Министерство образования Республики Беларусь
УО «Полесский государственный университет»

**О.В. СИДСКАЯ**

# ОСНОВЫ ДИСКРЕТНОЙ МАТЕМАТИКИ И ТЕОРИИ АЛГОРИТМОВ

Учебно-методическое пособие
для студентов специальности
«Информационные системы и технологии»
(по направлениям)

Пинск
ПолесГУ
2020

УДК 519.6:510.5(075.8)
ББК 22.176 + 22.12я73
С34

Р е ц е н з е н т ы:
кандидат технических наук, доцент Н.Н. Коваленко;
кандидат физико-математических наук, доцент П.А. Павлов

У т в е р ж д е н о
научно-методическим советом ПолесГУ

**Сидская, О.В.**
С34 Основы дискретной математики и теории алгоритмов : учебно-методическое пособие / О.В. Сидская. – Пинск : ПолесГУ, 2020. – 75 с.

ISBN 978-985-516-424-2

Содержит краткие теоретические сведения, примеры и задачи по следующим разделам дискретной математики и теории алгоритмов: множества, отношения, алгебры, логические функции, элементы математической логики, элементы теории графов, методы комбинаторного поиска, основы теории алгоритмов и автоматов.

Может быть использовано для студентов специальности информационные системы и технологии (по направлениям).

УДК 519.6:510.5(075.8)
ББК 22.176 + 22.12я73

ISBN 978-985-516-424-2

## СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

В настоящее время применение математических моделей систем и процессов в природе и обществе и обществе является актуальным и своевременным.

Основы дискретной математики, изучаемые студентами, являются важнейшей составляющей математической подготовки будущих специалистов по управлению, так как дискретные модели тесно связаны с дискретными способами обработки информации, которые стали преобладающими в кибернетике.

# ГЛАВА 1. МНОЖЕСТВА. ОТНОШЕНИЯ. АЛГЕБРЫ

## 1.1. Способы задания множеств. Операции над множествами. Декартово произведение множеств

*M1*±– множество всех натуральных чисел: *1, 2, 3, ...* . В дальнейшем будем обозначать его *N*; элементы *N* – натуральные числа. Часто *0* также считают натуральным числом множество, полученное добавлением *0* к *N*, будем обозначать *N0*.

*M2* – множество всех натуральных чисел, не превосходящих *100*.

*M3* – множество всех решений уравнения $sin\ x = 1$; элементы *M3* – числа, являющиеся решениями этого уравнения.

*M4* – множество всех чисел вида $П/2 \pm kП$, где $k \in N0$.

*M5* – множество всех действительных чисел (в дальнейшем *R*).

*M6* – футбольная команда «Зенит» (т.е. множество ее футболистов).

*M7* – множество всех футбольных команд высшей лиги в сезоне 2004 г.

Множество может быть задано перечислением (списком своих элементов), порождающей процедурой или описанием характеристических свойств, которыми должны обладать его элементы.

***Операции над множествами***

Два множества *A* и *B* равны $(A = B)$, если они состоят из одних и тех же элементов. Например, если $A = \{1, 2, 3, 4\}$, $B = \{3, 1, 4, 2\}$, то $A = B$.

*Объединением (суммой) множеств A* и *B* называется множество $A \cup B$, элементы которого принадлежат хотя бы одному из этих множеств. Например, если $A = \{1, 2, 4\}$, $B = \{3, 4, 5, 6\}$, то $A \cup B = \{1, 2, 3, 4, 5, 6\}$.

*Пересечением (произведением) множеств A* и *B* называется множество $A \cap B$, элементы которого принадлежат как множеству *A*, так и множеству *B*. Например, если $A = \{1, 2, 4\}$, $B = \{3, 4, 5, 2\}$, то $A \cap B = \{2, 4\}$.

*Разностью* множеств $A$ и $B$ называется множество $AB$, элементы которого принадлежат множеству $A$, но не принадлежат множеству $B$. Например, если $A=\{1,2,3,4\}$, $B=\{3, 4, 5\}$, то $AB=\{1, 2\}$.

*Симметричной разностью* множеств $A$ и $B$ называется множество $A \Delta B$, являющееся объединением разностей множеств $AB$ и $BA$, т.е. $A \Delta B=(AB) \cup (BA)$. Например, если $A=\{1, 2, 3, 4\}$, $B = \{3, 4, 5, 6\}$, то $A \Delta B = \{1, 2\} \cup \{5, 6\} = \{1, 2, 5, 6\}$.

1) переместительные законы пересечения и объединения (коммутативность):
   $A \cap B = B \cap A$; $A \cup B = B \cup A$;

2) сочетательные законы пересечения и объединения (ассоциативность):
   $(A \cap B) \cap C = A \cap (B \cap C)$; $(A \cup B) \cup C = A \cup (B \cup C)$;

3) $A \cap A = A$ $A \cup A = A$;

4) $A \cap \varnothing = \varnothing$ $A \cup \varnothing = A$;

5) $A \cap U = A$ $A \cup U = U$;

6) распределительные законы (дистрибутивность):
$(A \cup B) \cap C=(A \cap C) \cup (B \cap C)$ $(A \cap B) \cup C=(A \cup C) \cap (B \cup C)$;

7) законы включения:
$A \cap (B \cup C) \subset (A \cap B) \cup (A \cap C)$ $(A \cap B)(A \cap C) \subset A \cap (B \cup C)$;

вычитание и дополнение также обладают рядом свойств:

8) $A' \cap A = \varnothing$ $A' \cup A = U$;

9) $(A \cap B)' = A' \cup B'$ $(A \cup B)' = A' \cap B'$;

10) $\varnothing' = U$ $U' = \varnothing$;

11) $(AB)C = A(B \cup C)$ $(AB)C = (AC)B$;

12) $(AB) \cup B = A \cup B$ $(AB) \cap C = (A \cap B)(B \cap C)$;

13) $A(B \cup C) = (AB) \cap (AC)$ $A(B \cap C) = (AB) \cup (AC)$.

*Декартовым (или прямым) произведением множеств A и B называется* такое результирующее множество пар вида $(x, y)$, построенных таким образом, что первый элемент из множества $A$, а второй элемент пары – из множества $B$.

Общепринятое обозначение: $A \times B = \{(x, y) \mid x \in A, y \in B\}$.

*Рассмотрим несколько свойств декартова произведения.* Если $A$, $B$ – конечные множества, то $A \times B$ – конечное. И наоборот, если одно из множеств-сомножителей бесконечное, то и результат их произведения – бесконечное множество. Количество элементов в декартовом произведении равно произведению чисел элементов множеств-сомножителей (в случае их конечности, разумеется):

$$|A \times B| = |A| \bullet |B|.$$

Коммутативный закон не выполняется, т.к. пары элементов результата декартова произведения упорядочены:

$$A \times B \neq B \times A.$$

Ассоциативный закон не выполняется:

$$(A \times B) \times C \neq A \times (B \times C).$$

Имеет место дистрибутивность относительно основных операций на множествах:

$$(A * B) \times C = (A \times C) * (B \times C), \; * \in \{\cap, \cup, \setminus\}.$$

***Пример 1.1:***

Положим, $A = \{1, 2\}$, $B = \{3, 4\}$. Тогда результат декартова произведения можно записать так:

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

а $$B \times A = \{(3, 1), (3, 2), (4, 1), (4, 2)\}.$$

Если в предыдущем примере положить $B = A$, очевидно, что

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

Возьмём $A = \{x \in R \mid 0 \leq x \leq 5\}$, $B = \{x \in R \mid 5 \leq x \leq 10\}$.

Тогда $A \times B = \{(x, y) \in R^2 \mid 0 \leq x \leq 5 \wedge 5 \leq x \leq 10\}$.

Множества декартова произведения могут и не быть привычными числовыми множествами:

$$A = \{\circ, \diamond\}, B = \{2, 8\}, A \times B = \{(\circ, 2), (\circ, 8), (\diamond, 2), (\diamond, 8)\}.$$

## 1.2. Бинарные и *n*-арные отношения. Свойства бинарных отношений. Отношения эквивалентности и порядка. Функции, соответствия, отображения. Алгебраические структуры

*Декартовым произведением* множеств $X$ и $Y$ называется множество $X \times Y$ всех упорядоченных пар $(x, y)$, таких, что

$$x \in X,\ y \in Y.$$

*Соответствием* между множествами $X$ и $Y$ (или соответствием из $X$ в $Y$) называется любое подмножество декартова произведения $X \times Y$.

Если множества $X$ и $Y$ совпадают, то соответствие между множествами $X$ и $Y$ называют также *бинарным отношением* на множестве $X$.

*n-арные отношения* – это отношения, заданные на декартовом произведении более чем двух множеств.

***Свойства бинарных отношений***

Отношение $R$ называется *рефлексивным*, если для любого $a \in M$ имеет место $aRa$. Главная диагональ его матрицы содержит только единицы.

Отношение $R$ называется *антирефлексивным*, если ни для какого $a \in M$ не выполняется $aRa$. Главная диагональ его матрицы содержит только нули.

Отношения ≤ и «иметь общий делитель» рефлексивны, отношения ≤ и «быть сыном» антирефлексивны. Отношение «быть симметричным относительно оси $x$» не является ни рефлексивным, ни антирефлексивным: точка плоскости симметрична сама себе, если она лежит на оси $x$, и несимметрична сама себе в противном случае.

Отношение $R$ называется *симметричным*, если для пары $(a,b) \in M^2$ из $aRb$ следует $bRa$ (иначе говоря, для любой пары $R$ выполняется либо в обе стороны, либо не выполняется вообще). Матрица симметричного отношения симметрична относительно главной диагонали: $c_{ij} = c_{ji}$ для любых $i$ и $j$.

Отношение $R$ называется *антисимметричным*, если из $a_iRa_j$ и $a_jRa_i$ следует, что $a_i = a_j$.

Отношение *«быть симметричным относительно оси x»* является симметричным: если первая точка симметрична второй, то и вторая симметрична первой.

Пример антисимметричного отношения – отношение $\leq$: действительно, если $a \leq b$ и $b \leq a$, то $a = b$. Нетрудно убедиться в том, что отношение $R$ симметрично тогда и только тогда, когда $R = R^{-1}$.

Отношение $R$ называется *транзитивным*, если для любых $a, b, c$ из $aRb$ и $bRc$ следует $aRc$. Отношения «равенство», $\leq$, «жить в одном городе» транзитивны; отношение «быть сыном» нетранзитивно. Отношение «пересекаться», т.е. «иметь непустое пересечение», заданное на системе множеств, также нетранзитивно. Например, $\{1, 2\}$ пересекается с $\{2, 3\}$, $\{2, 3\}$ пересекается с $\{3, 4\}$, однако $\{1, 2\}$ и $\{3, 4\}$ не пересекаются. Отношение называется отношением эквивалентности (или просто эквивалентностью), если оно рефлексивно, симметрично и транзитивно.

***Пример 1.2:***

a) все классы эквивалентности по отношению равенства $E$ состоят из одного элемента;
b) формулы, описывающие одну и ту же элементарную функцию, находятся в одном классе эквивалентности по отношению равносильности; в этом примере счетны: само множество формул, множество классов эквивалентности (т.е. индекс разбиения) и каждый класс эквивалентности;
c) разбиение множества треугольников по отношению конгруэнтности имеет континуальный индекс, причем каждый класс также имеет мощность континуум; конгруэнтность является отношением эквивалентности на множестве треугольников;
d) разбиение $N$ по отношению «иметь общий остаток от деления на $7$» имеет конечный индекс $7$ и состоит из $7$ счетных классов $0, 7, 14, \ldots; 1, 8, 15, \ldots; 2, 9, 16, \ldots; \ldots; 6, 13, 20, \ldots$

*Отношения порядка.* Отношение называется *отношением нестрогого порядка*, если оно рефлексивно, антисимметрично и транзитивно. Отношение называется *отношением строгого порядка*, если оно антирефлексивно, антисимметрично и транзитивно. Оба типа отношений называются отношениями порядка. Элементы *a, b* сравнимы по отношению порядка *R*, если выполняется *aRb* или *bRa*. Множество *M*, на котором задано отношение порядка, называется *линейно упорядоченным*, если любые два элемента *M* сравнимы, и *частично упорядоченным* в противном случае.

***Пример 1.3:***

a) отношения $\leq$ и $\geq$ для чисел являются отношениями нестрогого порядка, отношения < и > – отношениями строгого порядка – оба отношения линейно упорядочивают множества *N* и *R*;
b) определим отношения *J* и < на $R_n$ следующим образом: $(a1, \ldots, an) \leq (b1, \ldots, bn)$, если $a1 \leq b1, \ldots, an \leq bn$; $(a1, \ldots, an) < (b1, \ldots, bn)$, если $(a1, \ldots, an) \leq (b1, \ldots, bn)$ и хотя бы в одной координате *i* выполнено $a_j < b_j$. Эти отношения определяют частичный порядок на $Rn : (5, 1/2, –3) < (5, 2/3, –3); (5, 1/2, –3)$ и $(5, 0, 0)$ несравнимы. $(5, 2/3, –3); (5, 1/2, –3)$ и $(5, 0, 0)$ несравнимы;
c) на системе подмножеств множества *M* отношение включения *H* задает нестрогий частичный порядок, а отношение строгого включения *M* задает строгий частичный порядок; например, $\{1, 2\}\ M\ \{1, 2, 3\}$, а $\{1, 2\}$ и $\{1, 3, 4\}$ несравнимы;
d) отношение подчиненности на предприятии задает строгий частичный порядок. В нем несравнимыми являются сотрудники разных отделов.

***Функции, соответствия, отображения***

*Соответствия*

Соответствием между множествами *A* и *B* называется подмножество $G \subseteq A \times B$.

Если $(a, b) \in G$, то говорят, что *b* соответствует, а при соответствии *G*.

Множество $np_1G$ называется *областью определения* соответствия, множество $np_2G$ – *областью значений* соответствия.

Если $np_1G = A$, то соответствие называется *всюду определенным* или *полностью определенным* (в противном случае соответствие называется *частичным*); если $np_2G = B$, то соответствие называется *сюръективным* (сюръекцией).

Множество всех $b \in B$, соответствующих элементу $a \in A$, называется *образом a* в *B* при соответствии *G*.

Множество всех $a$, которым соответствует $b$, называется *прообразом* $b$ в $A$ при соответствии $G$. Если $G \subseteq np_1G$, то образом множества $C$ называется объединение образов всех элементов $C$. Аналогично определяется прообраз множества $D$ для любого $D \subseteq np_2G$.

Соответствие $G$ называется *инъективным* (инъекцией), если прообразом любого элемента из $np_2G$ является единственный элемент из $np_1G = A$.

Соответствие $G$ называется *функциональным* (или *однозначным*), если образом любого элемента из $np_1G$ является единственный элемент из $np_2G$.

Соответствие $G$ между $A$ и $B$ называется *взаимно однозначным*, или *биекцией* (иногда пишут *«1–1-соответствие»*), если оно всюду определено, сюръективно, функционально и инъективно.

*Отображения и функции*

*Функцией* называется функциональное соответствие если функция $f$ устанавливает соответствие между множествами $A$ и $B$, то говорят, что функция $f$ имеет тип $A \rightarrow B$ (обозначение $f : A \rightarrow B$).

Каждому элементу $a$ из своей области определения функция $f$ ставит в соответствие единственный элемент $b$ из области значений. Это обозначается хорошо известной записью $f(a) = b$. Иногда, если это не вызывает неудобств, используют обозначения $fa$ или $af$.

Элемент $a$ называется *аргументом функции*, $b$ – *значением функции на* $a$.

Полностью определенная функция $f: A \rightarrow B$ называется *отображением* $A$ *в* $B$.

***Пример 1.4:***

a) функция $f(x) = 2^x$ является отображением $N$ в $N$ и $N_0$ на $M2^n$;
b) всякая нумерация счетного множества является его отображением на $N$;
c) функция $f(x) = \sqrt{x}$ не полностью определена, если ее тип $N \rightarrow N$, и полностью определена, если ее тип $N \rightarrow R$ или $R^+ \rightarrow R$ ($R^+$ – положительное подмножество $R$);
d) пусть зафиксирован список $\{a_1, \ldots, a_n\}$ всех элементов конечного множества $A$. Тогда любой вектор $v_i = (a_{i1}, \ldots, a_{in})$ из $A_n$

можно рассматривать как описание функции $f_i : A \to A$ (т.е. преобразования $A$), определяемой следующим образом: $f_i(a_j) = a_{ij}$, т.е. значение $f_i$ для $a_j$ равно $j$-й компоненте $v_j$. Число всех преобразований $A$ равно, следовательно, $|A^n| = n^n$. Аналогично всякую функцию типа $N \to N$ можно представить бесконечной последовательностью элементов $N$, т.е. натуральных чисел; отсюда нетрудно показать, что множество всех преобразований счетного множества континуально.

*Алгебраические структуры*

Всюду определенная (тотальная) функция $\phi : M^n \to M$ называется $n$-арной ($n$-местной) операцией на $M$. Если операция $\phi$- бинарная (т.е. $\phi : M^n \times M \to M$), то будем писать $a\phi b$ вместо $\phi(a, b)$.

*Замечание:* Такая форма записи называется инфиксной.

Множество $M$ вместе с набором операций $\sum = \{\phi_1, \ldots, \phi_m\}$, $\phi_i : M^n \to M$, где $n_i$-арность операции $\phi_i$, называется *алгебраической структурой*, *универсальной алгеброй* или просто *алгеброй*. Множество $M$ называется основным (несущим) множеством, или *основой (носителем);* вектор арностей $(n_1, \ldots, n_m)$ называется *типом.* Множество операций $\sum$ называется *сигнатурой.* Запись $(M; \sum)$ или $(M; \phi_1, \ldots, \phi_m)$.

