Пусть M – множество, P – множество принадлежности, P=[0; 1]; A и B – нечеткие множества. Говорят, что A и B дополняют друг друга если $A = \bar{B}$, $B = \bar{A}$.

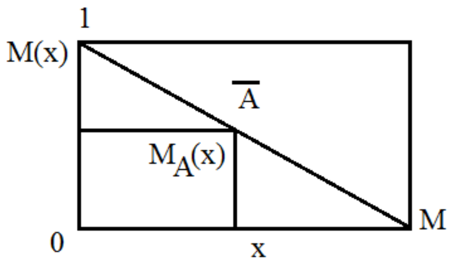
$\bar{B}$ - дополняет A, $\bar{A}$ – дополняет B.

$$A = \bar{B},\ B = \bar{A} \mid \forall x \in M \qquad \mu_A(x) = 1 - \mu_B(x)$$

Имеет место закон двойного дополнения: $\bar{\bar{A}} = A$.

Для нечетких множеств можно построить визуальное представление в следующем виде:





При визуальном представлении используется прямоугольная система координат. Прямоугольная система координат, на оси ординат которой откладывается μ(x), а на оси абсцисс в произвольном порядке расположены элементы x множества M.

Принадлежность каждого элемента определяется величиной его ординаты заштрихованная часть в первом случае изображает подмножество AcM.

Пример.

$M = \{x_i \mid_{i=\overline{1,5}}\}, P = [0,1]$

$A = \{ \left(\frac{x_1}{0.1}\right), \left(\frac{x_2}{0.6}\right), \left(\frac{x_3}{0.4}\right), \left(\frac{x_4}{0}\right), \left(\frac{x_5}{1}\right)\}$

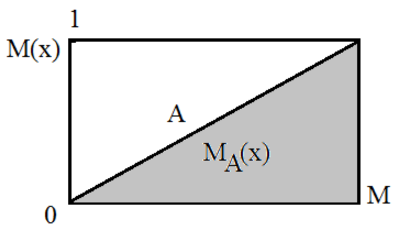
$\bar{A} = \{ \left(\frac{x_1}{0.9}\right), \left(\frac{x_2}{0.4}\right), \left(\frac{x_3}{0.6}\right), \left(\frac{x_4}{1}\right), \left(\frac{x_5}{0}\right)\}$

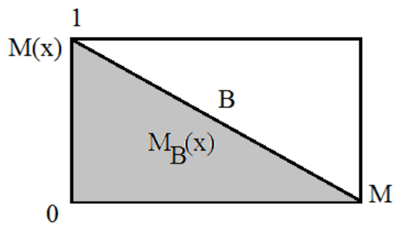
2. Пересечение.

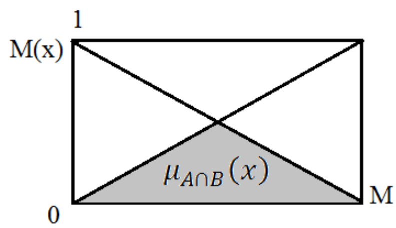
Пусть M – множество, а P – множество принадлежностей, A и B – два нечетких множества.

Пересечение $A \cap B$ определяют как наибольшее нечеткое подмножество, содержащее одновременно A и B.

$$A \cap B \mid \forall x \in M, \mu_{A\cap B}(x) = \min\left(\mu_A(x), \mu_B(x)\right)$$







Пример.

$M = \{x_i \mid_{i=\overline{1,5}}\}, P = [0,1]$

$A = \{ \left(\frac{x_1}{0.2}\right), \left(\frac{x_2}{0.8}\right), \left(\frac{x_3}{1}\right), \left(\frac{x_4}{0.5}\right), \left(\frac{x_5}{0}\right)\}$

$B = \{ \left(\frac{x_1}{0.6}\right), \left(\frac{x_2}{0.4}\right), \left(\frac{x_3}{0.2}\right), \left(\frac{x_4}{0.7}\right), \left(\frac{x_5}{0.9}\right)\}$

$A \cap B = \{ \left(\frac{x_1}{0.2}\right), \left(\frac{x_2}{0.4}\right), \left(\frac{x_3}{0.2}\right), \left(\frac{x_4}{0.5}\right), \left(\frac{x_5}{0}\right)\}$

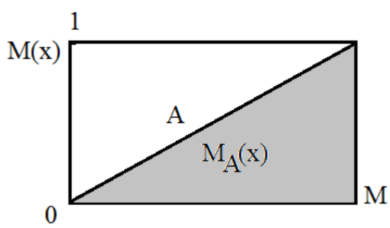
3. Объединение

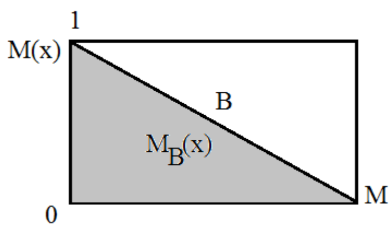
Пусть M – множество, P – множество принадлежностей, A и B – 2 нечетких множества.

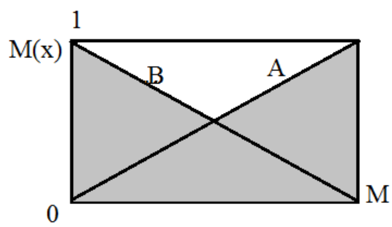
Определим объединение A и B как множество, которое содержит A и B.

$$A \cup B \mid \forall x \in M, \mu_{A\cup B} = \max\left(\mu_A(x), \mu_B(x)\right)$$

Пример. A и B заданы как и в предыдущем примере.







$A \cup B = \{ \left(\frac{x_1}{0.6}\right), \left(\frac{x_2}{0.8}\right), \left(\frac{x_3}{1}\right), \left(\frac{x_4}{0.7}\right), \left(\frac{x_5}{0.9}\right)\}$

4. Разность A\B.

$$A\backslash B \mid \forall x \in M, \mu_{A\backslash B}(x) = \min\left(\mu_A(x), 1 - \mu_B(x)\right) \text{ или } A\backslash B = A \cap \bar{B}$$

Пример.

$\bar{B} = \{ \left(\frac{x_1}{0.4}\right), \left(\frac{x_2}{0.6}\right), \left(\frac{x_3}{0.8}\right), \left(\frac{x_4}{0.3}\right), \left(\frac{x_5}{0.1}\right)\}$

$A\backslash B = A \cap \bar{B} = \{ \left(\frac{x_1}{0.2}\right), \left(\frac{x_2}{0.6}\right), \left(\frac{x_3}{0.8}\right), \left(\frac{x_4}{0.3}\right), \left(\frac{x_5}{0}\right)\}$

Введенные операции удовлетворяют тем же законам, что и те же операции для множеств.

1. Закон коммутативности

$A \cup B = B \cup A$

$A \cap B = B \cap A$

2.

$A \cup (B \cup C) = (A \cup B) \cup C$

$A \cap (B \cap C) = (A \cap B) \cap C$

3.

$A \cup A = A$

$A \cap A = A$

4.

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

5. Действия с константами.

$A \cup \emptyset = A$

$A \cap \emptyset = \emptyset$

6. Закон двойного дополнения

$\bar{\bar{A}} = A$

7. Законы де Моргана.

$\overline{A \cup B} = \bar{A} \cap \bar{B}$

$\overline{A \cap B} = \bar{A} \cup \bar{B}$

8. Законы поглощения

$A \cup A \cap B = A$

$A \cap (A \cup B) = A$