Дискретная математика

Лекции

Тема 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МНОЖЕСТВ

1.1. Основные понятия и определения теории множеств

Под *множеством* понимается любое собрание определенных и различных между собой объектов, мыслимое как единое целое.

Множества будем обозначать заглавными буквами латинского алфавита; объекты, которые образуют множества, будем называть *элементами* множества и обозначать малыми буквами латинского алфавита. Если элемент *x* принадлежит множеству X, то этот факт записывается в виде $x \in X$, если не принадлежит, то $x \notin X$. Как правило, считается, что все элементы множества различны. Множество с повторяющимися элементами называется *мультимножеством.*

Множество, содержащее конечное число элементов, называется *конечным;* в противном случае множество называется бесконечным. Количество элементов конечного множества называется *мощностью* и обозначается $|X|=n$, если множество X содержит n элементов. Если множество не содержит ни одного элемента, то оно называется *пустым* и обозначается $\varnothing$.

Для произвольных множеств X и Y можно определить два типа отношений – *отношение равенства и отношение включения.*

Два множества считаются равными, если они состоят из одних и тех же элементов. Принято обозначение X=Y, если X и Y равны, и X≠Y- иначе.

Легко видеть, что для любых множеств X, Y, Z справедливы соотношения

$$X = X$$

$$X = Y \rightarrow Y = X$$

$$(X = Y \text{ и } Y = Z) \rightarrow X = Z$$

Если каждый элемент множества X является элементом множества Y, то говорят, что X включено в Y и обозначают $X \subseteq Y$:

$$X \subseteq Y \Leftrightarrow (x \in X \to x \in Y).$$

В этом случае говорят, что множество X является *подмножеством* множества Y. В частности X и Y могут совпадать, поэтому $\subseteq$ называется также отношение *нестрогого включения.* Отметим некоторые свойства подмножества, вытекающее из его определения:

$$X \subseteq X$$

$$(X \subseteq Y \text{ и } Y \subseteq Z) \to X \subseteq Z.$$

Если $X \subseteq Y$ и $X \neq Y$, то говорят, что X есть *собственное подмножество* Y и обозначают $X \subset Y$, отношение между множествами в этом случае называется отношением *нестрогого включения.* Для отношения строгого включения справедливо

$$(X \subset Y \text{ и } Y \subset Z) \to X \subset Z.$$

Невключение подмножества X в множество Y обозначается $X \not\subset Y (X \not\subseteq Y)$.

Заметим, что если X является подмножеством Y и наоборот, то X и Y состоят из одних и тех же элементов, поэтому

$$X = Y \Leftrightarrow (X \subseteq Y \text{ и } Y \subseteq X).$$

Для каждого множества X существует множество, элементами которого являются различные подмножества множества X. Такое множество называется *семейством множества* или *булеаном* и обозначается $\beta(X)$. Так как $\varnothing$ включено в любое множество, то $\varnothing \subseteq \beta(X)$.

ПРИМЕР 1.

Пусть $X = \{x_1, x_2, x_3\}$. Тогда

$\beta(X) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{x_1, x_2, x_3\}, \varnothing\}$. Если в рамках некоторого рассуждения рассматриваются подмножества некоторого множества, то оно называется *универсальным,* или *универсумом* и обозначается *E*.

Множество может быть задано различными способами: перечислением элементов в скобках { } (для конечных множеств) или указанием их свойств, однозначно определяющих принадлежность элементов данному множеству, при этом используется запись

$$X = \{x \mid x \text{ обладает свойством } P(x)\}$$

(выражение в скобках читается: множество всех элементов x, которые обладают свойством P(x). Так, множество натуральных чисел N={1,2,…} может быть описано следующим образом:

$$\mathrm{N}=\{\, i \mid \text{ если целое } i \in N, \text{ то } i+1 \in N, i \geq 1\}.$$

Кроме того, множество можно задать с помощью характеристической функции, значения которой указывают является ли (да или нет) x элементом множества X:

$$\mu_x(x)=\begin{cases} 1, x \in X \\ 0, x \notin X \end{cases}$$

Заметим, что для любых элементов $\mu_{\varnothing} = 0$; $\mu_E = 1$.

ПРИМЕР 2.

Пусть на универсуме E={a, b, c, d, e} определено множество X={a, c, d}, тогда

$$\mu_X(a)=1, \mu_X(b)=0, \mu_X(c)=1, \mu_X(d)=1, \mu_X(e)=0.$$

Декартовым (прямым) произведением множеств X и Y называется множество упорядоченных пар вида

$$\mathrm{X} \times \mathrm{Y}=\{(x,y) \mid \; x \in \mathrm{X} \text{ и } y \in \mathrm{Y}\}.$$

ПРИМЕР 3.

Пусть X=$\{x_1, x_2, x_3\}$, $\mathrm{Y}=\{y_1, y_2\}$

Тогда $\mathrm{X} \times \mathrm{Y}=\{(x_1, y_1),(x_1, y_2),(x_2, y_1),(x_2, y_2),(x_3, y_1),(x_3, y_2)\}$.

Две пары (x, y) и (u, v) считаются равными тогда и только тогда, когда $x=u$ и $y=v$.

1.2. Операции над множествами

Для получения новых множеств из уже существующих используют операции над множествами. Рассмотрим основные из них.

Суммой или *объединением* множеств X и Y называется множество $\mathrm{X} \cup \mathrm{Y}$ или $\mathrm{X}+\mathrm{Y}$, все элементы которого являются элементами множества X или Y:

$$X \cup Y = \{x \mid x \in X \text{ или } x \in Y\}.$$

Произведением или *пересечением* множеств X и Y называется множество $X \cap Y$ или $X \cdot Y$, элементы которого являются элементами обоих множеств X и Y:

$$X \cap Y = \{x \mid x \in X \text{ и } y \in Y\}.$$

Дополнением множества X называется множество $\overline{X}$ всех тех элементов *x*, которые не принадлежат множеству X:

$$\overline{X} = \{x \mid x \in E \text{ и } x \notin X\}.$$

Разностью множеств X и Y называется множество X\Y всех тех элементов X, которые не принадлежат Y:

$$X \backslash Y = \{x \mid x \in X \text{ и } x \notin Y\} = X \cap \overline{Y}.$$

Дополнение множества X представляется с помощью операции разности следующим образом: $\overline{X} = E \backslash X$.

Симметричной разностью множества X и Y называется множество $X \Delta Y$ всех элементов X и Y которые не входят одновременно в эти оба множества

$$X \Delta Y = (X \backslash Y) \cup (Y \backslash X) = (X \cup Y) \backslash (X \cap Y).$$

ПРИМЕР 4.

Пусть на универсуме E={*a, b, c, d, e, f, g*} определены множества X={*a, c, d, f*} и Y={*b, d, e, f*}.

Тогда X∪Y={ *a, b, c, d, e, f*}, X∩Y={*d, f*}, $\overline{X}$={*b, e, g*}, X\Y={*a, c*}, Y\X={*b, e*}, XΔY={*a, b, c, e*}.

Свойства операций над множествами:

1. $X \cup Y = Y \cup X$, $X \cap Y = Y \cap X$ (*коммутативность*);

2. $X \cup (Y \cup Z) = (X \cup Y) \cup Z$,

   $X \cap (Y \cap Z) = (X \cap Y) \cap Z$ (*ассоциативность*);

3. $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$ (*дистрибутивность*);

4. $X \cup \varnothing = X, \quad X \cap \varnothing = \varnothing;$

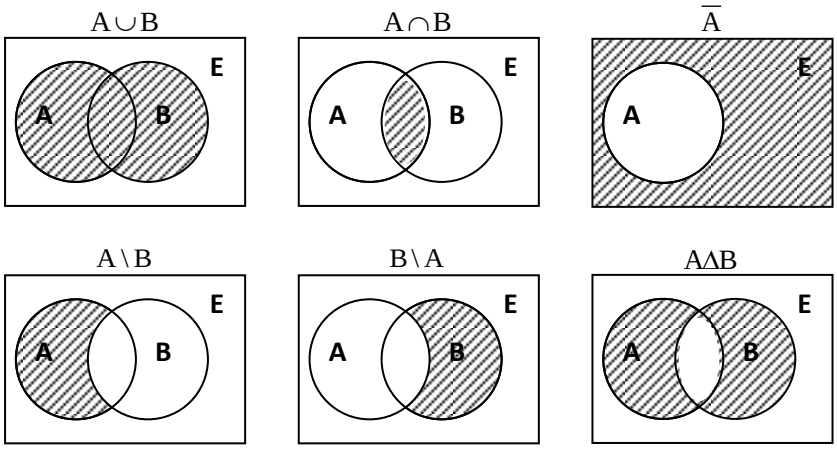
5. $X \cup E = E, \; X \cap E = X;$

6. $X \cup \overline{X} = E, \; X \cap \overline{X} = \varnothing$ (*комплиментарность*);

7 . $X \cup X = X, \; X \cap X = X$ (*идемпотентность*).

1.3. Круги Эйлера

Наглядно графически представить операции над множествами позволяют круги Эйлера (которые в литературе еще называются диаграммами Венна). Внутри прямоугольной области, соответствующей универсальному множеству, изображаются пересекающиеся окружности, каждая из которых соответствует тому или иному множеству. Примеры изображения операций над множествами с помощью кругов Эйлера:

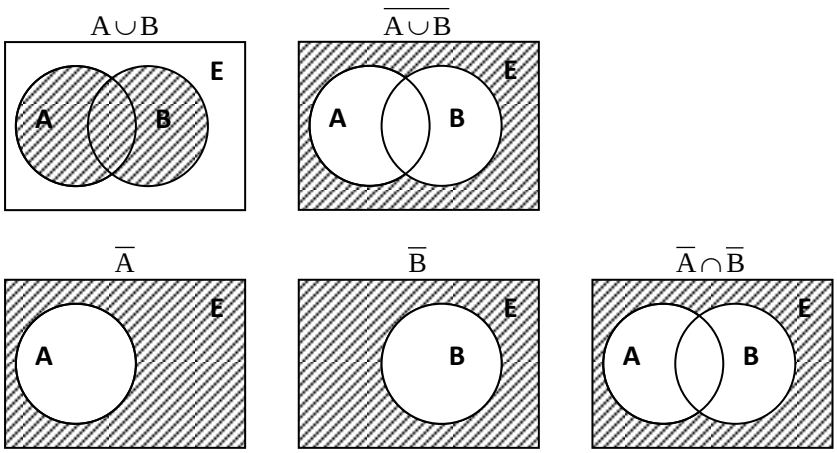


С помощью кругов Эйлера доказывать тождества и решать ряд прикладных задач. Рассмотрим примеры.

ПРИМЕР 5.

Доказать тождество $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

Решение. Изобразим с помощью кругов левую и правую части равенства и из совпадения изображений делаем заключение об истинности тождества.



ПРИМЕР 6.

Из 100 туристов, отправляющихся в заграничное путешествие, немецким языком владеют 30 человек, английским - 28, французским - 42. Английским и немецким одновременно владеют 8 человек, английским и французским - 10, немецким и французским - 5, всеми тремя языками - 3. Сколько из туристов не владеют ни одним языком?

Решение: Выразим условие этой задачи графически. Обозначим кругом тех, кто знает английский, другим кругом - тех, кто знает французский, и третьим кругом - тех, кто знают немецкий.

Всеми тремя языками владеют три туриста, значит, в общей части кругов вписываем число 3. Английским и французским языком владеют 10 человек, а 3 из них владеют еще и немецким. Следовательно, только английским и французским владеют 10-3=7 человек.

Аналогично получаем, что только английским и немецким владеют 8-3=5 человек, а немецким и французским 5-3=2 туриста. Вносим эти данные в соответствующие части.

Определим теперь, сколько человек владеют только одним из перечисленных языков. Немецкий знают 30 человек, но 5+3+2=10 из них владеют и другими языками, следовательно, только немецкий знают 20 человек. Аналогично получаем, что одним английским владеют 13 человек, а одним французским - 30 человек.

По условию задачи всего 100 туристов. 20+13+30+5+7+2+3=80 туристов знают хотя бы один язык, следовательно, 20 человек не владеют ни одним из данных языков.

Тема 2. ОТНОШЕНИЯ

Для выражения взаимодействий и связей между элементами множеств в математике используется понятие отношения.

2.1. Понятие отношения

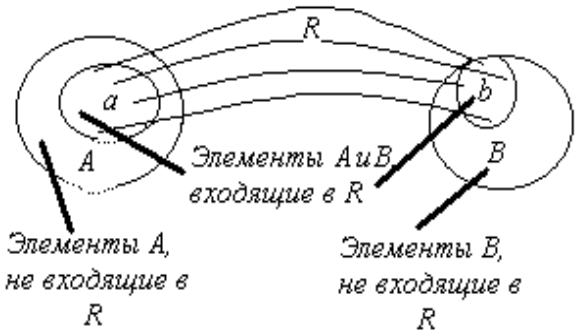
n – *местным* или n – *арным* отношением R на множествах $A_1, A_2, ..., A_n$ называется любое подмножество прямого произведения этих множеств, то есть $R \subseteq A_1 \times A_2 \times ... \times A_n$. Другими словами, элементы этих множеств $x_1, x_2, ..., x_n$ связаны отношением R тогда и только тогда, когда n упорядоченных чисел $(x_1, x_2, ..., x_n) \in R$.

Отношение $R \subseteq A^n$ называется n – *местным* на множестве A.

При $n = 0$ отношение R задает фиксированный элемент множества A. При $n = 1$ отношение R представляет собой подмножество множества A и называется *унарным отношением* или *свойством*. При $n = 2$ отношение R называется *бинарным* или *соответствием*. При $n = 3$ отношение *тернарное* и т. д.

ПРИМЕР 1.

Пусть имеются две группы студентов специальности «Прикладная математика» 1-го и 2-го курса и студенты этих групп образуют множества A и B. Тогда унарное отношение R, заданное на A имеющее смысл «быть отличником» отбирает из A те элементы (студентов), которые являются отличником. Если же отношение R бинарное и заданное на A и B, то оно имеет смысл всех пар отличников, среди которых один студент первого курса, а другой – второго. Графически данную ситуацию можно представить в виде:



В математике чаще всего используются бинарные отношение (соответствия). В дальнейшем рассматриваются в основном только такие отношения и при этом слово “бинарные” опускается.

Пусть A и B – два множества. *Соответствием* или *(бинарным) отношением* из множества A в множество B называется подмножество R прямого произведения $A \times B$, т.е. $R \subseteq A \times B$. Если $a \in A$, $b \in B$, находятся в отношении, то пишут: $(a,b) \in R$ или $R(a,b)$, а также в инфиксной форме aRb. При этом говорят, что b соответствует a при соответствии R или b находится в отношении R с a. Если $R=\varnothing$, то отношение называют *пустым*. Отношение $U = A \times B$ называют *полным*. Для любого множества A определяется *тождественное отношение* - $I = \{(a,a) | a \in A\}$.

