ОСНОВЫ ТЕОРИИ МНОЖЕСТВ

1. Способы задания множеств

Множество— совокупность различимых между собой объектов, объединяемых в целое некоторым общим признаком. Например: множества студентов, книг, законов, чисел и т. п.
Элементы — объекты, из которых состоит множество.
Обозначения:*A, B, C,...* — множества, a, *Ь, c,...* — элементы (точки) множеств.
Принадлежность:
$a\in A$ — a принадлежит множеству A;
$a\notin A$a не принадлежит множеству A.
Записью a1, a2, …, $a_n\in M$ пользуются в качестве сокращения для записи $a1\in M$, $a2\in M$, …, $a_n\in M$.
Задание множеств. Чтобы задать множество, нужно указать, какие элементы ему принадлежат. Это можно сделать различными способами:
Перечислением элементов: $A=\{ a_1, a_2, ... , a_k \}$;
Указанием характеристического свойства (хар. предикатом): M := {x | P(x)};
Порождающей процедурой: M := {x | x := f}.
При задании множеств перечислением обозначения элементов обычно заключают в фигурные скобки и разделяют запятыми. Характеристический предикат − это некоторое условие, выраженное в форме логического утверждения или процедуры, возвращающей логическое значение, и позволяющее проверить, принадлежит ли любой данный элемент множеству. Если для данного элемента условие выполнено, то он принадлежит определяемому множеству, в противном случае − не принадлежит. Порождающая процедура − это процедура, которая в процессе работы порождает некоторые объекты, являющиеся элементами определяемого множества.
Пример 1:
M9: = {1, 2, 3, 4, 5, 6, 7, 8, 9};
M9: = {n | n $\in$N& n < 10};
M9: = {n | for i from 0 to 8 do n := i + l}.

При задании множеств перечислением обозначения элементов иногда снабжают индексами и указывают множество, из которого берутся индексы. В частности, запись {ai]ki=l означает то же, что {ai,..., a^}, а запись M = {Ma}a(-A означает, что M является семейством, элементами которого являются множества Ma, причём индекс a «пробегает» множество A. Перечислением можно задавать только конечные множества. Бесконечные множества задаются характеристическим предикатом или порождающей процедурой.

Из определения множества следует, что в нем не должно быть неразличимых элементов. Поэтому во множестве не может быть одинаковых элементов. Запись множества {2, 2, 3, 5} следует рассматривать как некорректную и заменить ее на {2, 3, 5}. Так, множество всех простых делителей числа 60 равно {2, 3, 5}. Многие определения теории множеств удобно давать в виде математических выражений, содержащих некоторые логические символы:

∀ – символ, называемый квантором общности и означающий «любой», «каков бы ни был», «для всех»;
∃ – символ, называемый квантором существования и означающий «существует», «найдется хотя бы один»;
⇒ – символ следствия (импликации), означающий «влечет за собой»;
⇔ – символ эквивалентности (равносильности), означающий «то же самое, что»;
& (∧) – символ конъюнкции, одновременного выполнения условий, означающий «и»;
∨ – символ дизъюнкции, выполнения хотя бы одного из условий, означающий «или»;
¬( ) – символ отрицания, невыполнения условия, означающий «не».

2. Операции над множествами и их свойства. Диаграммы Венна

Множества A и B ***равны****, A=B,* тогда и только тогда, когда $A \subseteq B$ и $B \subseteq A$, т. е. состоят из одинаковых элементов, причем порядок следования элементов не имеет значения; в противном случае пишут $A \neq B$.

Объединением множеств A и B (обозначение $A \cup B$) называется множество, состоящее из всех тех и только тех элементов, которые принадлежат хотя бы одному из множеств A, B: $A \cup B = \{x \mid x \in A \text{ или } x \in B\}$. Если система содержит небольшое количество множеств, то их объединение описывается явно: $A \cup B \cup C \cup D$ и т. д.

Для объединения множеств существуют коммутативный и ассоциативный законы: $A \cup B = B \cup A$, $(A \cup B) \cup C = A \cup (B \cup C)$. Рассмотрим $A \cup \varnothing = A$. Очевидно, что $A \cup A = A$ для любого множества A. Для любых множеств A и B верно $A \subseteq (A \cup B)$, но неверно $A \in (A \cup B)$.

Пересечением множеств A и B называется множество C, состоящее из всех элементов, принадлежащих одновременно и A, и B: $C = A \cap B = \{x \mid x \in A \text{ и } x \in B\}$. Аналогично определяется пересечение (в том числе бесконечной) системы множеств. Для пересечения множеств существуют коммутативный и ассоциативный законы: $A \cap B = B \cap A$, $(A \cap B) \cap C = A \cap (B \cap C)$. Рассмотрим $A \cap \varnothing = \varnothing$. Очевидно, что $A \cap A = A$ для любого множества A. Для любых множеств A и B верно $A \cup (A \cap B) = A$, но неверно $A \cap (A \cup B) = A$.

Разностью *множеств A и B* называется множество C , состоящее из тех элементов множества *A,* которые не содержатся в множестве B: $C = A \setminus B = \{x \mid x \in A \text{ и } x \notin B\}$.

В отличие от операций объединения и пересечения данная операция определяется только для двух множеств. Для произвольных множеств A и B верны соотношения:

$A \setminus B = \varnothing \Leftrightarrow A \subseteq B$,

$A \setminus \varnothing = A$,

$A \setminus B = A \Leftrightarrow A \cap B = \varnothing$.

Дополнением множества A до универсального множества U (обозначение $\overline{A}$) называется множество всех элементов U, не принадлежащих A:

$\overline{A} = \{x \mid (x \in U) \& (x \notin A)\}$,

$\overline{A} = U \setminus A$.

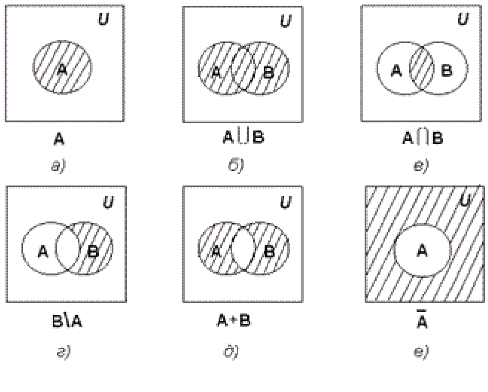
$A \cap \overline{A} = \varnothing$

$A \cup \overline{A} = U$

$\overline{\overline{A}} = A$.

Симметрической разностью множеств ***A*** и ***B*** (обозначение $A \Delta B$) называется множество $(A \setminus B) \cup (B \setminus A)$. Оно состоит из всех тех и только тех элементов универсального множества, которые либо принадлежат A и не принадлежат B, либо наоборот, принадлежат B, но не A.

Названные операции и свойства к ним могут быть проиллюстрированы **диаграммами Венна**



3. Декартово произведение множеств, булеан, мощность множества

Прямым произведением множеств A и B (обозначение $A \times B$) называется множество, состоящее из всех тех и только тех упорядоченных пар, первая компонента которых принадлежит множеству A, а вторая – множеству B.
$A \times B = \{(a, b) \mid a \in A, b \in B\}$.
Пусть A и B – отрезки соответственно длинами a и b вещественной оси. Прямое произведение $A \times B$ представляет собой заштрихованный прямоугольник с длиной a и шириной b в декартовой системе координат XOY (рис. 5).
Множество всех подмножеств множества **A** называется его ***булеаном*** (или множеством-степенью) и обозначается через P(A). **Пример**. Если $A = \{a, b, c\}$, то $P(A) = \{\emptyset, \{a\},\{b\},\{c\},\{a, b\},\{b, c\}, \{a, c\},\{a, b, c\}\}$.
Число элементов конечного множества называется его ***мощностью***. Если множество A содержит n элементов, то будем писать $|A| = n$. Если $A = \emptyset$, то $|A| = 0$.

4. Упорядоченные множества. Проекция множества

Кортеж − это последовательность элементов, т. е. совокупность элементов, в которой каждый элемент занимает определенное место. Часто кортеж называют вектором, а элементы, образующие кортеж, − его компонентами, или координатами. Компоненты нумеруются слева направо. Число компонент называется длиной, или размерностью, кортежа. Могут быть и бесконечные кортежи. В отличие от элементов множества координаты кортежа могут совпадать. Кортеж будем заключать в круглые скобки. Например, a = (a1, …, an) – кортеж длиной n с элементами a1, …, an. Иногда скобки и даже запятые опускаются. Два конечных кортежа равны, если имеют одинаковую длину и соответствующие компоненты.
Операция проектирования множества может применяться лишь к таким множествам, элементами которых являются кортежи одинаковой длины. Пусть V – множество кортежей одинаковой длины. Тогда проекцией множества V на i-ю ось называется множество проекций всех векторов из V на i-ю ось: npiV = {npiv | v∈V}. Аналогично определяется проекция множества V на несколько осей: пр i1,...,ikV= | v∈V = { i1,...,ikv | v∈V }. Пример 22. Пусть V := {1, 2, 3, 4, 5), (2, 1, 3, 5, 5), (3, 3, 3, 3, 3), (3, 2, 3, 4, 3)}. Тогда пр2V = {2, 1, 3}, пр2,4V = {(2, 4), (1, 5), (3, 3)}.

