Н.А. Ююкин С.И. Моисеев Г.Ф. Федотенко

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Учебное пособие

$$F_1, F_2, \ldots, F_n \mapsto G$$

Воронеж 2007

ГОУВПО «Воронежский государственный
технический университет»

Н.А. Ююкин С.И. Моисеев Г.Ф. Федотенко

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

Утверждено Редакционно-издательским советом
университета в качестве учебного пособия

Воронеж 2007

УДК 510(075)

Ююкмн Н.А. Математическая логика и теория алгоритмов: учеб. пособие/ Н.А. Ююкин, С.И. Моисеев, Г.Ф. Федотенко. Воронеж: ГОУВПО «Воронежский государственный технический университет», 2007. 90 с.

В учебном пособии рассматриваются краткие теоретические сведения по теории множеств, бинарным отношениям, алгебраическим системам, булевым функциям, *k* – значной логике, формальным системам, формализации понятия алгоритма и оценки его сложности.

Учебное пособие соответствует требованиям государственного образовательного стандарта высшего профессионального образования по направлению 090100 «Информационная безопасность», специальностям 090102 «Компьютерная безопасность», специальности 090105 «Комплексное обеспечение информационной безопасности автоматизированных систем», дисциплине «Математическая логика и теория алгоритмов»

Библиогр.: 8 назв.

Научный редактор д-р физ.-мат. наук, проф. И.Л. Батаронов

Рецензенты: кафедра высшей математике Воронежского института МВД России (зав. кафедрой д-р физ.-мат. наук, проф. В.В. Меньших);

д-р физ.-мат. наук, проф. В.Н. Нечаев

ВВЕДЕНИЕ

Данное пособие предназначено для студентов ВГТУ, обучающихся по специальностям:

090102 – Компьютерная безопасность;

090105 – Комплексное обеспечение информационной безопасности автоматизированных систем.

Дисциплина “Математическая логика и теория алгоритмов” обеспечивает приобретение знаний и умений в соответствии с Государственным общеобразовательным стандартом и при этом содействует фундаментализации образования, формированию мировоззрения и развитию логического мышления.

Математическая логика и теория алгоритмов представляет собой раздел математики, в котором изучаются основания математики, логики и общие свойства алгоритмов, включающие: формулы алгебры высказываний; представление булевых функций формулами; критерии полноты систем булевых функций; критерии полноты систем функций k-значной логики; классификацию функций k-значной логики; минимизацию булевых функций; исчисление высказываний и предикатов, основные подходы к формализации понятия алгоритма; понятие о сложности алгоритмов; вычислительные алгоритмы; дедуктивные процедуры вывода в логике первого порядка; принцип резолюций для логики высказываний и логики предикатов; реляционную алгебру и реляционное исчисление; а также вырабатываются практические навыки по использованию вышеприведенных понятий.

Целью курса является формирование у студентов теоретических знаний, практических умений и навыков в области моделирования процессов и явлений в естествознании и технике с возможностью употребления математических символов для выражения количественных и качественных отношений объектов, необходимых для выполнения служебной деятельности в области защиты информации на высоком профессиональном уровне.

Достижению данной цели служат следующие задачи:

- изучить максимально широкий круг понятий математической логики и теории алгоритмов;
- получить навыки решения учебных и практических задач;
- овладеть методами алгоритмизации;
- ознакомиться с основами формальных математических систем (теорий) и автоматическим доказательством теорем;
- выработать навыки постановки и решения информационных задач, моделирования и анализа информации.

Дисциплина “Математическая логика и теория алгоритмов” относится к числу прикладных математических дисциплин. Она основывается на знаниях, приобретенных студентами при изучении дисциплины “Алгебра”. Знания и навыки, полученные при изучении дисциплины “Математическая логика и теория алгоритмов”, используются при изучении общепрофессиональных и специальных дисциплин.

1. КРАТКИЕ СВЕДЕНИЯ О МНОЖЕСТВАХ, ОТНОШЕНИЯХ И АЛГЕБРАИЧЕСКИХ СИСТЕМАХ

1.1. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ

1.1.1. Основные понятия

В математике понятие множества принадлежит к числу первичных, то есть неопределяемых через более простые. Это понятие лишь проясняется, то есть даётся описание его основных свойств.

Множеством называется любая совокупность определенных и различимых между собой объектов нашей интуиции и интеллекта, мыслимая, как единое целое. Эти объекты называются элементами множества. Множества обозначаются большими буквами латинского алфавита, а их элементы – малыми. Запись $x \in X$ означает, что элемент x принадлежит множеству X, в противном случае пишут $x \notin X$.

В этом "определении" совокупность предметов рассматривается, как один общий объект и при этом предметы как бы собираются в один мешок, а дальше работают с этим мешком, как с единым целым, не задумываясь о его содержании. Такой подход известен в биологии, где растения и животные, классифицируются по видам, классам, отрядам и т. д. При этом внимание переносится с отдельных представителей на общие свойства группы, как совокупности. В языке это отражается в словах "компания", "стая", "стадо" и т.д.

В "определении" множества нет никаких ограничений на природу элементов. Это может быть множество студентов второго курса, множество пятен на солнце, множество зелёных яблок, множество звёзд на небе и так далее. Заметим, что в качестве элементов множеств могут быть также множества. Например: с одной стороны, группа студентов – это множество, состоящее из людей, а с другой стороны, эта группа является элементом множества всех групп в институте.

В математике часто используют числовые множества, элементами которого являются числа. Из школьной алгебры из-

вестны числовые множества N – натуральных, Z – целых, Q – рациональных, I – иррациональных, R – действительных чисел.

Геометрически множество действительных чисел изображается точками числовой оси, то есть прямой на которой выбрано: 1) начало отсчёта, 2) положительное направление и 3) единица масштаба.

Между множеством действительных чисел и точками числовой оси существует взаимно однозначное соответствие.

Множества точек X числовой оси называются:

$a \le x \le b$ – отрезком,

$a < x < b$ – интервалом,

$a < x \le b$, $a \le x < b$ – полуинтервалом,

Все указанные множества называются промежутками.

Всякий интервал, содержащий точку a, называется окрестностью точки a.

Если множество содержит конечное число элементов, то оно называется конечным, а в противном случае – бесконечным.

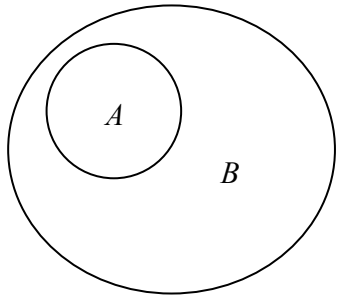
Конечное множество обычно задаётся перечислением его элементов с заключением их в фигурные скобки, то есть

$$X= \{x_1,x_2,,\ldots,x_n\}.$$

Однако перечисление элементов громоздко для описания больших множеств и не применимо для бесконечных множеств. Такие множества задаются с помощью характеристических свойств. Пусть $P(x)$ некоторое предложение, зависящее от x, которое может быть истинным или ложным в зависимости от x, тогда запись $X=\{x|P(x)\}$, означает, что $x \in X$ тогда и только тогда, когда $P(x)$ истинное утверждение.

Например: $A= \{1,2\}=\{x \in N \mid x<3\}$.

Подмножеством A множества B $(A \subset B)$ или (A содержится в B) называется множество A, каждый элемент которого принадлежит B. Графически это изображается с помощью кругов Эйлера, как показано на рисунке.



Множества A и B называются равными ($A=B$), если $A \subseteq B$ и $B \subseteq A$.

Множество, не содержащее ни одного элемента, называется пустым и обозначается $\emptyset$. Совокупность всех допустимых объектов в рамках решаемой задачи называется универсальным множеством и обозначается U.