Областью определения (*Dom R*) соответствия R, называется множество элементов $a \in A$, для каждого из которых, найдется хотя бы один элемент $b \in B$, такой, что aRb.

Областью значения (*Im R*) соответствия R называется множество элементов $b \in B$, для каждого из которых, найдется хотя бы один элемент $a \in A$, такой, что aRb.

Соответствие R называется *всюду определенным*, если *Dom* $R=A$, в противном случае – *частично определенным.* Соответствие называется сюръективным, если *Im* $R=B$.

Для каждого $a \in A$, множество элементов $b \in B$ таких, что aRb называется *образом элемента* $a \in A$ относительно R и обозначается $im_R a$.

Прообразом элемента $b \in B$ относительно R, называется множество элементов $a \in A$, таких, что aRb. Прообраз обозначается: $coim_R b$

В общем случае отношения (соответствия) могут быть заданы любым из двух способов, которые используются для задания множеств, т.е. перечислением элементов отношения или указанием их характеристических свойств.

Отношения, определенные на конечных множествах $A = \{a_1, a_2, ..., a_n\}, B = \{b_1, b_2, ..., b_m\}$, могут быть заданы:

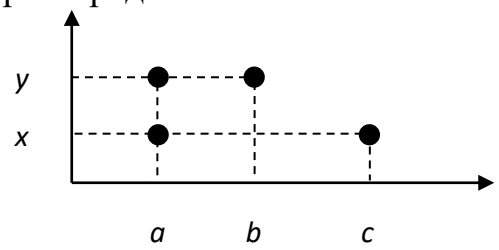
1) *Списком*, т.е. перечислением тех пар элементов, для которых это отношение выполнено. Например, если $A=\{a,b,c\}$ и $B=\{x,y\}$, то $R=\{(a,x),(a,y),(b,y),(c,x)\}$.
2) *Матрицей* [R] размерности $m \times n$, элементы которой

$$r_{ij} = \begin{cases} 1, если (a_i, b_j) \in R, \\ 0, если (a_i, b_j) \notin R, \end{cases}$$

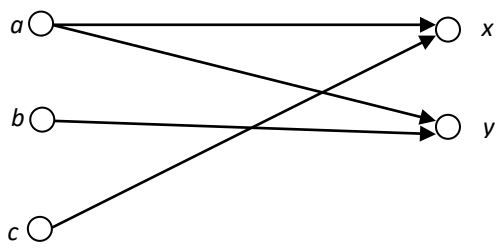
т.е. строки этой матрицы помечаются элементами из A, а столбцы – элементами из B, а на пересечении строки a_i со столбцом b_i стоит единица (1), если aRb; и нуль (0), - в противном случае. Тогда для выше приведенного примера имеем матрицу

	x	y
a	1	1
b	0	1
c	1	0

3) *Графиком* на координатной плоскости, горизонтальная и вертикальная оси которой представляют элементы множеств A и B соответственно.



4) *Графом*, в котором элементы множеств A и B изображаются точками на плоскости. Причем эти точки обозначаются теми же символами, что и соответствующие элементы. Точки a и b соединяются направленным отрезком от a к b, если aRb. Например, для предыдущего случая отношение R изображается ориентированным графом.



2.2. Свойства отношений

Пусть задано отношение на множестве A, т.е. $R \subseteq A^2 = A \times A$. Это отношение может удовлетворять одному из следующих свойств.

Рефлексивным называется отношение R, если для любого элемента a из множества A оно связано отношением R с самим же собой, то есть, если $a \in A$, то $(a,a) \in R$.

Антирефлексивным называется отношение R, если для любого элемента a из множества A не будет связано отношением R с самим же собой, то есть, если $a \in A$, то $(a,a) \notin R$.

Симметричным называется отношение R, если для любой пары элементов a и b из A, при выполнении отношения R для этих элементов, это отношение выполняется и при перестановки данных элементов местами, то есть для b и a: если для любых $(a,b) \in R$, то отсюда следует $(b,a) \in R$.

Антисимметричным называется отношение R, если для любой пары элементов a и b из A, при выполнении отношения R для этих элементов, это отношение никогда не выполняется при перестановки данных элементов местами, то есть, если для любых $(a,b) \in R$, то отсюда следует $(b,a) \notin R$.

Транзитивным называется отношение R, если для любых трех элементов a, b и c из A, при выполнении отношения R для элементов a и b и для элементов b и c, обязательно следует выполнение отношение R и для a и c, то есть из $(a,b) \in R$ и $(b,c) \in R$ всегда следует $(a,c) \in R$.

На основании приведенных свойств можно выделить особые группы отношений.

Отношение R называется отношением *эквивалентности*, если R рефлексивно, симметрично и транзитивно. Это отношение обозначается символами E и ~(тильда): aEb или $a \sim b$. Важное значение отношения эквивалентности состоит в том, что оно определяет признак, по которому происходит разбиение исходного множества на непересекающиеся подмножества, называемые классами эквивалентности. Пусть E – эквивалентность на множестве A. *Классом эквивалентности* элемента $x \in A$ называется множество $E(x) = \{y | xEy\}$. Классы эквивалентности E называются также E – классами. Отношение эквивалентности характеризует *одинаковость* объектов. Однако существуют ситуации, в которых требуется оценить *сходство* объектов. Этой цели служит отношение толерантности. Отношение, удовлетворяющее свойствам рефлексивности и симметричности, называется отношением *толерантности*.

В ряде случаев требуется указать старшинство, важность, “первичность” и другие подобные свойства объектов. Для этого служат различные виды отношения порядка.

Отношение R называется *предпорядком* или *квазипорядком*, если оно рефлексивно и транзитивно.

Отношение R называется отношением *порядка*, если оно антисимметрично и транзитивно. Отношение порядка может быть рефлексивным, и тогда оно называется отношением *нестрогого порядка*. Антирефлексивное отношение порядка называется отношением *строгого порядка*. Отношение порядка может быть полным (линейным), и тогда оно называется отношением *полного* или *линейного порядка*. Отношение порядка, не обладающее свойством полноты (линейности), называется отношением *частичного порядка*.

Отношение строгого порядка (полного или частичного) обозначается символом $<$, а отношение нестрогого порядка - $\leq$. Отношение порядка в общем случае обозначается знаком $\prec$.

Множество, на котором определено отношение частичного (полного) порядка называется *частично (вполне) упорядоченным*.

ПРИМЕР 2.

Пусть на множестве А студентов одной группы института задано бинарное отношение R. Тогда, если:

а) $(a,b) \in R$ есть «студенты a и b живут в одном районе», то оно рефлексивно (любой студент a живет в одном районе сам с собой), симметрично (если студенты a и b живут в одном районе, то b и a также живут в одном районе), транзитивно (если студенты a и b живут в одном районе и студенты b и c также живут в одном районе, то a и c также живут в одном районе). Следовательно, это отношение эквивалентности и делит всех студентов на группы, проживающие в разных районах (классы эквивалентности) а одинаковостью эквивалентности будет принадлежность к определенному району.

б) $(a,b) \in R$ есть «студент a старше студента b», то оно антирефлексивно (любой студент a не старше самого себя), антисимметрично (если студент a старше b, то b никогда не старше a), транзитивно (если студент a старше b и студент b старше c, то a будет всегда

старше c). Следовательно, это отношение строгого порядка и оно позволяет упорядочить всех студентов по возрасту.

в) $(a,b) \in R$ есть «студент a любит другого студента b», то оно только антирефлексивно (любой студент a не любит самого себя, предметом его любви должен быть другой студент), симметричность или антисимметричность отсутствует, так как если студент a любит b, то b может как любить, так и не любить a, не транзитивно (если студент a любит b и студент b любит c, то a не обязательно любит c). Следовательно, это отношение не относится к какой-либо группе.

ТЕМА 3. ЭЛЕМЕНТЫ КОМБИНАТОРИКИ

Комбинаторика занимается различного вида соединениями, которые можно образовать из элементов конечного множества. Некоторые элементы комбинаторики были известны в Индии еще во II в. до н. э. Индийцы умели вычислять числа, которые сейчас называют "сочетания". В XII в. Бхаскара вычислял некоторые виды сочетаний и перестановок. Предполагают, что индийские ученые изучали соединения в связи с применением их в поэтике, науке о структуре стиха и поэтических произведениях. Как научная дисциплина, комбинаторика сформировалась в XVII в. Все разнообразие комбинаторных формул может быть выведено из двух основных утверждений, касающихся конечных множеств – правило суммы и правило произведения.

Правило суммы.

Если имеется X элементов первого множества и Y элементов второго и эти множества не пересекаются, то один элемент из первого *или* второго множества можно выбрать X+Y способами.

ПРИМЕР 1.

а) Если на первой полке стоит X книг, а на второй Y, то выбрать книгу из первой *или* второй полки, можно X+Y способами.

б) Ученик должен выполнить практическую работу по математике. Ему предложили на выбор 17 тем по алгебре *и* 13 тем по геометрии. Сколькими способами он может выбрать одну тему для практической работы? Решение: X=17, Y=13. По правилу суммы 17+13=30 тем.

Правило произведения

Если имеется X элементов первого множества и Y элементов второго и эти множества не пересекаются, то один элемент из первого *и* один элемент второго множества можно выбрать X·Y способами.

ПРИМЕР 2.

а) Если на первой полке стоит 5 книг, а на второй 10, то выбрать одну книгу с первой полки *и* одну со второй можно 5·10=50 способами.

б) Переплетчик должен переплести 12 различных книг в красный, зеленый и коричневые переплеты. Сколькими способами он может это сделать? Решение: Имеется 12 книг и 3 цвета, значит по правилу произведения возможно 12·3=36 вариантов переплета.

в) Сколько существует пятизначных чисел, которые одинаково читаются слева направо и справа налево? Решение: В таких числах последняя цифра будет такая же, как и первая, а предпоследняя - как и вторая. Третья цифра будет любой. Это можно представить в виде <u>XYZYX</u>, где Y и Z -любые цифры, а X - не ноль. Значит по правилу произведения количество цифр одинаково читающихся как слева направо, так и справа налево равно 9·10·10=900 вариантов.

На основании приведенных правил и с использованием алгебраических выкладок можно вывести основные комбинаторные схемы, которые приводим ниже.

Размещения без повторений

Сколько можно составить телефонных номеров из 6 цифр каждый, так чтобы все цифры были различны?

Это пример задачи на размещение без повторений. Размещаются здесь 10 цифр по 6. А варианты, при которых одинаковые цифры стоят в разном порядке считаются разными.

Пусть *X* - множество, состоящие из *n* элементов. Из этого множества отбирается подмножество из *m* элементов ($m \le n$), при этом важен порядок следования элементов в отбираемом подмножестве. Размещением без повторений из *n* элементов по *m* называется число A_n^m различных способов, которыми можно отобрать и упорядочить *m* элементов из множества X.

Количество всех размещений из *n* элементов по *m* вычисляется по формуле $A_n^m = \dfrac{n!}{(n-m)!}$. Здесь величина *n*! называется «*n*-факториал»

(factorial анг. сомножитель) и равна произведению чисел натурального ряда от 1 до какого либо числа n: $n!=1\cdot2\cdot3\cdot...\cdot n$, при этом считается, что $0!=1$.

Согласно приведенной формуле, ответ на вышепоставленную задачу будет $A_{10}^{6}=\frac{10!}{(10-6)!}=151200$.

ПРИМЕР 3.

Сколькими способами 4 юноши могут пригласить четырех из шести девушек на танец? Решение: два юноши не могут одновременно пригласить одну и ту же девушку. И варианты, при которых одни и те же девушки танцуют с разными юношами считаются, разными, поэтому: $A_{6}^{4}=\frac{6!}{(6-4)!}=\frac{720}{2}=360$. Возможно 360 вариантов.

Перестановки.

В случае размещений без повторений, при которых число элементов исходного множества равно числу элементов в отбираемом подмножестве $n=m$ возникает важный частный случай, называемый перестановкой множества X.

Количество всех перестановок из n элементов обозначают $P_{n.}$, и равно $P_n=n!$

Действительно, при $n=m$: $P_n=A_n^m=A_n^n=\frac{n!}{(n-m)!}=\frac{n!}{0!}=n!$

ПРИМЕР 4.

а) Сколько различных шестизначных чисел можно составить из цифр 0, 1, 2, 3, 4,5, если цифры в числе не повторяются? Решение: Найдем количество всех перестановок из этих цифр: $P_6=6!=720$. Однако, число 0 не может стоять впереди числа, поэтому от этого числа необходимо отнять количество перестановок, при котором 0 стоит впереди. А это $P_5=5!=120$. В результате получаем $P_6-P_5=720-120=600$ различных чисел.

б) Вспомним басню Крылова «Квартет»:

«Проказница Мартышка, Осел, Козел, да косолапый Мишка, затеяли играть квартет … Стой, братцы стой! – кричит Мартышка, - погодите! Как музыке идти? Ведь вы не так сидите…». И так, и этак пересаживались герои басни – опять музыка на лад не идет. Вероятно, крыловские музыканты так и не перепробовали всех возможных мест. Однако способов не так уж и много.

Сколько? Здесь идет перестановка из четырех, значит, возможно $P_4=4!=24$ варианта перестановок.

Сочетания без повторений

Сочетанием без повторений называется такое размещение, при котором порядок следования элементов не имеет значения. Всякое подмножество X состоящее из *m* элементов, называется сочетанием из *n* элементов по *m*. Таким образом, количество вариантов при сочетании будет меньше количества размещений. Число сочетаний из *n* элементов по *m* обозначается C_n^m и вычисляется по формуле $C_n^m = \frac{A_n^m}{P_n^m} = \frac{n!}{(n-m)!\,m!}$.

Пример 5.

а) Сколько трехкнопочных комбинаций существует на кодовом замке (все три кнопки нажимаются одновременно), если на нем всего 10 цифр. Решение: Так как кнопки нажимаются одновременно, то выбор этих трех кнопок – сочетание. Отсюда возможно $C_{10}^3 = \frac{10!}{(10-3)!\cdot 3!} = \frac{8\cdot 9\cdot 10}{6} = 120$ вариантов.

б) У одного человека 7 книг по математике, а у второго – 9. Сколькими способами они могут обменять друг у друга две книги на две книги. Решение: Так как надо порядок следования книг не имеет значения, то выбор 2 книг - сочетание. Первый человек может выбрать 2 книги из семи: $C_7^2 = \frac{7!}{(7-2)!\cdot 2!} = \frac{6\cdot 7}{2} = 21$ способами. Второй человек может выбрать 2 книги из девяти: $C_9^2 = \frac{9!}{(9-2)!\cdot 2!} = \frac{8\cdot 9}{2} = 36$. Значит всего по правилу произведения возможно $21\cdot 36=756$ вариантов.