5. Соответствия, основные определения, способы задания

Пусть X и Y – два непустых множества. Если определен способ сопоставления элементов Y элементам X, то говорят, что между множествами X и Y установлено ***соответствие***. При этом совершенно необязательно, чтобы в сопоставлении участвовали все элементы множеств X и Y. Для того чтобы задать соответствие между множествами X и Y, нужно задать множество $Q \subseteq X \times Y$, определяющее закон, по которому осуществляется соответствие, т. е. перечисляющий все пары (x, y), участвующие в сопоставлении. Таким образом, соответствие q представляет собой тройку множеств q = (X, Y, Q), в которой $Q \subseteq X \times Y$. X называют областью отправления соответствия, вторую компоненту Y – областью прибытия соответствия, третью компоненту Q – графиком соответствия. С каждым соответствием неразрывно связаны еще два множества: множество **пр1Q**, называемое областью определения соответствия, которое состоит из всех элементов множества X, участвующих в сопоставлении, и множество **пр2Q**, называемое областью значений соответствия, которое состоит из всех элементов множества Y, участвующих в сопоставлении. Если $(x, y) \in Q$, то говорят, что элемент y соответствует элементу x. Геометрически это удобно изображать стрелкой, направленной от x к y. Если пр1Q = X, то соответствие называется всюду определенным, или **отображением X в Y** (в противном случае соответствие называется частичным). Если пр2Q = Y, то соответствие называется **сюръективным** (сюръекцией). Множество всех $y \in Y$, соответствующих элементу $x \in X$, называется **образом** x в Y при соответствии q. Множество всех $x \in X$, которым соответствует элемент $y \in Y$, называется **прообразом** y в X при соответствии q. Если $C \subseteq$ пр1Q, то образом множества C называется объединение образов всех элементов C. Аналогично определяется прообраз множества D для любого $D \subseteq$ пр2Q. Соответствие q называется **инъективным** (инъекцией), если любые различные x1 и x2 из пр1Q имеют различные образы и любые различные y1 и y2 из пр2Q имеют различные прообразы при соответствии q. Соответствие q называется **функциональным** (или однозначным), если образом любого элемента $x \in$ пр1Q является единственный элемент $y \in$ пр2Q. Соответствие q между множествами X и Y называется взаимно однозначным, или **биективным**, биекцией (иногда пишут «1-1-соответствие»), если оно всюду определено, сюръективно и инъективно. Однозначное отображение называется **функцией**. Функция является инъективной, если различным x1 и x2 из X соответствуют различные y1 и y2 из Y, и сюръективной, если она сюръективна как соответствие. Функция называется биективной, если она одновременно инъективна и сюръективна.

Способы задания: табличный, графический, матрицей, аналитический (т.е.формулой), способ перечисления пар .

6. Функции, способы задания, композиция функций

Пусть $f: X \to Y$ – функция. Будем обозначать $D(f)$ область определения функции и $E(f)$ область значений функции f. Каждому элементу $x \in X$ ставит в соответствие единственный элемент $y \in Y$. Это обозначается записью $f(x) = y$ либо $f: x \to y$. Элемент x называется аргументом функции, y – значением функции на x. Если $f(X)$ состоит из единственного элемента, то f называется функцией-константой. Тождественной функцией на множестве X называется функция $e: X \to X$, такая, что $e(x) = x$ для любого $x \in X$. Если $X, Y \subseteq R$, то функцию f называют вещественной.

Пусть даны функции $f: X \to Y$ и $g: Y \to Z$. Функция $h: X \to Z$ является **композицией** функций f и g, $h = g^{o}f$, если для любого $x \in X$ $h(x) = = g(f(x))$. Часто говорят, что функция h получена подстановкой f в g. Функция, полученная из f1, …, fn некоторой подстановкой их друг в друга и переименованием аргументов, называется **суперпозицией** f1, …, fn. Выражение, описывающее эту суперпозицию и содержащее функциональные знаки и символы аргументов, называется формулой.

Способы задания: табличный, графический, аналитический (т.е.формулой) .

7. Бинарные отношения и их свойства

Бинарное отношение - это отношение между двумя объектами. Бинарное отношение можно определить как совокупность упорядоченных пар, указывающих объекты, находящиеся в данном отношении. В общем случае, если два элемента a, b находятся в данном отношении R, то этот факт записывают (a, b) ϵ R или aRb. Если эти элементы не находятся в отношении R, то это записывают так: (a, b) ∉ R, или $a\overline{R}b$. Некоторым из наиболее известных отношений присваивают специальные названия и обозначения. Примеры: эквивалентность (==), отношение порядка (>) или (>), равенство (=), параллельность (||), перпендикулярность (┴) и т. д. Очевидно, что всякое бинарное отношение R можно рассматривать как подмножество прямого произведения некоторых множеств A и B: R$\underline{c}$ A×B. Левой областью бинарного отношения называют множество всех первых компонент упорядоченных пар, составляющих данное отношение, то есть R_={a| (a,b)ϵR}
Правой областью бинарного отношения R называют множество всех вторых компонент упорядоченных пар, составляющих данное отношение, то есть R_+={b| (a,b)ϵR}

Полем бинарного отношения R называют объединение его левой и правой областей: F(R)= R_ ∪ R_+. Бинарное отношение R^{-1} называют обратным к отношению R, если (a,b)ϵR^{-1} тогда и только тогда, когда (b,a)ϵR то есть R^{-1} ={(a,b)| (b,a)ϵR}.

Пересечением бинарного отношения R по элементу a ϵ F(R) называют совокупность всех вторых (различных) компонентов упорядоченных пар, составляющих данное отношение, и таких, у которых первой компонентой есть элемент a. Обозначение: R_a.

Композицией бинарных отношений R и S называют бинарное отношение T, состоящее из всех упорядоченных пар (a, b), для каждой из которых существует элемент ñ ϵ Ê₊∩S такой, что (a, c) ϵ R, (c, b) ϵ S (то есть aRc, cSb). Операцию композиции записывают так: T = R°S. Например, пусть R = {(1, 1), (1, 2), (2, 3), (3, 3)}, S = {(2, 4), (2, 5), (3, 2), (5, 5)}. Тогда R°S = {(1, 4), (1, 5), (2, 2), (3, 2)}, S°R = {(3, 3)}

Свойства бинарных отношений. Бинарное отношение R называют **рефлексивным**, если для любого элемента a ϵ F(R) имеет место aRa (находится в отношении R с самим собой). Бинарное отношение R называют **антирефлексивным**, если для любого элемента поля a ϵ F(R) имеет место $a\bar{R}a$. Бинарное отношение R **симметричное**, если из aRb следует bRa. Бинарное отношение R **асимметрично**, если из aRb следует bRa. Бинарное отношение R называют **антисимметричным**, если из aRb и bRa следует, что a = b. Бинарное отношение R называют **транзитивным**, если из aRb и bRc следует aRc. В противном случае отношение R называют нетранзитивным. Бинарное отношение называют отношением **эквивалентности**, если оно рефлексивно, симметрично и транзитивно.

Классом эквивалентности R a называют множество всех вторых компонентов упорядоченных пар отношения эквивалентности R, у которых первой компонентой является элемент a: R_a = {b | (a, b) ϵ R}.

8. Способы задания бинарных отношений

1. Бинарное отношение R можно задать перечислением всех упорядоченных пар, находящихся в отношении R. Очевидно, что такой способ задания отношений приемлем для относительно небольшого числа упорядоченных пар. Например: R1 = {(1, 1), (2, 2), (3, 3), (4, 4)}.

2. Задать формулой.

3. Графическое задание бинарного отношения предполагает графическое представление элементов левой и правой областей отношения в виде точек в этих областях, соединенных дугами (направленными отрезками). Каждая дуга представляет некоторую упорядоченную пару, находящуюся в данном отношении. Дуга начинается в точке, соответствующей первой компоненте упорядоченной пары, и заканчивается в точке, соответствующей второй компоненте упорядоченной пары.

4. Бинарное отношение можно задать в табличной форме. В этой таблице указываются все элементы поля отношения и соответствующие им пересечения данного отношения по выбранному элементу.

5. Бинарное отношение можно задать матрицей ||ai,j||, в которой строки и столбцы соответствуют полю отношения. В этой матрице i-я строка соотносится с некоторым элементом левой области отношения, а j-й столбец – с некоторым элементом правой области отношения. Тогда ai,j = 1, если соответствующие элементы находятся в данном отношении, и ai,j = 0 в противном случае.

9. Операции над бинарными отношениями

Если R – бинарное отношение, то в качестве универсального множества в этом случае рассматривают множество U = F(R) × F(R), где F(R) – поле отношения R. Если совместно рассматривается несколько бинарных отношений, то в качестве универсального множества рассматривают множество U = A × A, где A есть объединение полей каждого из рассматриваемых отношений. Так как всякое бинарное отношение – это множество упорядоченных пар, то над бинарными отношениями можно выполнять все теоретико-множественные операции: объединение, пересечение, разность, дополнение.

Композицией бинарных отношений R и S называют бинарное отношение T, состоящее из всех упорядоченных пар (a, b), для каждой из которых существует элемент $c \in R_+ \cap S_-$ такой, что aRc, cSb. Операцию композиции записывают так: T = R°S. Например, пусть R = {(1, 1), (1, 2), (2, 3), (3, 3)}, S = {(2, 4), (2, 5), (3, 2), (5, 5)}. Тогда R°S = {(1, 4), (1, 5), (2, 2), (3, 2)}, S°R = {(3, 3)}.

ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

1. Высказывания и операции над ними

Высказывание – это повествовательное предложение, о котором можно сказать истинно оно или ложно. Каждое из высказываний принято обозначать латинской буквой. Если высказывание представляет собой одно утверждение - его называют элементарным или простым. Используя такие логические операции, как не, или, и, можно построить новые, так называемые составные высказывания компонуя более простые.

Логической операцией наз-ся построение нового высказывания из исходных высказываний

Отрицанием произвольного высказывания x называется высказывание вида ($\bar{x}$), чье истинностное значение строго противоположно значению X. Определяющая таблица истинности отрицания высказывания приведена в табл.

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

Конъюнкцией или логическим умножением двух высказываний X и У называют составное высказывание вида z=x^y(x*y). Оно принимает истинное значение только в том случае, когда истинны обе его составные части.

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкцией или логическим сложением двух высказываний X и У называется составное высказывание z=x˅y(x+y). Оно истинно, если хотя бы одна из ее составных частей имеет истинное значение.

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Импликацией двух высказываний X и У называется составное высказывание z=x→y. Оно будет ложью когда x – истинно, а y – ложно, в остальных случаях будет истиной.

x	y	$x \to y$
0	0	0
0	1	0
1	0	1
1	1	0

Эквиваленцией двух высказываний X и У называется составное высказывание z=x↔y. Оно истинно, только когда оба высказывания либо истины, либо ложны.

x	y	$x \leftrightarrow y$
0	0	1
0	1	0
1	0	0
1	1	1

Исключающее двух высказываний X и У называется составное высказывание z=x(+)y. Оно истинно, только когда оба высказывания не совпадают.