Совокупность всех подмножеств множества A называется множеством-степенью и обозначается $P(A)$, то есть $P(A)=\{B \mid B \subseteq A\}$.

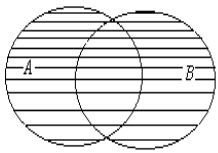
1.1.2. Операции над множествами

Объединение $A \cup B$ есть множество всех элементов принадлежащих A или B. Следует иметь ввиду, что существуют два значения союза или:

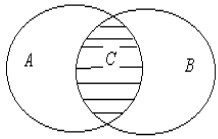
1) «исключающее или» - либо то, либо другое и третьего не дано;

2) «не исключающее или» - то или другое, или то и другое вместе.

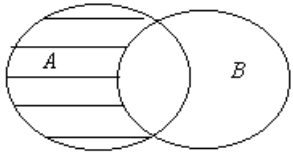
В определении объединения множеств подразумевается второе “не исключающее или”, то есть элемент может принадлежать только A, только B, а также одновременно этим множествам



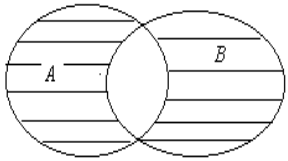
Пересечение множеств C=A$\cap$*B*, есть множество элементов принадлежащих *A* и *B*.



Разность *A**B*, есть множество, состоящее из элементов *A*, не входящих в множество *B*.



Симметричная разность $A\Delta B$



Прямым (декартовым) произведением двух множеств *AxB* называется множество всех упорядоченных пар (*a,b*), в которых первый элемент $a \in A$, а второй $b \in B$.

Степенью множества *A* называется его прямое произведение самого на себя, то есть $A^n = \underbrace{A \times A \times ... \times A}_{n-раз}$.

Дополнением множества *A* называется множество $\bar{A}$, состоящее из элементов универсального множества, не являющихся элементами множеством *A*.

1.1.3. Алгебраические свойства операций над множествами

1. идемпотентность

1.1. $A \cup A = A$ 1.2. $A \cap A = A$

2. коммутативность

2.1. $A \cup B = B \cup A$ 2.2 $A \cap B = B \cap A$

3. ассоциативность

3.1. $A \cup (B \cup C) = (A \cup B) \cup C$ 3.2 $A \cap (B \cap C) = (A \cap B) \cap C$

4 дистрибутивность

4.1. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ 4.2 $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

5. поглощение

5.1. $A \cup (A \cap B) = A$ 5.2. $A \cap (A \cup B) = A$

6. свойства нуля

6.1. $A \cup \varnothing = A$ 6.2. $A \cap \varnothing = A$

7. свойства единицы

7.1. $A \cup U = U$ 7.2. $A \cap U = A$

8. инволютивность

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

9. законы де Моргана

9.1. $\overline{A \cup B} = \overline{A} \cap \overline{B}$ 9.2 $\overline{A \cap B} = \overline{A} \cup \overline{B}$

10. дополнительность

10.1. $A \cup \bar{A} = U$ 10.2 $A \cap \bar{A} = \varnothing$

1.2. ОТНОШЕНИЯ

1.2.1. Понятие отношения

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

n – местным (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 $(x_1, x_2, ..., x_n) \in R$.

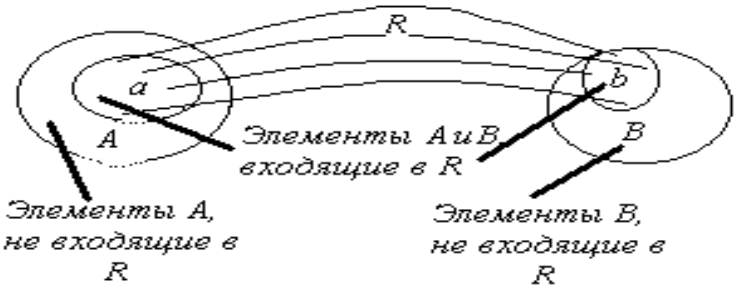
Отношение $R \subseteq A^n$ называется n – местным на множестве A.

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.

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 - $I = \{(a,a) | a \in A\}$.

Принадлежность элементов a и b отношению R наглядно можно представить в следующем виде



Областью определения (*DomR*) 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 (*ImR*) 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 *DomR=A*, в противном случае – частично определенным. Соответствие называется сюръективным, если *ImR=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: $coim_R b$

Заметим, что $DomR = \bigcup_{b \in B} coim_R b$ и $\mathrm{Im}R = \bigcup_{a \in A} im_R a$.

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.

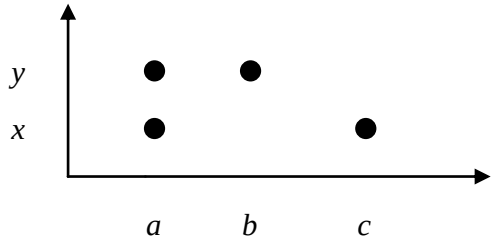
Отношения, определенные на конечных множествах $A = \{a_1, a_2, ..., a_n\}, B = \{b_1, b_2, ..., b_m\}$, могут быть заданы:

1) Списком, т.е. перечислением тех пар элементов, для которых это отношение выполнено. Например, если $A=\{a,b,c\}$ и $B=\{x,y\}$, то $R=\{(a,x),(a,y),(b,y),(c,x)\}$.

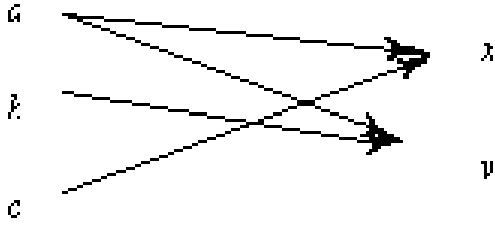
2) Матрицей [R] размерности $m \times n$, элементы которой $r_{ij} = \begin{cases} 1, если (a_i, b_j) \in R, \\ 0, если (a_i, b_j) \notin R, \end{cases}$ т.е. строки этой матрицы помечаются элементами из A, а столбцы – элементами из B, а на пересечении строки a_i со столбцом b_i стоит единица (1), если 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.



1.2.2. Операции над отношениями

Так как отношения из A в B задаются подмножествами $R \subseteq A \times B$, следовательно для них определены те же теоретико-множественные операции, что и над множествами:

1) Объединение $R_1 \cup R_2 = \{(a,b) \mid (a,b) \in R_1 \quad или \quad (a,b) \in R_2\}$.
2) Пересечение $R_1 \cap R_2 = \{(a,b) \mid (a,b) \in R_1 \quad и \quad (a,b) \in R_2\}$.
3) Разность $R_1 \setminus R_2 = \{(a,b) \mid (a,b) \in R_1 \quad и \quad (a,b) \notin R_2\}$.
4) Дополнение $\overline{R} = \{(a,b) \mid (a,b) \notin R\}$.

Заметим, что операции объединения, пересечения и дополнения бинарных отношений удовлетворяют законам идемпотентности, коммутативности, ассоциативности, поглощение, инволюции и законам де Моргана.

Над отношениями могут также осуществляться другими алгебраические операции:

5) Обратное отношение $R^{-1} = \{(a,b) \mid (b,a) \notin R\}$.
6) Произведение (композиция) отношений

$R_1 \circ R_2 = \{(a,b) \mid a \in A$ и $b \in B$ и $\exists c \in C \quad (a,c) \in R_1$ и $(c,b) \in R_2\}$.

7) Степень отношения $R^n = \underbrace{R \times R \times \ldots \times R}_{n-раз}$.