*Замечание:* Операции $\phi_i$ *конечностны (финитарны),* сигнатура $\sum$ конечна. Носитель не обязательно конечен, но не пуст. Если в качестве $\phi_i$ допускается не только функция, но отношения, то множество $M$ вместе с набором операций и отношений называется *моделью.* В приложениях обычно используется следующее обобщение понятия алгебры.

Пусть $M = \{M_1, \ldots, M_n\}$ – множество *основ* $\sum = \{\phi_1, \ldots, \phi_n\}$ – сигнатура, причем $\phi_i : M_{i1} \times \ldots \times M_{in} \to M_j$. Тогда $(M; \sum)$ называется *многоосновной алгеброй*. Другими словами, многоосновная алгебра имеет несколько носителей, а каждая операция сигнатуры действует из прямого произведения некоторых носителей в некоторый носитель.

**Задачи**

1.1. Доказать равенства:

a) $A \backslash (A \backslash B) = A \cap B$;

b) $(A \backslash B) \backslash C = A \backslash (B \cup C)$;

c) $(A \backslash B) \cap C = (A \cap C) \backslash (B \cap C)$.

1.2. Доказать включения:

a) $A \cup (B \setminus C) \supset (A \cup B) \setminus C$;

b) $(A \cup C) \setminus B \subset (A \setminus B) \cup C$.

1.3. Решить систему уравнений:

$$\begin{cases} A \cap X = B, \\ A \cup X = C, \end{cases} \quad \text{где } B \subseteq A \subseteq C.$$

1.4. Доказать, что система уравнений $\begin{cases} A \cap X = \varnothing, \\ B \cap \overline{X} = \varnothing. \end{cases}$ имеет решение тогда и только тогда, когда $B \subseteq \overline{A}$, при условии решением системы является любое множество $X$, такое, что $B \subseteq X \subseteq \overline{A}$.

1.5. Построить бинарное отношение:

a) рефлексивное, симметричное, нетранзитивное;

b) рефлексивное, антисимметричное, нетранзитивное;

c) рефлексивное, несимметричное, транзитивное;

d) нерефлексивное, антисимметричное, транзитивное.

1.6: a) доказать, что всякое частично упорядоченное множество содержит не более одного наибольшего (наименьшего) элемента;

b) построить пример частично упорядоченного множества, имеющего точно один минимальный элемент, но не имеющего наименьшего элемента.

1.7. Установите истинность или ложность каждого из следующих высказываний:

a) $\{1,2\} \subseteq \{3,1,2\}$;
c) $\{1\} \in \{\{1\},2,3\}$;

b) $\{1\} \subseteq \{1,2,\{1,2,3\},\{1,3\}\}$;
d) $\varnothing \subseteq \{\varnothing\}$.

1.8. Пусть $A = \{1,2,3,4,5,6,7\}$;
$B = \{4,5,6,7,8,9,10\}$;
$C = \{2,4,6,8,10\}$;
$U = \{1,2,3,4,5,6,7,8,9,10\}$.

Определите множество $(C \cap B) \setminus A$:

a) $\{8,10\}$;
c) $\{6,10\}$;
e) $\{1,2,3,4,5,6,7,8,10\}$.

b) $\{9,10\}$;
d) $\{4,5,6\}$;

1.9: a) определите, соответствует или нет выражение $(\overline{A \cup B})$ множеству, изображенному с помощью диаграммы Эйлера-Венна;

b) определите, соответствует или нет выражение $(A \cup B) \setminus C$ множеству, изображенному с помощью диаграммы Эйлера-Венна.

1.10. В олимпиаде по математике приняло участие 10 учеников класса, в олимпиаде по биологии – 7 человек, а в олимпиаде по физике – 9 человек. Известно, что в олимпиадах по математике и биологии участвовало 4 ученика, а в олимпиадах по математике и физике – 5 учеников, а во всех трех олимпиадах – 2 ученика. Сколько школьников участвовали в олимпиадах по физике и биологии, если всего участников олимпиад было 17 человек?

1.11. Известно, что из 100 студентов живописью увлекается 28 человек, спортом – 42 человека, музыкой – 30, живописью и спортом – 10, живописью и музыкой – 8, спортом и музыкой – 5, живописью, спортом и музыкой – 3. Найти количество студентов, занимающихся только музыкой. Найти количество студентов, занимающихся только живописью. Найти количество студентов, занимающихся спортом и музыкой. Найти количество студентов, занимающихся только спортом.

1.12. Из 25 учеников класса «отлично» по английскому языку за полугодие получили 10 человек, а «отлично» по математике получили 14 человек, 7 учеников получили «отлично» по обоим предметам. Сколько учеников класса не имеют отличной оценки ни по математике, ни по английскому языку?

1.13. В ящике лежат 120 деталей. Из них на автомате № 1 – 82 штуки, на автомате № 2 – 23 штуки, на автомате № 3 – 42 штуки. 18 деталей было обработано на автоматах № 1 и № 2, 17 на автоматах № 2 и № 3. 10 деталей прошли обработку на всех трех автоматах. Сколько деталей не обработано ни на одном из автоматов?

## Вопросы

1. Выпишите элементы множества $M = (A - B) U (A \cap B)$ для множеств $A = \{1, 2, 3, 4, 5\}$, $B = \{3, 4, 5\}$ ; для множеств $A = \{2, 4, 5\}$, $B = \{1, 4\}$ ; для множеств $A = \{1\}$, $B = \varnothing$.

2. Дайте определение разбиения множества. Приведите все разбиения для множества $A = \{a, b, c\}$.
3. Какие множества называются эквивалентными? В каких случаях эквивалентны конечные и бесконечные множества?
4. Дайте определение счетного множества.
5. Что такое мощность множества? Дать определение.
6. Чему в математике служат отношения?
7. Как классифицируются отношения в зависимости от числа связей между элементами множества?
8. Дайте определение бинарного отношения.
9. Что представляет собой декартово произведение множеств?
10. Выпишите декартовы произведения множеств $A = \{a, b\}$, $B = \{2, 3\}$ ; декартового квадрата $A = \{1, a\}$.
11. Сколько элементов включает декартовый квадрат множества $A = \{1, 2, \ldots, i, \ldots, n\}$ ?
12. Дайте определение бинарного, тернарного и $n$-арного отношения в терминах множеств.
13. Что понимают под рефлексивными и антирефлексивными отношениями?
14. Как характеризуются симметричные, асимметричные и антисимметричные отношения?
15. Дайте определение транзитивного отношения.

# ГЛАВА 2. ЛОГИЧЕСКИЕ ФУНКЦИИ

## 2.1. Способы задания логических функций. Булевы функции двух переменных. Алгебра булевых функций

Различают табличный, матричный, графический и аналитический способы задания. При табличном задании используются так называемые «таблицы истинности» логических функций, в которых указываются значения функций на всём множестве комбинаций их аргументов. Т.о., число столбцов в таблице истинности определяется числом аргументов и числом функций, а количество строк – по формуле:

$$N = 2^N. \tag{1}$$

Матричный способ задания (или задание функций с помощью булевых матриц) основан на графическом отображении всего множества комбинаций аргументов функции на «плоскости» (в двумерном пространстве).

Булева матрица представляет собой прямоугольник с соотношением сторон 1 : 2 (при нечётном числе аргументов функции) или квадрат (при чётном числе аргументов), разделённые на элементарные квадраты (клетки). Число клеток в матрице всегда кратно степени двойки и определяется формулой (1). Графический способ задания логических функций основан на использовании *n*-мерных кубов. Размерность куба определяется числом *n* аргументов функции, например, функцию от трёх аргументов можно задать трёхмерным кубом, каждая вершина которого соответствует определённой комбинации аргументов.

Аналитический способ задания функций используется наиболее широко для отыскания функциональных схем синтезируемых устройств. Логическая переменная в алгебре логики может принимать одно из двух возможных значений: *TRUE* – истина, *FALSE* – ложь.

Логическая функция определяется как *n-местная функция*, определенная на множестве истинных значений < *Истина (True)*, *Ложь (False)* > и принимающая значения в этом множестве. Если последовательность логических переменных обозначить как $X = (x_1, x_2, ..., x_n)$ и назвать двоичным набором, то под функцией алгебры логики следует понимать однозначное отображение множества всевозможных наборов* на множество $Y = < 0,1 >$. Если две функции алгебры логики $f1(x_1, x_2, ..., x_n)$ и $f2(x_1, x_2, ..., x_n)$ принимают на всех возможных наборах одинаковые значения, то они называются *равными (эквивалентными)*. К элементарным функциям обычно относят: функцию инверсии (отрицания), конъюнкцию, дизъюнкцию, импликацию, штрих Шеффера и стрелку Пирса. Новые функции АЛ можно получить из известных функций либо путем перенумерации аргументов, либо путем подстановки в функцию новых функций вместо аргументов. Функция АЛ, полученная из функций $f_1, f_2, ..., f_k$ с помощью этих правил, называется суперпозицией функций $f_1, f_2, ..., f_k$.

### *Алгебра булевых функций*

*Булевы функции* – функции, аргументы и значения которых принимают значения истина и ложь. Истину и ложь будем обозначать соответственно *1* и *0*. Т.о., функция $n$ аргументов $f$ есть

$$f : \{0, 1\} \times \{0, 1\} \times ... \times \{0, 1\} \mid \{_z\}\, n \to \{0, 1\}.$$

Аргументы этих функций будем называть логическими переменными и обозначать буквами $x$, $y$ и $z$, возможно с индексами. Множество всех булевых функций (функций алгебры логики) будем обозначать $P^2$.

***Пример 2.1:*** Табличное задание функции $f$:

| | $x$ | $y$ | $z$ | $f(x,y,z)$ |
|---|---|---|---|---|
| | 0 | 0 | 0 | 1 |
| | 0 | 0 | 1 | 0 |
| | 0 | 1 | 0 | 1 |
| $2^3$ | 0 | 1 | 1 | 0 |
| | 1 | 0 | 0 | 0 |
| | 1 | 0 | 1 | 1 |
| | 1 | 1 | 0 | 0 |
| | 1 | 1 | 1 | 1 |

Всего существует $2^3$ различных наборов значений трёх переменных. Если их нумеровать от $0$ до $2^3 - 1$, то набор с номером $i$ оказывается представлением числа $i$ в двоичной системе счисления. Всего различных функций от трёх аргументов – $2^{2*2*2}$. В общем случае число строк в таблице для функции от $n$ аргументов равно $2^n$. Число различных булевых функций от $n$ аргументов – $2^2$.

*Определение 1*. Будем говорить, что функция $f(x_1, x_2, ..., x_n)$ не зависит существенно от $x_n$ ($x_n$ – несущественная переменная функции $f(x_1, x_2, ..., x_n)$), если на любых значений $\alpha 1, \alpha 2, ..., \alpha n{-}1 \in \{0, 1\}$ выполняется равенство $f(\alpha 1, \alpha 2, ..., \alpha n{-}1, 0) = f(\alpha 1, \alpha 2, ..., \alpha n{-}1, 1)$. Переменные функции $f$, которые не являются несущественными, называют существенными переменными и говорят, что функция $f$ существенно от них зависит.

*Определение 2.* Будем говорить, что формулы $U$ и $B$ эквивалентны, и писать $U = B$, если $f_U = f_B$ с точностью до несущественных переменных. Рассмотрим основные функции, используемые в качестве элементарных функций в алгебре логики. Всего существует четыре различные функции от одной переменной: тождественный ноль – $f(x) = 0$; тождественная единица – $f(x) = 1$; тождественная функция или тождественный $x$–$f(x) = x$; отрицание $x$ или *«не x»* – $f(x) = \neg x$ также обозначается $x$:

| $x$ | $0$ | $1$ | $x$ | $\neg x$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |

Из них тождественный ноль и тождественная единица не зависят существенно от $x$. Т.е. фактически это две функции без аргументов – константы: $f = 0$ и $f = 1$.

Рассмотрим основные булевы функции от двух переменных:

| $x$ | $y$ | $x \vee y$ | $x \wedge y$ | $x \oplus y$ | $x \supset y$ | $x \equiv y$ | $x \mid y$ | $x \downarrow y$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

$f(x, y) = x \vee y$ – дизъюнкция, логическое «или». $f(x, y) = x \wedge y$ – конъюнкция, логическое «и», логическое умножение. Также можно использовать обозначения $x\&y$ или $xy$. $f(x, y) = x \oplus y$ – сложение по *модулю 2*, логическое исключающее «или». Также можно использовать обозначение $x+y$, $f(x, y) = x \supset y$ – импликация, «если, то». Также можно использовать обозначение $x \to y$. $3f(x, y) = x \equiv y$ – эквивалентность. Также можно использовать обозначение: $x \sim y$. $f(x, y) = x \mid y$ – штрих Шеффера; $f(x, y) = x \downarrow y$ – стрелка Пирса.

Получим систему элементарных функций:

$$P = \{0, 1, x, x, x \vee y, xy, x \oplus y, x \supset y, x \equiv y, x \mid y, x \downarrow y\}.$$

Указанные функции будем теперь также называть операциями.

*Определение 3.* Формула, задающая тождественно истинную функцию, называется тавтологией.

*Определение 4.* Формула, задающая тождественно ложную функцию, называется противоречием.

*Определение 5.* Формула называется выполнимой, если для нее существует набор аргументов, на котором она принимает значение «1».

***Дизъюнктивная нормальная форма***

*Определение 6.* Введем следующее обозначение: $x\ \sigma = x,$ $\sigma = 1,\ x,\ \sigma = 0$. Также будем говорить $x$ в степени $\sigma$, имея в виду запись $x\ \sigma$, определенную выше.

*Утверждение 1.* Пусть $f(x_1, ..., x_n) \in P2$ и $f6 = 0$.

Тогда $f(x_1, ..., x_n) = \_(\sigma_1, ..., \sigma_n)\ f(\sigma_1, ..., \sigma_n) = 1\ x\ \sigma_1\ 1 \cdot\cdot\cdot x\ \sigma\ n\ (1)$.

*Определение 7.* Формула вида $x\ \sigma_{i1}\ i_1\ x\ \sigma_{i2}\ i2\ \cdot\ \cdot\ \cdot\ x\ \sigma_{ik}\ ik$, где $xij$ – логическая переменная; $\sigma_{ij}$ – логическая константа; $i_1 < i_2 < ... < i_k$ называется конъюнктом.

*Определение 8.* Если $f(x1, ..., x_n)$ представлена в виде $f(x_1, ..., x_n) = K_1 \vee K_2 \vee ... \vee K_s$, где $K_1, K_2, ..., K_s$ – различные конъюнкты, то говорят, что $f$ представлена в дизъюнктивной нормальной форме (*ДНФ*). Если в каждый $K_i$ входят все переменные $x_1, ..., x_n$, то говорят, что $f$ представлена в совершенной дизъюнктивной нормальной форме (*СДНФ*).

## 2.2. Нормальные формы логических функций. Полнота и замкнутость. Критерий полноты системы логических функций

*Конъюнктивным одночленом* от переменных $X_1, X_2...X_n$ называется конъюнкция этих переменных или их отрицаний.

Здесь «или» употребляется в неисключающем смысле, т.е. в конъюнктивный одночлен может входить одновременно и переменная, и ее отрицание.

Приведем несколько примеров конъюнктивных одночленов:

$$X_1 \wedge X_1,\ X_1 \wedge \neg X_2 \wedge X_3,\ X_2 \wedge \neg X_1 \wedge X_3 \wedge \neg X_2 \wedge X_5.$$

*Дизъюнктивным одночленом* от переменных $X_1, X_2, \ldots, X_n$ называется дизъюнкция этих переменных или их отрицаний (и здесь союз «или» употребляется в неисключающем смысле).

Приведем примеры дизъюнктивных одночленов:

$$X_1 \vee \neg X_2 \vee X_3,\ X_2 \vee X_2,\ X_1 \vee \neg X_2 \vee X_3 \vee \neg X_6 \vee X_5.$$

*Дизъюнктивной нормальной формой* называется дизъюнкция конъюнктивных одночленов, т.е. выражение вида $K_1 \vee K_2 \vee \ldots \vee K_p$, где все $K_i$, $i = 1, 2, \ldots, p$, являются конъюнктивными одночленами (необязательно различными).

Аналогично *конъюнктивной нормальной формой* называется конъюнкция дизъюнктивных одночленов $D_1 \wedge D_2 \wedge \ldots \wedge D_q$, где все $D_j$, $j = 1, 2, \ldots, q$ являются дизъюнктивными одночленами (необязательно различными).

Среди множества дизъюнктивных (равно как и конъюнктивных) нормальных форм, которыми обладает данная формула алгебры высказываний, существует уникальная форма – она единственна для данной формулы. Это т.н. *совершенная дизъюнктивная нормальная форма (среди конъюнктивных форм* – совершенная конъюнктивная нормальная форма).

Одночлен (конъюнктивный или дизъюнктивный) от переменных $X_1, X_2, \ldots, X_n$ называется совершенным, если в него от каждой пары $X_i$, $\neg X_i$ $(i = 1, 2, \ldots, n)$ входит только один представитель $(X_i$ или $\neg X_i)$.

Нормальная форма (дизъюнктивная или конъюнктивная) от переменных $X_1, X_2, \ldots, X_n$ называется совершенной от этих переменных, если в нее входят лишь совершенные одночлены (конъюнктивные или дизъюнктивные соответственно) от $X_1, X_2, \ldots, X_n$.

*Определение 1*. Система функций $P \subseteq P^2$ называется полной, если любую функцию из $P^2$ можно представить в виде формулы над $P$.