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

2. Формулы алгебры высказываний и порядок выполнения операций. Таблицы истинности

С помощью логических операций над высказываниями можно строить различные формулы. Порядок операций: $\neg,\wedge,\vee,(+),\to,\leftrightarrow$. Всякое высказывание, кот. м.б. получено из элементарных высказываний путём применения конечного числа логических операций, наз. ***формулой алгебры логики.*** Логическое значение формулы определяется заданными логическими значениями входящих в неё элементарных высказываний. Если ставится задача определить все возможные формулы в зависимости от значения входящих в неё – решается через построение **таблицы истинности**.

3. Равносильности логических формул

Две формулы алгебры логики А и В называются равносильными, если они принимают одинаковые логические значения при любом наборе значений входящих в формулы элементарных высказываний (переменных).

Обозначение. А≡В. Формула А называется **тождественно истинной** (тавтологией), если она принимает значение 1 при всех значениях входящих в её переменных. Формула А называется **тождественно ложной**, если она принимает значение 0 при всех значениях входящих в её переменных.

Свойства: 1) А=А –рефлексивность, 2) Если А=В, то В=А – симметричность, 3) А=В, В=С, то А=С – транзитивность 4)x*/+y=y*/+x – коммунитативность, 5) (x*/+y)*/+z=(x*/+z)*/+y – ассоциативность, 6) (x*/+y)+/*z=(x+/*z)*/+ (y+/*z) – дистрибутивность.

1. Основные равносильности.

- $x \wedge x \equiv x, x \vee x \equiv x$ – законы идемпотентности;
- $x \wedge 1 \equiv x$;
- $x \vee 1 \equiv 1$;
- $x \wedge 0 \equiv 0$;
- $x \vee 0 \equiv x$;
- $x \wedge \bar{x} \equiv 0$ – закон противоречия;
- $x \vee \bar{x} \equiv 1$ – закон исключенного третьего;
- $\bar{\bar{x}} \equiv x$ – закон снятия двойного отрицания;
- $x \vee (y \wedge x) \equiv x,\ x \wedge (y \vee x) \equiv x$ – законы поглощения.

1. Равносильности, выражающие одни логические опера

- $x \leftrightarrow y \equiv (x \to y) \wedge (y \to x)$;
- $x \to y \equiv \bar{x} \vee y$;
- $\overline{x \wedge y} \equiv \bar{x} \vee \bar{y}$;
- $\overline{x \vee y} \equiv \bar{x} \wedge \bar{y}$;
- $x \wedge y \equiv \overline{\bar{x} \vee \bar{y}}$;
- $x \vee y \equiv \overline{\bar{x} \wedge \bar{y}}$.

Формулы расщепления : $(x*y)+(x*\bar{y})=x$ и $(x+y)*(x+\bar{y})=x$

4. **Булевы функции и способы их задания**

Булевой функцией (*БФ*) или функцией алгебры логики (*ФАЛ*), от n переменных называется функция $f(x_1,x_2,x_3,\dots,x_n)$, которая на любом наборе своих аргументов может принимать одно из двух значений: 0 или 1. Под набором аргументов понимается совокупность значений переменных, каждая из которых может быть равна 0 или 1. Для функции от n аргументов количество возможных наборов равно $2^\wedge 2^n$

1) Задание булевой функции таблицей истинности

	x_1	x_2	...	x_n	$f(x_1, x_2, ..., x_n)$
0	0	0	...	0	$f(0, 0, ..., 0)$
1	0	0	...	1	$f(0, 0, ..., 1)$
...	...	...	...	...	...
a	a_1	a_2	...	a_n	$f(a_1, a_2, ..., a_n)$
...	...	...	...	...	...
$2^n - 1$	1	1	...	1	$f(1, 1, ..., 1)$

2) Задание булевой функции характеристическими множествами. Так называются два множества:

M^1_f, состоящее из всех наборов, на которых функция принимает значение 1, то есть $M^1_f = \{\alpha \in B^n : f(\alpha) = 1\}$;

M^0_f, состоящее из всех наборов, на которых функция принимает значение 0, то есть $M^0_f = \{\alpha \in B^n : f(\alpha) = 0\}$.

3) Задание булевой функции вектором ее значений.

$\varphi_f = f(0,0, \dots, 0) f(0,0, \dots, 1) \dots f(1,1, \dots, 1)$.

4) Задание булевой функции матрицей Грея. Булево пространство задается матрицей Грея, и наборы (клетки матрицы), на которых булева функция $f(x_1, \dots, x_n)$ принимает значение 1, отмечаются и называются *точками*.

5) Интервальный способ задания булевой функции. Булеву функцию $f(x_1, \dots, x_n)$ можно задать множеством интервалов $I_f = \{I_1, I_2, \dots, I_k\}$, объединение которых образует характеристическое множество M^1_f, то есть $I_1 \cup I_2 \cup \dots \cup I_k = M^1_f$. Множество интервалов I_f называется *достаточным* для функции $f(x_1, \dots, x_n)$.

6) Задание булевой функции формулами.

5. Дизъюктивные формы представления логический функций. Приведение к ДНФ

ДНФ логической функции называется дизъюнкция любого конечного множества попарно различных элементарных конъюнкций. $f(x_1, x_2, x_3) = x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee \bar{x}_1 x_3$.

Алгоритм приведения формулы к ДНФ:

1. Выражают все логические операции в формуле через дизъюнкцию, конъюнкцию и отрицание.

2. Используя законы Де-Моргана переносят все отрицания к переменным и сокращают двойные отрицания.

3. Используют закон дистрибутивности конъюнкции относительно дизъюнкции, преобразуют формулу так, чтобы все конъюнкции встречались раньше дизъюнкции.

6. Совершенная нормальная дизъюктивная форма и её свойства

Совершенная дизъюнктивная нормальная форма формулы (СДНФ) - это дизъюнкция коньюнкций, обладающая свойствами:

1. Каждое логическое слагаемое формулы содержит все переменные, входящие в функцию F(x1,x2,...xn)

2. Все логические слагаемые формулы различны

3. Ни одно логическое слагаемое не содержит переменную и её отрицание

4. Ни одно логическое слагаемое формулы не содержит одну и ту же переменную дважды. СДНФ можно получить или с помощью таблиц истинности или с помощью равносильных преобразований.

7. Конъюнктивные формы представления логических функций. Приведение к КНФ

Что значит нормальна форма: Нормальная форма логической формулы не содержит знаков импликации, эквиваленции и отрицания неэлементарных формул.
Конъюнктивная нормальная форма, т. е. конъюнкция нескольких дизъюнкций
КНФ: $(x \vee \bar{y} \vee z) \wedge (y \vee z)$

Для приведение к КНФ:

1. Привести функцию к виду, используя базовые операции – конъюнкцию, дизъюнкцию и инверсию;

2. преобразовать инверсию с помощью законов де Моргана до отдельных букв;

3. уничтожить все суммы произведений, используя законы поглощения, с помощью второго закона дистрибутивности.

8. Совершенная нормальная коньюнктивная форма и её свойства

Совершенно конъюнктивная НФ - конъюнкция дизъюнкций, причём в каждой дизъюнкции (в каждой скобке) присутствуют все переменные, входящие в формулу, либо их отрицание, нет одинаковых дизъюнкций, в каждой дизъюнкции нет одинаковых слагаемых.

Свойства: 1. в ней нет одинаковых элементарных дизъюнкций

2.в каждой дизъюнкции нет одинаковых пропозициональных переменных

3.каждая элементарная дизъюнкция содержит каждую пропозициональную букву из входящих в данную КНФ пропозициональных букв.

9. Базис представления логических функций(Функционально полная система)

Функционально полная система логических элементов - это такой набор элементов, используя который можно реализовать любую сколь угодно сложную логическую функцию. Поскольку любая логическая функция представляет собой комбинацию простейших функций - дизъюнкции, конъюнкции и инверсии, то набор из элементов трех типов, реализующих соответственно функции И, ИЛИ и НЕ, естественно, является функционально полным.

Система логических функций $\sum = \{f_1, \dots, f_m\}$ называется *Функционально полной системой*, если любая логическая функция может быть выражена через функции $f_1, \dots, f_m$ с помощью их суперпозиции

10. Процедуры приведения ДНФ к КНФ и наоборот

Алгоритм этого перехода следующий: ставим над ДНФ два отрицания и с помощью правил де Моргана (не трогая верхнее отрицание) приводим отрицание ДНФ снова к ДНФ. При этом приходится раскрывать скобки с использованием правила поглощения (или правила Блейка). Отрицание (верхнее) полученной ДНФ (снова по правилу де Моргана) сразу дает нам КНФ:

$$xy \vee y\bar{z} = \overline{\overline{xy \vee y\bar{z}}} = \overline{\overline{xy} \cdot \overline{y\bar{z}}} = \overline{(\bar{x} \vee \bar{y})(\bar{y} \vee z)}$$

$$= \overline{\bar{y} \vee \bar{x}z} = y \cdot (\overline{\bar{x}z}) = y \cdot (x \vee \bar{z}).$$

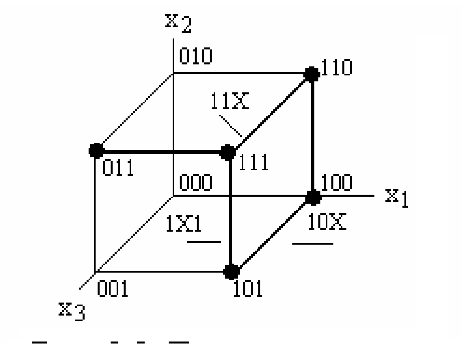
Заметим, что КНФ можно получить и из первоначального выражения, если вынести y за скобки;

б) переход от КНФ к ДНФ. Этот переход осуществляется простым раскрытием скобок (при этом опять-таки используется правило поглощения)

$(x \vee \bar{y} \vee \bar{z})(\bar{x} \vee \bar{y})(\bar{y} \vee \bar{z}) = (\bar{y} \vee \bar{x}\bar{z})(x \vee \bar{y} \vee \bar{z}) = \bar{y} \vee \bar{x}\bar{z}$. Таким образом, получили ДНФ.