в) При игре в домино 4 игрока делят поровну 28 костей. Сколькими способами они могут это сделать? Первый игрок делает выбор 7 костей из 28 костей. Второй забирает 7 из 28-7=21 костей, третий 7 из 21-7=14 костей, а четвертый игрок забирает оставшиеся кости. Следовательно, возможно $C_{28}^7 \cdot C_{21}^7 \cdot C_{14}^7$ комбинаций.

Размещения и сочетания с повторениями

Часто в задачах по комбинаторике встречаются множества, в которых какие-либо компоненты повторяются. Если в отбираемом подмножестве элементы могут повторяться, то это либо размещения с повторениями (порядок следования элементов важен), либо сочетания с повторениями (порядок не важен). Размещения и сочетания с повторениями обозначаются теми же буквами, но над ними ставится значок ~. Размещения с

повторениями вычисляются по формуле $\tilde{A}_n^m = n^m$, а для сочетания с повторениями – по формуле $\tilde{C}_n^m = \frac{(m+n-1)!}{m!(n-1)!} = C_{m+n-1}^n$.

ПРИМЕР 6.

а) Сколько трехзначных чисел можно составить из цифр 1, 2, 3, 4, 5? Решение: Так как порядок цифр в числе существенен, цифры могут повторяться, то это будут размещения с повторениями из пяти элементов по три, а их число равно $\tilde{A}_5^3 = 5^3 = 125$.

б) В кондитерском магазине продавались 4 сорта пирожных: эклеры, песочные, наполеоны и слоеные. Сколькими способами можно купить 7 пирожных. Решение: Покупка не зависит от того, в каком порядке укладывают купленные пирожные в коробку. Покупки будут различными, если они отличаются количеством купленных пирожных хотя бы одного сорта. Следовательно, количество различных покупок равно числу сочетаний четырех видов пирожных по семь - $\tilde{C}_7^4 = \frac{(7+4-1)!}{7!(4-1)!} = \frac{10!}{7!\cdot 3!} = 120$.

в) Обезьяну посадили за пишущую машинку с 45 клавишами, определить число попыток, необходимых для того, чтобы она наверняка напечатала первую строку романа Л.Н. Толстого «Анна Каренина», если строка содержит 52 знака и повторений попыток не будет? Решение: порядок букв имеет значение. Буквы могут повторяться. Значит, всего есть $\tilde{A}_{45}^{52} = 45^{52}$ вариантов.

Тема 4. АЛГЕБРА ВЫСКАЗЫВАНИЙ

4.1. Общие понятия. Операции над высказываниями (законы логики)

Логика (в переводе с древнегреческого) означает слово, выражающее мысль. Логика - это наука о способах мышления. Математическая логика служит для создания алгоритмов логического вывода.

Особенность математической логики заключается в использовании математического языка символов и формул. Это приводит к устранению двусмысленности, которая свойственна естественным языкам.

Математическая логика имеет дело с высказываниями. Высказывание - это повествовательное предложение, о котором имеет смысл говорить, что оно истинно или ложно, но ни то и другое вместе. Высказывания обозначаются большими буквами латинского алфавита, а их значения, то есть истина и ложь – 1 и 0 соответственно.

Логическая структура сложного высказывания. В процессе рассуждений (не только в математики) из одних высказываний формируются другие сложные высказывания, полученные из элементарных с помощью

частицы «НЕ» или путем соединения элементарных высказываний с помощью союзов «И», «ИЛИ», «ЕСЛИ..,ТО» ТОГДА И ТОЛЬКО ТОГДА, КОГДА» и другие. Эти союзы обозначают определенные логические связи между высказываниями, при этом их можно рассматривать как операции над высказываниями. В алгебре высказываний вводятся операции, которые позволяют судить об истинности или ложности сложного высказывания по истинности и ложности составляющих его высказываний.

1. Отрицание $\overline{A}$ - означает, что высказывание, которое истинно, когда A - ложно и ложно, когда A - истинно. В обычной речи соответствует частице «НЕ».
2. Конъюнкция $A \wedge B$ означает высказывание, которое истинно в том случае, когда A и B оба истинны. Соответствует связки “И”.
3. Дизъюнкция $A \vee B$ означает, что высказывание истинно в том случае, когда истинно одно из высказываний A или B. Соответствует связке «ИЛИ».
4. Импликация $A \rightarrow B$ означает высказывание, которое ложно тогда и только тогда, когда A - истинно, а B - ложно. Соответствует связки «ЕСЛИ..,ТО…».
5. Эквивалентность $A \leftrightarrow B$ означает высказывание, которое истинно тогда и только тогда, когда A и B оба истинны или оба ложны. Соответствует связке «ТОГДА И ТОЛЬКО ТОГДА, КОГДА».

Для проведения доказательств в математической логики используются таблицы истинности.

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$	$\overline{A}$
0	0	0	0	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	0	0
1	1	1	1	1	1	0

4.2. Формулы логики высказываний

Понятие формулы

Задача математической логики дать принцип рассуждения, позволяющий механическим путем решать вопрос: можно ли некоторую цепочку рассуждений считать правильной? При этом важна только форма высказывания составляющих эту цепочку, но не их содержание и смысл.

Формальный язык, который описывает логику высказываний, является поэтому чисто синтаксическим и он не требует, чтобы формулы логики высказываний имели какую-то семантику (смысл).

Алфавитом логики высказываний называется любое непустое множество, содержащее следующие символы: логические переменные x_1, x_2,..,x_n; логические связки $\wedge,\vee,-,\rightarrow,\leftrightarrow$, а также символы скобок (,).

Словом в любом алфавите называется любая последовательность символов (возможно пустая). Слово в алфавите логики высказываний называется формулой, если оно удовлетворяет следующему индуктивному определению:

1. Любая логическая переменная – формула.
2. Если A и B – формулы, то $\overline{A}$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ – тоже формулы.
3. Никаких других формул, кроме тех, которые указанны в п.1 и п.2, нет.

Иногда понятие формулы расширяется за счет введения в них важных логических операций:

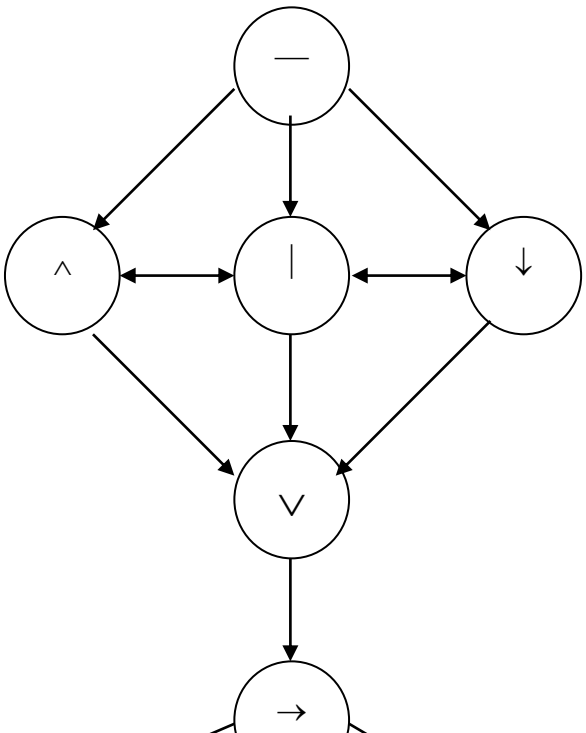
$A|B$ – штрих Шеффера (антиконъюнкция), т. е. $A|B = \overline{A \wedge B}$.

$A\downarrow B$ – стрелка Пирса (антидизъюнкция), т. е. $A \downarrow B = \overline{A \vee B}$.

$A\oplus B$ – “кольцевая сумма” (“логическое сложение”, “сложение по модулю 2”) – $A \oplus B = \overline{A \leftrightarrow B}$.

Для упрощения записи в формулах логики высказываний приняты следующие соглашения относительно расстановки скобок:

1. Внешние скобки не пишутся.
2. На множестве операций логики высказываний вводятся отношения: транзитивное - “быть более сильным” и эквивалентности - “быть равносильным”. Они представлены на рисунке, причем “более сильные” находятся выше, а ‘равносильные” – на одном уровне.



Равносильность формул

Пусть A и B - две формулы и $\{x_1, x_2,..,x_n\}$ – множество всех логических переменных, входящих в формулу A или, и B эти формулы называются <u>равносильными</u>, если при любом распределении истинности значений переменных $x_1, x_2,..,x_n$ они принимают одинаковые значения. Равносильность формул логики высказываний обозначается так $A \equiv B$.

Различие знаков $\leftrightarrow$ и $\equiv$ состоит в том, что символ $\leftrightarrow$ является символом формального языка, а символ $\equiv$ обозначает отношение на множестве формул.

Отношение равносильности, есть отношение эквивалентности, т. к. оно рефлексивно, т. е. $A \equiv A$, симметрично $A \equiv B \Rightarrow B \equiv A$ и транзитивно $A \equiv B$, $B \equiv C \Rightarrow A \equiv C$.

Для любых формул A, B и C справедливы следующие равносильности:

1. $A \wedge B \equiv B \wedge A$
2. $A \wedge A \equiv A$
3. $A \wedge (B \wedge C) \equiv (\mathrm{A} \wedge \mathrm{B}) \wedge C$
4. $A \vee B \equiv B \vee A$
5. $A \vee A \equiv A$
6. $A \vee (B \vee C) \equiv (A \vee B) \vee C$
7. $A \vee (B \wedge C) \equiv (A \vee \mathrm{B}) \wedge (A \vee C)$
8. $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
9. $A \wedge (A \vee B) \equiv A$
10. $A \vee (A \wedge B) \equiv A$
11. $\overline{\overline{A}} \equiv A$
12. $\overline{A \vee B} \equiv \overline{A} \wedge \overline{B}$
13. $\overline{A \wedge B} \equiv \overline{A} \vee \overline{B}$
14. $A \equiv (A \wedge B) \vee (A \wedge \overline{B})$
15. $A \equiv (A \vee B) \wedge (A \vee \overline{B})$
16. $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$
17. $A \rightarrow B \equiv \overline{A} \vee B \equiv \overline{A \wedge \overline{B}}$
18. $A \vee B \equiv \overline{A} \rightarrow B \equiv \overline{\overline{A} \wedge \overline{B}}$

19. $A \wedge B \equiv \overline{A \to \overline{B}} \equiv \overline{\overline{A} \vee \overline{B}}$

Тождественно – истинные формулы

Формула, которая истинна независимо от того, какие значения принимают встречающиеся в ней высказывательные (логические) переменные называется тавтологией (или тождественно - истинной формулой).

Формула является тавтологией тогда и только тогда, когда соответствующая истинная функция принимает значение 1,

т. е. в ее таблице истинности столбец стоящий под этой формулой состоит только из единиц.

Формула называется выполнимой, если на некотором наборе распределения истинностных значений переменных она принимает значение 1.

Формула называется тождественно-ложной или противоречием, если она ложна независимо от того, какие значения принимают встречающиеся в ней высказывательные переменные, т. е. в таблице истинности, столбец стоящий под этой формулой состоит полностью из 0.

Формула называется опровержимой, если при некотором распределении истинных значений переменных она принимает значение 0.

Следствия из этих определений:

А тавтология тогда и только тогда, когда А является неопровержимым;

А тождественно – ложное тогда и только тогда, когда А является выполнимой;

А тавтология тогда и только тогда, когда $\overline{A}$ тождественно – ложное;

А тождественно – ложное тогда и только тогда, когда $\overline{A}$ – тавтология.

Наиболее важными тавтологиями являются (A, B, C – произвольные формулы):

1. $A \vee \overline{A}$;
2. $A \to A$;
3. $A \to (B \to A)$;
4. $(A \to B) \to ((A \to B) \to (A \to C))$;
5. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$;
6. $(A \wedge B) \to A$; $(A \wedge B) \to B$;
7. $A \to (B \to (A \wedge B))$;
8. $A \to (A \vee B)$; $B \to (A \vee B)$;
9. $(\overline{B} \to \overline{A}) \to ((\overline{B} \to A) \to B)$;
10. $((A \to B) \to A) \to A$.

4.3. Функции алгебры логики

Абстрактное определение булевой алгебры

Алгебра $<A; +, \cdot, \bar{\ }>$ называется алгеброй Буля, Если для любых $a, b, c \in A$ ее операции удовлетворяют аксиомам:

1. коммутативности

1.1. $a + b = b + a$;

1.2 $ab = ba$;

2. ассоциативности

2.1. $a + (b + c) = (a + b) + c$ –;

2.2 $a\,(bc) = (ab)\,c$;

3. дистрибутивности

3.1. $a\,(b + c) = ab + ac$;

3.2 $a + (bc) = (a + b)\,(a + c)$;

4. идемпотентности

4.1. $a + a = a$;

4.2 $aa = a$;

5. инволюции $\bar{\bar{a}} = a$;

6. поглощения

6.1. $a + (ab) = a$;

6.2. $a\,(a + b) = a$;

7. де Моргана

7.1. $\overline{a + b} = \bar{a}\ \bar{b}$;

7.2. $\overline{ab} = \bar{a} \vee \bar{b}$;

8. нейтральности

8.1. $(a + a)\,b = b$;

8.2. $a\ \bar{a} + b = b$.

Если обозначить элемент $0 = a\ \bar{a}$, а $1 = a + \bar{a}$, тогда выполняются равенства:

$$0 = 1;\quad \bar{1} = 0;\quad a + 1 = 1;\quad a \cdot 1 = a;\quad a + 0 = a;\quad a \cdot 0 = 0.$$

Булева алгебра называется <u>вырожденной</u>, если 0 и 1 совпадают. В таком случае $a = a \cdot 1 = a \cdot 0 = 0$, следовательно, она не содержит никаких других элементов, а, значит, состоит ровно из одного элемента. Всякая невырожденная булева алгебра содержит два нейтральных элемента 0 (нулевой элемент) и 1 (единичный элемент).

В определении булевой алгебры нет указания на существование такой алгебры. Рассмотрим конкретные примеры:

1. <u>Двоичная модель.</u>

Это самая простая модель булевой алгебры, которая содержит только два элемента 0 и 1, а ее операции вводятся с помощью таблиц значений.

+	0	1
0	0	1
1	1	1

·	0	1
0	0	0
1	0	1

	$\bar{\ }$
0	1
1	0

Данная модель является наиболее важной для компьютерных приложений.

2. Модель исчисления высказываний.

A – множество высказываний с логическими операциями конъюнкции, дизъюнкции и отрицания. Обозначая противоречие 0, а тавтологию – 1, можно проверить, что данное множество с указанными операциями является булевой алгеброй.