*Замечание 1*. Удобно доказывать полноту системы функций, показывая, что она сводится к уже известной полной системе.

***Теорема 1.*** Пусть даны две системы функций $P \subseteq P^2$ и $Q \subseteq P^2$. Пусть $P$ – полная система функций, и любая функция из $P$ представима в виде формулы над $Q$. Тогда $Q$ – полная система функций.

*Критерий полноты системы функций.* Существует пять классов функций: $T_0$, $T_1$, $S$, $M$, $L$. Каждый из этих классов функций замкнут и, как можно видеть из таблицы, ни один не совпадает с $P^2$:

| | $T_0$ | $T_1$ | $S$ | $M$ | $L$ |
|---|---|---|---|---|---|
| $\neg x$ | – | – | + | – | + |
| $0$ | + | – | – | + | + |
| $1$ | – | + | – | + | + |
| $xy$ | + | + | – | + | – |

***Теорема Поста:*** Для полноты системы функций $P \subseteq P^2$ необходимо и достаточно, чтобы $P$ не лежал полностью ни в одном из классов $T_0$, $T_1$, $S$, $M$, $L$:

$$P \not\subseteq T_0,\ P \not\subseteq T_1,\ P \not\subseteq S,\ P \not\subseteq M,\ P \not\subseteq L\,.$$

*Доказательство. Необходимость.* Если система функций $P$ лежит полностью в одном из классов $R \in \{T_0, T_1, S, M, L\}$, то, поскольку все эти классы замкнуты и не совпадают с $P^2$, $[P] \subseteq [R] \neq P^2$. Тогда система $P$ не является полной.

***Пример 2.2:***

Требуется проверить на полноту систему функций $P = \{0, 1, xy, x \oplus y \oplus z\}$. Рассмотрим принадлежность функций $P$ классам $T_0$, $T_1$, $S$, $M$, $L$ и заполним таблицу:

| | $T_0$ | $T_1$ | $S$ | $M$ | $L$ |
|---|---|---|---|---|---|
| $0$ | + | – | – | + | + |
| $1$ | – | + | – | + | + |
| $xy$ | + | + | – | + | – |
| $x \oplus y \oplus z$ | + | + | + | – | + |

Рассмотрим, например, проверку функции $x \oplus y \oplus z$:

a) $0 \oplus 0 \oplus 0 = 0 \Rightarrow x \oplus y \oplus z \in T_0$;

b) $1 \oplus 1 \oplus 1 = 1 \Rightarrow x \oplus y \oplus z \in T_1$;

c) $\overline{\bar{x} \oplus \bar{y} \oplus \bar{z}} = 1 \oplus (1 \oplus x) \oplus (1 \oplus y) \oplus (1 \oplus z) = x \oplus y \oplus z \Rightarrow$ ; $x \oplus y \oplus z \in S$;

d) $(1, 0, 0) \prec (1, 1, 0)$, но $1 = 1 \oplus 0 \oplus 0 > 1 \oplus 1 \oplus 0 = 0 \Rightarrow x \oplus y \oplus z \notin M$;

e) очевидно, функция является линейной: $x \oplus y \oplus z \in L$.

Теперь, заполнив и проанализировав таблицу, можно убедиться, что система функций $P$ является полной, т.к. в каждом столбце, соответствующем одному из классов, присутствует хотя бы один минус. В то же время ни одно подмножество $P$ полной системой не является, поскольку, если вычеркнуть в таблице хотя бы одну строку, появится столбец, не имеющий минуса.

*Определение.* Пусть $M$ – замкнутый класс функций. Пусть $B \subseteq M$. $B$ называется *базисом класса $M$*, если:

1) $[B] = M; f$;
2) $\forall f \in B \Rightarrow [B \setminus \{f\}]\, f = M$.

## 2.3. Минимизация логических функций. Метод минимизации Квайна – Мак-Класски. Метод Блейка

1. *Метод Квайна* – способ представления функции в *ДНФ* или *КНФ* с минимальным количеством членов и минимальным набором переменных. Преобразование функции можно разделить на два этапа:
   - на первом этапе осуществляется переход от канонической формы (*СДНФ* или *СКНФ*) к т.н. сокращённой форме;
   - на втором этапе – переход от сокращённой формы к минимальной форме.

*Первый этап (получение сокращённой формы).* Представим, что заданная функция *f* представлена в *СДНФ*.

Для осуществления первого этапа преобразование проходит два действия:

- операция склеивания;
- операция поглощения.

Операция склеивания сводится к нахождению пар членов, соответствующих виду $w \cdot x$ или $w \cdot \bar{x}$, и преобразованию их в следующие выражения:

$$w \cdot x \vee w \cdot \bar{x} = w \cdot (x \vee \bar{x}) = w.$$

Результаты склеивания $w$ теперь играют роль дополнительных членов. Необходимо найти все возможные пары членов (каждый член с каждым). Потом выполняется операция поглощения. Она основана на равенстве:

$$w \vee w \cdot z = w \cdot (1 \vee z) = w$$

(член $w$ поглощает выражение $w \cdot z$).

Вследствие этого действия из логического выражения вычёркиваются все члены, поглощаемые другими переменными, результаты которых получены в операции склеивания. Обе операции первого этапа могут выполняться до тех пор, пока это может быть осуществимо.

*Второй этап (табличный) (получение минимальной формы).* Как и на первом этапе, в полученном равенстве могут содержаться члены, устранение которых никаким образом не повлияет на конечный результат.

Следующий этап минимизации – удаление таких переменных. Представленная таблица содержит значения истинности функции. По ней будет собрана следующая *СДНФ*:

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $f(x_1, x_2, x_3, x_4)$ |
|---|---|---|---|---|
| *0* | *0* | *0* | *0* | *1* |
| *0* | *0* | *0* | *1* | *1* |
| *0* | *0* | *1* | *0* | *1* |
| *0* | *0* | *1* | *1* | *0* |
| *0* | *1* | *0* | *0* | *0* |
| *0* | *1* | *0* | *1* | *0* |
| *0* | *1* | *1* | *0* | *1* |
| *0* | *1* | *1* | *1* | *0* |
| *1* | *0* | *0* | *0* | *0* |
| *1* | *0* | *0* | *1* | *0* |
| *1* | *0* | *1* | *0* | *0* |
| *1* | *0* | *1* | *1* | *0* |
| *1* | *1* | *0* | *0* | *0* |
| *1* | *1* | *0* | *1* | *0* |
| *1* | *1* | *1* | *0* | *1* |
| *1* | *1* | *1* | *1* | *1* |

Собранная по этой таблице *СДНФ* выглядит следующим образом:

$$f(x_1, x_2, x_3, x_4) =$$
$$= \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 \vee \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \vee \overline{x_1} \cdot x_2 \cdot x_3 \cdot \overline{x_4} \vee x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4} \vee x_1 \cdot x_2 \cdot x_3 \cdot x_4 .$$

Конечное выражение достигается за счёт повторного использования операций склеивания и поглощения:

$$\overline{x_1}\cdot\overline{x_2}\cdot\overline{x_3}\vee\overline{x_1}\cdot\overline{x_2}\cdot\overline{x_4}\vee\overline{x_1}\cdot x_3\cdot\overline{x_4}\vee x_2\cdot x_3\cdot\overline{x_4}\vee x_1\cdot x_2\cdot x_3.$$

Члены этого выражения являются простыми импликантами выражения. Переход от сокращённой формы к минимальной осуществляется с помощью импликантной матрицы.

Члены *СДНФ* заданной функции вписываются в столбцы, а в строки – простые импликанты, т.е. члены сокращённой формы. Отмечаются столбцы членов *СДНФ*, которые поглощаются отдельными простыми импликантами.

В следующей таблице простая импликанта $\overline{x_1}\cdot\overline{x_2}\cdot\overline{x_3}$ поглощает члены $\overline{x_1}\cdot\overline{x_2}\cdot\overline{x_3}\cdot\overline{x_4}$ и $\overline{x_1}\cdot\overline{x_2}\cdot\overline{x_3}\cdot x_4$ (в первом и во втором столбцах поставлены крестики).

*Использование метода для получения минимальной КНФ.* Для получения Минимальной конъюнктивной нормальной формы (*МКНФ*), используя метод Квайна, вводятся следующие критерии:

- для минимизации берётся не *СДНФ*, а *СКНФ* функции;
- склеиваемые пары членов меняются на: $w\vee x$ или $w\vee\bar{x}$;
- правило операции поглощения выглядит следующим образом:

$$z\cdot(z\vee y)=z\vee z\cdot y=z\cdot(1\vee y)=z.$$

2. *Метод Квайна – Мак-Класки.* Метод Мак-Класки состоит из двух основных этапов:

*1)* Нахождение всех простых импликант логической функции, используя правило склеивания:
   *a)* $(X\&Y)\vee(X\&\bar{Y})\equiv X;$
   *b)* $(X\vee Y)\&(X\vee\bar{Y})\equiv X.$

*2)* Минимизации полученного множества простых импликант (задача нахождения оптимального покрытия).

*Метод Мак-Класки. Первый этап.* Разделить двоичные векторы области единиц логической функции на секции в соответствии с их индексами. Индекс двоичного вектора равен числу единиц, входящих в состав этого вектора. Составить таблицу интервалов, используя правило склеивания. Склеивать между собой только те двоичные векторы, которые отличаются друг от друга только в одной координате (ближайшие векторы). Склеивание происходит по этой координате. Ближайшие векторы могут находиться только в соседних секциях таблицы. В конце первого этапа получают все простые импликанты логической функции.

*Метод Мак-Класки. Второй этап.* В ходе второго этапа полученное множество простых импликант минимизируют, т.е. выбирают минимальное количество простых импликант, которое позволяет покрыть всю область единиц логической функции (типичная задача нахождения оптимального покрытия).

***Пример 2.3:***

Пусть задана логическая функция:

$$f(x_1, x_2, x_3 x_4) = \sum(0,2,5,7,8,14,15)_1.$$

Найти *МДНФ* методом Мак-Класки.

*Решение:* Выпишем двоичные векторы области единиц логической функции и найдем их индексы:

$$v_1(x_1, x_2, x_3 x_4) = \{(0000), (0010), (0101), (0110), (0111), (1000), (1001), (1110), (1111)\}.$$

*Первый этап* – нахождение всех простых импликант логической функции.

Разделим двоичные векторы на секции в соответствии с их индексами, получим:

| *Индекс* | *Интервал* |
|---|---|
| *0* | *0000* |
| *1* | *0010*<br>*1000* |
| *2* | *0101*<br>*0110*<br>*1001* |
| *3* | *0111*<br>*1110* |
| *4* | *1111* |

Склеиваем между собой ближайшие векторы соседних секций, пока это возможно:

| индекс | интервал | | интервал | | | интервал | |
|---|---|---|---|---|---|---|---|
| 0 | 0000 | 0-1 | 00-0 | A1 | 2-3-3-4 | -11- | A6 |
| 1 | 0010 | | -000 | A2 | | ~~-11-~~ | |
| | 1000 | 1-2 | 0-10 | A3 | | | |
| 2 | 0101 | | 100- | A4 | | | |
| | 0110 | 2-3 | 01-1 | A5 | | | |
| | 1001 | | 011- | | | | |
| 3 | 0111 | | -110 | | | | |
| | 1110 | 3-4 | -111 | | | | |
| 4 | 1111 | | 111- | | | | |

Все оставшиеся не склеенными интервалы образуют множество простых импликант логической функции. Простые импликанты обозначаем A1, A2, ...

Сокращенная *ДНФ* = дизъюнкция всех простых импликант;

Сокращенная *ДНФ* = $A_1 \vee A_2 \vee A_3 \vee A_4 \vee A_5 \vee A_6$.

*Второй этап* – минимизация полученного множества простых импликант логической функции. Все векторы области единиц должны быть покрыты простыми импликантами (в каждом столбце хотя бы один *«x»*), и импликант должно остаться минимальное количество – *«x»* – вектор входит в данный простой импликант:

| Impli-kant | 0000<br>0 | 0010<br>2 | 0101<br>5 | 0110<br>6 | 0111<br>7 | 1000<br>8 | 1001<br>9 | 1110<br>14 | 1111<br>15 |
|---|---|---|---|---|---|---|---|---|---|
| A1 00-0 | x | x | | | | | | | |
| ~~A2 -000~~ | ~~x~~ | | | | | ~~x~~ | | | |
| ~~A3 0-10~~ | | ~~x~~ | | ~~x~~ | | | | | |
| A4 100- | | | | | | x | x | | |
| A5 01-1 | | | x | | x | | | | |
| A6 -11- | | | | x | x | | | x | x |

Все векторы области единиц должны быть покрыты простыми импликантами (в каждом столбце хотя бы один *«x»*), и импликант должно остаться минимальное количество.

Оптимальное покрытие: $A_1, A_4, A_5, A_6$.

*МДНФ*: $A_1 \vee A_2 \vee A_3 \vee A_4 \vee A_5 \vee A_6 = \overline{X_1}\&\overline{X_2}\&\overline{X_4} \vee X_1\&\overline{X_2}\&\overline{X_3} \vee \overline{X_1}\&X_2\&X_4 \vee X_2\&X_3$.

3. *Метод Блейка – Порецкого.* Метод позволяет получать сокращенную *ДНФ* булевой функции $f$ из ее произвольной *ДНФ*. Базируется на применении формулы обобщенного склеивания: $A \cdot x + B \cdot \bar{x} = A \cdot x + B \cdot \bar{x} + A \cdot B$, справедливость которой легко доказать.

Действительно, $A \cdot x = A \cdot x + A \cdot B \cdot x$, $\quad B \cdot \bar{x} = B \cdot \bar{x} + A \cdot B \cdot \bar{x}$.

Следовательно, $A \cdot x + B \cdot \bar{x} = A \cdot x + A \cdot B \cdot x + B \cdot \bar{x} + A \cdot B \cdot \bar{x} = A \cdot x + B \cdot \bar{x} + A \cdot B \cdot$

В основу метода положено следующее утверждение: если в произвольной *ДНФ* булевой функции $f$ произвести все возможные обобщенные склеивания, а затем выполнить все поглощения, то в результате получится сокращенная *ДНФ* функции $f$.

*Рассмотрим пример:*

Пусть булева функция $f$ задана произвольной *ДНФ*:

$$f_{ДНФ} = \overline{x_1} \cdot \overline{x_2} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2.$$

Необходимо, используя метод Блейка – Порецкого, получить сокращенную *ДНФ* функции $f$. Проводим обобщенные склеивания. Легко видеть, что первый и второй элемент исходной *ДНФ* допускают обобщенное склеивание по переменной $x_1$.

В результате склеивания получим:

$$\overline{x_1} \cdot \overline{x_2} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} = \overline{x_1} \cdot \overline{x_2} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2}.$$

Первый и третий элемент исходной *ДНФ* допускают обобщенное склеивание как по переменной $x_1$, так и по $x_2$.

После склеивания по $x_1$ имеем:

$$x_1 \cdot x_2 + x_1 \cdot x_2 = x_1 \cdot x_2 + x_1 \cdot x_2 + x_2 \cdot x_2.$$

После склеивания по $x_2$ имеем:

$$\overline{x_1} \cdot \overline{x_2} + x_1 \cdot x_2 = \overline{x_1} \cdot \overline{x_2} + x_1 \cdot x_2 + \overline{x_1} \cdot x_1.$$

Второй и третий элементы *ДНФ* допускают обобщенное склеивание по переменной $x_2$.

После склеивания получаем:

$$x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 + x_1 \cdot x_3.$$

Выполнив последнее обобщенное склеивание, приходим к *ДНФ*:

$$f = \overline{x_1} \cdot \overline{x_2} + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 + x_1 \cdot \overline{x_3}.$$

После выполнения поглощений получаем:

$$f = \overline{x_1} \cdot \overline{x_2} + \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 + x_1 \cdot \overline{x_3}.$$

Попытки дальнейшего применения операции обобщенного склеивания и поглощения не дают результата.

Следовательно, получена сокращенная *ДНФ* функции $f$.

Далее задача поиска минимальной *ДНФ* решается с помощью импликантной матрицы точно так же, как в методе Квайна.

## 2.4. Теоретико-множественное представление логических функций. Визуально матричный метод минимизации

*Теоретико-множественные представления.* Теоретико-множественные представления базируются на понятиях: множество, элементы множества и отношения на множествах.

В множестве могут быть выделены подмножества. Из двух и более множеств или подмножеств можно, установив отношения между их элементами, сформировать новое множество, состоящее из элементов, качественно отличающихся от элементов исходных множеств.

При применении теоретико-множественных представлений для отображения сложных систем и процессов в них наиболее общими формальными характеристиками являются абстрактные знаковые формулы, с помощью которых удобно отображать многоуровневое строение систем.

***Пример 2.4:***

Система $S$ может быть отображена в совокупность множеств, описываемую теоретико-множественной формулой:

$$S = \langle I, C, A, P, E, U, X, W, Z \rangle,$$

где $I = \{i\}$ – совокупность (вектор) входных информационных сигналов;

$A = \{a\}$ – множество выходных действий (актуаций) системы;

$E = \{e\}$ – множество элементов, из которых составлена данная система;

$U = \{u\}$ – множество отношений между элементами $E$ системы;

$C = \{c\}$ – множество состояний (ситуаций) системы;

$X = \{x\}$ – множество признаков, характеризующих состояние элементов и отношений;

$W = \{w\}$ – множество свойств, т.е. $w$ – свойства свойств системы: вес, цена, значимость каждого свойства по отношению к другим свойствам;

$Z = \{z\}; P = \{p\}$ – множество возмущений $z$, действующих на систему.