11.Геометрическое представление логических функций. Контактные схемы

В геометрическом смысле каждый набор значений_переменных $(x_1, x_2, \ldots, x_n)$ можно рассматривать как n – мерный вектор, определяющий точку в n –мерном пространстве. Все множество двоичных наборов значений аргументов образует геометрическое множество вершин n –мерного единичного куба. Выделяя вершины, на которых значение функции равно 1, можно получить геометрический образ истинностной функции.



Контактная схема (англ. *contact circuit*) представляет собой [ориентированный ациклический граф](), на каждом ребре которого написана переменная или ее отрицание.
При операции конъюнкции(умножение) идёт последовательное соединение контактов, при операции дизъюнкция – параллельное. При отрицании – ключ замкнут.

12. Минимизация логических функций. Правила минимизации

Минимальной формой представления переключательной функции называют такую форму, которая не допускает больше никаких упрощений.

Основные этапы метода гиперкубов

- Построить таблицу истинности.
- Выписать все гиперкубы из M1(f) и импликанты.
- Взять простые импликанты.
- Построить таблицу накрытия.
- Из оставшихся простых импликантов создать тупиковую ДНФ.

Возьмем в качестве примера следующую булеву функцию.

$$f(x,y,z) = xy \vee \overline{x}z \vee yz$$

	x	y	z	$f(x,y,z)$
0	0	0	0	0
1	0	0	1	**1**
2	0	1	0	0
3	0	1	1	**1**
4	1	0	0	0
5	1	0	1	0
6	1	1	0	**1**
7	1	1	1	**1**

$M_1(K)$	K
$\{1\}$	$\bar{x}\bar{y}z$
$\{3\}$	$\overline{x}yz$
$\{6\}$	$xy\overline{z}$
$\{7\}$	xyz
$\{1,3\}$	$\overline{x}z$
$\{3,7\}$	yz
$\{6,7\}$	xy

	1	3	6	7
$\overline{x}z$	1	1		
yz		1		1
xy			1	1

1) Построим таблицу истинности для нее:
2) Выпишем все гиперкубы, лежащие в M1(f) и соответствующие им импликанты:
3) Выбираем простые импликанты и строим таблицу их накрытия:
4) Поскольку импликанта yz перекрывается другими, ее можно изъять из выражения. Выходит, что тупиковая ДНФ функции имеет вид:

$$f(x,y,z) = xy \vee \overline{x}z$$

13. Принцип двойственности в булевой алгебре

Булева функция f*(x1, …, xn) называется двойственной булевой функции f(x1, …, xn), если она получена из f(x1, …, xn) инверсией всех аргументов и самой функции.

Пример: $(x \vee y)^* = \overline{(\overline{x} \vee \overline{y})} = \overline{\overline{x}}\,\overline{\overline{y}} = x\,y.$ • Функция f называется самодвойственной если **f*=f.**

ТЕОРЕМА (**Закон двойственности**)
Если формула f1 равносильна формуле f2 , то формула f1* равносильна формуле f2*.
ТЕОРЕМА (**Принцип двойственности**)
Двойственная к булевой формуле может быть получена заменой констант 0 на 1, 1 на 0, /\ на \/, \/ на /\ и сохранением структуры формулы.

МАТЕМАТИЧЕСКАЯ КОМБИНАТОРИКА

1. Правила суммы и произведения

Пусть X – конечное множество такое, что |X| = n. Тогда говорят, что объект x из X может быть выбран n способами. Пусть X1, …, Xn – попарно непересекающиеся множества, то есть Xi ∩ Xj = ∅ при любых i ≠ j. Тогда, очевидно, выполняется равенство $\left|\bigcup_{i=1}^{n} X_i\right| = \sum_{i=1}^{n} |X_i|$.

В комбинаторике этот факт называется **правилом суммы.** Для n = 2 оно формулируется следующим образом: «Если объект x может быть выбран m способами, а объект y – другими n способами, то выбор “либо x, либо y” может быть осуществлен m + n способами».

Правило произведения: Если объект x1 может быть выбран n1 способами, после чего объект x2 может быть выбран n2 способами и для любого i, где 2 ≤ i ≤ m − 1, после выбора объектов x1, ..., xi объект xi+1 может быть выбран ni+1 способами, то выбор кортежа (x1, x2, ..., xm) длины m может быть осуществлен n1 · n2 · … · nm способами. Сформулируем частный случай этого правила для кортежа длины 2: «Если объект x может быть выбран m способами и после каждого из таких выборов объект y в свою очередь может быть выбран n способами, то выбор упорядоченной пары (x, y) может быть осуществлен m · n способами».

2. Размещения

Размещениями называют комбинации, составленные из n различных элементов по m элементов, которые отличаются либо составом элементов, либо их порядком. Число всех возможных размещений вычисляется по формуле:

$$A_n^m = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-m+1) = \frac{n!}{(n-m)!}$$

(п, к)-размещением с повторениями называется упорядоченная (п, А;)-выборка, элементы в которой могут повторяться;

(п, к)-размещением без повторений называется упорядоченная (п, /с)-выборка, элементам в которой повторяться запрещено;

3. Перестановки

Перестановками называют комбинации, состоящие из одних тех же n различных элементов и отличающиеся только порядком их расположения.
Число всех возможных перестановок находится по формуле:
$P_n = 1 \cdot 2 \cdot \ldots \cdot n = n!$,

4. Сочетания

Сочетаниями называют комбинации, составленные из n различных элементов по m элементов, которые отличаются хотя бы одним элементом.
Число всех возможных сочетаний находится по формуле:
$C_n^m = \frac{n!}{(n-m)! \cdot m!}$.

• неупорядоченная (n,k)-выборка с повторяющимися элементами называется (n,k)-сочетанием с повторениями

• неупорядоченная (n,k)-выборка без повторяющихся элементов называется (n,k)-сочетанием без повторений.

5. Бином Ньютона

Числа C(n, k) возникают как коэффициенты при раскрытии скобок в биноме (a + b)^n.

В общем случае, раскрывая скобки в биноме (a + b)^n, мы будем получать члены вида

a^{n-k} b^k (где к принимает каждое из значений от 0 до n) при перемножении символов b, взятых из k скобок, и a, взятых из оставшихся (n — k) скобок. Так как есть ровно C(n,k) способов выбора k скобок из n, то у нас будет в точности C(n,k) членов вида a^{n-k} b^k при A: = 0, 1, ...,n. Следовательно $(a+b)^n = C(n, 0)a^n + C(n, 1)a^{n-1}b + C(n, 2)a^{n-2}b^2 + \cdots + C(n, n)b^n.$

Эта формула называется биномом Ньютона. Ровно поэтому коэффициенты C(n,k) часто называют биномиальными коэффициентами.

Биномиальные коэффициенты полезно выстроить в так называемый треугольник Паскаля

$$
\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
\cdots & & \cdots & & \cdots & & \cdots & & \cdots & & \cdots
\end{array}
$$

6. Размещения с повторением

Предположим, что мы берем элементы x1, x2, ..., xn из множества X мощности к. Каждый такой набор принято называть выборкой объема k из n элементов или, иначе, (n, k)-выборкой. Выборка называется упорядоченной, если порядок следования элементов в ней задан. При этом две упорядоченные выборки, различающиеся лишь порядком следования элементов, считаются разными. Если же порядок следования элементов в выборке не имеет значения, то выборка называется неупорядоченной.

(n, k)-**размещением с повторениями** называется упорядоченная (n, k)-выборка, элементы в которой могут повторяться;

Пример. Всевозможные размещения с повторениями из трех элементов a, b, c по 2:

$aa, ab, ac, ba, bb, bc, ca, cb, cc.$

Теорема. Число всевозможных размещений с повторениями из n элементов по k равно

$n^k.$

7. Сочетания с повторением

Предположим, что мы берем элементы x1, x2, ..., xn из множества X мощности к. Каждый такой набор принято называть выборкой объема k из n элементов или, иначе, (n, k)-выборкой. Выборка называется упорядоченной, если порядок следования элементов в ней задан. При этом две упорядоченные выборки, различающиеся лишь порядком следования элементов, считаются разными. Если же порядок следования элементов в выборке не имеет значения, то выборка называется неупорядоченной.

Неупорядоченная (n, k)-выборка с повторяющимися элементами называется (n, k)-**сочетанием с повторениями**.

Теорема. Число сочетаний с повторениями из n элементов по k выражается формулой

$$\tilde{C}_n^k = \frac{(n+k-1)!}{k!(n-1)!} = C_{n+k-1}^k.$$

Пусть имеется три элемента (n = 3): a, b и c. Тогда из этих трёх элементов можно составить шесть сочетаний с повторениями по два элемента (k = 2): ab, ac, bc, aa, bb, cc (порядок неважен!)

8. Перестановка с повторением

Если в основном множестве k элементов a1,a2,...ak и выборка n элементов составляется так:

элемент a1 повторяется n1 раз,

элемент a2 повторяется n2 раз,

...

элемент ak повторяется nk раз,

такие выборки называются перестановками с повторениям.

Их возможное количество вычисляется вычисляется по формуле:

$$\overline{P_n} = P_{n_1,n_2,\ldots,n_k} = \frac{n!}{n_1!\cdot n_2!\cdot\ldots\cdot n_k!}$$

Пример:

Сколько различных пятибуквенных слов можно составить из букв слова «манна»?

Решение:

В слове буквы а и н повторяются 2 раза, а буква м один раз.