Заметим, что:

1) $[R_1 \cup R_2] = [R_1] + [R_2]$, где сложение элементов матриц осуществляется по правилам 0+0=0, 1+1=1+0=0+1=1.

2) $[R_1 \cap R_2] = [R_1] * [R_2]$, где умножение матриц осуществляется поэлементно с обычными правилами умножения чисел.
3) $[R_1 \circ R_2] = [R_1] \cdot [R_2]$, где умножение матриц производится по обычному правилу умножения матриц.
4) $[R^{-1}] = [R]^T$, где символ T означает транспонирование матрицы.

1.2.3. Свойства отношений на множестве

Пусть задано отношение на множестве A, т.е. $R \subseteq A^2 = A \times A$, тогда отношение R называется:

рефлексивным, если $\forall a \in A (a,a) \in R$
антирефлексивным, если $\forall a \in A (a,a) \notin R$
симметричным, если $\forall (a,b) \in R \Rightarrow (b,a) \in R$
антисимметричным, если $\forall (a,b),(b,a) \in R \Rightarrow a = b$
транзитивным, если $\forall a,b,c \in A \ aRb \text{ и } bRc \Rightarrow aRc$
полным или линейным, если $\forall a,b \in A \ a \neq b \Rightarrow aRb \vee bRa$

Для указанных отношений справедливы следующие утверждения:

R рефлексивно $\Leftrightarrow I \subseteq R$
R антирефлексивно $\Leftrightarrow R \cap I = \varnothing$
R симметрично $\Leftrightarrow R = R^{-1}$
R антисимметрично $\Leftrightarrow R \cap R^{-1} \subseteq I$
R транзитивно $\Leftrightarrow R \circ R \subseteq R$
R полно $\Leftrightarrow R \cup I \cup R^{-1} = U$

Заметим, что:

в матрице рефлексивного отношения все диагональные элементы равны единице, а антирефлексивного – нулю. Для симметричного отношения справедливо $[R]^T = [R]$. В случае антисимметричного отношения матрица $[R \cap R^{-1}] = [R] * [R^{-1}]$ имеет все элементы вне главной диагонали равные нулю. Для транзитивного отношения верно утверждение $[R \circ R] = [R]$.

1.2.4. Отношения эквивалентности, толерантности и порядка

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. Множество $A/E = \{E(x) | x \in A\}$ называется фактор-множеством множества A по отношению к E. Множество A/E является разбиением множества A. Обратно, если $\{A_i\}$ - некоторое разбиение множества A, то можно задать соответствующее ему отношение эквивалентности E по следующему правилу: $xEy \Leftrightarrow x, y \in A_i$ для некоторого i.

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 $R \subseteq A^2$ называется предпорядком или квазипорядком, если R рефлексивно и транзитивно.

Отношение $R \subseteq A^2$ 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 поряд-

ка, не обладающее свойством полноты (линейности), называется отношением частичного порядка.

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

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

1.3. ПОНЯТИЕ АЛГЕБРАИЧЕСКОЙ СИСТЕМЫ

1.3.1. Понятие отображения

Частным видом отношения является отображение (функциональное соответствие).

Отображение – это закон, по которому каждому элементу x некоторого заданного множества X, однозначно соответствует определенный элемент y, другого заданного множества Y. Такое соответствие записывается в виде $y=f(x)$ или $f:x\to y$, при этом говорят, что отображение f действует из X в Y и пишут $f:X\to Y$. Элемент $y=f(x)$ называется образом элемента x, а x называется прообразом элемента y.

Отображение числового множества в числовое называется функцией.

Когда множества X и Y не числовые, отображение называется оператором. Отображение не числового множества в числовое называется функционалом. Отображение $f:X\to X$ называется преобразованием множества X.

Иногда рассматривают отображения f, определённое на некотором подмножестве $D_f \subseteq X$. В этом случае D_f называется областью определения отображения f. Подмножество $Im\ f=\{f(x) \mid x\in X\}$ множества Y называется областью значений (образом)f.

Сужением (ограничением) отображения $f:X\to Y$, на подмножество $A\subset X$, называется отображение $f_A(x)$, заданное равенством $f_A(x)=f(x)$, для всех $x\in A$.

Расширением (продолжением, распространением) отображения $f:X\to Y$ на множестве $B\supset X$ (X – является подмножест-

вом B) называется любое отображение $f_B: B \to Y$, совпадающее с f на множестве X.

Если заданы три множества X,Y,Z и два отображения $f: X \to Y$ и $g: Y \to Z$, то существует отображение $h: X \to Z$, которое определяется равенством $h(x)=g(f(x))$. Это отображение называется композицией (суперпозицией, произведением) отображений и обозначается gf. Композиция отображений обладает свойством ассоциативности, то есть $h(gf)=(hg)f$.

Отображение называется тождественным , если $f(x)=x$, для всех $x \in X$. Отображение $f: X \to Y$ называется инъективным, если для любых двух элементов $x_1, x_2 \in X$, таких что $x_1 \neq x_2$ следует, что $f(x_1) \neq f(x_2)$. Отображение называется сюръективным, если $Imf=Y$. Отображение, одновременно являющееся инъективным и сюръективным называется биективным.

1.3.2. Алгебраическая операция

Отображение f: $A^n \to A$ называется n-местной алгебраической операцией на множестве A. Очевидно, что n-местная алгебраическая операция на множестве A является $(n+1)$ – местным отношением на множестве A. При $n=0$ операция f: $A^0 \to A$ есть $\{(\varnothing, a)\}$ для некоторого $a \in A$. Эта операция называется константой на множестве A и отождествляется с некоторым элементом a этого множества. При $n=1$ операция f называется унарной, а при $n=2$ - бинарной.

Примерами унарных операций являются:

1. Элементарные функции одного аргумента - $e^x, \log x, \sin x$ и другие;
2. Операция над множествами – дополнение $\overline{A}$;
3. Операции над отношениями – дополнение $\overline{R}$, обратное отношение R^{-1}, составное отношение $R^2 = R \circ R$ и другие.

Примерами бинарных операций являются:

1. Арифметические операции – сложение, вычитание, умножение, деление, возведение в степень.

2. Операции над множествами – пересечение, объединение и разность.

3. Операция композиции функций, отображений, отношений.

Обозначим символом **T** произвольную бинарную алгебраическую операцию. Тогда операция над элементами $a,b \in A$, дающая результат $c \in A$ записывается в виде $a\mathbf{T}b = c$.

Свойства бинарных операций:

1. Ассоциативность - $(a\mathbf{T}b)\mathbf{T}c = a\mathbf{T}(b\mathbf{T}c)$. (Выполнение этого условия означает, что скобки в этом выражении можно не расставлять.)

2. Коммутативность - $a\mathbf{T}b = b\mathbf{T}a$.

3. дистрибутивность - $\mathbf{T}(b\perp c)=(a\mathbf{T}b)\perp(a\mathbf{T}c)$ - дистрибутивность операции T слева относительно операции $\perp$ и $(a\perp b)\mathbf{T}c=(a\mathbf{T}c)(b\mathbf{T}c)$ – дистрибутивность операции T справа относительно операции $\perp$

Способы задания операций

Унарные операции задаются:

1. Перечнем всех аргументов $a \in A$ и соответствующих им значений $b \in A$:

$$f = \left| \begin{array}{c} a_1 a_2,...,a_n \\ b_1 b_2,...,b_n \end{array} \right|$$

2. Списком всех пар “аргумент – значение” для всех возможных значений аргументов:

$f = \{(a_1,b_1),(a_2,b_2),...,(a_n,b_n)\}$.

3. Формулой $f(a)=b$, например $\lg a = b$.