Представление системы полной формулой не всегда возможно и целесообразно. Обычно системы описываются сокращенными формулами в зависимости от требований полноты описания.

При отображении системы осуществляется ее декомпозиция – выделение групп (множеств) элементов, обладающих одинаковыми (в рамках определенных ограничений) свойствами.

Выделив множества, можно производить соответствующие операции над ними, т.е., ставя их в определённые отношения друг с другом, перейти к композиции системы:

$$\{x\}\ R_1\ \{c\}\ R_2\ \{e\}\ R_j\ \{u\}\ R_k\ \{a\}\ R_n\ \{i\}.$$

Символом $R$ здесь обозначаются отношения между элементами или множествами в случае, если не определен характер этих отношений.

*Визуально-матричный метод минимизации.* Логическая функция может быть представлена на матричной форме своих характеристических множеств.

Матричная форма – таблица, каждая клетка которой соответствует одному из наборов таблицы истинности.

Левая верхняя клетка, которой соответствуют *а*, *в* (указаны выше) и *с* (указана слева), очевидно, отвечает *первой строке* таблицы с *номером 0*. Правая нижняя клетка карты, которой соответствуют переменные *а*, *в* и *с*, отвечает строке таблицы с *номером 1*, и т.д.

**Таблица 1 – Таблица истинности**

| | $a$ | $в$ | $c$ | $f$ | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **0** | $\bar{a}\cdot\bar{в}\cdot\bar{c}$ | $a\vee в\vee c$ |
| 1 | 0 | 0 | 1 | **0** | $\bar{a}\cdot в\cdot c$ | $\bar{a}\vee в\vee c$ |
| 2 | 0 | 1 | 0 | **0** | $\bar{a}\cdot в\cdot\bar{c}$ | $a\vee\bar{в}\vee c$ |
| 3 | 0 | 1 | 1 | **1** | $a\cdot в\cdot\bar{c}$ | $\bar{a}\vee\bar{в}\vee\bar{c}$ |
| 4 | 1 | 0 | 0 | **0** | $\bar{a}\cdot\bar{в}\cdot c$ | $a\vee в\vee\bar{c}$ |
| 5 | 1 | 0 | 1 | **1** | $a\cdot\bar{в}\cdot c$ | $\bar{a}\vee в\vee\bar{c}$ |
| 6 | 1 | 1 | 0 | **1** | $\bar{a}\cdot в\cdot c$ | $a\vee\bar{в}\vee\bar{c}$ |
| 7 | 1 | 1 | 1 | **1** | $a\cdot в\cdot c$ | $\bar{a}\vee\bar{в}\vee\bar{c}$ |

Карта Карно для функции, рассмотренной выше в *примере 2*, представлена в **Таблице 2**:

**Таблица 2 – Карта Карно**

| $a\,в$ / $c$ | | 00<br>$\bar{a}\cdot\bar{в}$ | 01<br>$\bar{a}\cdot в$ | 11<br>$a\cdot в$ | 10<br>$a\cdot\bar{в}$ |
|---|---|---|---|---|---|
| $\bar{c}$ | 0 | 0 | 0 | 1 | 0 |
| $c$ | 1 | 0 | 1 | 1 | 1 |

В клетки карты заносим *0* или *1* в соответствии с **Табл. 1**.

Далее необходимо в карте выделить один или несколько прямоугольников, включающих возможно большее число клеток с *«1»*. При этом прямоугольники могут содержать *2n* клеток, т.е. *1, 2, 4, 8* и т.д. Одна и та же клетка может входить в несколько прямоугольников. В нашем случае таких прямоугольников можно выделить три, каждому из них соответствует один член искомого уравнения.

Для вертикального прямоугольника можно записать:

$$a\cdot в\cdot c\vee a\cdot в\cdot\bar{c}=a\cdot в\cdot(c\vee c)=a\cdot в.$$

Для двух оставшихся получаем $в\cdot c$ и $a\cdot c$; т.е. ту переменную, которая повторяется дважды – один раз *«0»*, другой *«1»* – исключаем, а ту, которая не меняется, оставляем.

Сокращенная *ДНФ* функции:

$$y = a \cdot в \vee a \cdot c \vee в \cdot c.$$

## Задачи

1. Сколько всего страниц в книге, если сумма всех цифр номеров страниц равна 2395?
2. Что получится, если сложить: начало конца, конец начала, первое ноября, хвост кота, середину вторника, перед осы, начало леса и конец ночи?
3. В ящике лежат лимоны. Сначала из него взяли половину всех лимонов и половину лимона, затем половину остатка и ещё половину лимона. Наконец, ещё достали половину нового остатка и половину лимона. После этого в ящике остался 31 лимон. Сколько лимонов было в ящике вначале?
4. Один получил свое имя за свои габариты, другой – за способность давать информацию, третий – за свою дислокацию, а четвертый и вовсе предпочитает оставаться инкогнито. А как зовут пятого?
5. Ученик не заметил знака умножения между двумя трёхзначными числами и написал одно шестизначное число. Результат получился в три раза больше. Найдите эти числа.

## Вопросы и задачи

1. Приведите примеры предложений, которые не являются высказываниями.
2. Какие значения может принимать высказывание?
3. Чем отличаются первичные и вторичные высказывания?
4. Как может быть задана логическая функция?
5. Приведите таблицу истинности функции «конъюнкция $x_1$, $x_2$».
6. Как на основании таблицы истинности функции получить СДНФ? Постройте ее для следующей таблицы:

| $x_1$ | $x_2$ | $F$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

7. Как на основании таблицы истинности функции получить *СКНФ*? Постройте ее для такой таблицы:

| $x_1$ | $x_2$ | $F$ |
|---|---|---|
| *0* | *0* | *1* |
| *0* | *1* | *0* |
| *1* | *0* | *1* |
| *1* | *1* | *1* |

8. Дайте определение булевой алгебры логических функций.
9. Справедливы ли сочетательный, переместительный и распределительный законы для операций булевой алгебры?
10. Объясните, почему $x \wedge 0 = 0$, $x \wedge 1 = x$, $x \vee 0 = x$, $x \vee 1 = 1$?
11. Минимизируйте функцию: $F(x_1, x_2) = (\overline{x_1} \wedge x_2) \vee (x_1 \wedge x_2)$.
12. Какая система функций называется функционально полной?
13. Приведите пример функционально полных систем.
14. Дайте определение алгебры Жегалкина.
15. Дайте определение замыкания множества логических функций.

# ГЛАВА 3. ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ

## 3.1. Основные понятия логики высказываний. Тождественно истинные формулы логики высказываний и их формальный вывод

*Высказывание* – всякое повествовательное предложение, утверждающее что-либо о чём-либо, и при этом мы можем сказать, истинно оно или ложно в данных условиях места и времени.

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

Высказывания, которые получаются из элементарных с помощью грамматических связок «не», «и», «или», «если …, то …», «тогда и только тогда», принято называть *сложными*.

Считается, что каждое высказывание либо истинно, либо ложно и ни одно высказывание не может быть одновременно истинным и ложным.

Элементарные высказывания обозначаются малыми буквами латинского алфавита: *x, y, z, …, a, b, c, …*; истинное значение высказывания – буквой *и* или цифрой *1*, а ложное – буквой *л* или цифрой *0*.

*Определение формулы исчисления высказываний:*

1. Всякая переменная *x, y, z, …* является формулой.
2. Если *A* и *B* – формулы, то слова *(A & B), (A v B), (A ¬ в)*, $\bar{A}$ – также формулы.
3. Никакая другая строчка символов не является формулой.

Переменные высказывания будем называть *элементарными формулами*. Приведем примеры формул исчисления высказываний. Переменные высказывания *x, y, z* являются формулами, согласно *п. 1 определения* формулы.

Но тогда слова *(x & y)*, *(x v z), (y z)*, $\bar{x}$ являются формулами согласно *п. 2 определения*.

По этой же причине будут формулами слова:

$$\overline{(x \& y)},\ ((x \vee z)\&(y \to z)).$$

*Определение.* Выводом из конечной совокупности формул $H$ называется всякая конечная последовательность формул $B_1, B_2, \ldots, B_k$, всякий член которой удовлетворяет одному из следующих трех условий:

1) он является одной из формул совокупности $H$;
2) он является доказуемой формулой;
3) он получается по правилу заключения из двух любых предшествующих членов последовательности $B_1, B_2, \ldots, B_k$.

## 3.2. Основные понятия логики предикатов. Применение выражений логики предикатов для описания некоторых отношений

*Одноместным предикатом* $P(x)$ называется произвольная функция переменного $x$, определенная на множестве $M$ и принимающая значения из множества $\{1.0\}$. Множество $M$, на котором определен предикат $P(x)$, называется *областью определения предиката*. Множество всех элементов $x \in M$, при которых предикат принимает значение «истина», называется *множеством истинности предиката* $P(x)$, т.е. множество истинности предиката $P(x)$ – это множество $I_P = \{x: x \in M, P(x) = 1\}$.

*Двухместным предикатом* $P(x, y)$ называется функция двух переменных $x$ и $y$, определенная на множестве $M = M_1 \times M_2$ и принимающая значения из множества $\{1,0\}$. Аналогично определяется $n$-местный предикат.

*Определение 1.* Конъюнкцией двух предикатов $P(x)$ и $Q(x)$ называется новый предикат $P(x)\&Q(x)$, который принимает значение «истина» при тех и только тех значениях $x \in M$, при которых каждый из предикатов принимает значение «истина» и принимает значение «ложь» во всех остальных случаях. Очевидно, что областью истинности предиката $P(x)\&Q(x)$ является общая часть областей истинности предикатов $P(x)$ и $Q(x)$, т.е. пересечение $I_P \cap I_Q$.

*Определение 2.* Дизъюнкцией двух предикатов $P(x)$ и $Q(X)$ называется новый предикат $P(x) \vee Q(x)$, который принимает значение «ложь» при тех и только тех значениях $x \in M$, при которых каждый из предикатов принимает значение «ложь» и принимает значение «истина» во всех остальных случаях.

*Определение 3.* Отрицанием предиката $P(x)$ называется новый предикат $P(x)$, который принимает значение «истина» при всех значениях $x \in M$, при которых предикат $P(x)$ принимает значение «ложь», и принимает значение «ложь» при тех значениях $x \in M$, при которых предикат $P(x)$ принимает значение «истина». Из этого определения следует, что $I_{\bar{P}} = M \setminus I_P = CI_P$ .

*Определение 4.* Импликацией предикатов $P(x)$ и $Q(x)$ называется новый предикат $P(x) \to Q(x)$. который является ложным при тех и только тех значениях $x \in M$, при которых одновременно $P(x)$ принимает значение «истина», а $Q(x)$ – значение «ложь» и принимает значение «истина» во всех остальных случаях. Т.к. при каждом фиксированном $x \in M$ справедлива равносильность $P(x) \to Q(x) \equiv \bar{P}(x) \vee Q(x)$ , то $I_{P \to Q} = CI_P \cup I_Q$.

***Пример 3.1:***

Среди следующих предложений выделить предикаты и для каждого из них указать область истинности, если $M = R$ для одноместных предикатов и $M = R \times R$ для двухместных предикатов:

1) $x + 5 = 1$;
2) при $x = 2$ выполняется равенство $x^2 - 1 = 0$;
3) $x^2 - 2x + 1 = 0$;
4) существует такое число $x$, что $x^2 - 2x + 1 = 0$;
5) $x + 2 < 3x - 4$;
6) однозначное число $x$ кратно $3$;
7) $(x + 2)\ (3x - 4)$;
8) $x^2 + y^2 > 0$.

***Решение:***

1. Предложение является одноместным предикатом $P(x)$, $I_p = \{4\}$.
2. Предложение не является предикатом. Это ложное высказывание.
3. Предложение является одноместным предикатом $P(x)$, $I_p = \{1\}$.
4. Предложение не является предикатом. Это истинное высказывание.
5. Предложение является одноместным предикатом $P(x)$, $I_p = (3; +\infty)$.

6. Предложение является одноместным предикатом *P(x)*, $I_p = \{0; 3; 6; 9\}$.
7. Предложение не является предикатом.
8. Предложение является двухместным предикатом *Q(x, y)*, $I_Q = R \times R \setminus \{(0, 0)\}$.

***Пример 3.2:***

Дана формула $\forall x(P(x)\&Q(x) \to R(x))$, где предикаты *P(x), Q(x)* и *R(x)* определены на множестве *N*. Найти ее значение, если:

1. *P(x):* «число *x* делится на *3*», *Q(x):* «число *x* делится на *4*», *R(x):* «число *x* делится на *2*».
2. *P(x):* «число *x* делится на *3*», *Q(x):* «число *x* делится на *4*», *R(x):* «число *x*: делится на *5*».

***Решение.*** В обоих случаях конъюнкция *P(x)&Q(x)* есть утверждение, что число *x* делится на *12*. Но тогда при всех *x*, если число *x* делится на *12*, то оно делится и на *2*; значит, в <u>*случае 1 (примере 3.2)*</u> формула истинна. Т.к. из делимости числа *x* на *12* не при всех *x* следует делимость числа *x* на *5*, то в <u>*случае 2 (примере 3.2)*</u> формула ложна.

**Задачи**

1. Пусть даны предикаты: *P(x)*: «*x* – четное число» и *Q(x)*: «*x* кратно *3*», определенные на множестве *N*. Найти области истинности предикатов:
   1) $P(x)\ \&\ Q(x)$;
   2) $P(x) \vee Q(x)$;
   3) $\overline{P}(x)$;
   4) $P(x) \to Q(x)$.
2. Даны предикаты:

$$P(x): x^2 + x + 1 > 0 \text{ и } Q(x): x^2 - 4x + 3 = 0,$$

определенные на множестве *R*. Требуется установить, какие из следующих высказываний истинны и какие ложны:

1) $\forall x P(x)$;
2) $\exists x P(x)$;
3) $\forall x Q(x)$;
4) $\exists x Q(x)$.

3. Пусть предикат $Q(x, y)$: «$x \vdots y$» определен на множестве $N \times N$. Показать, что высказывания $\forall y \exists x Q(x,y)$ и $\exists x \forall y Q(x,y)$ имеют различные логические значения.

4. Вычислить значение формулы: $\forall x \exists y P(x,y) \to \exists x \forall y P(x,y)$, если предикат $P(x, y)$ имеет значение $P(x, y)$ – «число $x$ меньше числа $y$» – и определен на множестве $M = N \times N$.

5. Найти отрицание следующих формул:
   *1)* $\forall x(P(x) \& Q(x))$;
   *2)* $\exists x(P(x) \vee Q(x))$;
   *3)* $\forall x \exists y(R(x,y) \to L(x,y))$.

6. Пусть $A = x_1 \& \overline{x_2} \to x_3$ и наборы значений переменных *(1, 0, 1)* и *(1, 0, 0)*. Тогда $\{x_1, \overline{x_2}, x_3\} \vdash A$, а $\{x_1, \overline{x_2}, \overline{x_3}\} \vdash \overline{A}$. Записать выводы в обоих случаях.

**Вопросы**

1. Дайте определение предиката.
2. Что представляет собой предметная область предиката и какие значения может принимать предикат?
3. В чем отличие предиката от булевой функции?
4. Каков смысл кванторов общности и существования?
5. Как определяется истинность предиката?
6. Какая формула логики предикатов называется выполнимой? противоречивой?
7. Какие формулы называются эквивалентными?
8. В чем состоит сложность определения выполнимости формулы?

# ГЛАВА 4. ЭЛЕМЕНТЫ ТЕОРИИ ГРАФОВ

## 4.1. Основные понятия и определения. Матричное представление. Маршруты, цепи, циклы, разрезы

***Основные понятия теории графов***

*Граф* – это система, которая интуитивно может быть рассмотрена как множество кружков и множество соединяющих их линий (геометрический способ задания графа – **Рис. 1**).

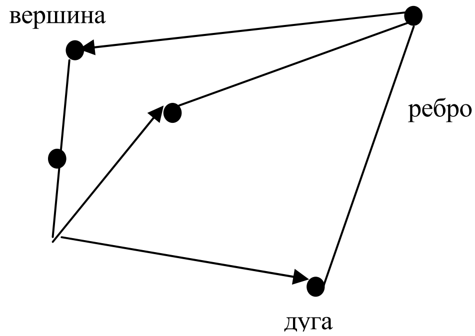



**Рис. 1 – Геометрический способ задания графа**

Кружки называются вершинами графа, линии со стрелками – дугами, без стрелок – рёбрами. Граф, в котором направление линий не выделяется (все линии являются ребрами), называется *неориентированным* (**Рис. 2**); граф, в котором направление линий принципиально (линии являются дугами) называется *ориентированным* (**Рис. 3**).

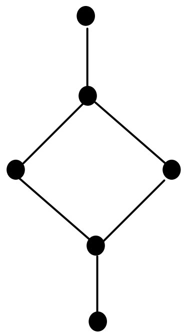


**Рис. 2 – Неориентированный граф**

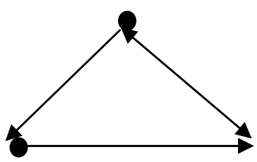


**Рис. 3 – Ориентированный граф**

Теория графов может рассматриваться как раздел дискретной математики (точнее, теории множеств), и тогда определение графа следующее.

***Граф*** **–** это конечное множество *X*, состоящее из *n* элементов $X = \{1,\ 2, \ldots\ n\}$, называемых вершинами графа, и подмножество *V* декартова произведения $X \times X$, называемое множеством дуг.