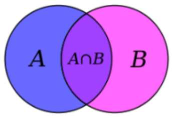
$$\overline{P_5} = P_{2,2,1} = \frac{5!}{2!\cdot 2!\cdot 1!} = \frac{2\cdot 3\cdot 4\cdot 5}{2\cdot 2} = 30$$

9. Формулы включений и исключений

Формула включений-исключений — комбинаторная формула, позволяющая определить мощность объединения конечного числа множеств, которые в общем случае могут пересекаться друг с другом. Например, в случае двух множеств AA, BB формула включений-исключений имеет вид:

$|A \cup B| = |A| + |B| - |A \cap B|$.

Действительно, сумме $|A|+|B|$ элементы пересечения$A \cap B$ учтены дважды, и чтобы компенсировать это мы вычитаем $|A \cap B|$ из правой части формулы



Таким же образом и в случае $n>2$ множеств процесс нахождения количества элементов объединения $A_1 \cup A_2 \cup \ldots \cup A_n$ состоит во включении всего, затем исключении лишнего, затем включении ошибочно исключенного и т. д. Отсюда и происходит название формулы. Формула включений-исключений утверждает:

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i} |A_i| - \sum_{i<j} |A_i \cap A_j| + \sum_{i<j<k} |A_i \cap A_j \cap A_k| - \ldots$$

$$\ldots + (-1)^{n-1} |A_1 \cap A_2 \cap \ldots \cap A_n|.$$

ОСНОВЫ ТЕОРИИ ГРАФОВ

1. Графы, основные понятия и определения

Пусть V – непустое множество и E – набор пар элементов множества V, причем в парах могут быть одинаковые элементы и допускается повторение пар. Тогда совокупность (V, E) называется графом G. Будем обозначать иногда этот граф как G(V, E). Элементы множества V называются вершинами графа, а элементы множества E – ребрами. Ребра графа могут представляться как неупорядоченными парами {vi, vj}, так и упорядоченными (vi, vj). В последнем случае ребро называется ориентированным, или дугой, vi – начальной вершиной (началом), vj – конечной вершиной (концом) данной дуги. Ребро {vi, vi} или дуга (vi, vi) называется петлей. Граф, состоящий из вершин и соединяющих их ребер, называется неориентированным, а граф, состоящий из вершин и соединяющих их дуг, – ориентированным (орграфом). Графы, содержащие как ребра, так и дуги, именуются смешанными. Вершины, соединенные между собой хотя бы одним ребром или дугой, называются смежными. Аналогично, два ребра, имеющие хотя бы одну общую вершину, называются смежными. 4 и e5. Если ребро ek соединяет две вершины, т. е. ek = {vi, vj}, ek = (vi, vj) или ek = (vj, vi), то ребро ek называется инцидентным вершинам vi и vj или вершины vi и vj называются инцидентными ребру ek. Граф называется конечным, если множества его вершин и ребер конечны (пустое множество тоже рассматривается как конечное). Назовем граф обыкновенным или простым, если в нем отсутствуют петли и кратные ребра (дуги). Граф, имеющий кратные ребра (дуги), называется мультиграфом, а граф, в котором есть хотя бы одна петля, называется псевдографом. Два графа G и H изоморфны (записывается G≡H), если между их множествами вершин существует взаимно однозначное соответствие, сохраняющее смежность. Если ребра ориентированы, то их направления также должны соответствовать друг другу. Обыкновенный неориентированный граф называется полным, если любые две различные его вершины смежные. Обыкновенный орграф называется полным, если в нем любые две различные вершины соединены парой антипараллельных дуг. Число ребер неориентированного графа, инцидентных вершин v, называется степенью, или порядком, этой вершины. При подсчете числа ребер, инцидентных вершине v, некоторую неопределенность вносит петля, так как ее можно считать и как единственное, и как двойное ребро. Будем обозначать степень вершины v через d(v) (или deg(v)). Изолированными называются вершины, которые не являются концами ребер и не связаны ни между собой, ни с другими вершинами. Изолированность вершины v в неориентированном графе эквивалентна условию d(v) = 0. Вершина степени 1 (единица) называется концевой, или висячей, вершиной, если петля считается двойной. Маршрутом в графе называется чередующаяся последовательность вершин и рёбер. Если vq — Vk, то маршрут замкнут, иначе — открыт. Если все рёбра различны, то маршрут называется цепью. Замкнутая цепь называется циклом, замкнутая простая цепь называется простым циклом. Число циклов в графе G обозначается z(G). Граф без циклов называется ациклическим. Для орграфов цепь называется путем, а цикл — контуром. Цепь(цикл) наз-ся гаментоновым, если он проходит через все вершины графа по одному разу.

2. Способы представления графов

Матрицей смежности вершин неориентированного графа G называется квадратная матрица A(G) = (aij) порядка p (p – количество вершин графа), элементы aij которой равны числу ребер, соединяющих вершины νi и νj (при этом петля может означать одно или два ребра по договоренности).

Матрицей смежности вершин ориентированного графа G называется квадратная матрица A(G) = (aij) порядка p (p – количество вершин графа), элементы aij которой равны числу дуг, исходящих из вершины νi и заходящих в вершину νj.

Матрицей инцидентности неориентированного графа G называется матрица B(G) = (bij) размерности p × q (p и q – количество вершин и ребер графа соответственно), элементы которой определяются следующим образом: 1, если вершина инцидентна ребру ; 0, если вершина не инцидентна ребру. **Матрицей инцидентности орграфа** G с вершинами p и ребрами q называется матрица B(G) = (bij) размерности p × q, элементы которой определяются следующим образом: 1, если вершина начало дуги ; -1, если вершина конец дуги ; 1, если начало; 0, если вершина не инцидентна дуге.

Список смежности. Представляет собой структуру данных, которая для каждой вершины графа хранит список смежных с ней вершин. Список представляет собой массив указателей, i-ый элемент которого содержит указатель на список вершин, смежных с i-ой вершиной.

Список инцидентности. Эта структура содержит для каждой вершины $v \in V$ список таких вершин $u \in V$, что существует дуга (v,u) в орграфе или ребро (v,u) в неориентированном графе. Каждый элемент такого списка содержит два поля: информацию о вершине и указатель на следующий элемент в списке.

3. Изоморфизм графов

Два графа G и H изоморфны (записывается GH), если между их множествами вершин существует взаимно однозначное соответствие, сохраняющее смежность. Если ребра ориентированы, то их направления также должны соответствовать друг другу. Очевидно, что отношение изоморфизма графов является эквивалентностью, т. е. оно рефлексивно, симметрично и транзитивно. Следовательно, множество всех графов разбивается на классы так, что графы из одного класса попарно изоморфны, а графы из разных классов не изоморфны. Изоморфные графы естественно отождествлять, т. е. считать совпадающими (их можно изображать одним рисунком). На практике доказывается с помощью перестановки строк и столбцов матрицы смежности.

4. Частичные графы. Подграфы

Граф H(x) называется **частичным** для графа G(X), если все ребра H(X) являются ребрами G(X) и множество вершин графа H(X) совпадает с множеством вершин графа G(X) Частичный граф содержит **часть ребер** (дуг). Подграф содержит **часть вершин** вместе с ребрами, соединяющими эти вершины. Граф называется связным, если любые две его вершины можно соединить цепью. Ребро графа называется перешейком, если его удаление приводит к тому, что граф становится несвязным. Граф из одних перешейков называется деревом.

5. Маршруты, цепи, циклы

Маршрутом в графе называется чередующаяся последовательность вершин и рёбер. Если vq — Vk, то маршрут замкнут, иначе — открыт. Если все рёбра различны, то маршрут называется цепью. Замкнутая цепь называется циклом, замкнутая простая цепь называется простым циклом. Число циклов в графе G обозначается z(G). Граф без циклов называется ациклическим. Для орграфов цепь называется путем, а цикл — контуром. Цепь(цикл) наз-ся гаментоновым, если он проходит через все вершины графа по одному разу.

6. Эйлеровы графы

Граф называется **эйлеровым**, если для всякой вершины графа найдется маршрут начинающейся и заканчивающейся в этой вершине и проходящий через каждое ребро только один раз. Такой маршрут называется эйлеровым циклом.

Теорема. Граф является Эйлером тогда и только тогда, когда степени всех его вершин четные. **алгоритм**:начнем маршрут P в произвольной вершине a графа G и будем продолжать его, насколько возможно, все время через новые ребра,. Так как к каждой вершине подходит четное число ребер, этот процесс может окончиться только в исходной вершине не a.(чистка улиц)

7. Гамильтоновы графы

Граф называется гамильтоновым, если для каждой вершины графа найдется маршрут начинающейся и заканчивающей в этой вершине и проходящий через все вершины только один раз. Такой маршрут называется гамильтоновым циклом. (поезд и станции)

Алгоритма нет, но:

1. Выбрать исходную вершину и включить её в маршрут.

2. Выбрать вершину ближайшую к данной по весу, включить её в маршрут.

3. Выбрать не использованные вершины ближайшую к последней, включить её в маршрут.

4. Вернуться к шагу 2 если не использованы все вершины.

5. Включить в маршрут исходную вершину.

8. Связность. Цикломатическое число

1.Неориентированный граф считается связным, если из любой вершины есть путь в любую другую вершину (путь может состоять из любого количества рёбер) - Чтобы граф с n вершинами был связным, он должен иметь не менее (n-1) рёбер. - Если граф имеет не менее (n*n - 3n + 4)/2 рёбер, то он гарантированно связный. - Если граф связный, у него обязательно есть вершины степени не менее 2, то есть вершины, каждая из которых имеет не менее двух смежных вершин. - Если граф связный и без циклов (то есть это дерево), то удаление любого ребра приведёт к потере связности. **2.**
Величина $k = N - n + 1$ называется **цикломатическим числом** связного графа. N – ребра, n – вершины. В формуле q=1 – для связного графа, q – количество компонентов связности.

9. Плоские и планарные графы. Свойства планарных графов

Плоский граф – это граф, который нарисован на плоскости так, что никакие два его ребра не пересекаются. **Планарный граф –** это граф, изоморфный плоскому графу. Для связного плоского графа справедливо следующее соотношение между количеством вершин V(G), рёбер E(G) и граней, формула Эйлера:

F(G) - |V(G)|-|E(G)|+|F(G)|=2

Плоский граф – это граф уложенный на плоскость, а планарный граф – это граф который можно уложить на плоскость.