Бинарные операции задаются:

1. Таблицей (таблица Кэли). Слева и сверху таблицы выписываются все значения аргументов, а на пересечении строк и столбцов – результат операции. Например, для операции “сложение по модулю 3” на множестве {0,1,2} имеет следующий вид:

$\oplus_3$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

2. Списком, путем перечисления всех троек (a,b,c).

Например, для предыдущей операции:

$\oplus_3 = \{(0,0,0), (0,1,1), (0,2,2), (1,0,1), (1,1,2), (1.2,0), (2,0,2), (2,1,0), (2,2,1)\}$

3. Формулой $f(a,b)=c$ или $a\mathbf{T}b=c$. Например: $a \oplus_3 b = c$, где $\oplus_3$ - операция сложения по модулю 3.

1.3.3. Общие сведения об алгебраических системах

Алгебраическая система – это множество с определенными на нем операциями и отношениями. Алгебраическая система – это объект $\boldsymbol{A} = \langle A, O, R \rangle$, где A - непустое множество, O - семейство алгебраических операций, а R - семейство отношений, заданных на множестве A. Множество A называется носителем алгебраической системы, а его элементы – элементами системы. Алгебраическая система называется конечной, если множество A конечно.

Множества $\Omega = O \bigcup R$ всех главных операций и отношений в A называется сигнатурой в A. Алгебраическая система называется универсальной алгеброй, если множество ее отношений пусто $(O = \varnothing)$, и называется реляционной системой или моделью, если пусто множество основных операций $(R = \varnothing)$.

Пусть $\boldsymbol{A} = \langle A, O, R \rangle$ - алгебраическая система, в которой $O = \{o_i, i \in I\}, R = \{r_j, j \in J\}$, причем o_i есть m_i – арная операция, а r_j есть n_j – местное отношение. Тогда последовательность целых чисел $\langle m_i; n_j \rangle$ называется типом алгебраической системы $\boldsymbol{A}$.

Пусть $\boldsymbol{A} = \langle A, O, R \rangle$ и $\boldsymbol{B} = \langle B, O', R' \rangle$ - две алгебраические

системы одного и того же типа. Отображение $\varphi: A \to B$, называется гомомофизмом системы ***A*** в систему ***B***, если выполняются следующие условия:

1. $\varphi(o_i(x_1, x_2, ..., x_{m_i})) = o'(\varphi(x_1), \varphi(x_2), ..., \varphi(x_{m_i}))$ и

2. $r_j(x_1, x_2, ..., x_{n_j}) \Rightarrow r_j'(\varphi(x_1), \varphi(x_2), ..., \varphi(x_{n_j}))$ для всех $o_i \in O$ и $r_j \in R$.

Если $\varphi: A \to B$ - гомоморфизм, то его обозначают $\varphi: \boldsymbol{A} \to \boldsymbol{B}$.

Гомоморфизм φ: ***A***→***B*** являющийся инъекцией, называется мономорфизмом. Гомоморфизм φ: ***A***→***B***, являющийся сюръекцией, называется эпиморфизмом и при этом система ***B*** называется гомоморфным образом системы ***A***. Гомоморфизм φ: ***A***→ ***A*** называется эндоморфизмом. Сюръективный мономорфизм φ: ***A***→***B***, для которого φ^{-1} – гомоморфизм, называется изоморфизмом и обозначается φ: ***A***↔***B***. Изоморфизм φ: ***A***↔***A*** называется автоморфизмом системы ***A***.

Понятие изоморфизма – одно из важнейших понятий в современной математике. При изоморфизме сохраняются действие всех основных операций и отношений. Это позволяет переносить изучение свойств с одной системы на другую.

1.4. РЕЛЯЦИОННАЯ АЛГЕБРА

1.4.1. Алгебра отношений

Алгебра отношений – это алгебра, носителем которой является множество отношений ***P***={P_1, P_2, ...,P_m, ...}. Сигнатура $\sum$ алгебры отношений состоит из символов двухместных операций объединения $\cup$, пересечения $\cap$, разности \ и декартова произведения × отношений.

Отношения P_i и P_j называются совместимыми, если $P_i, P_j \subseteq A^n$ для некоторого множества A и числа $n \in N$.

Объединением $P_i \cup P_j$ двух совместимых отношений P_i и P_j называется множество всех кортежей X, каждый из которых

принадлежит хотя бы одному из этих отношений: $P_i \cup P_j = \{X | X \in P_i$ или $X \in P_j\}$.

Пересечением $P_j \cap P_i$ двух совместимых отношений P_i и P_j называется множество всех кортежей X, принадлежащих как отношению P_i,, так и отношению P_j: $P_i \cap P_j = \{X | X \in P_i$ и $X \in P_j\}$.

Разностью $P_i \backslash P_j$ двух совместимых отношений P_i и P_j называется множество всех кортежей X, длины n принадлежащих отношению P_i и не принадлежащих P_j: $P_i \backslash P_j = \{X | X \in P_i$ и $X \notin P_j\}$.

Пример. Если $P = \{(a,b,d), (b,c,e)\}$, $Q = \{(a,b,d), (b,d,e)\}$, то $P \cup Q = \{$(a,b,d), $(b,c,e), (b,d,e)\}$, $P \cap Q = \{(a,b.d)\}$, $P \backslash Q = \{(b,c,e)\}$.

Декартовым произведением $P_i \times P_j$ двух отношений P_i и P_j называются множество всех кортежей z таких, что z – конкатенация кортежей $x \in P_i$ и $y \in P_j$: $z = \hat{xy}$, где $\hat{xy} = (x_1,...,x_r, y_1,...,y_s)$, если $x=(x_1,...,x_r)$, $y=(y_1,...,y_s)$. Итак $P_i \times P_j = \{\hat{xy} | x \in P_i, y \in P_j\}$.

Пример Если $P=\{(a,b),(b,c)\}$, $Q=\{(b,c,a),(c,a,a)\}$, то $P \times Q = \{(a,b,b,c,a),(a,b,c,a,a),(b,c,b,c,a),(b,c,c,a,a)\}$.

Алгебры отношений используются для формализации реальных объектов. Рассмотрим, применение алгебры отношений при создании информационного обеспечения – разработке реляционной базы данных. Основой построения реляционной базы данных является двумерная таблица. Каждый i – й столбец таблицы соответствует i – му домену. Если n – местное отношение R_n содержится в $D_1 \times D_2 \times \ldots D_n$, то i – м *доменом* отношения R_n, где $i = 1,\ldots,n$, называется множество D_i.

Строка таблицы соответствует кортежу значений доменов, находящихся в отношении R_n.

Пример. Рассмотрим 4-местное отношение R_4 (расписание экзаменов).

R_4	D_1	D_2	D_3	D_4
1	А – 1	Информатика	10 января	Ауд. 320
2	А - 2	Физика	10 января	Ауд. 324
3	А – 2	Анализ	15 января	Ауд. 324
4	А – 1	Физика	16 января	Ауд. 320
5	А – 1	Анализ	21 января	Ауд. 324
6	А - 2	Информатика	21 января	Ауд. 320

Отношение R_4 является подмножеством декартова произведения $D_1 \times D_2 \times D_3 \times D_4$, и поэтому каждое из множеств D_i является доменом:

- домен D_1 (группа) содержит значения А-1, А-2, то есть $D_1 = \{А – 1, А – 2\}$;
- домен D_2 (дисциплина) - D_2 ={Информатика, Физика, Анализ};
- домен D_3 (дата) – D_3 = {10 янв., 16 янв., 21 янв.};
- домен D_4 (аудитория) – D_4 = {Ауд.320, Ауд. 324}.

Порядок столбцов в таблице фиксирован, а строки в общем случае располагаются произвольно. Цифры первого столбца 1,2,…,6 являются идентификаторами отношения R_4.