3. Теоретико–множественная модель.

Пусть A – непустое множество, тогда множество-степень $P(A)$, является булевой алгеброй, элементами которой служат различные подмножества множества A. При этом операции + булевой алгебры соответствует объединение множеств, операции · – пересечение множеств, а $\bar{\ }$ – дополнение A. При такой интерпретации булевой алгебры выполняются все аксиомы и при этом множества A и $\varnothing$ являются соответственно единичным и нулевым элементами алгебры.

Булевы функции

Рассмотрим еще одну модель булевой алгебры.

Пусть M – произвольная булева алгебра с базисными операциями +, ·, $\bar{\ }$. Рассмотрим множества n – местных функций : f: $M^n \to M$, т. е. $(x_1,x_2,..,x_n) \to f(x_1,x_2,..,x_n)$.

Тогда по определению:

$f_1 + f_2$: $(x_1,x_2,..,x_n) \to f_1(x_1,x_2,..,x_n) + f_2(x_1,x_2,..,x_n)$;

$f_1 \cdot f_2$: $(x_1,x_2,..,x_n) \to f_1(x_1,x_2,..,x_n) \cdot f_2(x_1,x_2,..,x_n)$;

$\bar{f} : (x_1, x_2, ..., x_n) \to \overline{f(x_1, x_2, ..., x_n)}$.

Множество, таким образом определенных функций, вместе с введенными операциями, является булевой алгеброй и оно называется булевыми функциями.

Две постоянные функции:

0: $(x_1,x_2,..,x_n) \to 0$;

1: $(x_1,x_2,..,x_n) \to 1$.

Они являются соответственно нулевым и единичным нейтральными элементами.

Если булева алгебра M двухэлементна, т. е. содержит два элемента 1 и 0, то булевы функции называются двоичными функциями.

Если в двухэлементной булевой алгебре элементы 1 и 0 интерпретируются как “включено” и “выключено” соответственно, то двоичные функции называются переключательными функциями.

Любую логическую функцию можно задать таблицей истинности, в левой части которой выписаны все возможные наборы значений ее аргументов, т. е. x_1, x_2, .., x_n, а правая часть представляет собой столбец значений функций соответствующий этим наборам.

Число всех возможных различных наборов значений n - переменных логической функции $f(x_1,x_2,..,x_n) = 2^n$ (равно числу всех возможных двоичных векторов) длины n. Число всех различных функций n – переменных равно числу возможных расстановок 0 и 1 в столбце с 2^n сторонами, т. е. $|P_2(n)| = 2^{2^n}$.

Особую роль в алгебре логике играют логические функции одной и двух переменных – унарные и бинарные логичекие операции. Эти функции интерпретируются естественными логическими связками “НЕ”,”И”,”ИЛИ” и т. д. Они широко используются при описании систем, явлений, формализации рассуждений и т. п.

Множества всех логических функций одной переменной $P_2(1)$ - унарных логических операций представляют таблицу истинности $|P_2(1)| = 2^2 = 4$,

x	φ_0	φ_1	φ_2	φ_3
0	0	0	1	1
1	0	1	0	1
Обозначение функции	0	x	$\overline{x}$	1

где $\varphi_0(x), \varphi_3(x)$ – константы 0 и 1 соответственно.

Эти функции не зависят от переменной x и, поэтому, x в этих функциях называется фиктивной (несущественной) переменной для этих функций.

$\varphi_1(x) = x$ – повторение переменной.

$\varphi_2(x) = \overline{x}$ – отрицание переменной.

Множество всех логических функций двух переменных $P_2(2)$, то есть, бинарных логических операций, представляется таблицей истинности, содержащей $|P_2(2)| = 2^{2^2}$ функций, из которых шесть имеют фиктивные переменные.

x_1	x_2	φ_0	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
		0	$\wedge$	$\overline{\rightarrow}$	x_1	$\overline{\leftarrow}$	x_2	$\oplus$	$\vee$

x_1	x_2	φ_8	φ_9	φ_{10}	φ_{11}	φ_{12}	φ_{13}	φ_{14}	φ_{15}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1
		$\downarrow$	$\leftrightarrow$	$\bar{x}_2$	$\leftarrow$	$\bar{x}_1$	$\rightarrow$	$\mid$	1

$\varphi_1(x_1, x_2)$ – конъюнкция (логическое умножение $x_1 \cdot x_2$);

$\varphi_7(x_1, x_2)$ – дизъюнкция (логическое сложение $x_1 + x_2$).

Логические функции трех переменных и более обычно задаются (наряду с таблицей истинности) формулами, состоящими из символов переменных и знаков унарных и бинарных операций.

Наиболее употребляемыми являются операции $\bar{\ }$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$, $\oplus$, $\mid$, $\downarrow$. Значение любой логической формулы, содержащей знаки этих операций можно вычислить для любого набора значение переменных, используя выше приведенные таблицы истинности.

Таким образом, формула, наряду с таблицей истинности, служит способом задания и вычисления функций. В общем случае формула описывает логическую функцию как суперпозицию других более простых функций.

Эквивалентными или равносильными называются формулы, представляющие одну и ту же функцию. Эквивалентность формул в алгебре логики обозначается знаком =. Стандартный метод установления эквивалентности двух формул состоит в следующем:

1. По каждой формуле восстанавливается таблица истинности;
2. Полученные таблицы сравниваются по каждому набору значений переменных. Стандартный метод требует $2 \cdot 2^n$ вычислений.

4.4. Нормальные формы булевых функций

Разложение булевых функций

При изучении алгебры логики возникает вопрос: любая ли функция алгебры логики может быть выражена в виде формулы? Или другими словами: нельзя ли свести все булевы функции к какому-нибудь меньшему числу “базисных булевых функций”? Ответ на этот вопрос – положительный. Например, все булевы функции можно представить как композицию только трех функций:

1. Двухместной конъюнкции $x_1 \wedge x_2$ (логического умножения $x_1\, x_2$);
2. Двухместной дизъюнкции $x_1 \vee x_2$ (логического сложения $x_1 + x_2$);
3. Одноместная функция отрицания $\bar{x}$.

Для решения поставленного вопроса вводится обозначение $x^{\sigma} = (x \wedge \sigma) \vee (\bar{x} \vee \bar{\sigma})$, где σ - параметр равный 0 или 1.

Очевидно, что: $x^{\sigma} = \begin{cases} \bar{x}, \text{ при } \sigma = 0 \\ x, \text{ при } \sigma = 1 \end{cases}$.

Выражение x^{σ} называется <u>литерой</u>. Литеры x и $\bar{x}$ называется <u>контрарными</u>.

<u>Конъюнктом или элементарной конъюнкцией</u> называется конъюнкция литер.

<u>Дизъюнктом или элементарной дизъюнкцией</u> называется дизъюнкция литер.

Например, формулы $x \vee \bar{y} \vee \bar{z}$ и $x \vee y \vee x \vee \bar{x}$ – дизъюнкты; формула $\bar{x}_1 \wedge x_2 \wedge x_3$ и $x_1 \wedge x_2 \wedge \bar{x}_1$ – конъюнкты.

<u>Дизъюнктивной нормальной формой (ДНФ)</u> называется дизъюнкция конъюнктов.

<u>Конъюнктивной нормальной формой (КНФ)</u> называется конъюнкция дизъюнктов. Например, формула: $x\ \bar{y} \vee y\, z$ – ДНФ, а формула $(x \vee y \vee z)(x \vee z)\, y$ – КНФ;

<u>Теорема:</u> Любая формула алгебры логики эквивалентна некоторой ДНФ. В подтверждении этого рассмотрим следующий алгоритм.

<u>Алгоритм приведения формулы к ДНФ:</u>

1. Выражаем все логические операции, участвующие в построении формулы через дизъюнкции, конъюнкции и отрицания, используя для этого следующие эквивалентности:
   $\varphi \to \psi = \bar{\varphi} \vee \psi$, $\varphi \leftrightarrow \psi = (\bar{\varphi} \vee \psi)(\varphi \vee \bar{\psi})$, а также определения операций $|, \downarrow, \oplus$.
2. Используя законы де Моргана, переносим все отрицания к переменным и сокращаем двойные отрицания по правилу $\bar{\bar{\varphi}} = \varphi$.
3. Используя закон дистрибутивности $\varphi(\psi \vee \chi) = \varphi\psi \vee \varphi\chi$, преобразуем формулу так, чтобы все конъюнкции выполнялись раньше дизъюнкций.

В результате применения п. 1 – 3, получаем ДНФ данной формулы.

<u>Пример</u>. Приведем к ДНФ формулу $\varphi = (x \to y) \downarrow \overline{(y \to z)}$.

Для этого проделаем следующее:

1. Выразим логические операции $\to$, $\downarrow$ через операции $\wedge$, $\vee$, $\bar{\ }$
   $\varphi = (\bar{x} \vee y) \downarrow \overline{(\bar{y} \vee x)} = \overline{(\bar{x} \vee y) \vee \overline{(\bar{y} \vee z)}}$.

2. В полученной формуле перенесем отрицание к переменным и сократим двойные отрицания

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

3. Используя закон дистрибутивности, приводим формулу к ДНФ: $\varphi = (x \wedge \bar{y} \wedge \bar{y}) \vee (x \wedge \bar{y} \wedge z)$.

<u>Теорема.</u> Любая формула эквивалентна некоторой КНФ.

<u>Алгоритм приведения формулы к КНФ:</u>

Этот алгоритм включает в себя п. 1 – 2 приведения формулы к ДНФ, а п. 3 заменяется

3'. Используем закон дистрибутивности

$$\varphi \vee (\psi \wedge \chi) = (\varphi \vee \psi)(\varphi \vee \chi),$$

т. е. преобразуем формулы так, чтобы все дизъюнкции выполнялись раньше, чем конъюнкции.

<u>Пример</u>. Приведем к КНФ формулу

$$\varphi = (x \to y) \wedge ((\bar{y} \to z) \to \bar{x}).$$

Для этого выполним:

1. Преобразуем формулу φ к формуле, не содержащей знака $\to$:

$$\varphi = (\bar{x} \vee y) \wedge ((\bar{\bar{y}} \vee z) \to \bar{x}) = (\bar{x} \vee y) \wedge (\overline{(\bar{\bar{y}} \vee z)} \vee \bar{x})$$

2. В полученной формуле перенесём отрицание к переменным и сократим двойное отрицание, т. е.

$$\varphi = (\bar{x} \vee y) \wedge ((\bar{y} \wedge \bar{z}) \vee \bar{x}).$$

3. По закону дистрибутивности получаем:

$$\varphi = (\bar{x} \vee y) \wedge (\bar{y} \vee \bar{x}) \wedge (\bar{z} \vee \bar{x}) = (\bar{x} \vee y) \wedge (\bar{x} \vee \bar{y}) \wedge (\bar{x} \vee \bar{z})$$

Данная формула является КНФ!

Преобразуем далее эту формулу, используя закон дистрибутивности:

$$\varphi = (\bar{x} \vee (y \wedge \bar{y})) \wedge (\bar{x} \vee \bar{z}) = \bar{x} \wedge (\bar{x} \vee \bar{z})$$

Используя закон поглощения, получим:

$$\varphi = \bar{x}.$$

Эта формула одновременно является ДНФ и КНФ!

Совершенные ДНФ и КНФ

Любая булева функция может иметь бесконечно много представлений в виде ДНФ и КНФ.

Особое место среди них занимают совершенные ДНФ (СДНФ) и совершенные КНФ (СКНФ).

Пусть $(x_1,x_2,..,x_n)$ – набор логических переменных.

$\Delta = (\delta_1,\delta_2,..,\delta_n)$ – набор 0 и 1.

<u>Конституентой нуля</u> набора Δ называется дизъюнкт, он обозначается $K^0(\delta_1,\delta_2,.., \delta_n) = x_1^{1-\delta 1} \vee x_2^{1-\delta 2} \vee x_n^{1-\delta n}$.

<u>Конституентой единицы</u> набора Δ называется конъюнкт и обознается $K^1(\delta_1,\delta_2,..,\delta_n) = x^{\delta 1},x^{\delta 2},..,x^{\delta n}$.

Заметим, что $K^1(\delta_1,\delta_2,..,\delta_n) = 1$, $[K^0(\delta_1,\delta_2,..,\delta_n) = 0]$, тогда и только тогда, когда $x_1 = \delta_1$, $x_2 = \delta_2,...,x_n = \delta_n$.

<u>СДНФ</u> называется дизъюнкция некоторых конституент единицы, среди которых нет одинаковых.

<u>СКНФ</u> называется конъюнкция некоторых конституент нуля, среди которых нет одинаковых.

Таким образом, СДНФ (СКНФ) – это ДНФ (КНФ), содержащие в каждом конъюнкте (дизъюнкте) все переменные x_i из набора $(x_1,x_2,..,x_n)$, которые входят ровно один раз либо как x_i, либо как $\overline{x_i}$.

Например:

Формула $x_1\overline{x}_2x_3$ – конституента единицы $K^1(1,0,1)$; формула $x \vee y \vee \overline{z}$ – конституента нуля $K^0(0,0,1)$.

Формула $x_1\overline{x}_2x_3 \vee \overline{x}_1x_2x_3\,\mathrm{x}_1$ – СДНФ.

Формула $(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y} \vee z) \wedge (\overline{x} \vee y \vee z)$ – СКНФ.

Формула $x_1\overline{x_2}x_3 \vee \overline{x_1}x_2x_3 \vee x_1\overline{x_2}x_3$ – не является СДНФ.

<u>Пример</u>: найти СДНФ и СКНФ функции $f(x,y,z)$ заданной таблицей истинности:

X	0	0	0	0	1	1	1	1
Y	0	0	1	1	0	0	1	1
Z	0	1	0	1	0	1	0	1
f (x,y,z)	1	0	0	1	0	1	0	1

По теореме о функциональной полноте СДНФ имеет вид:

$$\varphi_1 = \overline{x}\,\overline{y}\,\overline{z} \vee \overline{x}\,y\,z \vee x\,\overline{y}\,z \vee x\,y\,z,$$

а СКНФ –

$$\varphi_2 = (x \vee y \vee \overline{z})(x \vee \overline{y} \vee z)(\overline{x} \vee y \vee z)(\overline{x} \vee \overline{y} \vee z).$$

Таким образом, для нахождения СДНФ и СКНФ исходной формулы φ, составляется ее таблица истинности, а затем по ней строится требуемая совершенная нормальная форма.

При этом, имея, например, СДНФ φ_1 для функции *f* можно составить ее таблицу истинности, а по ней найти СКНФ φ_2. Часто такой способ нахождения СДНФ и СКНФ бывает достаточно трудоемким.