10. Раскраска графа. Хроматическое число

При раскраске элементам графа ставятся в соответствие метки с учётом определенных ограничений; эти метки традиционно называются «цветами». В простейшем случае такой способ окраски **вершин графа**, при котором любым двум смежным вершинам соответствуют разные цвета, называется **раскраской вершин**.

Аналогично **раскраска рёбер** присваивает цвет каждому ребру так, чтобы любые два смежных ребра имели разные цвета. Наконец, **раскраска областей** планарного графа назначает цвет каждой области, так, что каждые две области, имеющие общую границу, не могут иметь одинаковый цвет. Наименьшее число красок, необходимое для правильной раскраски графа G называется *хроматическим числом* графа G.

11. Операции над графами

1) **Дополнением графа** $G_1(V_1,E_1)$ (обозначение — $\overline{G_1}(V_1,E_1)$)называется граф $G_2(V_2,E_2)$, где $V_2 = V_1$ & $E_2 = \overline{E_1} = \{e \in V_1 \times V_1 \mid e \notin E_1\} = V \times V \setminus E$.
2) **Объединением** (дизъюнктным) графов $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ (обозначение — $G_1(V_1,E_1) \cup G_2(V_2,E_2)$, при условии $V_1 \cap V_2 = \varnothing$) называется граф $G(V,E)$, где $V = V_1 \cup V_2$ & $E = E_1 \cup E_2$.
3) **Соединением графов** $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ (обозначение — $G_1(V_1,E_1) + G_2(V_2,E_2)$, при условии $V_1 \cap V_2 = \varnothing$) называется граф $G(V,E)$, где $V = V_1 \cup V_2$ & $E = E_1 \cup E_2 \cup \{e = (v_1, v_2) \mid v_1 \in V_1$ & $v_2 \in V_2\}$.
4) **Удаление вершины** v из графа $G_1(V_1,E_1)$ (обозначение — $G_1(V_1,E_1) - v$ при условии $v \in V_1$) даёт граф $G_2(V_2,E_2)$, где
$$V_2 = V_1 - v \ \& \ E_2 = E_1 \setminus \{e = (v_1, v_2) \mid v_1 = v \lor v_2 = v\}.$$
5) **Удаление ребра** e из графа $G_1(V_1,E_1)$ (обозначение — $G_1(V_1,E_1) - e$ при условии $e \in E_1$) даёт граф $G_2(V_2,E_2)$, где $V_2=V_1$, $E_2 = E_1 - e$.
6) **Добавление вершины** v в граф $G_1(V_1,E_1)$ (обозначение — $G_1(V_1,E_1) + v$ при условии $v \notin V_1$) даёт граф $G_2(V_2,E_2)$, где $V_2 = V_1 + v$, $E_2 = E_1$.
7) **Добавление ребра** e в граф $G_1(V_1,E_1)$ (обозначение — $G_1(V_1,E_1) + e$ при условии $e \notin E_1$) даёт граф $G_2(V_2,E_2)$, где $V_2=V_1$, $E_2 = E_1 + e$.
8) **Стягивание** (правильного) подграфа A графа $G_1(V_1,E_1)$ (обозначение — $G_1(V_1,E_1)/A$ при условии $A \subset V_1, v \notin V_1$) даёт граф $G_2(V_2,E_2)$, где
$$V_2 = (V_1 \setminus A) + v,$$
$$E_2 = E_1 \setminus \{e = (u, w) \mid u \in A \lor w \in A\} \cup \{e = (u, v) \mid u \in \Gamma(A) \setminus A\}.$$
9) **Размножение вершины** v графа $G_1(V_1,E_1)$ (обозначение — $G_1(V_1,E_1) \uparrow v$ при условии $v \in V_1, v' \notin V_1$) даёт граф $G_2(V_2,E_2)$, где
$$V_2 = V_1 + v' \ \& \ E_2 = E_1 \cup \{(v, v')\} \cup \{e = (u, v') \mid u \in \Gamma^+(v)\}.$$
10) **Отожествление вершин.** Пусть u и v – две вершины графа $G = (V, E)$. Удалим эти вершины из графа G и добавим новую вершину x, соединив ее ребром с каждой вершиной, входящей в объединение окружений вершин u и v в исходном графе G. Построенный граф получился из графа G отождествлением вершин u и v. Отождествление вершин u и v называется стягиванием ребра (u,v), если $(u,v) \in E(G)$.
11) **Расщепление вершины.** Пусть v – вершина графа G. Рассмотрим два множества $N_1(v)$ и $N_2(v)$, объединение которых совпадает с окружением $N(v)$ вершины v. Удалив вершину v, добавим новые вершины v_1, v_2 и ребро (v_1, v_2). Соединим v_1 с каждой вершиной из $N_1(v)$, а v_2 – с каждой вершиной из $N_2(v)$. Произведенная операция называется расщеплением вершины v, а полученный граф обозначается G_v^*.
12) **Дублирование вершин.** При дублировании графа G получим новый граф G2 следующим образом: добавим вершину u’ и соединим эту вершину со всеми вершинами, которые смежны с вершиной u.
13) **Разбиение ребра.** При разбиении ребра l(u,v) графа G получаем новый граф следующим образом: удалим ребро e(u,v) из множества графа, добавим вершину w. Добавим рёбра (u,w), (w,v).

14) **Пересечение графов.** Пусть **графов** $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ – произвольные графы. Пересечением $G_1(V_1,E_1) \cup G_2(V_2,E_2)$, называется граф со множеством вершин $V = V1 \cap V2$и множеством ребер $E=E1 \cap E2$.
15) **Композиция графов.** Пусть $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$– два ориентированных графа с одними и теми же множествами вершин V. Композицией (иногда произведением) G1(G2) графов G1 и G2 называется ориентированный граф с множеством вершин V, в котором существует дуга (vi, νj) тогда и только тогда, когда для некоторой вершины $\nu \in V$ существуют дуги $(\nu i, \nu) \in E1$ и $(\nu, \nu j) \in E2$. Операция композиции может быть выполнена в матричной форме.
16) **Декартово произведение графов.** $V(G1\times G1) = V(G1) \times V(G2)$.

12. Деревья, основные понятия, определения и теоремы

Связный граф без циклов называется (свободным) деревом. Деревья являются в некотором смысле простейшим классом графов.

Ориентированные деревья. Ориентированным деревом (или ордеревом, или корневым деревом) называется орграф со следующими свойствами.

1. Существует единственный узел r, полустепень захода которого равна 0, $d^+(r) = 0$. Он называется корнем ордерева.

2. Полустепень захода всех остальных узлов равна 1, $d^+(v) = 1$.

3. Каждый узел достижим из корня.

Упорядоченные деревья. Множества T1, ..., Tk в эквивалентном определении ордерева являются поддеревьями. Если относительный порядок поддеревьев T1, ..., Tk фиксирован, то ордерево называется упорядоченным.

Последовательное дерево, представляющее собой простую цепь, и звездное дерево (или **куст**), в котором одна из вершин (центр) смежна со всеми остальными вершинами. **Лесом** называют граф, связные компоненты которого являются деревьями.

Свойства: 1. Каждое дерево с   вершинами имеет в точности   ребро. 2. Граф является деревом тогда и только тогда, когда каждая пара различных вершин графа соединяется одной и только одной простой цепью. 3.У каждого дерева найдется висячая вершина. 4.При удалении любого ребра дерева оно распадается на связные компоненты, являющиеся либо изолированными вершинами, либо деревьями. При добавлении в дерево любого нового ребра в нем образуется простой цикл, и оно перестает быть деревом.

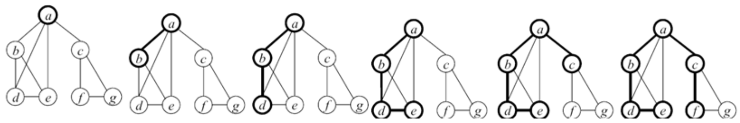
Полные бинарные деревья. Полным бинарным деревом будем называть такое дерево, в котором каждая вершина имеет не более двух «сыновей», а заполнение вершин осуществляется в порядке от верхних уровней к низшим, причем на одном уровне заполнение вершин производится слева направо. Верхним считается уровень с номером 1 (самый высокий).

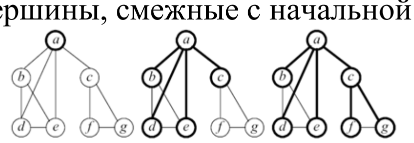
13. Остовное дерево минимального веса и способы его построения

Подграф в G, являющийся деревом и включающий в себя все вершины G, называется остовным деревом. Остовное дерево в графе G строится просто: выбираем произвольное его ребро и последовательно добавляем другие ребра, не создавая при этом циклов, до тех пор, пока нельзя будет добавить никакого ребра, не получив при этом цикла. Для построения остовного дерева в графе из n вершин необходимо выбрать ровно n - 1 ребро. На языке теории графов нам нужно в нагруженном графе найти остовное дерево наименьшего общего веса. Такое дерево принято называть минимальным остовным деревом или, сокращенно, МОД. **Алгоритм Прима**: берем любую вершину, всякий раз ищется минимальное по весу ребро, один конец которого — уже взятая в граф вершина, а другой конец — ещё не взятая, и это ребро добавляется в граф(если таких рёбер несколько, можно взять любое). Этот процесс повторяется до тех пор, пока граф не станет содержать все вершины (или, что то же самое,   ребро).

14. Обходы вершин графа: поиск в ширину и поиск в глубину

В глубину: когда возможные пути по *ребрам*, выходящим из вершин, разветвляются, нужно сначала полностью исследовать одну ветку и только потом переходить к другим.



Таким образом, основная идея **поиска в ширину** заключается в том, что сначала исследуются все вершины, смежные с начальной вершиной (*вершина* с которой начинается обход). 