*Ориентированным графом G (орграфом)* называется совокупность *(X, V)*. *Неориентированным* графом называется совокупность множеств *X* и множества неупорядоченных пар элементов, каждый из которых принадлежит множеству *X*. Дугу между вершинами *i* и *j*, где $i, j \in N$, будем обозначать *(i, j)*. Число дуг графа будем обозначать $m(V = (v_1, v_2, \ldots v_m))$..

*Подграфом* называется часть графа, образованная подмножеством вершин вместе со всеми рёбрами (дугами), соединяющими вершины из этого множества. Если в графе удалить часть рёбер (дуг), то получим частичный граф (**Рис. 4**).

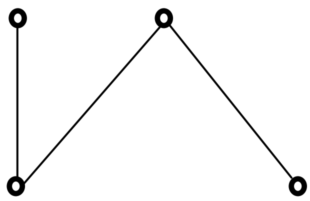


**Рис. 4 – Подграф**

Две вершины называются *смежными*, если они соединены ребром (дугой). Граф называется *полным*, если каждые две вершины его соединены одним и только одним ребром.

Граф, для которого из $(i,\ j) \in V$ следует $(i,\ j) \in V$, называется *симметричным.* Если из $(i,\ j) \in V$ следует $(i,\ j) \notin V$, то соответствующий граф называется *антисимметричным.*

Вершина степени *0*, либо в точности одна вершина степени $n - 1$.

***Матричный способ задания графов***

Квадратная матрица, элементами которой являются нули и единицы, а также некоторое число *m*, называется матрицей смежности графа $G(V,E)$ тогда и только тогда, когда её элементы образуются по следующему правилу:

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}$$

Элемент $a_{ij}$, стоящий на пересечении $v_i$-й строки и $v_j$-го столбца, равен единице, если имеется ребро, идущее из вершины $v_i$ в вершину $v_j$, и $a_{ij}$ равен нулю в противном случае. Элемент $a_{ij}$ равен единице, если при вершине $v_i$ имеется петля, и равен нулю в противном случае. Элемент $a_{ij}$ равен некоторому числу *m*, где *m* – число рёбер графа, идущее из вершины $v_i$ в вершину $v_j$.

Т.о., если граф $G(V, E)$ задан одним из указанных способов: аналитическим, геометрическим или матричным, всегда можно перейти к любому другому способу задания. Наиболее часто для задания графа используется аналитический и матричный способы, а геометрический способ служит для иллюстрации полученных результатов.

***Маршруты, цепи, циклы, разрез***

*Маршрутом в графе* называется чередующаяся последовательность вершин и рёбер, в которой любых два соседних элемента инцидентны: $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$.

Если $v_0 = v_k$, то маршрут замкнут; в противном случае открыт.

*Путём называется* последовательность дуг (в ориентированном графе) – такая, что конец одной дуги является началом другой дуги.

*Простой путь* – путь, в котором ни одна дуга не встречается дважды.

*Контур* – путь, у которого конечная вершина совпадает с начальной вершиной.

*Длиной пути* (контура) называется число дуг пути (или сумма длин его дуг, если последние заданы).

*Цепью* называется множество рёбер (в неориентированном графе), которые можно расположить так, что конец (в этом расположении) одного ребра является началом другого.

Другое определение: *цепь* – последовательность смежных вершин. Замкнутая цепь называется *циклом*.

Можно определить простые и элементарные цепи. Элементарная цепь (цикл, путь, контур), проходящая через все вершины графа, называется *гамильтоновой цепью.*

Простая цепь (цикл, путь, контур), содержащая все рёбра (дуги) графа называется *цепью Эйлера*.

Если любые две вершины графа можно соединить цепью, то граф называется ***связным***.

Если граф не является связным, то его можно разбить на связные подграфы, называемые *компонентами.*

*Связностью графа* называется минимальное число рёбер, после удаления которых граф становится несвязным.

***Теорема 1.*** Рассмотрим граф $G = (V, E)$: неориентированный, связный $(p = 1)$ и *немультиграф* $(s = 1)$. Тогда количество фундаментальных разрезов в точности равно $\rho(G) = n-1$.

*Доказательство.* В доказательстве может помочь описание алгоритма построения фундаментальной системы разрезов, которое дано ниже.

*Алгоритм построения фундаментальной системы разрезов (для графа такого, как в условии теоремы):*

1. Построим любое остовное дерево $T$ исходного графа $G$.
2. Удалим некоторое ребро $e$ из остовного дерева. Понятно, что остов распадется на две компоненты связности. Вершины первой компоненты обозначим $V_1$, а второй – $V_2$.
3. Этому ребру тогда будет соответствовать разрез $S_e = (V_1, V_2)$.
4. Все разрезы $\{S_e\}$ (соответствующие всем ребрам, взятым из остова) образуют фундаментальную систему разрезов.

## 4.2. Обходы в графе. Кратчайшие пути. Эйлеровы и гамильтоновы циклы. Раскраска и планарность

### ***Эйлеровы графы***

Замкнутая цепь в графе $G$ называется эйлеровой цепью, если она содержит все ребра и все вершины графа.

Граф, содержащий эйлерову цепь, будет называться *эйлеровым графом.*

Иными словами, эйлеров граф – это *связный граф*, в котором имеется замкнутая цепь, проходящая точно один раз через каждое его ребро.

Эйлер предложил рассмотреть граф, изображенный на **Рис. 5**.

***Теорема 1*** *(Эйлер, 1736).* Для неодноэлементного связного графа *G* следующие условия эквивалентны:

1) *G* – Эйлеров граф;
2) каждая вершина графа *G* имеет четную степень;
3) множество всех ребер графа *G* можно разбить на циклы.

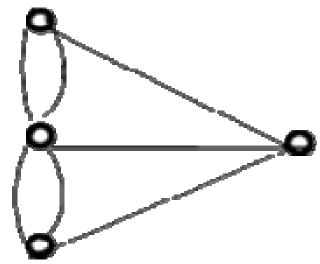


**Рис. 5 – Граф, предложенный Эйлером**

*Следствие:* пусть *G* – произвольный граф, содержащий $2l$ вершин нечетной степени, где $l >= 1$. Тогда множество всех ребер графа можно разбить на *l* цепей, каждая из которых соединяет две вершины нечетной степени.

*Утверждение,* вытекающее из предыдущей теоремы и следствия: связный граф *G* является полуэйлеровым графом тогда и только тогда, когда *G* содержит не более двух вершин нечетной степени.

*Следствие:* пусть связный граф *G* содержит две вершины нечетной степени *u* и *v*. Тогда существует *(u, v)*-цепь, содержащая все ребра графа *G*. Граф называется произвольно вычерчиваемым из вершины *v*, если любая его цепь с началом в вершине *v* может быть продолжена до эйлеровой цепи графа *G*. Разумеется, если графа произвольно вычерчиваем из вершины *v*, то он является эйлеровым графом.

***Теорема 2.*** Неодноэлементный эйлеров граф *G* является произвольно вычерчиваемым из вершины *v* тогда и только тогда, когда вершина *v* принадлежит любому циклу графа *G*.

Опираясь на предыдущую теорему, можно описать строение всех графов, произвольно вычерчиваемых из вершин *v*.

Для этого возьмем произвольный лес *H*, т.е. ациклический граф, не содержащий вершину *v*. Каждую вершину нечетной степени из *H* соединим некоторым нечетным числом кратных ребер с *v*, а каждую вершину четной степени – четным числом (не исключая *0*) кратных ребер с *v*, причем каждую изолированную вершину из *H* обязательно соединим с *v* (см. **Рис. 6**).

В вершине $v$ можно соединить несколько петель. Полученный граф $G$:

1) связен;
2) имеет только вершины четной степени;
3) является произвольно вычерчиваемым из вершины $v$ (как эйлеров граф, у которого вершина $v$ принадлежит всем циклам).

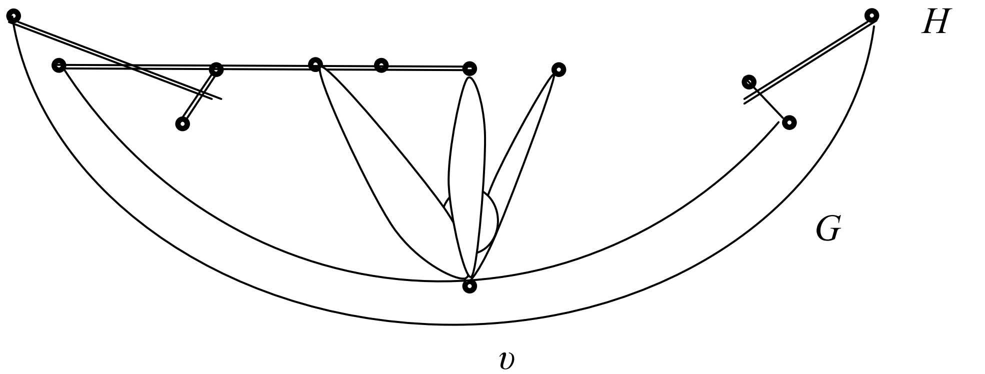



**Рис. 6 – Полученный граф**

***Гамильтоновы графы***

Гамильтоновой цепью графа называется его простая цепь, которая проходит через каждую вершину графа точно один раз. Цикл графа, проходящий через каждую его вершину, называется гамильтоновым циклом. Граф называется гамильтоновым, если он обладает гамильтоновым циклом.

Пусть граф $G$ имеет $n$ вершин $v_1, v_2, \dots, v_n$. Положим, $d_i = deg\ v_i$ $(i = 1, 2, \dots, n)$, и будем считать, что вершины графа упорядочены таким образом, что $d_1 \le d_2 \le \dots \le d_n$.

Последовательность $d_1, d_2, \dots, d_n$ называют последовательностью степеней графа $G$. Будем говорить, что числовая последовательность $d_1 \le d_2 \le \dots \le d_n$ мажорируется числовой последовательностью $d_1' \le d_2' \le \dots \le d_n'$, если $d_i \le d_i''$ для любого $i = 1, 2, \dots, n$.

***Теорема Хватала*** (1972). Пусть $G$ – обыкновенный граф, $d_1 \le d_2 \le \dots \le d_n$ – его последовательность степеней и $n \ge 3$. Если для любого $k$ верна импликация $d_k \le k < n/2 \rightarrow d_{n-k} \ge n-k$, то граф $G$ гамильтонов.

Пусть $G$ – обыкновенный граф, $d_1 \le d_2 \le \dots \le d_n$ – его последовательность степеней и $n \ge 3$.

Граф G гамильтонов, если выполнено одно из условий:

1) $d_k \geq n/2$ для любого $k = 1, \ldots, n$ (***Теорема Дирака***);
2) $deg\ u + deg\ v \geq n$ для любых двух различных несмежных вершин $u, v$ графа $G$ (***Теорема Оре, 1960***);
3) $d_k > k$ для любого натурального числа $k$ такого, что $1 \leq k < n/2$.

*Кратчайшие пути.* Задача о кратчайшем пути, т.е. о нахождении пути минимальной длины в ориентированном графе, имеет несколько вариантов:

1) найти кратчайший путь из заданной вершины $s$ в заданную вершину $t$;
2) найти кратчайшие пути из вершины $s$ во все остальные вершины графа;
3) найти кратчайшие пути между всеми упорядоченными парами вершин.

*Алгоритм Дейкстры:*

1. Положить $d(s) = 0$ и $d(x) = \infty$ для всех вершин $x \neq s$. Пометить $s$ и положить $y = s$.
2. Для всех непомеченных вершин $x$, смежных с $y$, пересчитать $d(x)$ по формуле: $d(x) = min\ \{d(x),\ d(y) + l(y,\ x)\}$. Если $d(x) = \infty$ для всех непомеченных вершин, закончить алгоритм: путей в эти вершины нет.
3. Пометить ту из вершин $x$, для которой $d(x)$ минимально. Пометить ребро, ведущее в $x$, которое дало оценку $d(x)$. Положить $y = x$.
4. а) если $y = t$, закончить алгоритм. Иначе перейти к *шагу 2*;
   б) если непомеченных вершин нет, закончить алгоритм.
   Иначе перейти к *шагу 2*.

Основное свойство величин $d(x)$, порождаемых алгоритмом Дейстры, описывается следующим утверждением:

Пусть $x_i$ – вершина $T_k$, помеченная в $i$-м цикле. Тогда:

1) $d_z(x_{i-1}) \leq d_z(x_i)$ для всех $i \leq k$;
2) для любой вершины $x$, непомеченной после $k$-го цикла, $d_z(x_k) \leq d(x)$.

***Теорема.*** Для любого $k < n$ $T_k$ является деревом кратчайших путей, т.е. путь в $T_k$, ведущий из $s$ в любую его вершину $x$, является кратчайшим путем из $s$ в $x$ в исходном графе $G$, $d_z(x)$ – его вершина.

*Раскраска элементов графа* в $k$ цветов, или *k-раскраска* – это разбиение элементов графа на $k$ классов. Рассматривают раскраски вершин и ребер неориентированных графов, а также раскраски граней плоских карт. Правильная раскраска – раскраска, где любая пара смежных элементов окрашена в разные цвета. Задача раскраски – найти минимальное число цветов, достаточное для правильной раскраски.

Граф называется *k-раскрашиваемым*, если существует его правильная вершинная *k-раскраска*.

Хроматическим числом $\chi(G)$ графа $G$ называется минимальное число $k$, для которого граф $G$ *k-раскрашиваемый*.

Граф $G$ называется *k-хроматическим*, если $\chi(G) = k$, и бихроматическим, если $\chi(G) = 2$.

Хроматический граф изображен на **Рис. 7**:

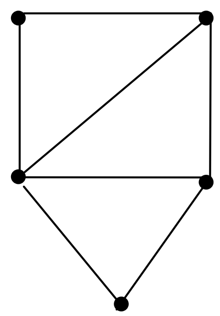


**Рис. 7 – Хроматический граф**

***Теорема.*** Для любого простого графа $\chi(G) \leq \Delta + 1$. Это означает, что любой граф можно раскрасить в $\Delta + 1$ цветов. Метод заключается в следующем: дано множество $\Delta + 1$ цветов. Первой вершине присвоим произвольный цвет из этого множества, в дальнейшем каждой выбираемой вершине $v$ присваиваем цвет, который не присвоен смежным с ней вершинам. Это всегда возможно, т.к. $d(v) \leq \Delta$.

***Теорема Брукса.*** Для графов и циклов нечетной длины $\chi(G) = \Delta + 1$. Для всех остальных графов $\chi(G) \leq \Delta$.

*Плоский граф* – граф, вершины которого являются точками плоскости, а ребра – непрерывными плоскими линиями без самопересечений, соединяющими соответствующие вершины так, никакие два ребра не имеют общей точки, кроме инцидентной им обоим вершины.

Примеры таких графов можно увидеть на рисунке ниже:

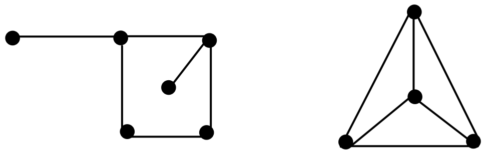
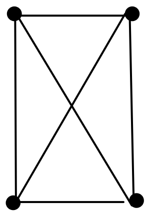


**Рис. 8 – Примеры плоского графа**

Любой граф, изоморфный плоскому графу, будем называть планарным. Граф, изображенный на рисунке ниже, является планарным.



*Очевидны следующие утверждения:*

1) всякий подграф планарного графа планарен;
2) граф планарен тогда и только тогда, когда каждая его связная компонента – планарный граф.

**Рис. 9 – Пример планарного графа**

О планарных графах говорят, что они укладываются на плоскости (имеют плоскую укладку).

*Жордановая кривая* – непрерывная спрямляемая линия, не имеющая самопересечений.

Граф $G$ укладывается в пространство $L$, если существует такое отображение вершин и ребер графа $G$ соответственно в точки и жордановы кривые этого пространства, что различным вершинам соответствуют различные точки, а кривые, соответствующие различным ребрам, пересекаются только в инцидентных этим ребрам вершинам.

***Пример 4.1:***

*Задача о трех домах и трех колодцах.* Имеются три дома 1, 2, 3 и три колодца 4, 5, 6. Каждый хозяин пользуется любым из трех колодцев. В некоторый момент обитатели домов решили проложить дорожки к колодцам так, чтобы исключить встречи на дорожках, т.е. чтобы дорожки не пересекались.

Возможно ли это? Возможно ли построить плоскую укладку графа?

Все попытки нарисовать девять непересекающихся дорожек, соединяющих дома с колодцами, заканчиваются неудачей. На самом деле не только существуют непланарные графы, но верно утверждение, приводимое ниже без доказательства.

*Утверждение:* почти все графы не являются планарными.

## 4.3. Деревья. Построение остовных деревьев. Алгоритмы анализа графов – поиск в глубину и в ширину

*Дерево* – связный (ориентированный или неориентированный) граф, не содержащий циклов (для любой вершины есть один и только один способ добраться до любой другой вершины).

Древовидная структура – тип организации, в котором каждый объект связан с хотя бы одним другим. Дерево и получило название «дерево», поскольку, будучи нарисованным, выглядит как дерево, только перевёрнутое «вверх ногами».