Для нахождения СДНФ можно воспользоваться следующим алгоритмом:

1. Сначало формулу приводят к ДНФ.
2. Затем преобразуем ее конъюнкты в конституенты единицы с помощью действий:
   а) если в конъюнкты входят некоторые переменные вместе со своим отрицанием, то этот конъюнкт удаляется из ДНФ;
   б) если в конъюнкт одна и та же литера x^{δ} входит несколько раз, то удаляются все литеры кроме одной;
   в) если в некоторый конъюнкт $x_1^{\delta 1},.., x_k^{\delta k}$ не входит переменная *y*, которая должна входить, то этот конъюнкт заменяется на эквивалентную формулу
   $x_1^{\delta 1},.., x_k^{\delta k} \wedge (y \vee \overline{y})$ и, применяя закон дистрибутивности получаем ДНФ. Если недостающих переменных несколько, то для каждой из них к конъюнкту добавляется формула ($y \vee \overline{y}$);
   г) если в полученной ДНФ имеется несколько одинаковых конституент единицы, то оставляют только одну из них. В результате получается СДНФ.

Например, найти СДНФ для ДНФ $\varphi = x\overline{x} \vee x \vee yzy$.

Данная формула представляет собой ДНФ. После выполнения пунктов 2а) и 2б), получим: $\varphi = x \vee yz$. Добавляя недостающие переменные в конъюнкты, получим:

$$\varphi = (x(y \vee \overline{y})) \vee (yz(x \vee \overline{x})) = xy \vee x\overline{y} \vee xyz \vee \overline{x}yz =$$
$$= xy(z \vee \overline{z}) \vee x\overline{y}(z \vee \overline{z}) \vee xyz \vee \overline{x}yz =$$
$$= xyz \vee xy\overline{z} \vee x\overline{y}z \vee x\overline{y}\overline{z} \vee xyz \vee \overline{x}yz$$

Удаляя дублирующую конституенту единицы, получаем окончательно:

$\varphi = xyz \vee xy\overline{z} \vee x\overline{y}z \vee x\overline{y}\overline{z} \vee \overline{x}yz$.

Алгоритм приведения КНФ к СКНФ аналогичен алгоритму приведения ДНФ к СДНФ.

4.5. Минимизация булевых функций

Проблема минимизации булевых функций

Для всякой булевой функции такой, что $f \neq 0$ существует ДНФ. Это может быть, например, СДНФ. В общем случае функция алгебры логики может быть представлена в виде ДНФ не единственным образом. В связи с этим возникает возможность выбора более предпочтительной ДНФ, в частности, наиболее простой или минимальной ее реализации.

ДНФ называется минимальной (МДНФ), если она содержит по сравнению с другими эквивалентными ей формами минимальное количество букв (при подсчете учитывается каждое вхождение буквы в формулу).

В простейшем случае минимизацию можно осуществить, выписав все ДНФ для этой функции и выбрав среди них минимальную. Однако такой метод очень трудоемок. Существуют более оптимальные способы решения поставленной задачи.

Элементарная конъюнкция φ_k называется импликантой булевой функции f, если $\varphi_k \vee f = f$.Импликанта φ_k называется простой импликантой, если после отбрасывания любой буквы из φ_k получается конъюнкция, не являющаяся импликантой функции f. А дизъюнкция всех простых импликант функции f называется сокращенной ДНФ функции f.

ДНФ называется кратчайшей, если она имеет наименьшее число элементарных конъюнкций среди всех ДНФ, эквивалентных ей. Для любой заданной функции, сокращенная ДНФ является единственной. Однако она может быть избыточной из-за того, что сокращенная ДНФ может содержать лишние импликанты, удаление которых не меняет таблицы истинности. Если из сокращенной ДНФ удалить все лишние импликанты, то полученная ДНФ называется тупиковой. Заметим, что представление функции в тупиковой форме в общем случае неоднозначно. Выбор из всех тупиковых форм формы с наименьшим числом вхождений переменных дает минимальную ДНФ (МДНФ).

Метод Квайна

В данном методе для нахождения МДНФ булевой функции вводится три операции.

1) Операция полнового склеивания

$$\varphi x \vee \varphi \bar{x} = \varphi(x \vee \bar{x}) = \varphi$$

2) Операция неполного склеивания

$$\varphi x \vee \varphi \bar{x} = \varphi(x \vee \bar{x}) \vee \varphi x \vee \varphi \bar{x} = \varphi \vee \varphi x \vee \varphi \bar{x}$$

3) Операция элементарного поглощения

$$\varphi x^{\delta} \vee \varphi = \varphi; \text{ где } \delta \in \{0,1\}$$

<u>Теорема Квайна.</u> Если исходя из СДНФ произвести все возможные операции неполного склеивания, а потом элементарного поглощения, то в результате получается сокращенное ДНФ, т.е. дизъюнкция всех простых импликант.

<u>Пример</u>. Пусть функция $f(x,y,z)$ задана в форме СДНФ

$$\varphi = \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz$$

Тогда произведя в два этапа все возможные операции неполного склеивания, а затем элементарного поглощения имеем:

$\varphi = \bar{x}y(\bar{z} \vee z) \vee y\bar{z}(\bar{x} \vee x) \vee yz(\bar{x} \vee x) \vee xz(\bar{y} \vee y)\, xy(\bar{z} \vee z) \vee \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}z \vee$

$xy\bar{z} \vee xyz = \bar{x}y \vee y\bar{z} \vee yz \vee xzxy \vee \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz = y(\bar{x} \vee x)$

$\bar{x}y \vee y(\bar{z} \vee z) \vee xz \vee \bar{x}y \vee y\bar{z} \vee yz \vee xz \vee \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz = \bar{x}y \vee y \vee xz \vee$

$\bar{x}y \vee y\bar{z} \vee yz \vee xy \vee \bar{x}y\bar{z} \vee \bar{x}yz \vee x\bar{y}z \vee xy\bar{z} \vee xyz = y \vee xz$

Таким образом сокращенная ДНФ функции f задается формулой $y \vee xz$.

На практике при выполнении операций неполного склеивания на каждом этапе можно не писать члены, участвующие в этих операциях, а писать результаты всевозможных полных склеиваний и конъюнкты, не участвующие ни в каком склеивании.

Пусть функция $f(x,y,z)$ задана СДНФ

$$\varphi = \bar{x}\bar{y}\bar{z} \vee \bar{x}\bar{y}z \vee x\bar{y}z \vee xyz$$

Тогда, произведя операции полного склеивания, а затем

элементарного поглощения получим

$$\varphi = \bar{x}\bar{y}(z \vee \bar{z}) \vee \bar{y}z(\bar{x} \vee x) \vee xz(\bar{y} \vee y) = \bar{x}\bar{y} \vee \bar{y}z \vee xz$$

Для получения минимальной ДНФ из сокращенной ДНФ используется матрица Квайна, которая строится следующим образом: В заголовках столбцов таблицы записываются конституенты СДНФ, а в заголовках строк простые импликанты из полученной сокращенной ДНФ. В таблице звездочками отмечаются те пересечения строк и столбцов, для которых конъюнкт, стоящий в заголовке строки, входит в конституенту единицы, являющейся заголовком столбца.

Импликанты	Конституенты единицы			
	$\bar{x}\bar{y}\bar{z}$	$\bar{x}\bar{y}z$	$x\bar{y}z$	xyz
$\bar{x}\bar{y}$	*	*		
$\bar{y}z$		*	*	
xz			*	*

В тупиковую ДНФ выбирается минимальное число простых импликант, дизъюнкция которых сохраняет все конституенты единицы, т.е. находит столбец матрицы Квайна содержит звездочку, стоящую на пересечении со строкой, соответствующей одной из выбранных импликант.

В качестве минимальной ДНФ выбирается тупиковая, имеющая наименьшее число вхождений переменных, т.е.

$$\varphi = \bar{x}\bar{y} \vee xz$$

В силу принципа двойственности для булевых алгебр все приведенные понятия и рассуждения очевидным образом можно преобразовать для нахождения минимальных конъюктивных нормальных форм (МКНФ)

Минимизация с помощью карт Карно

При числе входов, не превышающих четырех, наилучшим методом нахождения МДНФ является составление карт Карно. Рассмотрим данный

метод на примере:

Шаг 1. Составляется таблица истинности заданной формулы. Пусть это будет:

A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

В таблице должны быть представлены все возможные сочетания аргументов и соответствующие значения функции.

Шаг 2. Составляется карта Карно. Она представляет собой таблицу, близкую к таблице истинности, но содержащую переменные, расположенные по двум осям. Переменные должны быть расположены таким образом, чтобы при переходе от каждого квадрата к соседнему менялось бы состояние только одной переменной.

Шаг 3. Отметим на карте овалами группы, содержащие единицы. Три овала определяют логическое выражение

C \ *AB*	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$Q = AB + BC + AC.$$

Замечания.

1. Для получения простых логических выражений следует объединять в группы единицы, расположенные в 2, 4, 8 и т. п. соседних квадратах.
2. Чем крупнее блок, – тем проще логика.
3. Следует иметь в виду, что в карте Карно для образования блоков единиц стыкуются также ее края, например:

		$A\,B$			
		00	01	11	10
$C\,D$	00	0	0	0	0
	01	0	0	0	0
	11	1	0	0	1
	10	1	0	0	1

Выделенные квадраты карты Карно описываются выражением $Q = \overline{B}C$.

1.6. Полнота и замкнутость булевых функций

Полнота

Любая функция алгебры логики может быть выражена в виде формулы через элементарные функции $\overline{x}$, $x_1 \wedge x_2$, $x_1 \vee x_2$. Такими свойствами обладают и некоторые другие системы элементарных функций.

Система функций $\{f_1, f_2, \ldots f_n\}$ из P_2 (множество всех булевых функций) называется (функционально) полной, если любая булева функция может быть записана в виде формулы через функции этой системы.

Примеры:

1) P_2 – множество всех булевых функций – полная система

2) Система $\{\overline{x}, x_1 \wedge x_2, x_1 \vee x_2\}$ – полная система

Заметим, что система $\{0,1\}$ не является полной.

<u>Теорема</u>. Пусть даны две системы функций из P_2. $A = \{f_1, f_2 \ldots\}$ и $B = \{g_1, g_2 \ldots\}$, относительно которых известно, что система функций A полна и каждая ее функция выражается в виде формулы через функции системы B. Тогда система B является полной.

Опираясь на эту теорему можно установить полноту ряда систем и тем самым расширить список примеров полных систем.

3) Система $\{\bar{x}, x_1 \wedge x_2\}$ является полной, т.к. известно, что 2) полна и $x_1 \vee x_2 = \overline{\bar{x}_1 \wedge \bar{x}_2}$.

4) Система $\{\bar{x}, x_1 \vee x_2\}$ является полной, т.к. полна система 2) и, кроме того, $x_1 \wedge x_2 = \overline{\bar{x}_1 \vee \bar{x}_2}$.

5) Система $\{x_1 \mid x_2\}$ является полной, т.к. взяв за систему A систему 3), а за систему B систему 5) и определив $\bar{x} = x \mid x$ и $x_1 \wedge x_2 = \overline{x_1 \mid x_2} = (x_1 \mid x_2) \mid (x_1 \mid x_2)$.

6) Система $\{0, 1, x_1x_2, x_1 \oplus x_2\}$ является полной. Для этого за систему A берется система 3), а за B – система 6).

При этом: $x_1 \oplus 1 = \bar{x}_1$.

$$x_1x_2 = x_1 \wedge x_2$$

Приведенные примеры показывают, что существует целый ряд полных систем. Каждая из них может быть принята за множество элементарных функций. Таким образом, для задания булевых функций можно использовать различные языки формул. Какой из них является более удобным, зависит от характера рассматриваемой задачи.

Полином Жегалкина

Т.к. формула, построенная в системе 6) состоит из констант 0, 1 и функций x_1,x_2 и $x_1 \oplus x_2$, то после раскрытия скобок выражение формулы переходит в полином по модулю 2 (полином Жегалкина).

<u>Теорема</u>. Каждая функция из P_2 может быть выражена при помощи полинома по модулю 2. Т.е. в виде $f(x_1,x_2,\ldots,x_n) = \bigoplus_{(i_1,i_2,\ldots,i_n)} x_{i1} \cdot x_{i2} \cdots x_{in} \oplus c$, где в каждом наборе ($i_1$, i_2,…,i_n) все i_k (k=1,…,n) различны и суммирование ведется по несовпадающему набору значений. Представление булевой функции в виде полинома Жегалкина единственно с точностью до порядка слагаемых.

Полином Жегалкина называется нелинейным, если он содержит произведения переменных, и линейным – если он их не содержат.

Для получения полинома Жегалкина булевой функции используются аксиомы булевой алгебры и равенства, выражающие операции отрицания, $\wedge$, $\vee$ через операции $\oplus$, $\odot$.

$\bar{x} = x \oplus 1$

$x \wedge y = xy$

$x \vee y = x \oplus y \oplus xy$

Можно доказать тождества:

1) $x \oplus y = y \oplus x$; $\quad x \cdot y = y \cdot x$

2) $(x \oplus y) \oplus z = x \oplus (y \oplus z)$; $\quad (x \cdot y) \cdot z = x \cdot (y \cdot z)$

3) $x \oplus x = 0$; $\quad x \cdot x = x$

4) $x\,(y \oplus z) = x\,y \oplus x\,z$

5) $0 \oplus x = x$

6) $0 \cdot x = 0$

7) $1 \cdot x = x$

8) $x \oplus \bar{x} = 1$

9) $x \cdot \bar{x} = 0$

<u>Пример</u>. Определить полином Жегалкина для функции

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

Используя выше приведенные формулы, получим: $f\,(x, y, z) = (\bar{x}\,\bar{y}\,z \oplus \bar{x}yz \oplus \bar{x}\,\bar{y}z\,\bar{x}yz) \vee x\,\bar{y}\,\bar{z} = (\bar{x}\,\bar{y}\,z \oplus \bar{x}yz) \vee x\,\bar{y}\,\bar{z} = \bar{x}\,\bar{y}z \oplus \bar{x}yz \oplus x\,\bar{y}\,\bar{z}) \oplus (\bar{x}\,\bar{y}\,z \oplus \bar{x}yz)\,x\,\bar{y}\,\bar{z} = \bar{x}\,\bar{y}\,z \oplus \bar{x}yz \oplus x\,\bar{y}\,\bar{z} \oplus \bar{x}\,\bar{y}z\,x\,\bar{y}\,\bar{z} \oplus \bar{x}yz\,x\,\bar{y}\,\bar{z} = \bar{x}\,\bar{y}\,z \oplus \bar{x}yz \oplus x\,\bar{y}\,\bar{z} = \bar{x}z\,(y \oplus \bar{y}) \oplus x\,\bar{y}\,\bar{z} = \bar{x}\,\bar{z} \cdot 1 \oplus x\,\bar{y}\,\bar{z} = \bar{x}\,\bar{z} \oplus x\,\bar{y}\,\bar{z} = (x \oplus 1)\,z \oplus x\,(y \oplus 1)\,(z \oplus 1) = xz \oplus z \oplus (xy \oplus x)\,(z \oplus 1) = xz \oplus z \oplus xyz \oplus xz \oplus xy \oplus x = \underline{x \oplus z \oplus xy \oplus xyz}$ - полином Жегалкина

Заметим, что если в эквивалентности $\varphi \vee \psi = \varphi \oplus \psi \oplus \varphi\psi$ формулы φ и ψ являются различными конституентами единицы, то их произведения $\varphi\psi = 0$. Тогда $\varphi \vee \psi = \varphi \oplus \psi$. Следовательно, для получения полинома Жегалкина из совершенной ДНФ можно сразу заменить $\vee$ на $\oplus$.