15. Задача о кратчайшем пути в орграфе. Алгоритм Форда

Находит кратчайшие пути от одной вершины графа до всех остальных во взвешенном графе. Вес ребер может быть отрицательным.
ЗАДАЧА
Дан ориентированный или неориентированный граф G со взвешенными рёбрами. Требуется найти кратчайшие пути от выделенной вершины s до всех вершин графа.
АЛГОРИТМ
.Перед началом работы алгоритма, для всех вершин, кроме стартовой, расстояние полагается равным бесконечности.
.В цикле проверяется необходимость производить релаксацию для конкретного ребра (сравнение текущего пути, с заново посчитанным).
.Если текущая метка вершины больше чем метка нового пути, то она изменяется в его сторону, иначе остается неизменной.
.Алгоритм заканчивает свою работу, только если на одном из его очередных шагов не было не проведено ни одной релаксации.

$$a_{ij} = \begin{cases} 0, & \text{если вершины } v_i \text{ и } v_j \text{ совпадают,} \\ \infty, & \text{если вершины } v_i \text{ и } v_j \text{ не смежны,} \\ w, & \text{если вершины } v_i \text{ и } v_j \text{ смежны, и } w - \text{вес ребра } (v_i, v_j). \end{cases}$$

16. Отношение порядка между вершинами орграфа

В любом ориентированном графе без циклов можно установить отношение порядка между его вершинами. Рассмотрим сеть G=(J,U) с одним источником S и одним стоком T. В сети n вершин. Рангом вершины i назовем max число дуг в пути, которое связывает источник – вершину S, с данной вершиной i.

Правильной считается такая нумерация вершин в сети, при которой все номера вершин r-го ранга будут меньше номеров вершин (t+1)-го ранга. Рассмотрим одни из способов, который называется методом вычеркивания дуг.

Исходной вершине сети, т.е. источнику присваиваем номер 1. Затем вычеркнем (удалим) все дуги выходящие из вершины 1. После этого в сети окажутся конечное число вершин без входящих дуг. Назовем эти вершины, вершины 1-го ранга и присвоим ми в произвольном порядке номера 2,…,k_1. Затем вычеркнем (удалим) все дуги, выходящие из верши 1-го ранга. В сети опять окажется конечное число вершин без входящих дуг. Назовем их вершинами 2-го ранга и присвоим им в произвольном порядке номера k_1+1,…,k_2.

И т.д., процесс нумерации вершин прекращается тогда, когда будет пронумерована последняя вершина сети – сток.

17. Задача о пути максимальной длины в орграфе

Задача ставится следующим образом.
Задан конечный ориентированный граф без контуров G(X,U).
Каждой дуге графа “u” ставится в соответствие длина дуги l(u).
Требуется определить длиннейший путь, соединяющий две вершины графа x_0 и x_n.
Алгоритм
Каждая вершина графа получает числовую метку, которая может меняться конечное число раз. Установившаяся метка – величина длиннейшего пути из вершины x_0 в данную вершину x_j. В частности, установившаяся метка вершины x_n есть величина длиннейшего пути из x_0 в x_n.
Чтобы определить искомый путь, нужно рассмотреть последовательность шагов, на каждом из которых ищется одна из дуг длиннейшего пути между x_0 и x_n.
Алгоритм состоит в последовательном проведении следующих этапов:
1. Полагаем $\lambda_0 = 0$; $\lambda_i = -\infty$ $(i = 1,\dots,n)$.
2. Ищем дугу (x_i, x_j) такую, что $\lambda_j - \lambda_i \le l(x_i, x_j)$. Если такой дуги нет, то не существует пути, соединяющего x_0 и x_n. Если такая дуга найдется, то изменяем метку l_j на $l'_j = l_j + l(x_0, x_j)$.
3.Продолжаем процедуру пункта 2 до тех пор, пока метки вершин x_i не перестанут меняться.
Установленные метки обозначим l^*_i. При этом могут встретиться два случая:
1) $l^*_n = -\infty$, это соответствует тому, что пути, соединяющего вершины x_0 и x_n, не существует;
2) l^*_n- конечное число. Оно равно длине пути максимальной длины из x_0 в x_n.
Сам путь находим, отмечая вершины, по которым достигается максимум, т.е. те вершины, для которых $\lambda^*_j = \lambda^*_i + l(x_i, x_j)$
Если между вершинами графа-сети установлено отношение порядка, т.е. они “правильно” занумерованы, то решение задачи можно получить за один шаг, произведя подсчет меток с учетом следующей формулы:

$$\lambda_j = \max_i \{\lambda_i + l(x_i, x_j)\}$$

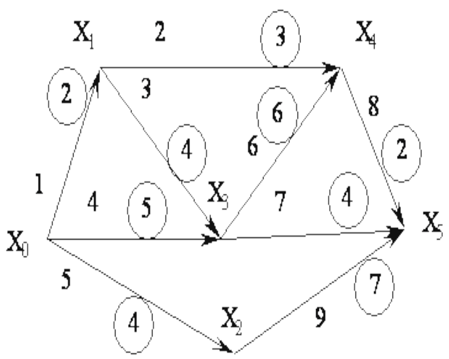
Пример.
Определим длиннейший путь на графе, изображенном на рис.3.3.1, а также его длину.
Вначале полагаем для вершины x_0□$_0$=0 и □$_j=-\infty$ для вершин x_i (i=1,…,5).
Затем, т.к. □$_1$-□$_0=-\infty$ <l(x_0,x_1), меняем метку вершины x_1, т.е. □$_1$, на
□’$_1$=□$_0$+l(x_0,x_1)=2 (x_0)
Аналогично □’$_2$=□$_0$+l(x_0,x_2)=4.
Чтобы найти метку вершины x_3, пользуясь формулой (3.3.2)
□’$_3$=max{[□’$_1$+l(x_1,x_3)],[□$_0$+l(x_0,x_3)]}=max{(2+4),(0+5)}=6 (x1)
Справа в скобках отмечаем вершины, по которым достигается максимум длины.
Аналогично
□’$_4$=max{[□’$_1$+l(x_1,x_4)],[□’$_3$+l(x_3,x_4)]}=max{(2+3),(6+6)}=12 (x_3)
□’$_5$=max{[□’$_3$+l(x_3,x_5)],[□’$_4$+l(x_4,x_5)],[□’$_2$+l(x_2,x_5)]= (x_4)
=max{(6+4),(12+2),(4+7)}=14
Искомый путь имеет длину l(□)=□$^*_5$=14. Причем в x_5 он идет из вершины x_4, в x_4 из x_3, в x_3 из x_1, в x_1 из x_0: x_5,x_4,x_3,x_1,x_0. Следовательно □=(x_0,x_1,x_3,x_4,x_5).
Путь максимальной длины называют критическим путем. Следовательно, критический путь в рассмотренном примере есть □=(x_0,x_1,x_3,x_4,x_5), а его длина l(□)=14.

18. Сетевое планирование. Задача о скорейшем пути завершения проекта

Сетевое планирование — метод анализа сроков (ранних и поздних) начала и окончания нереализованных частей проекта, позволяет увязать выполнение различных работ и процессов во времени, получив прогноз общей продолжительности реализации всего проекта. Задача сетевого планирования состоит в том, чтобы графически, наглядно и системно отобразить и оптимизировать последовательность и взаимозависимость работ, действий или мероприятий, обеспечивающих своевременное и планомерное достижение конечных целей.

Рассмотрим некоторый проект – совокупность операций (работ), составляющий некоторый многошаговый процесс. Пусть данные о строительстве приведены в следующей таблице

Виды работ	Какие работы следуют за перечисленными	Продолжитель-ность работ
1	2,3	2
2	8	3
3	6,7	4
4	6,7	5
5	9	4
6	8	6
7	–	4
8	–	2
9	–	7



Эту информацию о проекте представим в виде графа-сети. Дугами графа будем изображать работы, а вершинами графа – некоторые события. Назовем элементарными событиями начало и конец каждой работы, а некоторую совокупность элементарных событий – событием.
Вход графа – событие, заключающееся в начале всего проекта. Оно является событием, стоящим в начале одной или нескольких работ, а именно тех, которые не следуют ни за какими другими, т.е. работ, с которых может быть начато строительство. В нашем примере такими работами являются №1,4,5 (их нет во 2-ом столбце). Выходом графа будет являться событие, заключающееся в окончании работ, за которыми не следуют никакие другие работы, т.е. в окончании всего проекта. В данном примере – это работы №7,8,9. Все другие вершины графа есть события, заключающиеся в окончании одних и начале других работ. На графе номер работы обозначен числом вне кружка. Число, обведенное кружком, есть продолжительность данной работы.
Путь максимальной длины из вершины x_0 в x_i есть скорейшее время наступления события x_i. В самом деле, событие x_3, например, соответствующее началу 6-й и 7-й работ, может произойти только после окончания 3-й и 4-й работ, а следовательно, и после окончания 1-й, т.к. для выполнения 3-й работы необходимо окончание 1-й работы. Следовательно, скорейшее время наступления события x_3 есть max{5,(2+4)}=6.
Скорейшее время наступления события 5 есть скорейшее время окончания проекта в целом и равно длине пути максимальной длины из вершины x_0 в x_5.
Итак, если x_0 и x_n есть вход и выход графа-сети, соответствующего данному проекту, то для определения наиболее раннего срока окончания всех работ нужно найти путь максимальной длины из x_0 в x_n, т.е. критический путь, и определить его длину. Время, соответствующее скорейшему окончанию работ, т.е. скорейшему завершению проекта, называется критическим временем данного проекта. Оно численно совпадает с длиной критического пути из x_0 в x_n.
В приведенном примере критический путь, проходящий через вершины x_0,x_1,x_3,x_4,x_5, имеет длину, равную 14 l(m)=14, т.е. критическое время данного проекта равно 14.
Работы, составляющие критический путь, называются критическими работами (операциями). От своевременного выполнения критических операций зависит срок завершения проекта. Они не допускают запаздывания в исполнении в отличие от некритических операций.