Граф, изображённый на **Рис. 10**, является примером дерева:

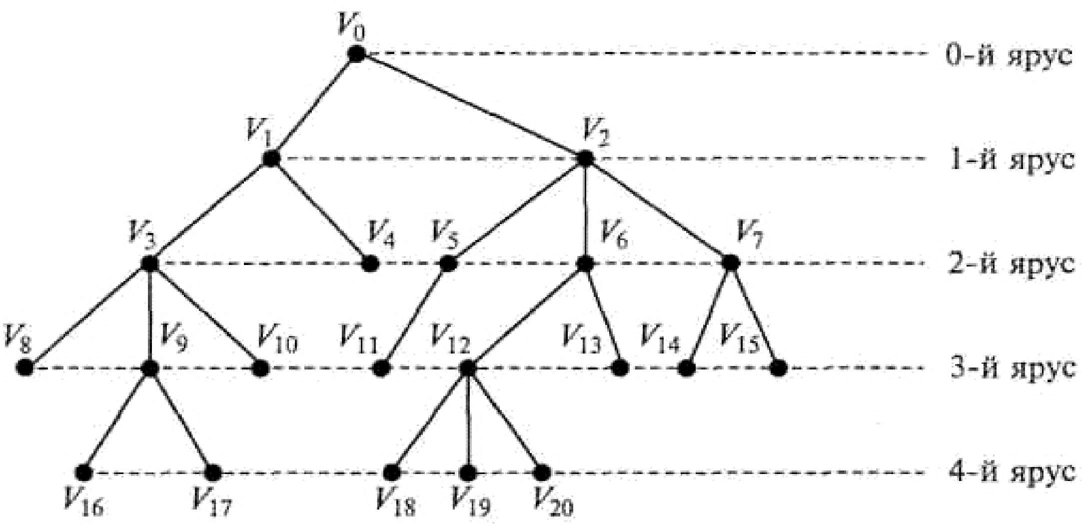



**Рис. 10 – Пример дерева**

Для каждой пары вершин дерева – *узлов* – существует единственный маршрут, поэтому вершины удобно классифицировать по степени удалённости от корневой вершины. Расстояние до корневой вершины $V_0$ называется *ярусом* $s$ вершины, $s=d(V_0V)$.

***Теорема.*** Граф $G(V, X)$ $(|V| = n > 1)$ является деревом тогда и только тогда, когда выполняется хотя бы одно из условий:

– граф $G(V, X)$ связан и не содержит циклов;
– граф $G(V, X)$ не содержит циклов и имеет $n - 1$ ребро;
– граф $G(V, X)$ связен и имеет $n - 1$ ребро;
– граф $G(V, X)$ не содержит циклов, но добавление ребра между несмежными вершинами приводит к появлению одного и только одного элементарного цикла;
– граф $G(V, X)$ связный, но утрачивает это свойство после удаления любого ребра;
– в графе $G(V, X)$ всякая пара вершин соединена цепью, и только одной.

Дерево называется *деревом с корнем*, если одна вершина выделена и расположена выше остальных.

Вершины, расположенные под одной вершиной, называются ее *сыновьями*, а сама вершина *отцом*.

Вершины, не имеющие сыновей, называются *листьями*.

Вершины, отличные от корня и листьев, называют *внутренними*.

Дерево, корнем которого является одна из вершин данного дерева, называется *поддеревом*.

*Ориентированное (направленное) дерево* – ацикличный орграф (ориентированный граф, не содержащий циклов), в котором только одна вершина имеет нулевую степень захода (в неё не ведут дуги), а все остальные вершины имеют *степень захода 1* (в них ведётся ровно по одной дуге).

Вершина с нулевой степенью захода называется *корнем дерева*, вершины с нулевой степенью исхода (из которых не исходит ни одна дуга) называются *концевыми вершинами* или *листьями*.

*Неориентированное дерево (*или просто *дерево)* – это конечный связный граф с выделенной вершиной (корнем) без циклов. Дерево не имеет петель и кратных рёбер.

*Лес* – это граф, компоненты которого являются деревьями. Деревья используются для описания структур организаций, предприятий и др. Такие структуры называются иерархическими.

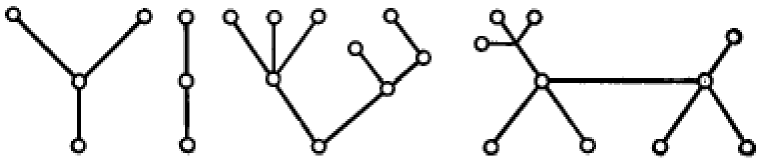


**Рис. 11 – Лес**

*Остовное дерево* – ациклический связный подграф данного связного неориентированного графа, в который входят все его вершины.

*Свойства* остовного дерева:

- Любое остовное дерево в графе с $n$ вершинами содержит ровно $n-1$ ребро.
- Число остовных деревьев в полном графе на $n$ вершинах выражается знаменитой формулой Кэли: $n^{n-2}$.
- В общем случае число остовных деревьев в произвольном графе может быть вычислено при помощи т.н. *матричной теоремы о деревьях*.

### *Алгоритмы анализа графов – поиск в глубину и поиск в ширину*

*Поиск в глубину (DFS)* – один из методов обхода графа.

Стратегия поиска в глубину, как и следует из названия, состоит в том, чтобы «идти вглубь» графа, насколько это возможно.

Алгоритм поиска описывается рекурсивно: перебираем все исходящие из рассматриваемой вершины рёбра. Если ребро ведёт в вершину, которая не была рассмотрена ранее, то запускаем алгоритм от этой нерассмотренной вершины, а после возвращаемся и продолжаем перебирать рёбра. Возврат происходит в том случае, если в рассматриваемой вершине не осталось рёбер, которые ведут в нерассмотренную вершину. Если после завершения алгоритма не все вершины были рассмотрены, то необходимо запустить алгоритм от одной из нерассмотренных вершин.

*Поиск в ширину (BFS)* – метод обхода графа и поиска пути в графе. Поиск в ширину является одним из неинформированных алгоритмов поиска.

*Алгоритм поиска в ширину.* Поиск в ширину работает путём последовательного просмотра отдельных уровней графа, начиная с узла-источника $u$. Рассмотрим все рёбра $(u,v)$, выходящие из узла $u$. Если очередной узел $v$ является целевым узлом, то поиск завершается – в противном случае узел $v$ добавляется в очередь.

После того как будут проверены все рёбра, выходящие из узла $u$, из очереди извлекается следующий узел $u$; процесс повторяется.

## 4.4. Независимые и доминирующие множества. Паросочетания и покрытия. Алгоритмы их нахождения

*Независимое множество вершин (внутренне устойчивое множество)* – это множество вершин графа $G$ – такое, что любые две вершины в нем не смежны, т.е. никакая пара вершин не соединена ребром. Множество $S \subset V$, удовлетворяющее условию $S \cap E(S) = \varnothing$ *(1)*, называется *независимым множеством вершин*.

Для графа, изображенного на **Рис. 12**, независимыми являются $\{x_7, x_8, x_2\}$, $\{x_3, x_1\}$ множества вершин.

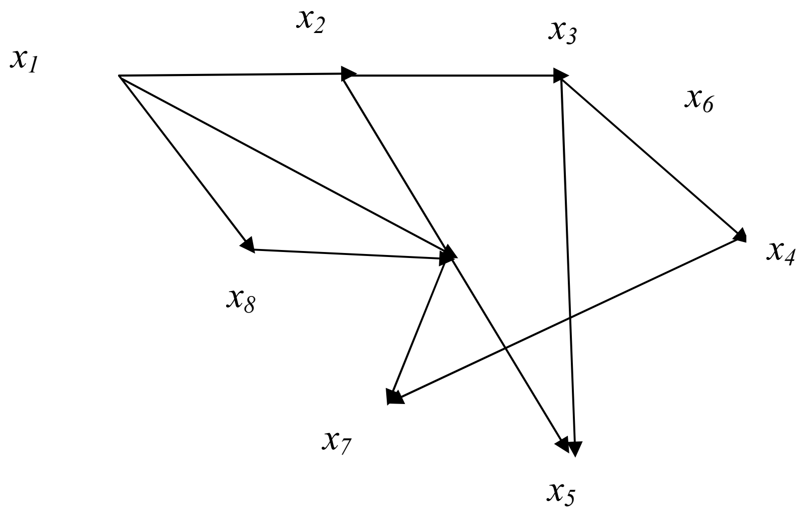



**Рис. 12 – Независимое множество вершин**

Независимое множество называется *максимальным*, когда нет другого независимого множества, в которое оно бы входило.

Для графа $G = (V, E)$ *доминирующее множество вершин* (называемое также *внешне устойчивым множеством*) есть множество вершин $S \subseteq E$, выбранное так, что для каждой вершины $x_j$, не входящей в $S$, существует дуга, идущая из некоторой вершины множества $S$ в вершину $x_j$.

Доминирующее множество называется *минимальным,* если нет другого доминирующего множества, содержащегося в нем.

Если $P$ – семейство всех минимальных доминирующих множеств графа, то число $\beta[G] = \min S \in P\,|S|$ называется *числом доминирования* графа $G$, а множество $S^*$, на котором достигается минимум, называется *наименьшим* доминирующим множеством.

Доминирующее множество называется *минимальным*, если его подмножество не является доминирующим.

*Паросочетанием в графе называется* множество ребер, попарно не имеющих общих вершин.

***Теорема 1.*** Для любого графа $G$ с $n$ вершинами, не имеющего изолированных вершин, справедливо равенство: $\pi(G) + p(G) = n$.

*Доказательство.* Пусть $M$ – наибольшее паросочетание в графе $G$. Обозначим через $W$ множество всех вершин графа, не покрытых ребрами этого паросочетания. Тогда $|W| = n - 2\pi(G)$. Очевидно, что $W$ – независимое множество (иначе $M$ не было бы наибольшим). Выберем для каждой вершины из $W$ какое-нибудь инцидентное ей ребро. Пусть $F$ – множество всех выбранных ребер.

Тогда $M\ U F$ – реберное покрытие и $M□F = n - \pi(G)$, следовательно, $p(G) \geq n - \pi(G)$.

***Алгоритмы нахождения***

1. *Волновой алгоритм* – алгоритм, позволяющий найти минимальный путь в графе с рёбрами единичной длины. Основан на алгоритме поиска в ширину. Применяется для нахождения кратчайшего пути в графе, в общем случае находит лишь его длину.
2. *Алгоритм Дейкстры.* Алгори́тм Де́йкстры *(Dijkstra's algorithm)* – алгоритм находит кратчайшее расстояние от

одной из вершин графа до всех остальных (если таковые имеются). Алгоритм работает только для графов без рёбер отрицательного веса. Алгоритм широко применяется в программировании и технологиях, его использует, например, протокол *OSPF* для устранения кольцевых маршрутов.

3. *Алгоритм Беллмана – Форда.* Алгоритм Беллмана – Форда применяется для нахождения кратчайшего расстояния от вершины *[s]* до остальных вершин.
4. *Алгоритм Флойда – Уоршелла*. Алгоритм Флойда – Уоршелла – динамический алгоритм для нахождения кратчайших расстояний между всеми вершинами взвешенного ориентированного графа. Процедура находит пути минимального веса в графе $G = (V,E)$, заданном весовой матрицей $W$, у которой элемент $W [i, j]$ равен весу ребра, соединяющего $i$-ю и $j$-ю вершины. При этом полагаем, что $W [i, i] = 0$ для всех $i$. Путь ищется для всех пар вершин графа. Для бесконечности используется число *GM* – его можно задавать в зависимости от конкретной задачи.
5. *Алгоритм Йена.* Работа алгоритма начинается с нахождения кратчайшего пути, для этого будем использовать уже описанный алгоритм Дейкстры. Второй путь ищем, перебирая кратчайшие отклонения от первого, третий – кратчайшие отклонения от второго и т.д. Алгоритм использует *массив p* для результирующего списка путей, и *массив l* для хранения соответствующих длин, при этом, если, начиная с некоторого $i$, элементы $l$ [$i$] равны $1$, значит, существует только $i–1$ кратчайших путей без петель.

## Задачи

1. В классе 30 человек, все они состоят в дружеских отношениях с одноклассниками. Может ли быть так, что у 9 из них имеется по 3 друзей (в этом классе), у 11 – по 4 друга, а у 10 – по 5 друзей?
2. В некоторой стране любые два города соединены либо авиалинией, либо железной дорогой.
   Докажите, что:

a) можно выбрать вид транспорта так, чтобы от любого города можно было добраться до любого другого, пользуясь только этим видом транспорта;
b) из некоторого города, выбрав один из видов транспорта, можно добраться до любого другого города не более чем с одной пересадкой (пользоваться можно только выбранным видом транспорта);
c) каждый город обладает свойством, указанным в *п.* b);
d) можно выбрать вид транспорта так, чтобы, пользуясь только им, можно было добраться из любого города в любой другой не более чем с двумя пересадками.

3. В стране 1993 города, и из каждого выходит не менее 93 дорог. Известно, что из любого города можно проехать по дороге в любой другой. Докажите, что это можно сделать не более чем с 62 пересадками. Дорога соединяет между собой два города.
4. Волейбольная сетка – прямоугольник 50×600 клеток. Какое наибольшее число верёвочек можно перерезать, чтобы сетка не распалась?
5. Написано 1997-значное число. Каждое двузначное число, образованное соседними цифрами, делится на 17 либо на 23? Последняя цифра числа – 1. Какова первая?
6. Докажите, что можно соединить $n$ точек $(n > 4)$ стрелками так, что из каждой точки в каждую можно было попасть, пройдя либо по одной стрелке, либо по двум (каждые две точки можно соединить стрелкой только в одном направлении; идти по стрелке можно только в указанном на ней направлении).
7. С помощью неравенства $E \leq 3V - 6$ можно доказать следующие три изящных факта …

## Вопросы

1. Дайте определение графа.
2. Приведите примеры ориентированного, неориентированного и смешанного графов, которые состоят из пяти вершин и пяти ребер.
3. Укажите на приведенных примерах смежные вершины, смежные ребра (дуги).

4. Дайте определение маршрута на графе. Как представляются маршруты?
5. Приведите пример графа с шестью вершинами, семью ребрами и перечислите все маршруты этого графа.
6. Какой маршрут называется цепью? Покажите на рисунке любого графа цепи и маршруты.
7. Какой маршрут называется циклическим? Приведите пример графа, содержащего цикл.
8. Какой граф называется связным? Несвязным? Приведите пример связного графа с пятью вершинами и пятью ребрами. Преобразуйте этот граф в несвязный.
9. Дайте определение дерева, леса, корневого дерева.
10. Приведите пример корневого бинарного дерева *высотой 2*. Сколько листьев содержит такое дерево? Сколько в нем вершин?
11. Приведите пример корневого тернарного дерева *высотой 2*. Сколько в нем вершин?
12. Как представляется граф при помощи матрицы инциденций? Приведите пример представления графа этим способом.
13. Как представляется граф при помощи векторов смежностей? Приведите пример представления графа этим способом.
14. Есть ли наилучший способ представления графа?

# ГЛАВА 5. КОМБИНАТОРНЫЕ ЗАДАЧИ И МЕТОДЫ КОМБИНАТОРНОГО ПОИСКА

## 5.1. Перечислительные и оптимизационные комбинаторные задачи. Комбинаторные конфигурации: перестановки и размещения

Можно выделить три типа задач комбинаторного анализа:

1) перечислительные задачи, связанные с подсчётом комбинаторных конфигураций, удовлетворяющих заданным условиям (существование внутренне устойчивых подмножеств вершин ранга);
2) задачи существования или несуществования конфигураций, удовлетворяющих заданным условиям и нахождение алгоритма построения конфигураций (существует ли Гамильтонов цикл в графе?);
3) экстремальные комбинаторные задачи, связанные с выделением из заданной совокупности конфигураций таких, которые обладали бы избранным свойством в наибольшей или наименьшей степени.

***Общие правила комбинаторики:***

1. *Правило суммы:* Если некоторый объект $A$ можно выбрать $n$ способами, а другой объект $B$ можно выбрать $m$ способами, то выбор «либо $A$, либо $B$» можно осуществить $(n + m)$ способами.
2. *Правило произведения. Основные комбинаторные конфигурации:* Размещением из $n$ по $k$ называется упорядоченный набор из $k$ различных элементов множества $\Omega$. В размещениях учитывается порядок следования элементов. Так два набора $(a1, a2, a3)$, $(a2, a3, a1)$ являются различными $3$-элементными размещениями из $6$-элементного множества $\{a1, a2, a3, a4, a5, a6\}$.

Говоря неформально, размещение – это расположение «предметов» на некоторых «местах» при условии, что каждое место занято в точности $1$ предметом и все предметы различны.

***Теорема.*** Число размещений из $n$ из $k$, $1 \le k \le n$, равно A k n = n(n–1) · · · · · (n–k + 1).

Сочетанием из $n$ по $k$ называется *k-элементное подмножество множества Ω*. Множество характеризуется составом, т.е. своими элементами, а не их порядком, поэтому подмножества $\{a1, a2\}$, $\{a2, a1\}$ множества $\Omega$ являются одинаковыми двухэлементными сочетаниями из $n$-элементного множества $\Omega$.

## 5.2. Производящие функции. Принцип включения и исключения. Методы комбинаторного поиска. Задача о кратчайшем покрытии

***Производящая функция*** – это функция, которая позволяет определить явный вид общего члена рассматриваемой числовой последовательности.

Так, пусть имеется последовательность $\{a_n\}$, для главного члена $a$ которой нужно найти общую формулу. Тогда введем функцию $A(x) = \sum_n a_n x^n$.

Если, используя свойства рассматриваемой последовательности, удастся решить составленные для $A(x)$ уравнения, то можно будет получить искомые элементы последовательности. Один из примеров применения производящих функций относится к *числам Фибоначчи*. Они являются решением рекуррентного уравнения $F_{n+1} = F_n + F_{n-1}$, где $F_0 = F_1 = 1$. Рассмотрим производящую функцию $F(x) = \sum_n f_n x^n$.