Замкнутость

Пусть A – некоторое подмножество функций из P_2. Замыканием A называется множество всех булевых функций, представленных в виде формул через функции множества A. Замыкание множества A обозначается $[A]$. Отметим, что замыкание инвариантно относительно операций введения и удаления фиктивных переменных.

<u>Примеры:</u> 1) $A = P_2$, $[A] = P_2$;

2) $A = \{1, x_1 \oplus x_2\}$. Замыканием этого множества будет класс L всех линейных функций, имеющих вид $f(x_1,x_2,\ldots,x_n) = c_0 \oplus c_1x_1 \oplus c_2 x_2 \oplus\ldots\oplus c_n x_n$, где $c_i = 0_i1$, причем (i =0,…,n)

Свойства замыкания:

1) $[A] = A$

2) $[[A]] = [A]$

3) если $A_1 \subset A_2$, то $[A_1] \subset [A_2]$

4) $[A_1 \cup A_2] \supset [A_1] \cup [A_2]$

Класс (множество) A называется (функционально) замкнутым, если $[A] = A$.

Примеры.

1) P_2 – замкнутый класс.

2) Класс $\{1, x_1 \oplus x_2\}$ – не замкнут.

3) L – замкнут, т.к. линейное выражение, составленное из линейных выражений, в свою очередь, является линейным выражением.

В терминах замыкания и замкнутого класса можно определить полноту (эквивалентную исходному определению), а именно: A – полная система, если $[A] = P_2$.

Теорема Поста

Ответ на вопрос о полноте произвольной системы F булевых функций дает теорема Поста. Для ее формулировки вводятся определения классов Поста.

1) P_0 – класс булевых функций, сохраняющих 0, т.е. функций $f(x_1, x_2,\ldots,x_n)$, для которых $f(0,\ldots 0) = 0$.

2) P_1 – класс булевых функций, сохраняющих единицу, т.е. $f(x_1, x_2,\ldots,x_n)$, для которых $f(1,\ldots,1) = 1$

3) S – класс самодвойственных функций

Функция $f^+(x_1,x_2,\ldots,x_n)$ называется двойственной по отношению к функции $f(x_1,x_2,\ldots,x_n)$, если $f^+(x_1,x_2,...,x_n) = \overline{f(\bar{x}_1,\bar{x}_2,...,\bar{x}_n)}$, т.е. функция, получающаяся из исходной при замене значения всех переменных и значений функций на противоположные. В таблице истинности заменяется 0 на 1 и 1 на 0.

Например:

1) $(x \vee y)^+ = x \wedge y$,

2) $(x \wedge y)^+ = x$,

3) $(\bar{x})^+ = \bar{x}$.

Функция $f(x_1, x_2,\ldots,x_n)$ называется самодвойственной, если $f^+(x_1, x_2,\ldots,x_n) = f(x_1, x_2,\ldots,x_n)$.

4) M – класс монотонных функций.

Булева функция $f(x_1, x_2,\ldots,x_n)$ называется монотонной, если для любых двух наборов $(\alpha_1, \alpha_2,\ldots,\alpha_n)$ и $(\beta_1, \beta_2,\ldots,\beta_n)$ из условия $\alpha_i \le \beta_i$ для всех $i = 1,\ldots,n$ следует, что $f(\alpha_1, \alpha_2,\ldots,\alpha_n) \le f(\beta_1, \beta_2,\ldots,\beta_n)$.

5) L – класс линейных функций, т.е. линейных полиномов Жегалкина.

Заметим, что каждый класс Поста замкнут относительно операций замены переменных и суперпозиции, т.е. с помощью этих операций из функций, принадлежащих данному классу можно получить только функции из этого класса.

<u>Теорема Поста</u>. Для того, чтобы система функций F была полной необходимо и достаточно, чтобы она целиком не содержалась ни в одном из пяти замкнутых классов P_0, P_1, S, M, L.

<u>Пример</u> $f(x,y) = x \mid y$.

Определим, к каким классам Поста относится $x \mid y$. Т.к. $f(0,0) = 1$, а $f(1,1) = 0$, то $f(x,y) \notin P_0$ и $f(x,y) \notin P_1$. Т.к. $f(1,0) \ne \overline{f(0,1)}$, то $f(x,y) \notin S$. Т.к. $f(0,0) > f(1,1)$, то $f(x,y) \notin M$. Полином Жегалкина для данной функции имеет вид $1 \oplus xy$, т.е.

эта функция нелинейна, следовательно $f(x,y) \notin L$.

Таким образом, имеем:

ФУНКЦИЯ	КЛАССЫ				
	P_0	P_1	S	M	L
$x\|y$	Нет	Нет	Нет	Нет	Нет

В силу теоремы Поста функция $x|y$ образует полную систему, т.е. с помощью штриха Шеффера можно получить любую булеву функцию.

Система булевых функций называется базисом, если она полна, а удаление любой функции из этой системы делают ее неполной.

Каждый базис содержит не более четырех булевых функций.

Примеры:

Следующие системы булевых функций являются базисами:

1) $\{\wedge, -\}$; $\{\vee, -\}$; $\{\rightarrow, -\}$; $\{|\}$; $\{\downarrow\}$; $\{\leftrightarrow, \vee, 0\}$; $\{\oplus, \wedge, \leftrightarrow\}$.

Широкий набор базисов открывает большие возможности при решении задач минимизации схем устройств дискретного действия, поскольку из базисных схем с помощью суперпозиций можно составить схему, соответствующую любой базисной функции.

Тема 5. ЛОГИКА ПРЕДИКАТОВ

5.1. Понятие предиката

Логика предикатов – логическая система, представляющая развитие алгебры высказываний. В логике предикатов, наряду с высказываниями, рассматриваются высказывательные функции или предикаты.

Предикатом от n переменных называется выражение $P(x_1,\ldots,x_n)$, которое становится высказыванием при подстановке в него вместо переменных $x_1,\ldots,x_n$ их значений из множества $(M_1,\ldots,M_n)$ соответственно.

Элементы множеств $(M_1,\ldots,M_n)$ называются предметами, а переменные.$x_1,\ldots,x_n$ – предметными переменными.

Множество всех упорядоченных наборов предметов длины n, т.е. x_n).

Если число $M = M_1 \times M_2 \times \ldots \times M_n$ называется полем предиката $P(x_1,\ldots,$ предметных переменных равно 0, то предикат является высказыванием.

Предметные переменные обозначаются малыми буквами конца латинского алфавита (эти буквы иногда снабжаются индексами):

$x, y, z, \ldots x_1, x_2, x_3.$

Предметные переменные обозначаются малыми буквами

начала латинского алфавита:

$a, b, c, \ldots a_1, b_1, c_1.$

Предметы обозначаются большими буквами латинского алфавита, с указанием предметных переменных, или без них:

$A(x), B, F(x,y), P(x_1, \ldots, x_n).$

Символами 0, 1 обозначаются ложь и истину соответственно.

<u>ПРИМЕРЫ:</u>

1) Выражение “x – простое число” не является высказыванием, но грамматически имеет форму высказывания. Из этого выражения можно получить высказывание, например, заменив x на число 1. В результате замены получается истинное высказывание. При замене x на число 6 получается ложное высказывание.

Таким образом “x – простое число” можно рассматривать как функцию $P(x)$, зависящую от переменной x. Область определения $P(x)$ – множество чисел, а область значения – высказывания. То есть $P(x)$ является предикатом.

2) Выражение “x больше y” можно рассматривать как предикат, т.е. функцию $Q(x,y)$, зависящую от x и y. Эта функция становится высказыванием после замены x и y на их значения.

К предикатам можно применить операции алгебры высказываний (конъюнкцию, дизъюнкцию, импликацию, эквивалентность, отрицание) и получить новые предикаты.

Если к предикатам “$x=y$” и “$x<y$”, обозначенным $A(x,y)$ и $B(x,y)$ соответственно, применить операцию конъюнкции, то получается новый предикат - $A(x,y) \wedge B(x,y)$

Таким образом, заменяя предметные переменные в предикатах их значениями из некоторого множества предметов, действительно получаем высказывания истинные или ложные.

5.2. Операции квантирования

Помимо операций алгебры высказываний, в логике предикатов есть две собственные операции, которые связаны непосредственно с природой самих предикатов.

Пусть дан предикат $P(x)$, зависящий от одной переменной и определенный на поле M.

1) Выражение $\forall x\ P(x)$ означает высказывание истинное только в том случае, когда предикат $P(x)$ истинен для всех предметов из поля M. Символ $\forall$ называется квантором общности. Выражение $\forall x\ P(x)$ читается так: «для всякого x P(x)».
2) Выражение $\exists x\ P(x)$ означает высказывание истинное только в том случае, когда предикат $P(x)$ истинен хотя бы для одного предмета из поля M. Данное выражение читается: «существует x такой, что $P(x)$». Символ $\exists$ называется квантором существования.

Рассмотрим примеры применения операции квантирования к предикатам. Пусть даны предикаты над полем натуральных чисел:

1) $x^2 = x \cdot x$, тогда $\forall x(x^2=x\cdot x)$ – истинное высказывание;
2) $x+2=7$, тогда $\forall x(x+2=7)$ – ложное высказывание;
3) $\exists x(x+2=7)$ – истинное высказывание;
4) $x+2=x$, тогда $\exists x(x+2=x)$ – ложное высказывание.

Операция квантирования обобщается на случай предикатов от n переменных. Рассмотрим предикат $G(x_1,\ldots,x_n)$ от переменных $x_1,\ldots,x_n$, заданный над полем $M = M_1 \times M_2 \times \ldots \times M_n$. Подставим вместо переменных $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ (исключая переменную x_i) предметы $a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n$ соответствующих множеств $M_1, M_2, \ldots, M_{i-1}, M_{i+1}, \ldots, M_n$. Предикат $G(a_1, a_2, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n)$ зависит от одной переменной x_i. Тогда выражение

$\forall x_i G(a_1, a_2, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n)$ есть высказывание, значение которого истинно, если предикат $G(a_1, a_2, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n)$ истинен для всякого предмета x_i из M_i. Следовательно, выражение $\forall x_i G(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ – является предикатом от (n-1) предметной переменной, не зависящей от x_i.

Аналогичные утверждения справедливы для квантора существования $\exists$.

5.3. Формулы логики предикатов

Из предикатов, используя операции отрицания, конъюнкции, дизъюнкции, импликации, эквивалентности, а также кванторов общности и существования, можно получить более сложные предикаты. Предметные переменные сложного предиката разбиваются на два класса:

1) связанные переменные, то есть те, по отношению к которым, применялись операции квантирования;
2) свободные переменные – все остальные.

Например, предикат $\forall xA(x,y))\vee\exists z(B(z,v))$ имеет переменные: x и z – связанные, а y и v – свободные.

Различают:

1. <u>определенные предикаты</u>, т.е. такие, значение которых - истина или ложь - известны для каждого набора значений свободных предметных переменных;
2. <u>переменные предикаты</u>, для которых их значения не определены.

Переменные предикаты обозначаются большими буквами конца латинского алфавита с указанием предметных переменных или без них

$$X,\ Y,\ Z,\ldots,X_1,\ X_2,\ X_3\ldots,X(x_1,\ldots x_n),\ v(x,y,z)\ldots.$$

В частности, переменный предикат от нуля предметных переменных есть переменное высказывание.

Применяя к переменным предикатам операции ($\wedge,\vee,\rightarrow,\leftarrow,\bar{\ },\forall,\exists$), получаем формулы логики предикатов. Например, выражение:

$(\forall xW(x,y)\vee X)\rightarrow U(z)$ – формула логики предикатов.

Равносильность формул логики предикатов

Рассматривая формулы логики предикатов, можно говорить

о равносильных формулах, т.е. о формулах, принимающих одно и тоже истинностное значение при замене всех свободных переменных предметами и переменных предикатов – определенными предикатами.

Пример. Рассмотрим формулы логики предикатов $\forall xW(x)$ и $\exists xW(x)$ над двумя типами полей:

1) пусть формулы $\forall xW(x)$ и $\exists xW(x)$ даны над полем $M_1=\{a\}$. В качестве значений переменного предиката $W(x)$ возьмем определенные предикаты одной переменной над полем M_1. Таких предикатов с точностью до истинностных значений существует два. Обозначим их как $A(x)$ и $B(x)$.

x	$A(x)$	$B(x)$
a	0	1

Истинностные значения формул $\forall xW(x)$ и $\exists xW(x)$:

$W(x)$	$\forall x W(x)$	$\exists x W(x)$
$A(x)$	0	0
$B(x)$	1	1

Таким образом, формулы $\forall x W(x)$ и $\exists x W(x)$ равносильны над полем M_1.

2) Пусть формулы $\forall x W(x)$ и $\exists x W(x)$ даны над полем $M_2 = \{a, b\}$. В качестве значений переменного предиката $W(x)$ необходимо взять определенные предикаты над полем $M_2 = \{a, b\}$. Таких предикатов существует четыре.

x	$I_1(x)$	$I_2(x)$	$I_3(x)$	$I_4(x)$
a	0	0	1	1
b	0	1	0	1

Истинностная таблица формул $\forall x W(x)$ и $\exists x W(x)$ имеет вид:

$W(x)$	$\forall x W(x)$	$\exists x W(x)$
$I_1(x)$	0	0
$I_2(x)$	0	1
$I_3(x)$	0	1
$I_4(x)$	1	1

Таким образом, формулы $\forall x W(x)$ и $\exists x W(x)$ не равносильны над полем M_2.

Если формулы равносильны над любым полем, то они называются <u>равносильными</u>.

Примеры равносильных формул:

1) $\overline{\forall x W(x)}$ и $\exists x \overline{W(x)}$

2) $\overline{\forall x \overline{W(x)}}$ и $\exists x W(x)$

3) $\overline{\exists x W(x)}$ и $\forall x \overline{W(x)}$

4) $\overline{\exists x \overline{W(x)}}$ и $\forall x W(x)$

Формулы логики предикатов, также как и формулы алгебры высказываний можно разбить на классы равносильных. Среди всех формул логики предикатов можно выделить формулы, истинные над любым полем. Их называют <u>тождественно истинными</u>. Например, формула $\forall x W(x) \to \exists x W(x)$ является тождественно истинной.