Таким образом, каждому отношению можно поставит в соответствие определенную таблицу.

1.4.2. Реляционная алгебра

Для преобразования отношений используется реляционная алгебра. Носитель реляционной алгебры представляет собой множество отношений, а набор операций кроме введенных операций $\cup$, $\cap$, \, $\times$ включает специальные операции над отношениями - выбор, проекцию и соединение.

Операция выбора представляет собой процедуру построения «горизонтального» подмножества отношения, то есть подмножества кортежей, обладающих заданным свойством.

Например с помощью операции выбора из отношения R_4 (расписание экзаменов), строится, например, отношение R_4' (расписание экзаменов по физике). Результатом операции вы-

бора являются строки, в которых домен D_2 представлен значением «Физика»,это 2-я и 4-я строки. В результате получается:

R_4	$_1$	D_2	D_3	D_4
2	А-2	Физика	10 января	Ауд. 324
4	А-1	Физика	16 января	Ауд. 320

Результатом операции проекции отношения $R_n \subseteq A_1 \times A_2 \times \ldots \times An$ на $A_{i1}, A_{i2}, \ldots, A_{im}$, где $\{i_1, \ldots, i_m\} \subseteq \{1,2,\ldots,n\}$ $i_j < i_k$ при $j<k$, называется множество

$\pi^{R_n}_{i_1,i_2,\ldots i_m} = \{(a_{i1}, a_{i2}, \ldots, a_{im}) | (a_1, a_2, \ldots, a_n) \in R_n\}$. Например, $\pi_1^{Rn} = \{a_1 | (a_1, a_2, \ldots, a_n) \in R_n\}$.

Операция проекции определяет построение «вертикального» подмножества отношения, то есть из кортежей удаляются координаты, соответствующие невыделенным доменам.

Например, проекция $\pi_{2,3}^{R4}$ отношения R_4 определяет множество пар, каждая из которых содержит название дисциплины и дату:

$\pi_{2,3}^{R4}$	D_2	D_3
1	Информатика	10 января
2	Физика	10 января
3	Анализ	15 января
4	Физика	16 января
5	Анализ	21 января
6	Информатика	21 января

Операция соединения по двум таблицам, имеющим общий домен, позволяет построить одну таблицу, каждая строка которой! образуется соединением двух строк исходных таблиц. Из заданных таблиц берут строки, содержащие одно и то же значение общего домена; общему домену ставится в соответствие один столбец.

Например, найдем по двум заданным таблицам R_4 и R'_4

R_4	D_1	D_2	D_3	D_4
1	А – 1	Информатика	10 января	Ауд. 320
2	А - 2	Физика	10 января	Ауд. 324

R'_4	D_1	D_2	D_3	D_4
1	А – 2	Анализ	15 января	Ауд. 324
2	А – 1	Физика	16 января	Ауд. 320

Результат соединения по домену D_1

R_7	D_1	D_2	D_3	D_4	D'_2	D'_3	D'_4
1	А – 1	Информат	10 янв	А - 320	Физика	16янв	А - 320
2	А - 2	Физика	10 янв	А - 324	Анализ	15янв	А - 324

В данном примере кортежи соединяются по условию равенства соответствующих координат, соединяющих кортежи, то есть $a = (a_1,..., a_i,..., a_n)$ и $b = (b_1, ...,b_i,..., b_n)$, когда $a_i = b_i$.

В зависимости от практических потребностей, операцию соединения можно определять и по другим правилам.

2. ОСНОВЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ

2.1. АЛГЕБРА ВЫСКАЗЫВАНИЙ

2.1.1. Общие понятия

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

Особенность математической логики состоит в использовании математического языка символов и формул. Это позволяет устранить двусмысленность, свойственную 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. Обычно, 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, так называемые сложные высказывания, полученные из элементарных с помощью частицы «НЕ» или путем соединения элементарных высказываний с помощью связок «И», «ИЛИ», «ЕСЛИ..,ТО» ТОГДА И ТОЛЬКО ТОГДА, КОГДА» и других. Эти связки обозначают определенные логические связи между высказываниями. Их можно рассматривать как некоторые операции над высказываниями. Эти операции позволяют судить об истинности или ложности сложного высказывания по истинности и ложности составляющих его высказываний.

2.1.2. Операции над высказываниями (законы логики)

1. Отрицание $\bar{A}$ - означает, что высказывание, которое истинно, когда A - ложно и ложно, когда A - истинно. В обычной речи оно 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».

Часто для проведения доказательств в математической логики используются таблицы истинности.

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$	$\bar{A}$
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

2.2. ФОРМУЛЫ ЛОГИКИ ВЫСКАЗЫВАНИЙ

2.2.1. 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.

Поэтому формальный язык, который описывает логику высказываний, является чисто синтаксическим. 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:

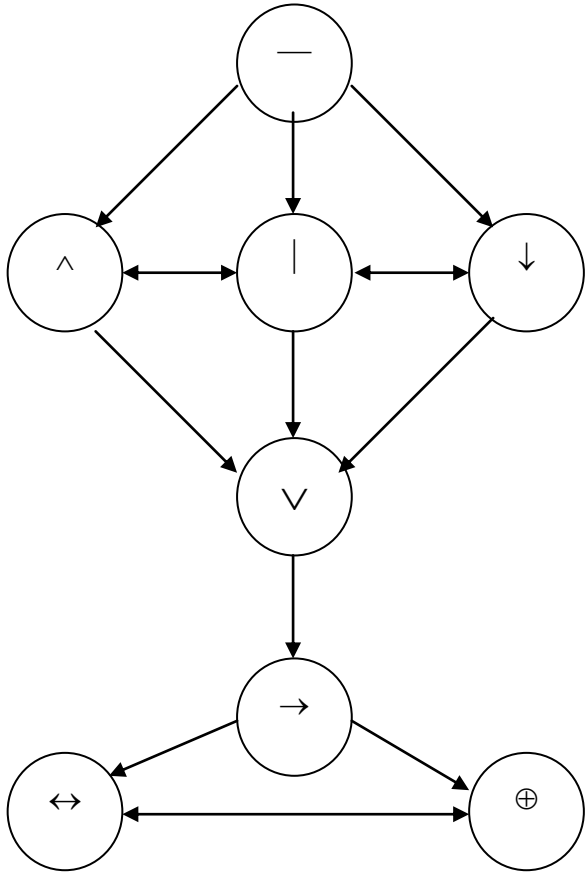
$A|B$ – штрих Шеффера (антиконъюнкция), т. е. $A|B = \overline{A \wedge B}$.

$A\downarrow B$ – 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.



2.2.2. Равносильность формул

Пусть A и B - две формулы и $\{x_1, x_2,..,x_n\}$ – множество всех пропозициональных (высказывательных) переменных, входящих в эти формулы. Формулы называются равносильными, если при любом распределении истинности значений переменных x_1, x_2,..,x_n они принимают одинаковые значения. Равносильность формул логики высказываний обозначается так $A=B$.

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

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

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

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

2.2.3. 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 одних 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:

- A тавтология тогда и только тогда, когда A является неопровержимым;
- A тождественно – ложное тогда и только тогда, когда A является выполнимой;
- A тавтология тогда и только тогда, когда $\overline{A}$ тождественно – ложное;
- A 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. $A \underline{\to} (A \underline{\vee} B)$; $B \underline{\to} (A \vee B)$;
9. $(\overline{B} \to \overline{A}) \to ((\overline{B} \to A) \to B)$;
10. $((A \to B) \to A) \to A$.

2.3. ФУНКЦИИ АЛГЕБРЫ ЛОГИКИ

2.3.1. 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. $(a + \bar{a})\,b = b$;

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

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

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

Булева алгебра называется вырожденной, 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. Для Рассмотрим конкретные примеры:

1. Двоичная модель.