Она удовлетворяет уравнению:

$$F(x) = 1 + x + \sum_{k=2:\infty} (F_{k-2} + F_{k-1})x^k = \ldots = 1 + (x + x^2)\, F(x).$$

Отсюда получаем, что $F(x) = (1 - x - x^2)^{-1}$. Далее находим разложение $1 - x - x^2 = (1 - ax)(1 - bx)$, где $a = (1 + \sqrt{5})/2$, $b = (1 - \sqrt{5})/2$. Далее, используя метод неопределенных коэффициентов, получаем $F(x) = \sum_{k=0:\infty} (a^{k+1} - b^{k+1})x^k / (a - b)$.

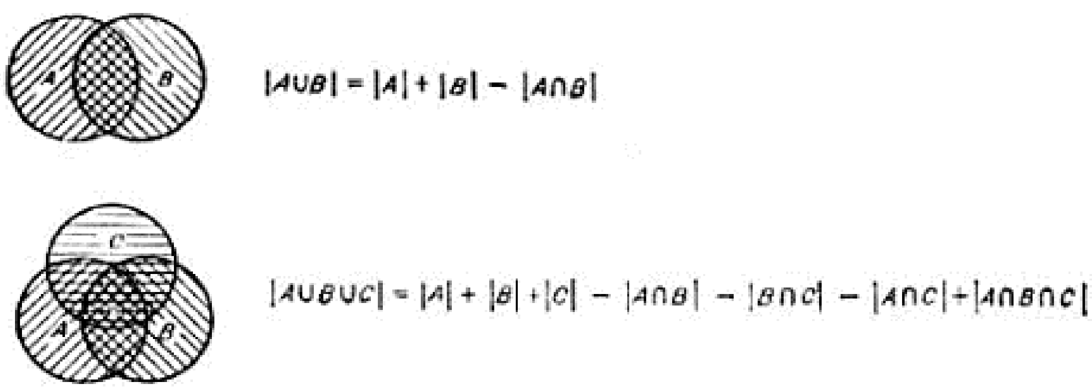
Интересно, что предел отношения $f_{n+1}/f_n$ равен $a$, т.е. «золотому сечению».

Также существует связь между числами Фибоначчи и треугольником Паскаля: можно выбрать линии, пересекающие узлы треугольника, так, чтобы сумма всех чисел на одной линии соответствовала *числу Фибоначчи*.

Другим примером являются *числа Каталана*. Они появляются в контексте следующей задачи: нужно найти число различных последовательных действий, чтобы вычислить сумму $S_0$, ..., $S_n$, складывая любые два рядом стоящих числа и помещая результат на их место. Тогда, если обозначить искомое число через $c_n$, то производящая функция будет выглядеть, как $C(x) = \sum_{n\,=\,0\,:\,\infty} c_n x^n$.

Следуя аналогичным рассуждениям, как и в случае с числами Фибоначчи, получим, что $C(x) = \sum_{n\,=\,0\,:\,\infty} C_{2n\,,\,n}\, x^n / (n + 1)$.

*Принцип включения и исключения.* Для общего представления приведем Диаграммы Венна для трех множеств $A$, $B$ и $C$:




Простые частные случаи принципа включения и исключения.

**Рис. 13 – Диаграммы Венна**

***Теорема*** $|\cup_{i=1\,:\,n} A_i| = \sum_{i\,=\,1\,:\,n}|A_i| - \sum_{1 \le i < j \le n}|A_i \cap A_j| + {}$ $+\sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k| - \ldots + (-1)^{n-1}|A_1 \cap \ldots \cap A_n|$.

*Доказательство.* Будем доказывать теорему по индукции. База индукции очевидно верна. Тогда пусть для $A_1, \ldots, A_{n-1}$ выполняется $|\cup_{i\,=\,1\,:\,n-1} A_i| = \sum_{i\,=\,1\,:\,n-1} |A_i| - \sum_{1 \le i < j \le n-1} |A_i \cup A_j| + {}$ $+ \sum_{1 \le i < j < k \le n-1} | A_i \cap A_j \cap A_k| - \ldots + (-1)^{n-2} |A_1 \cap \ldots \cap A_{n-1}|$.

Если мы применим эту формулу к сумме $(A_1 \cup \ldots \cup A_{n-1}) \cap A_n = {}$ $= \cup_{i\,=\,1\,:\,n-1} (A_i \cap A_n)$, то получим $|\cup_{i\,=\,1\,:\,n-1} A_i \cap A_n| = \sum_{i\,=\,1\,:\,n-1} |A_i \cap A_n| - {}$ $- \sum_{1 \le i < j \le n-1} |A_i \cap A_j \cap A_n| + \sum_{1 \le i < j < k \le n-1} | A_i \cap A_j \cap A_k \cap A_n| - \ldots + {}$ $-(-1)^{n-2} |A_1 \cap \ldots \cap A_{n-1} \cap A_n|$.

Отсюда уже получаем $|\cup_{i\,=\,1:n} A_i| = |(\cup_{i\,=\,1:n-1} A_i) \cup A_n| = {}$ $= |\cup_{i=1:n-1} A_i| + |A_n| - |\cup_{i\,=\,1:n-1} A_i \cap A_n| = \sum_{i\,=\,1:n} |A_i| - \sum_{1 \le i < j \le n} |A_i \cap A_j| + {}$ $+ \sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k| - \ldots + (-1)^{n-1} |A_1 \cap \ldots \cap A_n|$.

***Пример 5.1:***

Представьте, что перед вами на столе материализовалось яблоко, груша и банан. Выкладываем фрукты слева направо в следующем порядке: яблоко / груша / банан.

Сколькими способами их можно переставить?

*Пример решения:* три объекта можно переставить по формуле: $P_n = n!$; $P_3 = 3! = 6$-ю способами.

a) Сколькими способами можно выбрать один фрукт?

*Пример решения:* один фрукт можно выбрать, очевидно, 3 способами – взять либо яблоко, либо грушу, либо банан.

Формальный подсчёт проводится по формуле количества сочетаний:

$$C_n^r = \frac{n!}{r!(n-r)!}; \quad C_3^1 = \frac{3!}{1!(3-1)!} = \frac{6}{2*1} = 3 \text{ способами.}$$

b) Сколькими способами можно выбрать два фрукта?

*Пример решения:* Формальный подсчёт проводится по формуле количества сочетаний:

$$C_n^r = \frac{n!}{r!(n-r)!}; \; C_3^2 = \frac{3!}{2!(3-2)!} = \frac{3!}{2!*1!} = \frac{6}{2} = 3 \text{ способами.}$$

c) Сколькими способами можно выбрать три фрукта?

*Пример решения:* Формальный подсчёт проводится по формуле количества сочетаний:

$$C_n^r = \frac{n!}{r!(n-r)!}; \; C_3^3 = \frac{3!}{3!(3-3)!} = 1 \text{ способом.}$$

d) Сколькими способами можно взять хотя бы один фрукт?

*Пример решения:* Условие «хотя бы один» подразумевает, что нас устраивает 1 фрукт (любой) или 2 любых фрукта или все 3 фрукта: $C_3^1 + C_3^2 + C_3^3 = 3 + 3 + 1 = 7$ способами можно выбрать хотя бы один фрукт.

**Задачи**

1. В ящике находится 15 деталей. Сколькими способами можно взять 4 детали?
2. Сколькими способами из колоды в 36 карт можно выбрать 3 карты?

3. В группе 9 человек. Сколько можно образовать разных подгрупп при условии, что в подгруппу входит не менее 2 человек?
4. В пассажирском поезде 9 вагонов. Сколькими способами можно рассадить в поезде 4 человека – при условии, что все они должны ехать в различных вагонах?
5. Предприятие может предоставить работу по одной специальности 4 женщинами, по другой – 6 мужчинам, по третьей – 3 работникам независимо от пола. Сколькими способами можно заполнить вакантные места, если имеются 14 претендентов: 6 женщин и 8 мужчин?
6. Группу из 20 студентов нужно разделить на 3 бригады, причем в первую бригаду должны входить 3 человека, во вторую – 5 и в третью – 12. Сколькими способами это можно сделать?
7. Для участия в команде тренер отбирает 5 мальчиков из 10. Сколькими способами он может сформировать команду, если 2 определенных мальчика должны войти в команду?
8. В шахматном турнире принимали участие 15 шахматистов, причем каждый из них сыграл только одну партию с каждым из остальных. Сколько всего партий было сыграно в этом турнире?
9. Сколько существует разных исходов эксперимента, связанного с $n$ бросаниями монеты (исходы двух экспериментов считаются разными, если очерёдность выпадения гербов в этих экспериментах не совпадает с очерёдностью выпадения цифр)?
10. При игре в бридж между четырьмя игроками распределяется колода карт в 52 листа по 13 карт каждому игроку. Сколько существует различных способов раздать карты?
11. Сколько ожерелий можно составить из 7 бусинок разных размеров (надо использовать все 7 бусинок)?

**Вопросы**

1. Напишите формулу вычисления числа перестановок элементов множества, состоящего из $k$ элементов.

2. Определите, на сколько больше число перестановок из 7 элементов множества, чем из 5 его элементов?
3. Во сколько раз больше число перестановок из 7 элементов множества, чем из 5 его элементов?
4. Сколькими способами можно упорядочить множество чисел *{1, 2, 3, 4, ..., 2}* так, чтобы каждое число имело четный номер?
5. Сколько можно составить перестановок из *n* элементов некоторого множества, выбранные 2 элемента которых стоят рядом?
6. Составьте все перестановки множества чисел *{1, 2, 3, 4}* методом транспозиции *числа 1* с соседним элементом.
7. Составьте все перестановки множества чисел *{1, 2, 3, 4}* методом вставки числа 4 во всех перестановках чисел *{1, 2, 3}*.
8. Дайте определение перестановки с повторениями.
9. Напишите формулу для вычисления числа перестановок с повторениями для *n–элементного множества* с *r-группами* повторяющихся перестановок.
10. Сколько различных слов можно составить, переставляя буквы в слове «математика»?
11. Сколько пятибуквенных слов можно составить из букв *A*, *B*, *C*, если *A* повторяется в слове не более 2 раз, *B* – не более 1 раза и *C* – не более 3 раз?

# ГЛАВА 6. ОСНОВЫ ТЕОРИИ АЛГОРИТМОВ

## 6.1. Интуитивное понятие алгоритма и его уточнение в модели Маркова. Алгоритмическая модель Тьюринга

*Интуитивное понятие алгоритма* – одно из основных понятий математики, не допускающее определения в терминах более простых понятий.

*Конструктивный объект* – тот, который может быть набран весь целиком и представлен для рассмотрения.

*Неконструктивными объектами* являются, например, любые действительные числа, представления которых в десятичной записи $a_0\ a_1 \ldots a_n \ldots$ ни для какого $n$ из натуральных чисел не может быть целиком представлено для рассмотрения. Числа и не являются конструктивными объектами.

*Примеры алгоритмов*:

- алгоритмы арифметических действий с числами в двоичной системе счисления;
- алгоритм нахождения *НОД* 2 целых чисел, 2 многочленов от переменной $x$;
- алгоритм дифференцирования многочлена от переменной;
- алгоритм нахождения определителя квадратной матрицы.

***Примеры:***

- 10-я проблема Гильберта о разрешимости произвольного диофантового уравнения;
- проблема определения общезначимости произвольной формулы логики предикатов.

Нормальный алгоритм Маркова есть указание использовать ***упорядоченный список*** правил подстановки: $\alpha_i \Rightarrow \beta_i$, где $\alpha_i$ и $\beta_i$ – некоторые слова в алфавите $V_T$. Множество правил и порядок их использования позволяют выполнять преобразования исходного слова $P_o$ в заключительное слово $Q$, т.е. $P_o \rightarrow P_1 \rightarrow P_2 \rightarrow \ldots \rightarrow P_i \rightarrow \ldots \rightarrow Q$.

Для организации последовательного и упорядоченного просмотра правил последние должны быть индексированы $i \in \{1, 2, 3, \ldots\}$. Если слово $P_i$ есть цепочка вида $(\gamma_1 \alpha_i \gamma_2)$ в алфавите $V_m$, где $\gamma_1$ и $\gamma_2$ – слова в этом же алфавите и среди множества правил первым в упорядоченном списке есть правило $\alpha_i \Rightarrow \beta_i$, то нужно выполнить подстановку $(\gamma_1\alpha_i\gamma_2) \Rightarrow (\gamma_1\beta_i\, \gamma_2)$.

Распознаватели вхождения соединяются последовательно в соответствии с заданной последовательностью правил. Второй выход распознавателя вхождения при обнаружении $\alpha_i$ в слове $P_i$ передает информацию о слове $P_i = \gamma_1\alpha_i\gamma_2$ в $ОП_i$, где выполняется соответствующая замена слова $\alpha_i$ на слово $\beta_i$, т.е. $\gamma_1\alpha_i\gamma_2 \Rightarrow \gamma_1\beta_i\gamma_2 = P_{i+1}$.

*Машина Тьюринга* состоит из информационной ленты, считывающей и записывающей головки и управляющего устройства:

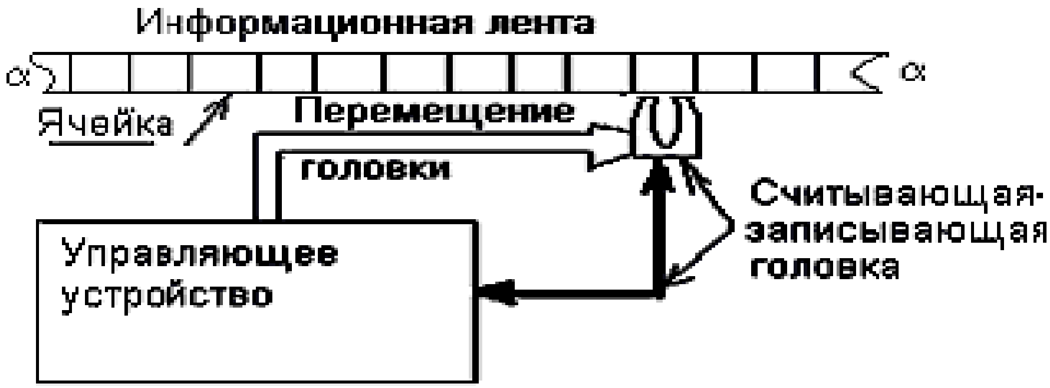



**Рис. 14 – Машина Тьюринга**

*Информационная лента* бесконечной длины представляет собой последовательность ячеек, в каждую из которых записан в точности только 1 символ из множества символов алфавита $V_T = \{a_1, a_2, \ldots, a_n\}$. Последовательность символов на ленте формирует слово $\alpha = (a_{12}\ldots)$. Пробел между словами также является символом множества $V_T$.

Информационная лента исполняет функции *внешней памяти* машины Тьюринга.

*Считывающая-записывающая головка* обозревает только одну ячейку информационной ленты, передает информацию о ее содержимом в управляющее устройство и по указанию последнего сохраняет или изменяет содержимое ячейки.

*Управляющее устройство* представляет собой механизм, который на каждом шаге вычисления находится в одном из множества состояний $Q = \{q_1, q_2, \ldots, q_m\}$. В зависимости от состояния $q_i$ и считанного символа $a_j$ управляющее устройство выдает команду на стирание или запись символа в обозреваемую ячейку, перевод управляющего устройства в новое состояние и перемещение головки на соседнюю ячейку информационной ленты. Поэтому состояния управляющего устройства называют «памятью машины Тьюринга», т.к. машина помнит все промежуточные состояния, которые привели машину из состояния $q_0$ (так обозначается начальное состояние) в некоторое состояние $q_i$.

## 6.2. Алгоритмически разрешимые и неразрешимые проблемы. Вычислительная сложность проблем. Некоторые алгоритмы на графах

***Алгоритмическая разрешимость*** – свойство формальной теории обладать алгоритмом, определяющим по данной формуле, выводима она из множества аксиом данной теории или нет. Теория называется разрешимой, если такой алгоритм существует, и неразрешимой, в противном случае. Вопрос о выводимости в формальной теории является частным, но вместе с тем, важнейшим случаем более общей проблемы разрешимости.

*Алгоритмически не разрешимые проблемы.* Свойство массовости алгоритмов означает, что они предназначены для решения некоторой массовой проблемы, формулируемой в виде отображения множества входных слов в соответствующие им выходные слова. Поэтому всякий алгоритм можно рассматривать как универсальное средство для решения целого класса задач.

В теории алгоритмов известны некоторые задачи, для решения которых не существует единого универсального приема. Однако алгоритмическая неразрешимость проблемы решения задач того или иного класса не означает невозможность решения любой конкретной задачи из этого класса.

*Вычислительная сложность проблем.* Теория сложности вычислений – раздел теории вычислений, изучающий объем работы, требуемой для решения вычислительной проблемы. Задача рассматривается как сложная, если решение проблемы требует большого количества ресурсов, независимо от алгоритма, используемого для ее решения. Теория формализует это интуитивное понятие, вводя математические модели вычислений для изучения этих проблем и количественной оценки объема ресурсов, необходимых для их решения, такие как время и используемая память.

*Представление машины Тьюринга графом.* При представлении машины Тьюринга посредством графа необходимо каждому состоянию поставить в соответствие вершину графа, а каждой команде – помеченную дугу.

Машина Тьюринга из рассмотренного примера инвертирования цепочки, состоящей из символов *0* и *1*, будет представлена в виде графа следующим образом:

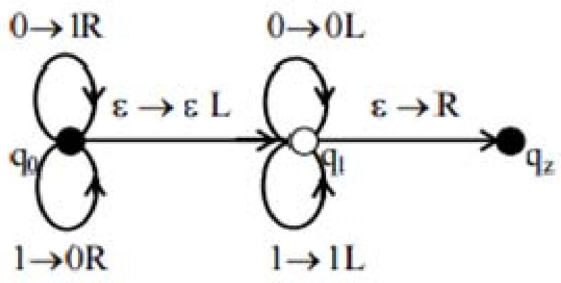



**Рис. 15 – Машина Тьюринга**

Начальная и конечная вершины графа обычно выделяются; на рисунке они отмечены черным кружком. Если на очередном такте работы машины Тьюринга символ на ленте не изменяется, то в правой части команды его можно не писать ($\varepsilon$ на ребре в состояние $q_z$).