6. ОСНОВЫ ТЕОРИИ ГРАФОВ

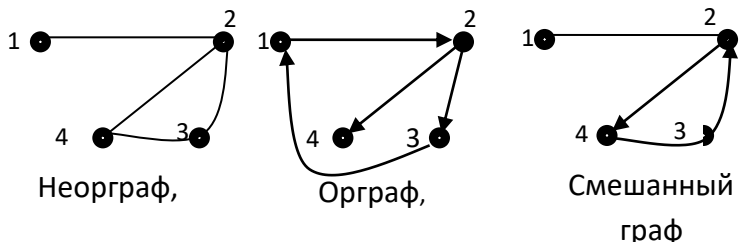
6.1. Основные понятия теории графов

Известно, что множество различных структур в юриспруденции, социальной сфере, теории управления и других науках можно изобразить с помощью абстрактных графических структур, какими, например, являются графики, диаграммы или схемы. Графическое преставление некоторой системы или явления, отображающее структуру связей между ее элементами, называется *графом*. Математиками были разработаны методы исследования свойств и характеристик таких структур, которые получили название *Теории графов*.

Графом $G=(X,U)$ будем называть совокупность двух конечных множеств: множества *вершин* $X=\{x_1,...,x_n\}$ и множества *ребер (дуг)* $U=\{u_1,...,u_m\}$, состоящего из некоторых пар элементов (x_i,x_j) множества X. Геометрически граф может быть представлен в виде рисунка, в котором вершинам соответствуют точки, а ребрам - линии, соединяющие некоторые из этих точек. Граф называется *помеченным*, если его вершинам приписаны некоторые метки, например, номера.

Если пары вершин (x_i, x_j) в множестве U являются неупорядоченными (т.е., порядок соединения вершин несущественен), то граф называется *неориентированным* (***неографом***), а пары (x_i, x_j) - *ребрами* этого графа. При этом вершины x_i, x_j называют *концами* (концевыми вершинами) ребра. В данном случае также говорят, что ребро (x_i, x_j) *соединяет* вершины x_i и x_j. Если пары вершин (x_i, x_j) в множестве U являются упорядоченными (т.е., порядок соединения вершин существенен), то граф называется *ориентированным* (***орграфом***), а пары (x_i, x_j) - *дугами*. Дуги орграфа, в отличие от ребер неографа, будем обозначать $< x_i, x_j >$. При этом x_i - *начало* (начальная вершина) дуги, x_j - *конец* (конечная вершина) дуги. Говорят также, что дуга $< x_i, x_j >$ *исходит* из вершины x_i и *заходит* в вершину x_j. Заметим, что $< x_i, x_j >$ и $< x_j, x_i >$ - это различные дуги в графе. При изображении орграфа дуги обозначают стрелками, указывающими их ориентацию (направление). Графы, в которых имеются и неориентированные ребра, и дуги, называются *смешанными*. Заметим, что любой неориентированный граф может быть представлен в виде орграфа путем замены каждого его ребра двумя противоположно направленными дугами.

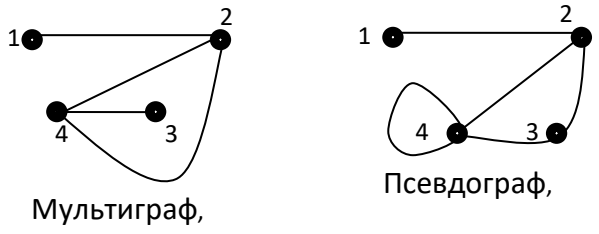
ПРИМЕР 1.



Пара вершин x_i, x_j может соединяться двумя или более ребрами (дугами одного направления). Такие ребра (дуги) называют *кратными*. Количество одинаковых ребер (x_i, x_j) (или дуг $< x_i, x_j >$) называется *кратностью* соответствующего ребра (дуги). *Петлей* называется дуга (ребро), с совпадающими начальной и конечной вершинами.

Граф с петлями и кратными ребрами (дугами) называется *псевдографом*. Граф с кратными ребрами (дугами) и без петель называется *мультиграфом*. Граф, не содержащий петель и кратных ребер, называется *простым* графом.

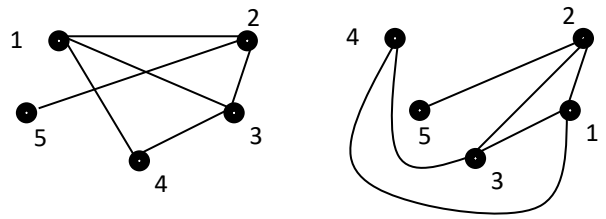
ПРИМЕР 2.



Два графа G и G' называются *изоморфными*, если они имеют одни и те же множества вершин $X = \{x_1, ..., x_n\}$ и множества ребер (дуг) $U = \{u_1, ..., u_m\}$, но изображены по разному.

Граф называется *плоским* или *планарным*, если для него можно найти такой изоморфный граф, ребра которого не пересекаются. Левый граф из примера 3 планарный, так, как его можно изоморфно изобразить без пересечения ребер (правый граф).

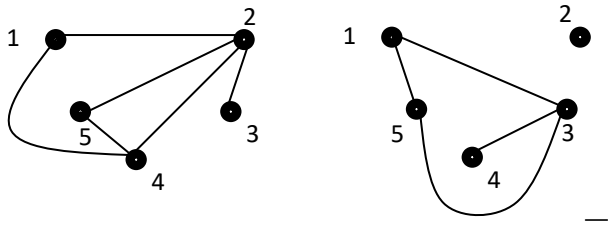
ПРИМЕР 3.



Изоморфные графы

Две вершины графа x_i и x_j называются *смежными*, если существует соединяющее их ребро (дуга). Два ребра (дуги) смежны, если они имеют общую вершину. Если ребро (дуга) u_k соединяет вершины x_i и x_j то говорят, что ребро (дуга) *инцидентно* вершинам x_i и x_j, а вершины x_i и x_j инцидентны ребру (дуге) u_k. Граф $\overline{G} = (X, \overline{U})$ называется *дополнением* простого графа G=(X,U), если множества вершин G и $\overline{G}$ совпадают, а две вершины смежные в $\overline{G}$ тогда и только тогда, когда они не смежные в G.

ПРИМЕР 4.



Граф G Дополнение $\overline{G}$

Число дуг, исходящих из вершины x_i ориентированного графа называется *полустепенью исхода* вершины x_i и обозначается $\rho^-(x_i)$. Число дуг, заходящих в вершину x_i, называется *полустепенью захода* вершины x_i и обозначается $\rho^+(x_i)$. В случае ориентированного псевдографа вклад каждой петли, инцидентной некоторой вершине x_i равен 1 как в $\rho^-(x_i)$, так и в $\rho^+(x_i)$.

В неориентированном графе число ребер, инцидентных данной вершине x_i, называется *степенью* (валентностью) вершины x_i и обозначается $\rho(x_i)$. Вершина графа, имеющая степень 0, называется

изолированной, а вершина, имеющая степень 1 - *висячей*. Для неориентированного псевдографа вклад каждой петли, инцидентной вершине x_i, в $\rho(x_i)$ равен 2. Число вершин нечетной степени в любом графе четно. Для неографа G из примера 3: $\rho(1)=2$; $\rho(2)=4$; $\rho(3)=1$; $\rho(4)=3$; $\rho(5)=2$.

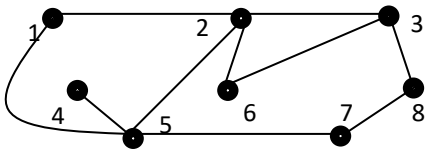
6.2. Маршруты, цепи, циклы на графах

Последовательность ребер неорграфа (x_1, x_2), ..., (x_{r-1}, x_r), в которой конец каждого предыдущего ребра совпадает с началом следующего, называется *маршрутом*, соединяющим вершины x_1 и x_r. Маршрут можно задавать перечислением вершин, в него входящих: $x_1 - x_2 - \ldots - x_{r-1} - x_r$. Аналогом маршрута для орграфа является *ориентированный маршрут* из x_1 в x_r представляющий собой последовательность дуг, в которой конец предыдущей дуги совпадает с началом следующей. При этом x_1 и x_r называются *начальной* и *конечной* вершинами маршрута. Если любые две вершины графа соединены маршрутом, то граф называется *связным*.

Маршрут является *замкнутым*, если начальная вершина совпадает с конечной, и *незамкнутым* в противном случае. Незамкнутый маршрут, в котором все ребра различны, называют *цепью*. Незамкнутый ориентированный маршрут, содержащий попарно различные дуги называется *путем*. Цепь (путь), в которой (в котором) все вершины попарно различны, называется *простой* (*простым*). Замкнутая цепь называется *циклом*, замкнутая простая цепь - *простым циклом*. Замкнутый путь называется *контуром*, замкнутый простой путь - *простым контуром*. Граф, не содержащий циклов, называют *ациклическим*.

Цикл (цепь) в графе G называется *гамильтоновым* (гамильтоновой), если он (она) проходит через каждую вершину графа ровно один раз. Граф G называется гамильтоновым, если он содержит гамильтонов цикл.

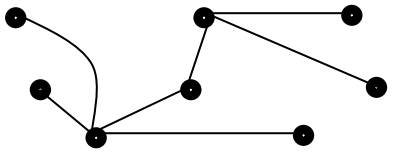
ПРИМЕР 5. Рассмотрим неориентированный граф:



Граф связный, так как можно найти маршрут между любыми двумя вершинами. Примером маршрута будет: 1-5-4-5-2-3-6. Замкнутый маршрут: 2-6-3-8-3-2. Пример цепи: 4-5-2-6-3-2-1. Простая цепь: 1-5-8-7-3. Цикл: 5-7-8-3-2-5. Данный граф является гамильтоновым, т.к. он содержит гамильтонову цепь 4-5-1-2-6-3-8-7, проходящую по одному разу через каждую вершину графа.

Связный ациклический граф, имеющий не менее двух вершин, называется *деревом*. Любое дерево на *n* вершинах содержит *n*-1 ребер и имеет по крайней мере две висячие вершины. Любые две вершины дерева связаны единственной простой цепью. *Ориентированным деревом* называется орграф без циклов, в котором имеется вершина x_0, из которой существует только один ориентированный путь в любую другую вершину.

ПРИМЕР 6. Дерево:

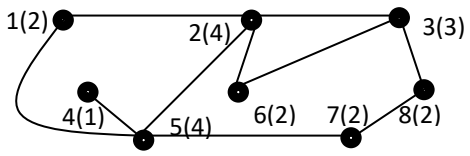


Цикл (цепь) в графе G называется *эйлеровым* (эйлеровой), если он (она) проходит по одному разу через каждое ребро этого графа. Если в графе существует эйлеров цикл, то его можно нарисовать, не отрывая карандаша от бумаги и не повторяя линий. Связный граф называется *эйлеровым*, если он содержит эйлеров цикл, и *полуэйлеровым*, если в нем существует незамкнутая эйлерова цепь. Согласно *теореме Эйлера*, связный граф является эйлеровым, тогда и только тогда, когда степени всех его вершин четны. Если число вершин нечетной степени в связном графе равно 2, то данный граф содержит эйлерову цепь. При этом вершины нечетной степени являются начальной и конечной вершинами цепи.

Наиболее распространенным алгоритмом построения эйлерова цикла является *алгоритм Флери*. Согласно этому алгоритму, необходимо начинать построение цикла с некоторой произвольной вершины графа и каждый раз

вычеркивать пройденное ребро. При выборе нескольких возможных вариантов выбирают тот, из которого можно вернуться в данную вершину. По ребру, в результате вычеркивания которого отсекается часть графа, необходимо проходить только в том случае, когда нет других возможностей.

ПРИМЕР 7. Граф из примера 5 имеет следующие степени вершин (указаны в скобках рядом с номером вершины):



Видно, что имеются две вершины с нечетной степенью. Следовательно граф имеет эйлерову цепь, начинающуюся и заканчивающуюся в вершинах 4 и 3.

Эта цепь, например, есть: 4-5-1-2-5-7-8-3-6-2-3.

6.3. Матричный способ задания графов

Кроме задания графа в виде графического изображения используются иные способы определения неориентированных графов. Наиболее часто используются методы задания графов через матрицы.

Матрицей в математике называется прямоугольная таблица с числами. Числа a_{ij} стоящие на пересечении i-й строки и j-го столбца, называются элементами матрицы. Номера i и j называются индексами элемента. Если число строк матрицы равно числу столбцов и равно n, то матрица называется квадратной порядка n. Элементы матрицы, у которых номер столбца равен номеру строки $i = j$, называются диагональными.

Матрицей смежности неориентированного графа G=(X,U) с n вершинами называется квадратная матрица A порядка n, элементы которой определяются следующим образом:

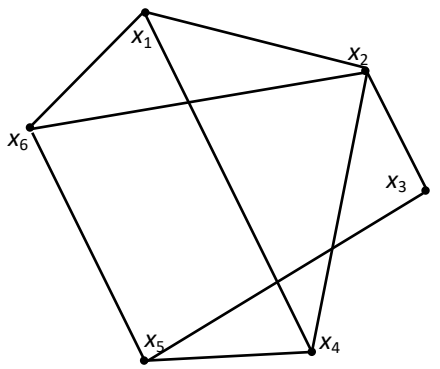
$$a_{ij} = \begin{cases} 1, & \textit{если вершины } x_i \textit{ и } x_j \textit{ соеденены ребром}; \\ 0, & \textit{в противном случае.} \end{cases}$$

Для ориентированного графа:

$$a_{ij} = \begin{cases} 1, & \textit{если из } x_i \textit{ в } x_j \textit{ идет дуга;} \\ 0, & \textit{в противном случае.} \end{cases}$$

В псевдографе $a_{ij} = k$, где k - кратность ребра (x_i, x_j) (дуги $< x_i, x_j >$). Петлям в матрице смежности соответствуют диагональные элементы с равными индексами, расположенные на главной диагонали. Сумма элементов матрицы A неориентированного графа G по i - ой строке (или по i - му столбцу) равна степени вершины $\rho(x_i)$. Для орграфа сумма элементов матрицы A по i - й строке равна $\rho^-(x_i)$, по i - му столбцу $\rho^+(x_i)$. Очевидно, что матрица смежности неографа является симметричной относительно главной диагонали.