Это самая простая модель булевой алгебры. Она 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

	¯
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 ее аксиомы. Множества A и $\varnothing$ являются соответственно единичным и нулевым элементами данной булевой алгебры.

2.3.2. 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:

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

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

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

Множество, таким образом определенных функций, вместе с введенными операциями, является булевой алгеброй, которое 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 соответствующие 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 функции, соответствующий этим наборам.

Число всех возможных различных наборов значений n - переменных логической функции $f(x_1,x_2,..,x_n) = 2^n$ (число 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 логические операции. Эти функции интерпретируются естественными логическими связками "НЕ","И","ИЛИ" и т. д. Они широко используются для описания систем, явлений, формализации рассуждений и т. п.

Множества всех логических функций одной переменной P_2 (1) - унарных логических операций представляется таблицей истинности содержащей 2^1 строк и $|P_2(1)| = 2^2 = 4$ столбцов,

x	φ_0	φ_1	φ_2	φ_3
0	0	0	1	1
1	0	1	0	1
Обозначение функций	0	x	$\bar{x}$	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.

$\varphi_2(x) = \bar{x}$ – 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
Обозначение функций		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
Обозначение функций		$\downarrow$	$\leftrightarrow$	$\bar{x}_2$	$\leftarrow$	$\bar{x}_1$	$\rightarrow$	$\mid$	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 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 логических 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. Если наборы совпадают, то формулы эквивалентны.

2.4. НОРМАЛЬНЫЕ ФОРМЫ БУЛЕВЫХ ФУНКЦИЙ

2.4.1. Разложение булевых функций

При изучении алгебры логики возникает вопрос: любая ли функция алгебры логики может быть выражена в виде формулы? Или другими словами: можно ли все булевы функции свести к какому-нибудь меньшему числу “элементарных булевых функций”? Ответ на этот вопрос – утвердительный. Например, все булевы функции можно представить в виде композиции 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 литерой. Литеры x и $\bar{x}$ называется контрарными.

Конъюнктом или элементарной конъюнкцией называется конъюнкция литер.

Дизъюнктом или элементарной дизъюнкцией 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.

Дизъюнктивной нормальной формой (ДНФ) называется дизъюнкция конъюнктов.

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

Заметим, что любая формула алгебры логики эквивалентна некоторой ДНФ. В подтверждении этого рассмотрим следующий алгоритм.

Алгоритм приведения формулы к ДНФ заключается в следующем:

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

Пример. 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 $\wedge$, $\vee$, $^-$
   $\varphi = (\overline{x} \vee y) \downarrow \overline{(\overline{y} \vee x)} = \overline{(\overline{x} \vee y) \vee \overline{(\overline{y} \vee z)}}$.
2. В полученной формуле перенесем отрицание к переменным и сократим двойные отрицания
   $\varphi = \overline{(\overline{x} \vee y)} \wedge \overline{\overline{(\overline{y} \vee z)}} = (\overline{\overline{x}} \wedge \overline{y}) \wedge (\overline{y} \vee z) = (x \wedge \overline{y}) \wedge (\overline{y} \vee z)$.
3. Используя закон дистрибутивности, приводим формулу к ДНФ: $\varphi = (x \wedge \overline{y} \wedge \overline{y}) \vee (x \wedge \overline{y} \wedge z)$.

Любая формула эквивалентна также некоторой КНФ.

Алгоритм приведения формулы к КНФ:

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.

Пример. 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 $\to$:

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

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:

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

Полученная в результате формула одновременно является ДНФ и КНФ!

2.4.2. 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.

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

Конституентой единицы 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 $x_1 = \delta_1$, $x_2 = \delta_2,...,x_n = \delta_n$.

СДНФ называется дизъюнкция некоторых конституент единицы, среди которых нет одинаковых.

СКНФ называется конъюнкция некоторых конституент нуля, среди которых нет одинаковых.

Другими словами, 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 $\bar{x}_i$.

Например:

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

Формула $x_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3$ x$_1$ – СДНФ.

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

Для решения задачи нахождения СДНФ и СКНФ исходной формулы φ, предварительно рассмотрим разложения булевой функции $f(x_1,x_2,..,x_n)$ по k переменным $(x_1,x_2,..,x_k)$ – называется разложением Шеннона.

Первая теорема Шеннона.

Любая булева функция $f(x_1,x_2,..,x_n)$ представима в виде 1-го разложения Шеннона:

$$f(x_1,x_2,...,x_n) = \bigvee_{\substack{по\ всем \\ \delta_1,\delta_2,...\delta_n}} x_1^{\delta_1} \wedge x_2^{\delta_2} \wedge ... \wedge x_k^{\delta_k} \wedge$$
$$\wedge\ f(\delta_1,\delta_2,...,\delta_k,x_{k+1},...,x_n).$$

Доказательство:

Заметим, что $x_i^{\delta_i} = 1$, тогда и только тогда, когда $x_i = \delta_i$. Рассмотрим произвольный набор переменных $(\alpha_1,\alpha_2,..,\alpha_n)$ и покажем, что левая и правая части разложения на этом наборе принимают одно и то же значение.

Левая часть дает $f(\alpha_1,\alpha_2,..,\alpha_n)$.

$$\text{Правая} \bigvee_{\substack{по\,всем \\ \delta_1,\delta_2,...,\delta_n}} \alpha_1^{\delta_1} \wedge \alpha_2^{\delta_2} \wedge ... \wedge \alpha_k^{\delta_k} \wedge f(\delta_1,\delta_2,...,\delta_k,\alpha_{k+1},...,\alpha_n) =$$

$$= \alpha_1^{\delta_1} \wedge \alpha_2^{\delta_2} \wedge ... \wedge \alpha_k^{\delta_k} \wedge f(\delta_1,\delta_2,...,\delta_k,\alpha_{k+1},...,\alpha_n) =$$

$$= f(\alpha_1,\alpha_2,...,\alpha_k,\alpha_{k+1},...,\alpha_n)$$

Аналогично рассуждая можно прийти ко второй теореме Шеннона.

Вторая теорема Шеннона.

Любая булева функция $f(x_1,x_2,..,x_n)$ представима в виде 2-го разложения Шеннона:

$$f(x_1,x_2,...,x_n) = \bigwedge_{\substack{по\,всем \\ \delta_1,\delta_2,...,\delta_n}} (x_1^{\delta_1} \vee x_2^{\delta_2} \vee ... \vee x_k^{\delta_k} \vee f(\delta_1,\delta_2,...,\delta_k,x_{k+1},...,x_n))$$

При $k=n$ для булевой функции $f(x_1,x_2,..,x_n) \neq 0$ из первой теоремы Шеннона получается ее представление в виде СДНФ:

$$f(x_1,x_2,...,x_n) = \bigvee_{\substack{по\,всем \\ f(\delta_1,\delta_2,...\delta_n)=1}} (x_1^{\delta_1} \wedge x_2^{\delta_2} \wedge ... \wedge x_n^{\delta_n}).$$

Аналогично для булевой функции $f(x_1,x_2,..,x_n)\neq 1$ получается ее представление в виде СКНФ:

$$f(x_1,x_2,...,x_n) = \bigwedge_{\substack{по\,всем \\ f(\delta_1,\delta_2,...\delta_n)=0}} (x_1^{\delta_1} \vee x_2^{\delta_2} \vee ... \vee x_n^{\delta_n}).$$

Приведённые формулы позволяют сформулировать теорему о функциональной полноте:

Для любой булевой функции f найдется формула φ, представляющая функцию f.