ТЕОРИЯ АЛГОРИТМОВ

1. Задача «Выполнимость»

Задача „Выполнимость"

Пусть $d_1, d_2, \ldots, d_n$ – некоторые дизъюнкты, тогда конъюнкция $d_1 \wedge d_2 \wedge \ldots \wedge d_n$ называется булевым выражением в КНФ или (NP-полная)

Постановка задачи:

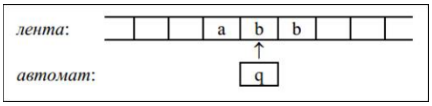
Пусть задано некоторое выражение в КНФ. Набор значений б.ф называют выполняющим, если на этом наборе б.ф $= 1$. Требуется найти выполняющий набор для выражения $d_1 \wedge d_2 \wedge \ldots \wedge d_n$. (SAT конференций SAT 2000, SAT 2007...) Имеются булевы выражения, для которых не существует выполняющего набора. Например, б.в

$(x \vee y \vee z) \wedge (x \vee \neg y) \wedge (y \vee \neg z) \wedge (z \vee \neg x) \wedge (\neg x \vee \neg y \vee \neg z)$ не является истинным ни для каких значений переменных.

Эта задача имеет многочисленные приложения в различных областях знаний!

2. Машина Тьюринга. Принцип работы. Основные команды

Машина Тьюринга — абстрактная вычислительная машина, предназначенная для формализации понятия алгоритма, имитирующая с помощью задания правил перехода других исполнителей, реализующих процесс пошагового вычисления; устройство, описание и схемы машины. Машина Тьюринга (МТ) состоит из двух частей — ленты и автомата. Лента используется для хранения информации. Она бесконечна в обе стороны и разбита на клетки, которые никак не нумеруются и не именуются. В каждой клетке может быть записан один символ или ничего не записано. Содержимое клетки может меняться — в неё можно записать другой символ или стереть находящийся в ней символ.



Автомат — это активная часть МТ. В каждый момент он размещается под одной из клеток ленты и видит её содержимое; это видимая клетка, а находящийся в ней символ — видимый символ; содержимое соседних и других клеток автомат не видит. Кроме того, в каждый момент автомат находится в одном из состояний, которые будем обозначать буквой «q» с номерами: «q1», «q2» и т. п. Находясь в некотором состоянии, автомат выполняет какую-то определённую операцию (например, перемещается направо по ленте, заменяя все символы «b» на «a»), находясь в другом состоянии — другую операцию. Автомат может выполнять три элементарных действия:

а) записывать в видимую клетку новый символ (менять содержимое других клеток автомат не может);

б) сдвигаться на одну клетку влево или вправо («перепрыгивать» сразу через несколько клеток автомат не может);

в) переходить в новое состояние.

Ничего другого делать автомат не умеет, поэтому все более сложные операции так или иначе должны быть сведены к этим трём элементарным действиям.

МТ работает тактами, которые выполняются один за другим. На каждом такте автомат МТ выполняет три следующих действия, причем обязательно в указанном порядке:

а) записывает некоторый символ «S» в видимую клетку (в частности, может быть записан тот же символ, что и был в ней, тогда содержимое этой клетки не меняется);

б) сдвигается на одну клетку влево (обозначение — L), либо на одну клетку вправо (обозначение — R), либо остается неподвижным (обозначение — H).

в) переходит в некоторое состояние «q» (в частности, может остаться в прежнем состоянии).

2. Универсальная кодировка машины Тьюринга

Каждая машина Тьюринга имеет три алфавита: внешний *A,* внутренний *Q* и алфавит сдвигов *S.* Все три алфавита - конечные множества, и можно считать, что

$A \cap Q = A \cap S = Q \cap S = \varnothing$.

Тогда любое слово над $A \cup Q \cup S$ однозначно разбивается на слоги (подслова) (возможно, и однобуквенные) над алфавитами *A, Q, S. В* частности, любую клетку программы машины как отображения $A \cup \{\Lambda\} \times Q \setminus \{q_0\} \to A \cup \{\Lambda\} \times Q \times S$ можно закодировать как пятибуквенное слово над $A \cup \{\Lambda\} \cup Q \cup S$, задающее соответствие $(\alpha, q_i) \to (\beta, q_j, s)$. Если условиться, что программа выписывается последовательно по столбцам сверху вниз, то мы получим стандартную запись программы в виде «длинного» слова над алфавитом $A \cup \{\Lambda\} \cup Q \cup S$. Поставим следующий вопрос: «Нельзя ли выбрать достаточно простой алфавит, с помощью слов которого можно будет кодировать все буквы, а значит, и слова над $A \cup \{\Lambda\} \cup Q \cup S$ (т.е. внешние слова и программу машины)?» Оказывается, в качестве такого простого алфавита, называемого универсальным, можно взять двухбуквенный алфавит {0; 1}. Опишем стандартную кодировку букв алфавита $A \cup \{\Lambda\} \cup Q \cup S$ словами над {0; 1} с помощью таблицы кодирования.

Кодировка с помощью этой таблицы называется стандартной. (Ясно, что стандартная кодировка допускает однозначное декодирование дли любых *A, Q, S.* Поэтому можно считать, что всегда применяется кодировка с помощью {0; 1} и что все машины работают со словами над алфавитом {0; 1}.

Буква	Код (слово) над {0; 1}
Сдвиги	-1
+1	
Буквы	Λ
a_1	
a_2	
...	...
a_k	$1\underbrace{0\ldots0}_{2k+4}1$
...	...
Состояния	q_0
q_1	
...	...
q_j	$1\underbrace{0\ldots0}_{2j+5}1$
...	...

3. Алгоритмически неразрешимые проблемы. Теория о самоприменимости машины Тьюринга. Тезис Тьюринга

Определение 15.12. Машина Тьюринга T называется самоприменимой, если она применима к слову $и_T$ стандартной записи на ленте своей программы.
Это определение разбило все множество машин Тьюринга МТна два класса: $МТ_S$ - класс самоприменимых и $МТ_{nS}$ - класс несамоприменимых машин.
Сформулируем проблему распознавания самоприменимости: «Существует ли такая машина Тьюринга S, которая умеет распознавать самоприменимость, т. е. по слову $и_T$, кодирующему программу машины T, сообщать, является ли машина T самоприменимой или нет?» Оказывается, имеет место следующая теорема.
Теорема 15.10. Проблема распознавания самоприменимости является алгоритмически неразрешимой по Тьюрингу проблемой.
Допустим противное, т. е. что существует машина S, решающая проблему самоприменимости, причем
$S(и_T)= 1$, если T - самоприменима,
$S(и_T)= 0$, если T - несамоприменима.

Если машина S существует, то ее можно перестроить в машину $\bar{S}$, которая в случае, когда $и_T$ - кодировка программы несамоприменимой машины, перерабатывает $и_T$ в слово «0» и останавливается, а в случае, когда $и_T$ - кодировка программы самоприменимой машины, «выпечатывает», уходя вправо, бесконечный хвост «1». Покажем, что существование машины $\bar{S}$, а значит и S, ведет к противоречиям. Для этого применим $\bar{S}$ к слову $u_{\bar{S}}$. Возможны два исхода:

а) после применения $\bar{S}$ к $u_{\bar{S}}$ машина напечатает 0 и остановится, но, с одной стороны, это означает (0) - несамоприменимость, а с другой (остановка машины) - самоприменимость;

б) в результате применения $\bar{S}$ к $u_{\bar{S}}$ идет без остановки печать бесконечного хвоста единиц. Хвост единиц означает теперь самоприменимость, а это противоречит тому, что машина не останавливается.
Полученное противоречие и доказывает теорему. В заключение опишем, что такое универсальная машина Тьюринга. В обычной машине Тьюринга программа «зашита» в устройство управления, а входная информации (условия задачи) записывается на бесконечной ленте. В универсальной машине в устройстве управления записана программа, реализующая алгоритм подражания, т. е. программную реализацию правил работы любой машины (см. п.15.2), вернее, правила обращения с ее программой. Тогда входной информацией для универсальной машины является пара - слово $и_т$, стандартно кодирующее машину-алгоритм, решающий данный класс задач Z; и слово v_z, кодирующее условие задачи $z \in Z$. Универсальная машина, используя $и_т$, перерабатывает v_z в $T(v_z)$. Ясно, что для построения универсальной машины потребуется использование техники машин с полулентами и универсального алфавита. Эта машина будет работать очень «вяло», теряя очень много времени на переходы от обрабатываемого слова к программе и обратно, однако ясно, что справедлива следующая теорема.

ОСНОВЫ ТЕОРИИ МНОЖЕСТВ

8. Способы задания множеств
9. Операции над множествами и их свойства. Диаграммы Венна
10. Декартово произведение множеств, булеан, мощность множества
11. Упорядоченные множества. Проекция множества
12. Соответствия, основные определения, способы задания
13. Функции, способы задания, композиция функций
14. Бинарные отношения и их свойства
15. Способы задания бинарных отношений
16. Операции над бинарными отношениями

ОСНОВЫ АЛГЕБРЫ ЛОГИКИ

2. Высказывания и операции над ними
3. Формулы алгебры высказываний и порядок выполнения операций. Таблицы искренности
4. Равносильности логических формул
5. Булевы функции и способы их задания
6. Дизъюктивные формы представления логический функций. Приведение к ДНФ
7. Совершенная нормальная дизъюктивная форма и её свойства
8. Конъюнктивные формы представления логических функций. Приведение к КНФ
9. Совершенная нормальная коньюнктивная форма и её свойства
10. Базис представления логических функций(Функционально полная система)
11. Процедуры приведения ДНФ к КНФ и наоборот
12. Геометрическое представление логических функций. Контактные схемы
13. Минимизация логических функций. Правила минимизации
14. Принцип двойственности в булевой алгебре

МАТЕМАТИЧЕСКАЯ КОМБИНАТОРИКА

6. Правила суммы и произведения
7. Размещения
8. Перестановки
9. Сочетания
10. Бином Ньютона
11. Размещения с повторением
12. Сочетания с повторением
13. Перестановка с повторением
14. Формулы включений и исключений