ПРИМЕР 8. Для графа, изображенного на рисунке, матрица смежности имеет вид:



$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Матрицей инцидентности неориентированного графа $G=(X,U)$ с n вершинами и m ребрами называется матрица B размера $n \times m$, элементы которой определяются следующим образом:

$$b_{ij} = \begin{cases} 1, & \textit{если вершина } x_i \textit{ инцидентна дуге } u_i; \\ 0, & \textit{в противоположном случае} \end{cases}$$

Для ориентированного графа:

$$b_{ij} = \begin{cases} 1, \text{ если вершина } x_i - \text{начальная вершина дуги } u_j; \\ -1, \text{ если вершина } x_i - \text{конечная вершина дуги } u_j; \\ 0, \text{ если вершина } x_i \text{ не инцидентна дуге } u_j. \end{cases}$$

Для графа на рисунке из примера 8, матрица инцидентности имеет вид:

$$B = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

7. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ АЛГОРИТМОВ

7.1. Алгоритм и его свойства

Слово «Алгоритм» происходит от algorithmi - латинского написания имени аль-Хорезми, под которым в средневековой Европе знали величайшего математика из Хорезма (город в современном Узбекистане) Мухаммеда бен Мусу, жившего в 783-850 гг. В своей книге «Об индийском счете» он сформулировал правила записи натуральных чисел с помощью арабских цифр и правила действий над ними столбиком. В дальнейшем алгоритмом стали называть точное предписание, определяющее последовательность действий, обеспечивающую получение требуемого результата из исходных данных. Алгоритм может быть предназначен для выполнения его человеком или автоматическим устройством. Создание алгоритма, пусть даже самого простого, - процесс творческий. Он доступен исключительно живым существам, а долгое время считалось, что только человеку. Другое дело - реализация уже имеющегося алгоритма. Ее можно поручить субъекту или объекту, который не обязан вникать в существо дела, а возможно, и не способен его понять. Такой субъект или объект принято называть формальным исполнителем. Примером формального исполнителя может служить стиральная машина-автомат, которая неукоснительно исполняет предписанные ей действия, даже если вы забыли положить в нее порошок. Человек тоже может выступать в роли формального исполнителя, но в первую очередь формальными исполнителями являются различные автоматические устройства, и компьютер в том числе. Каждый алгоритм создается в расчете на вполне конкретного исполнителя. Те действия, которые может совершать исполнитель, называются его допустимыми действиями. Совокупность допустимых действий образует систему команд

исполнителя. Алгоритм должен содержать только те действия, которые допустимы для данного исполнителя.

Данное выше определение алгоритма нельзя считать строгим - не вполне ясно, что такое «точное предписание» или «последовательность действий, обеспечивающая получение требуемого результата». Поэтому обычно формулируют несколько общих свойств алгоритмов, позволяющих отличать алгоритмы от других инструкций.

Такими свойствами являются:

Дискретность (прерывность, раздельность) - алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов. Каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего.

Определенность - каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Благодаря этому свойству выполнение алгоритма носит механический характер и не требует никаких дополнительных указаний или сведений о решаемой задаче.

Результативность (конечность) - алгоритм должен приводить к решению задачи за конечное число шагов.

Массовость - алгоритм решения задачи разрабатывается в общем виде, то есть, он должен быть применим для некоторого класса задач, различающихся только исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется областью применимости алгоритма.

Правила выполнения арифметических операций или геометрических построений представляют собой алгоритмы. При этом остается без ответа вопрос, чем же отличается понятие алгоритма от таких понятий, как «метод», «способ», «правило». Можно даже встретить утверждение, что слова «алгоритм», «способ», «правило» выражают одно и то же (т.е. являются синонимами), хотя такое утверждение, очевидно, противоречит “свойствам алгоритма”.

Само выражение «свойства алгоритма» не совсем корректно. Свойствами обладают объективно существующие реальности. Можно говорить, например, о свойствах какого-либо вещества. Алгоритм – искусственная конструкция, которую мы сооружаем для достижения своих целей. Чтобы алгоритм выполнил свое предназначение, его необходимо строить по определенным правилам. Поэтому нужно говорить все же не о свойствах алгоритма, а о правилах построения алгоритма, или о требованиях, предъявляемых к алгоритму.

Первое правило – при построении алгоритма прежде всего необходимо задать множество объектов, с которыми будет работать алгоритм. Формализованное (закодированное) представление этих объектов носит название данных. Алгоритм приступает к работе с некоторым набором данных, которые называются входными, и в результате своей работы выдает

данные, которые называются выходными. Таким образом, алгоритм преобразует входные данные в выходные.

Это правило позволяет сразу отделить алгоритмы от “методов” и “способов”. Пока мы не имеем формализованных входных данных, мы не можем построить алгоритм.

Второе правило – для работы алгоритма требуется память. В памяти размещаются входные данные, с которыми алгоритм начинает работать, промежуточные данные и выходные данные, которые являются результатом работы алгоритма. Память является дискретной, т.е. состоящей из отдельных ячеек. Поименованная ячейка памяти носит название переменной. В теории алгоритмов размеры памяти не ограничиваются, т. е. считается, что мы можем предоставить алгоритму любой необходимый для работы объем памяти.

В школьной «теории алгоритмов» эти два правила не рассматриваются. В то же время практическая работа с алгоритмами (программирование) начинается именно с реализации этих правил. В языках программирования распределение памяти осуществляется декларативными операторами (операторами описания переменных). В языке Бейсик не все переменные описываются, обычно описываются только массивы. Но все равно при запуске программы транслятор языка анализирует все идентификаторы в тексте программы и отводит память под соответствующие переменные.

Третье правило – дискретность. Алгоритм строится из отдельных шагов (действий, операций, команд). Множество шагов, из которых составлен алгоритм, конечно.

Четвертое правило – детерменированность. После каждого шага необходимо указывать, какой шаг выполняется следующим, либо давать команду остановки.

Пятое правило – сходимость (результативность). Алгоритм должен завершать работу после конечного числа шагов. При этом необходимо указать, что считать результатом работы алгоритма.

Итак, алгоритм – неопределяемое понятие теории алгоритмов. Алгоритм каждому определенному набору входных данных ставит в соответствие некоторый набор выходных данных, т. е. вычисляет (реализует) функцию. При рассмотрении конкретных вопросов в теории алгоритмов всегда имеется в виду какая-то конкретная модель алгоритма.

7.2. Виды алгоритмов

Алгоритм применительно к вычислительной машине – точное предписание, т.е. набор операций и правил их чередования, при помощи которого, начиная с некоторых исходных данных, можно решить любую задачу фиксированного типа.

Виды алгоритмов как логико-математических средств отражают указанные компоненты человеческой деятельности и тенденции, а сами алгоритмы в зависимости от цели, начальных условий задачи, путей ее

решения, определения действий исполнителя подразделяются следующим образом:

Механические алгоритмы, или иначе детерминированные, жесткие (например алгоритм работы машины, двигателя и т.п.);

Гибкие алгоритмы, например стохастические, т.е. вероятностные и эвристические.

Механический алгоритм задает определенные действия, обозначая их в единственной и достоверной последовательности, обеспечивая тем самым однозначный требуемый или искомый результат, если выполняются те условия процесса, задачи, для которых разработан алгоритм.

Вероятностный (стохастический) алгоритм дает программу решения задачи несколькими путями или способами, приводящими к вероятному достижению результата.

Эвристический алгоритм (от греческого слова “эврика”) – это такой алгоритм, в котором достижение конечного результата программы действий однозначно не предопределено, так же как не обозначена вся последовательность действий, не выявлены все действия исполнителя. К эвристическим алгоритмам относят, например, инструкции и предписания. В этих алгоритмах используются универсальные логические процедуры и способы принятия решений, основанные на аналогиях, ассоциациях и прошлом опыте решения сложных задач.

Линейный алгоритм – набор команд (указаний), выполняемых последовательно во времени друг за другом.

Разветвляющийся алгоритм – алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.

Циклический алгоритм – алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов.

Цикл программы – последовательность команд (серия, тело цикла), которая может выполняться многократно (для новых исходных данных) до удовлетворения некоторого условия.

Вспомогательный (подчиненный) алгоритм (процедура) – алгоритм, ранее разработанный и целиком используемый при алгоритмизации конкретной задачи. В некоторых случаях при наличии одинаковых последовательностей указаний (команд) для различных данных с целью сокращения записи также выделяют вспомогательный алгоритм.

8.3. Методы изображение алгоритмов

На практике наиболее распространены следующие формы представления алгоритмов:

– словесная (записи на естественном языке);

– графическая (изображения из графических символов);

– псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);

– программная (тексты на языках программирования).

Словесное описание алгоритма

Данный способ получил значительно меньшее распространение из-за его многословности и отсутствия наглядности.

Рассмотрим пример на алгоритме нахождение максимального из двух значений:

Определим форматы переменных X, Y, M, где X и Y – значения для сравнения, M – переменная для хранения максимального значения;

получим два значения чисел X и Y для сравнения;

сравним X и Y.

если X меньше Y, значит большее число Y.

Поместим в переменную M значение Y.

Если X не меньше (больше) Y, значит большее число X.

Поместим в переменную M значение X.

Словесный способ не имеет широкого распространения по следующим причинам:

– такие описания строго не формализуемы;
– страдают многословностью записей;
– допускают неоднозначность толкования отдельных предписаний.

Блок-схема алгоритма

А этот способ оказался очень удобным средством изображения алгоритмов и получил широкое распространение в научной и учебной литературе.

Структурная (блок-, граф-) схема алгоритма – графическое изображение алгоритма в виде схемы связанных между собой с помощью стрелок (линий перехода) блоков – графических символов, каждый из которых соответствует одному шагу алгоритма. Внутри блока дается описание соответствующего действия.

Графическое изображение алгоритма широко используется перед программированием задачи вследствие его наглядности, т.к. зрительное восприятие обычно облегчает процесс написания программы, ее корректировки при возможных ошибках, осмысливание процесса обработки информации.

Можно встретить даже такое утверждение: «Внешне алгоритм представляет собой схему – набор прямоугольников и других символов, внутри которых записывается, что вычисляется, что вводится в машину и что выдается на печать и другие средства отображения информации». Здесь форма представления алгоритма смешивается с самим алгоритмом.

Принцип программирования «сверху вниз» требует, чтобы блок-схема поэтапно конкретизировалась и каждый блок «расписывался» до

элементарных операций. Но такой подход можно осуществить при решении несложных задач. При решении сколько-нибудь серьезной задачи блок-схема «расползется» до такой степени, что ее невозможно будет охватить одним взглядом.

Блок-схемы алгоритмов удобно использовать для объяснения работы уже готового алгоритма, при этом в качестве блоков берутся действительно блоки алгоритма, работа которых не требует пояснений. Блок-схема алгоритма должна служить для упрощения изображения алгоритма, а не для усложнения.

В таблице приведены наиболее часто употребляемые символы.

Название символа	**Обозначение и пример заполнения**	**Пояснение**
Процесс	x=(a-b)/sin(l)	Вычислительное действие или последовательность действий
Решение	да a<b нет	Проверка условий
Модификация	i=1,50,2	Начало цикла
Предопределенный процесс	Расчет параметров	Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод	Ввод a,b,c	Ввод-вывод в общем виде
Пуск-останов	Начало	Начало, конец алгоритма, вход и выход в подпрограмму
Документ	Печать a,c	Вывод результатов на печать

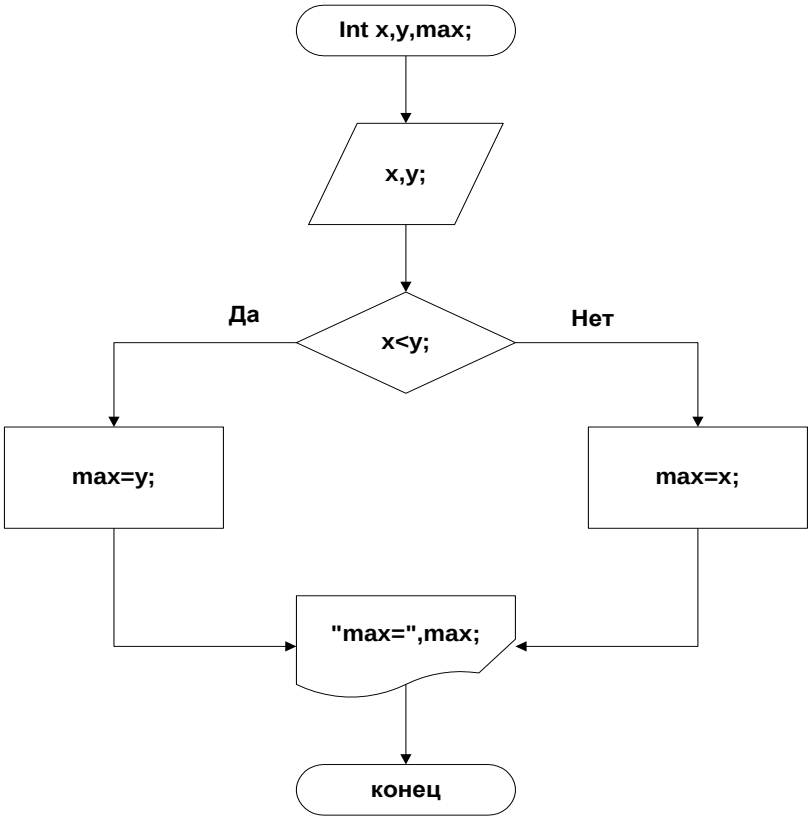
Блок «процесс» применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок «решение» используется для обозначения переходов управления по условию. В каждом блоке «решение» должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок «модификация» используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.

Блок «предопределенный процесс» используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

Например, блок - схемы алгоритма нахождения максимального из двух значений имеет вид:



Псевдокод

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Он занимает промежуточное место между естественным и формальным языками.

С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи.

В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд, рассчитанный на абстрактного исполнителя. Однако в

псевдокоде обычно имеются некоторые конструкции, присущие формальным языкам, что облегчает переход от записи на псевдокоде к записи алгоритма на формальном языке. В частности, в псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда. Они выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются. Единого или формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций.

Примером псевдокода является школьный алгоритмический язык в русской нотации, описанный в учебнике А.Г. Кушниренко и др. «Основы информатики и вычислительной техники».

Пример записи алгоритма на школьном алгоритмическом языке:

алг *Сумма квадратов (арг цел n, рез цел S)*
дано *| n > 0*
надо *| S = 1*1 + 2*2 + 3*3 + ... + n*n*
нач *цел i*
  ввод *n; S:=0*
  нц *для i от 1 до n*
    *S:=S+i*i*
  кц
  вывод *"S = ", S*
кон

Программное представление алгоритма

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенный произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.

Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы — компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на «понятном» ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем.

Следовательно, язык для записи алгоритмов должен быть формализован. Такой язык принято называть языком программирования, а запись алгоритма на этом языке — программой для компьютера.