1) Если $f \not\equiv 0$, то существует представление ее формулой φ, находящейся в СДНФ:

$$f(x_1,x_2,...,x_n) = \bigvee_{\substack{по\,всем \\ f(\delta_1,\delta_2,...\delta_n)=1}} K^1(\delta_1,\delta_2,...,\delta_n).$$

Такое представление единственно с точностью до порядка следования конституент единицы.

2) Если $f \not\equiv 1$, то существует представление ее функции φ, находящейся в СКНФ, а именно:

$$f(x_1,x_2,...,x_n) = \bigwedge_{\substack{по\,всем \\ f(\delta_1,\delta_2,...\delta_n)=1}} K^0(\delta_1,\delta_2,...,\delta_n)$$

и такое представление единственно с точностью до порядка следования конституент нуля.

Пример. 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:

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

а СКНФ – $\varphi_2 = (x \vee y \vee \bar{z})(x \vee \bar{y} \vee z)(\bar{x} \vee y \vee z)(\bar{x} \vee \bar{y} \vee z)$.

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 затем по ней найти СКНФ φ_2. Часто такой способ бывает трудоемким.

Для нахождения СДНФ можно воспользоваться следующим алгоритмом:

1. Сначала формулу приводят к ДНФ.
2. Затем преобразуют 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 $x_1^{\delta 1},.., x_k^{\delta k} \wedge (y \vee \bar{y})$ и, применяя закон дистрибутивности получаем ДНФ, содержащую переменную y. Если недос-

тающих переменных несколько, то для каждой из них к конъюнкту добавляется формула вида ($y \vee \bar{y}$);

г) если в полученной ДНФ имеется несколько одинаковых конституент единицы, то оставляют только одну из них. В результате получается СДНФ.

Пример. Найти СДНФ для ДНФ $\varphi = x\bar{x} \vee x \vee yzy$.

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

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

Удаляя дублирующую конституенту единицы, получаем окончательно:

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

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

2.5. МИНИМИЗАЦИЯ БУЛЕВЫХ ФУНКЦИЙ

2.5.1. Проблема минимизации булевых функций

Для всякой булевой функции такой, что $f \not\equiv 0$ 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 мини-

мальную. Однако такой метод очень трудоемок. Рассмотрим другие 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).

2.5.2. 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

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

Теорема Квайна. 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.

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

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

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

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

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

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

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

$$\vee \bar{x}\bar{y} \vee y\bar{z} \vee yz \vee xy \vee$$

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

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.

Пример. Пусть функция $f(x,y,z)$ задана СДНФ

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

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.

Для последнего примера матрица Квайна имеет вид

Импликанты	Конституенты единицы			
	$\bar{\bar{\bar{x}}}yz$	$\bar{\bar{x}}yz$	$\bar{x}yz$	xyz
$\bar{\bar{x}}y$	*	*		
$\bar{y}z$		*	*	
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. В указанном примере тупиковая и минимальная ДНФ совпадают и имеют вид

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

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

2.5.3. Минимизация с помощью карт Карно

При числе переменных, не превышающих четырех, наилучшим методом нахождения МДНФ является построение карт Карно. Данный метод рассмотрим на примере:

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

x	y	z	f
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 этой таблице должны быть представлены все возможные сочетания аргументов и соответствующие им значения функции.

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

x y / ***z***	00	01	11	10
0	0	0	1	0
1	0	1	1	1

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

$$f = xy \vee yz \vee xz$$

Замечания.

1. Для получения простых логических выражений следует объединять единицы в группы, 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:

		$x\,y$			
		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 $f = \bar{y}z$.

Другой пример карты Карно.

		$x\,y$			
		00	01	11	10
$z\,w$	00	1	0	0	1
	01	0	0	0	0
	11	0	0	0	0
	10	1	0	0	1

Логическое выражение для данной карты Карно - $f = \bar{y}\bar{w}$

2.6. ПОЛНОТА И ЗАМКНУТОСТЬ БУЛЕВЫХ ФУНКЦИЙ

2.6.1. Полнота

Ранее отмечалось, что любая функция алгебры логики может быть выражена в виде формулы через элементарные функции $\bar{x}$, $x_1 \wedge x_2$, $x_1 \vee x_2$. Однако, такими свойствами обладают и 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 фундаментально полных систем:

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.

Теорема. 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 $\{\overline{x}, x_1 \wedge x_2\}$ является полной, т.к. известно, что 2) полна и $x_1 \vee x_2 = \overline{\overline{x_1} \wedge \overline{x_2}}$.

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

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 $\overline{x} = x \mid x$ 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: $x_1 \oplus 1 = \overline{x}_1$, $x_1x_2 = x_1 \wedge x_2$

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.

2.6.2. 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 выра-

жение формулы переходит в полином по модулю 2 (полином Жегалкина).

Теорема. 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 $(i_1, i_2,\ldots,i_n)$ все i_k (k=1,…,n) различны и суммирование ведется по несовпадающему набору значений $c = \{0,1\}$. Представление булевой функции в виде полинома Жегалкина единственно с точностью до порядка следования 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 $\oplus$, $\odot$.

$\overline{x} = x \oplus 1$, $x \wedge y = xy$, $x \vee y = x \oplus y \oplus xy$

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

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

Пример. Определить полином Жегалкина для функции

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

Используя выше приведенные формулы, получим: $f(x, y, z) = (\overline{x}\,\overline{y}\,z \oplus \overline{x}yz \oplus \overline{x}\,\overline{y}z\ \overline{x}yz) \vee x\,\overline{y}\,\overline{z} = (\overline{x}\,\overline{y}\,z \oplus \overline{x}yz) \vee x\,\overline{y}\,\overline{z} = \overline{x}\,\overline{y}z \oplus \overline{x}yz \oplus x\,\overline{y}\,\overline{z}) \oplus (\overline{x}\,\overline{y}\,z \oplus \overline{x}yz)\ x\,\overline{y}\,\overline{z} = \overline{x}\,\overline{y}\,z \oplus \overline{x}yz \oplus x\,\overline{y}\,\overline{z} \oplus \overline{x}\,\overline{y}z\ x\,\overline{y}\,\overline{z} \oplus \overline{x}yz\ x\,\overline{y}\,\overline{z} = \overline{x}\,\overline{y}\,z \oplus \overline{x}yz \oplus x\,\overline{y}\,\overline{z} = \overline{x}z\,(y \oplus \overline{y}) \oplus x\,\overline{y}\,\overline{z} = \overline{x}\,\overline{z} \cdot 1 \oplus x\,\overline{y}$

$\overline{z} = \overline{x}\,\overline{z} \oplus x\,\overline{y}\,\overline{z} = (x \oplus 1)\,z \oplus x\,(y \oplus 1)\,(z \oplus 1) = xz \oplus z \oplus (xy \oplus x)\,(z \oplus 1) = xz \oplus z \oplus xyz \oplus xz \oplus xy \oplus x = \underline{x \oplus z \oplus xy \oplus xyz}$ - 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 $\oplus$.

2.6.3. Замкнутость

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

Примеры: 1) $A = P_2$, т. к. $[A] = P_2$;

2) $A = \{1, x_1 \oplus x_2\}$. Замыканием этого множества будет класс L всех линейных функций, имеющих вид $f(x_1,x_2,\ldots,x_n) = c_0 \oplus c_1x_1 \oplus c_2x_2 \oplus \ldots \oplus c_n x_n$, 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) 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 $[A] = P_2$.

2.6.4. 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 $f^+(x_1, x_2,...,x_n) = \overline{f(\overline{x}_1, \overline{x}_2,...,\overline{x}_n)}$, т.е. функция, получающаяся из исходной путем замены 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) $(\overline{x})^+ = \overline{x}$.