## 6.3. Понятие о конечном автомате. Интерпретация автоматов. Способы задания автоматов. Автоматы и теория алгоритмов

***Конечный автомат*** – абстрактный автомат, число возможных внутренних состояний которого конечно.

Конечные автоматы делятся на:

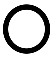
- детерминированные;
- недетерминированные.

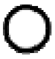
*Детерминированные конечные автоматы.* Детерминированным конечным автоматом (сокращенно *д. к. а.*) называется упорядоченная пятерка $A = \{Q, A, \delta, q_0, F\}$, состоящая из следующих объектов:

a) $Q = \{q_0, \ldots, q_m\}$ – конечный алфавит внутренних состояний автомата;
b) $A = \{a_0, \ldots, a_n\}$ – конечный входной алфавит автомата;
c) $\delta : Q \times A \rightarrow Q$ – функция перехода;
d) $q_0 \in Q$ – начальное состояние;
e) $F \subseteq Q$ – множество выделенных (конечных) состояний.

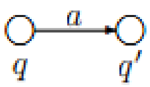
*Графическое изображение детерминированных автоматов.* Автоматы удобно изображать графически, используя следующие геометрические фигуры:

 > – начальное состояние;

 – выделенное состояние;

 – промежуточное состояние;

 > – одновременно начальное и выделенное состояние;



– такая дуга присутствует в автомате, если значение функции перехода $\Delta(q, a) = q'$;

– такая петля присутствует в автомате, если значение функции перехода $\delta(q, a) = q$.

*Определение.* Путем в детерминированном конечном автомате $A = \{Q, A, \delta, q_0, F\}$ назовем любую конечную последовательность $\{r_0, s_1, r_1, \ldots, s_k, r_k\}$, где $r_0, \ldots, r_k \in Q$; $s_1, \ldots, s_k \in A$ и $\delta(r_i, s_{i+1}) = r_{i+1}$ для всех $i < k$.

Если $\{r_0, s_1, r_1, \ldots, s_k, r_k\}$ – путь в автомате, то будем обозначать его через: $r_0 \xrightarrow{s_1} r_1 \xrightarrow{s_2} \ldots \xrightarrow{s_k} r_k$ и говорить, что слово $w = s_1 s_2 \ldots s_k$ читается вдоль дуг данного пути. В частности, пустое слово $\Lambda$ читается вдоль пути $\{r_0\}$, состоящего из одного состояния и не содержащего ни одной дуги. Пусть $A = \{Q, A, \delta, q_0, F\}$ – *д. к. а*.

Расширим функцию $\Delta$ до функции $\delta^* : Q \times A^* \rightarrow Q$ следующим образом индукцией по длине слова: $\delta^*(q, \Lambda) = q$, $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$, где $q \in Q$, $w \in A^*$, $a \in A$.

Говорят, что *д. к. а.* $A = \{Q, A, \delta, q_0, F\}$ распознает слово $w \in A^*$, если $\delta^*(q_0, w) \in F$.

Другими словами, слово $w = s_{12} \ldots s_k$ распознается автоматом, если в нем существует путь – $q_0 = r_0 \xrightarrow{s_1} r_1 \xrightarrow{s_2} \ldots \xrightarrow{s_k} r_k \in F$, такой, что он начинается в начальном состоянии $q_0$, вдоль его дуг читается слово $s_1s_2 \ldots s_k$ (в детерминированном автомате такой путь определяется однозначно по слову $w$) и заканчивается в некотором выделенном состоянии.

Через $T(A) = \{w \in A^* \mid \delta^*(q0, w) \in F\}$ будем обозначать язык всех слов, распознаваемых конечным автоматом $A$.

Язык $L \subseteq A^*$ называется автоматным, если существует конечный автомат $A$, такой, что $L = T(A)$.

***Недетерминированные конечные автоматы***

Недетерминированные конечные автоматы отличаются от детерминированных тем, что из любого состояния $q$ после считывания символа а возможны сразу несколько (или ни одного) переходов в состояния, принадлежащие множеству возможных состояний $\Delta(q, a)$.

Это означает, что автомат может продолжить дальнейший процесс чтения символов, перейдя в любое из возможных состояний.

*Недетерминированным конечным автоматом* (сокращенно *н. к. а.*) называется упорядоченная пятерка $A = \{Q, A, \Delta, q_0, F\}$, в которой $Q, A, q_0, F$ определяются и называются так же, как в детерминированном случае, а функция переходов $\Delta$ является функцией вида $\Delta: Q \times A \to P(Q)$, где $P(Q)$ – множество всех подмножеств $Q$.

*Графическое изображение недетерминированных автоматов.* При графическом изображении недетерминированных автоматов используются те же обозначения, что и в детерминированном случае. При этом из одного состояния автомата могут выходить сразу несколько (или нисколько) стрелок, помеченных одной и той же буквой алфавита.

Дуга, выходящая из состояния $q$, входящая в состояние $q^l$, помеченная символом $a$, присутствует в схеме автомата тогда и только тогда, когда $q^l \in \Delta(q, a)$.

Путь в недетерминированном конечном автомате $A = \{Q, A, \Delta, q_0, F\}$ определяется и обозначается так же, как в детерминированном случае, нужно лишь условие $\delta(r_i, s_i+1) = r_i+1$ заменить на условие $r_{i+1} \in \Delta(r_i , s_{i+1})$.

*Теорема (о детерминизации).* Для любого недетерминированного конечного автомата $A$ существует детерминированный конечный автомат $A$ такой, что $T(A) = T(A)$.

*Абстрактный автомат* – смысл состоит в реализации некоторого отображения множества слов входного алфавита в множестве слов выходного алфавита.

Известно, что конечный автомат представляет собой хотя и абстрактную, но с функциональной точки зрения довольно точную модель дискретного (цифрового) вычислительного или управляющего устройства.

*Входная буква* – это входной сигнал (точнее комбинация сигналов на всех входах устройства), *входное слово* – последовательность входных сигналов, поступающих в автомат в дискретные моменты времени (такты) $t = 1,2,3,\ldots;$ *выходное слово* – последовательность выходных сигналов, выдаваемых автоматом, *состояние автомата* – это комбинация состояний запоминающих элементов устройства.

*Способы задания автоматов.* Чтобы задать конечный автомат $S$, необходимо описать все элементы множества $S = \{A, X, Y, d, l\}$, т.е. необходимо описать входной, выходной алфавиты и алфавит состояний, а также функции переходов $d$ и выходов $l$. При этом среди множества $A = \{a_0, a_1, \ldots, a_n\}$ необходимо выделить начальное состояния $a_0$, в котором автомат находится в момент времени $t = 0$. Существует несколько способов задания работы автомата, но наиболее часто используются табличный и графический.

1. *Табличный способ.* При этом способе автомат Мили описывается двумя таблицами: таблицей переходов и таблицей выходов. Задание таблиц переходов и выходов полностью описывает работу конечного автомата, поскольку задаются не только сами функции переходов и выходов, но и также все три алфавита: входной, выходной и алфавит состояний. Для задания автомата Мура требуется

одна таблица, поскольку в этом автомате выходной сигнал однозначно определяется состоянием автомата.

2. *Графический способ задания автомата* (задание автомата с помощью графа). Этот способ основан на использовании ориентированных связных графов. Вершины графов соответствуют состояниям автомата, а дуги – переходам между ними. Две вершины графа $a_i$ и $a_s$ соединяются дугой, направленной от $a_i$ к $a_s$, если в автомате имеется переход из $a_i$ в $a_s$, т.е. $a_s = d(a_i, x_j)$.

В автомате Мили дуга отмечается входным сигналом $x_j$, вызвавшим переход, и выходным сигналом $y_g$, который возникает при переходе. Внутри кружочка, обозначающего вершину графа, записывается состояние. Например, для автомата Мили граф имеет вид **Рис. 16**, а для автомата Мура вид **Рис. 17**:

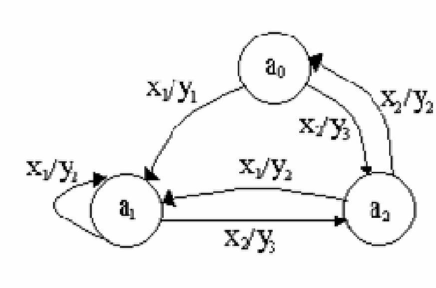



**Рис. 16 – Автомат Мили**

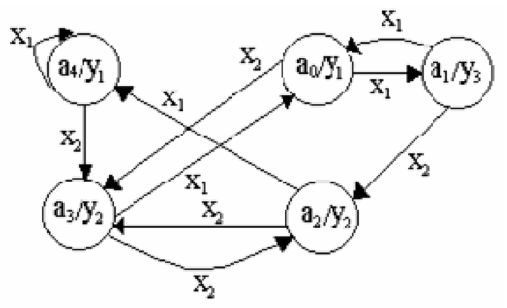



**Рис. 17 – Автомат Мура**

***Автоматы и теория алгоритмов***

Событие, представимое в автомате, – это множество, разрешимое автоматом, или, как говорят, автоматно-разрешимое множество. Для инициального автомата с выходами и заключительными состояниями вводится понятие автоматно-перечислимого множества: множество, перечислимое автоматом, – это множество всех выходных слов, образованных путями из начального состояния в заключительные состояния, иначе говоря, множество всех слов $S(a)$, таких, что $\delta(q_1, a)$ – заключительное состояние.

Если множество автоматно-перечислимо, то оно и автоматно-разрешимо, однако число состояний разрешающего автомата может экспоненциально возрасти.

***Пример 5.2:***

Доказать, что, если функция $f(x_1, x_2, \ldots, x_n)$ примитивно рекурсивна, то следующие функции также примитивно рекурсивны:

a) $\Psi(x_1, x_2, \ldots, x_n) = f(x_2, x_3, \ldots, x_n, x_1)$ – циклическая перестановка аргумента;

b) $\varphi(x_1, x_2, \ldots, x_{n-1}) = f(x_1, x_1, x_2, \ldots, x_{n-1})$ – отождествление аргументов.

*Решение:*

a) функция $\psi$ получается суперпозицией из $f$ и $I_n^m$:
$$\psi(x_1, x_2, \ldots, x_n) = f\big(I_n^2(x_1, x_2, \ldots, x_n), I_n^3(x_1, x_2, \ldots, x_n), \ldots, I_n^n(x_1, x_2, \ldots, x_n),$$
$$I_n^1(x_1, x_2, \ldots, x_n)\big);$$

b) функция $\varphi$ — суперпозиция $f$ и $I_{n-1}^m$: $\varphi(x_1, x_2, \ldots, x_{n-1}) =$
$$= f\big(I_{n-1}^1(x_1, x_2, \ldots, x_{n-1}), I_{n-1}^1(x_1, x_2, \ldots, x_{n-1}), \ldots, I_{n-1}^{n-1}(x_1, x_2, \ldots, x_{n-1})\big).$$

**Задачи**

1. Доказать, что функция, перечисляющая по порядку числа Фибоначчи (итальянский математик): $\begin{cases} f(0) = 0,\ f(1) = 1, \\ f(n+2) = f(n) + f(n+1) \end{cases}$ примитивно рекурсивна.
2. Доказать:
   *a)* $i\left(1 + \left[\frac{x}{2}\right]\right) = sgn(x)$;
   *b)* $i\left(1 + x + \left[\frac{x+1}{2}\right] + \left[\frac{x}{2}\right]\right) = 2x-1$;
   *c)* $i\left(1 + x + \left[\sqrt{4x+1}\right]\right) = x^2 + x$, где $f(x) = ig(x)$ – операция итерирования функций.
3. Построить машину Тьюринга для правильного вычисления функции $x + y$.
4. Пусть функция $f(x)$ и $g(x)$ правильно вычислимы. Показать, что функция $h(x) = f(g(x))$ правильно вычислима.
5. На ленте записаны два числа в двоичной системе счисления, разделенные звездочкой: *1011*101*. Определите, какую операцию проделает с ними машина Тьюринга, начиная из стандартного положения (крайняя правая ячейка, состояние $q_1$), если ее программа задается таблицей.

6. Докажите, что если машины Тьюринга $F$ и $G$ правильно вычисляют функции $f(y)$ и $g(x)$ соответственно, то композиция $H = FG$ этих машин правильно вычисляет сложную функцию: $h(x) = f(g(x))$.
7. Постройте машину Тьюринга (называемую «удвоение» и обозначаемую $Г$), которая перерабатывает слово *01*0* в слово *01*01*0*, причем в начальном и конечном положении обозревается крайняя левая ячейка.

**Вопросы**

1. Что называется вы×одом алгоритма?
2. Какие конкретные данные могут поступать на вход ЭВМ?
3. Правильно ли сказать, что алгоритм преобразует входные данные в результате решения задачи?
4. Какие объекты в самом общем виде являются входом алгоритма?
5. Как называются этапы, на которые разбивается алгоритм? Конечно ли их число?
6. Перечислите способы представления алгоритмов.
7. Что представляет собой блок-схема алгоритма?
8. Составьте алгоритм поиска наибольшего числа из множества трех чисел $a$, $b$, $c$.
9. Представьте алгоритм в виде блок-схемы и последовательности шагов.
10. Какие этапы включает анализ алгоритмов?
11. Какие способы используются для доказательства правильности алгоритмов?
12. Как осуществляется экспериментальная проверка правильности алгоритма? В чем ее недостаток?
13. Что представляют собой временные оценки алгоритма? Каковы пути их получения?
14. Как получают экспериментальные оценки времени работы алгоритма? В чем недостатки экспериментального метода?
15. Как улучшить достоверность экспериментальных оценок временной сложности алгоритмов?

16. Какие параметры алгоритма используются при теоритическом анализе его временной сложности?
17. Что представляет собой преобразование одной задачи в другую (сведение) и зачем оно осуществляется?
18. Какую возможность предоставляют разработчику алгоритмов понятия классов *NP* и *NP*-полных задач?
19. В чем смысл задачи о выполняемости булевой функции; к какому классу задач она относится: к *P* или *NP*?
20. Каков порядок доказательства того факта, что задача входит в класс *NP*-полных задач?

# ЛИТЕРАТУРА


1. Вороненко, А.А. Дискретная математика: задачи и упражнения с решениями : учебно-методическое пособие / А.А. Вороненко, В.С. Федорова. – М. : ИНФРА-М, 2014. – 104 с.
2. Емилечев, В.А. Лекции по теории графов : учеб. пособие / В.А. Емилечев [и др.] – 2-е изд., испр. – М. : Книжный дом «ЛИБРОКОМ», 2009. – 392 с.
3. Закревский, А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – М. : Физматлит, 2007. – 592 с.
4. Канцедал, С.А. Дискретная математика : учеб. пособие для студентов учреждений среднего профессионального образования / С.А. Канцедал – М. : ИД ФОРУМ : НИЦ Инфра-М, 2013. – 224 с.
5. Карпов, Ю.Г. Теория автоматов : учебник для вузов / Ю.Г. Карпов. – СПб. : Питер, 2002. – 206 с.
6. Котов, В.М. Дискретная математика. Специальный курс : пособие для студентов специальности 1-31 03 04 «Информатика» / В.М. Котов. – Минск : БГУ, 2010. – 115 с.
7. Кристофидес, Н. Теория графов. Алгоритмические подход / Н. Кристофидес. – М. : Мир, 1978. – 432 с.
8. Кузнецов, О.П. Дискретная математика для инженера : учеб. пособие / О.П. Кузнецов. – 6-е изд., стер. – СПб. : Лань, 2009. – 400 с.
9. Лихтарников, Л.М. Математическая логика. Курс лекций. Задачник-практикум и решения : учеб. пособие / Л.М. Лихтарников. – 4-е изд., стер. – СПб. : Лань, 2009. – 288 с.
10. Новиков, Ф.А. Дискретная математика для программистов / Ф.А. Новиков. – СПб. : Питер, 2000. – 304 с.
11. Осипова, В.А. Основы дискретной математики : учеб. пособие для студентов высших учебных заведений, обучающихся по направлению подготовки «Экономика» / В.А. Котов. – М. : ФОРМУ, 2013. – 159 с.

12. Просветов, Г.И. Дискретная математика: задачи и решения : учебно-практическое пособие / Г.И. Просветов. – 2-е изд., доп. – М. : Альфа-Пресс, 2013. – 240 с.
13. Шапорев, С.Д. Дискретная математика : курс лекций и практических занятий / С.Д. Шапорев. – СПб. : БХВ – Петербург, 2009. – 400 с.
14. Яблонский, С.В. Введение в дискретную математику / С.В. Яблонский. – М. : Наука, 1986. – 384 с.

*Учебное издание*

Сидская Ольга Владимировна

**Основы дискретной математики и теории алгоритмов**

Учебно-методическое пособие


Ответственный за выпуск *П.Б. Пигаль*

Редактор *Т.И. Андросюк*

Подписано в печать 16.07.2020 г. Формат 60×84/16.
Бумага офсетная. Гарнитура «Таймс». Ризография.
Усл. печ. л. 4,36. Уч.-изд. л. 2,46.
Тираж 77 экз. Заказ № 171.

Отпечатано в редакционно-издательском отделе
Полесского государственного университета
225710, г. Пинск, ул. Днепровской флотилии, 23.