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 $(\beta_1, \beta_2,\ldots,\beta_n)$ у которых $\alpha_i \le \beta_i$ 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.

Теорема Поста. Для того, чтобы система функций F была полной необходимо и достаточно, чтобы она полностью не со-

держалась ни в одном из пяти замкнутых классов P_0, P_1, S, M, L.

Пример $f(x,y) = x \mid y$.

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. $f(1,0) \neq \overline{f(0,1)}$, 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.

2.7. ЛОГИКА ПРЕДИКАТОВ

2.7.1. Понятие предиката

Логика предикатов – это логическая система, представляющая собой развитие алгебры высказываний. В логике пре-

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

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

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

Предметные константы обозначаются малыми буквами начала латинского алфавита:

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

Предикаты 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 истина соответственно.

Примеры:

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

Таким образом, выражение "x – простое число" можно рассматривать как функцию $P(x)$, зависящую от переменной x. При этом 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 двух предметных переменных 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.

Например, 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, можно получить новые высказывания, которые истинны или ложны.

2.7.2. Операции квантирования

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

Пусть дан предикат $P(x)$, зависящий от одной предметной переменной x 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 обобщаются 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. Следовательно 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. Таким образом, 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 $\exists$.

Таким образом, навешивание квантора на n – местный предикат приводит к уменьшению местности квантора на единицу.

2.7.3. Формулы логики предикатов

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

1) связанные переменные, то есть те, по отношению к которым, применяются 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.

Любую предметную переменную и предметную постоянную называют терм и обозначают *t*.

Если f_i^n - есть n – местный функциональный символ (функтор), а $t_1, t_2, ..., t_n$ - термы, то $f_i^n(t_1, t_2, ..., t_n)$ тоже является термом. Никаких других термов нет.

Если P_i^n - n – местный предикатный символ и $t_1, t_2, ..., t_n$ - термы, то $P_i^n(t_1, t_2, ..., t_n)$ - элементарная формула или атом. Т. е. Формула логики предикатов является атомарной, если в ней нет связанных переменных.

Различают:

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

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

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

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

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

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

2.7.4. 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 $\forall x W(x)$ и $\exists x W(x)$ над двумя типами полей:

1) пусть формулы $\forall x W(x)$ и $\exists x W(x)$ 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 $\forall x W(x)$ и $\exists x W(x)$:

$W(x)$	$\forall x W(x)$	$\exists x W(x)$
$A(x)$	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.

Обозначим их $I_1(x)$, $I_2(x)$, $I_3(x)$ и $I_4(x)$

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 равносильными.

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 тождественно истинными. Например, формула $\forall x W(x) \to \exists x W(x)$ является тождественно истинной.

Вопрос, является ли данная формула логики предикатов тождественно истинной, не удается решить эффективными методами, т.к. приходится использовать понятия бесконечности.

2.8. ФОРМАЛЬНЫЕ СИСТЕМЫ (ТЕОРИИ)

2.8.1. Определение формальной теории

Формальная теория T это:

1. Множество A символов образующих алфавит;

2. Множество F слов в алфавите A которые называются формулами;

3. Подмножество Ax формул ($Ax \subset F$), которые называются аксиомами;

4. Множество R отношений на множестве формул, которые называются правилами вывода.

Таким образом формальная теория T есть четвёрки $< A, F, Ax, R >$.

Обычно для образования символов A используются конечные множества.

Множество формул F задается индуктивным определением с помощью формальной грамматики. Это множество, как правило, бесконечно.

Множества A и F в совокупности определяют язык или сигнатуру формальной теории.

Множество аксиом Ax может быть конечным или бесконечным. Бесконечное множество аксиом задается с помощью конечного множества схем аксиом и правил порождения аксиом из схем аксиом.

Множество правил вывода R обычно конечно.

Факт вывода формулы G непосредственно из формул $F_1, F_2, ..., F_n$ по правилу вывода R записывается следующим образом:

$$\frac{F_1, F_2, ..., F_n}{G} R .$$

При этом формулы $F_1, F_2, ..., F_n$ называется посылками, а G - заключением.

Выводом формулы G из формул $F_1, F_2, ..., F_n$ формальной теории T называется последовательность формул $E_1, E_2, ..., E_n$, в которой последняя формула $E_n = G$, а любая другая формула E_i - либо аксиома, либо исходная формула $E_i = F_j$, либо непосредственно выводима из предыдущих формул.

Этот факт записывается так:

$$F_1, F_2, ..., F_n \mapsto G.$$

При этом $F_1, F_2, ..., F_n$ называются гипотезами вывода.

Если $\vdash G$, то формула G называется теоремой теории, то есть формула выводится из аксиом (без гипотез).

Интерпретацией формальной теории T в области интерпретации M называется функция $I : F \to M$, которая каждой формуле F формальной теории T однозначно сопоставляет некоторое содержательное высказывание относительно объектов множества M. Интерпретация I называется моделью формальной теории T, если все теоремы этой теории выполняются в интерпретации I.

Формула называется общезначимой (тавтологией) если она истина в любой интерпретации.

Формула называется противоречивой, если она ложна в любой интерпретации.

Формальная теория T называется непротиворечивой, если хотя бы одна ее теорема не является противоречием.

Формальная теория Т является полной (адекватной), если каждому истинному высказыванию M соответствует теорема в теории T.

Если для множества M существует формальная полностью не противоречивая теория T, то M называется аксиоматизируемым (формализуемым).

Система аксиом формально не противоречивой теории T называется независимой, если никакая из аксиом не выводится из остальных по правилам вывода теории T.

Формальная теория Т называется разрешимой, если существует алгоритм, который для любой формулы теории определяет, является ли эта формула теоремой теории или нет.

2.8.2. Исчисление высказываний

Исчисление высказываний – это формальная теория L в которой заданы:

1. Алфавит: а) $\bar{}, \to$ - связки

б) (,) – служебные символы

в) $a, b, ..., a_1, b_2$ – пропозициональные (высказывательные) переменные

2. Формулы: а) Переменные суть формулы.

б) Если A и B любые формулы, то $\overline{A}$ и $A \to B$ тоже формулы.

3. Аксиомы: а) $A_1 : (A \to (B \to A))$;

б) $A_2 : ((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$;

в) $A_3 : ((\overline{B} \to \overline{A}) \to ((\overline{B} \to A) \to B))$.

3) Правило: $\dfrac{A, A \to B}{B}$ Modus ponens (модус поненс - правило отделения).

Таким образом, множество аксиом теории L бесконечно, хотя задано тремя схемами аксиом. Множество правил вывода также бесконечно. Оно задано одной схемой.

Другие связки в исчислении высказываний вводятся определениями (а не аксиомами):

$$A \wedge B = \overline{A \to \overline{B}}, \quad A \vee B = \overline{A} \to B.$$

Любая формула, содержащая эти связки, рассматривается как синтаксическое сокращение собственной формулы теории L.

Отметим, что L формально не противоречива и полна на множестве высказываний.

2.8.3. Исчисление предикатов

Исчисление предикатов – это формальная теория P, в которой определены следующие компоненты:

1.Алфавит:

1) $\overline{\ }, \to$ – основные связки;

2) $\wedge$, $\vee$ – дополнительные связки;

3) (,) – служебные символы (открывающая и закрывающая скобки, а также запятая);

4) $\forall$, $\exists$ - кванторы всеобщности и существования;

5) $a,b,...,a_1,b_2,...$ - предметные константы;

6) $x,y,...,x_1,y_2,...$– предметные переменные;

7) $P,Q,...$ - предикаты;

8) $f,g,...$ - функциональные символы (функторы). С каждым предикатом и функтором связано некоторое натуральное число называемое кратностью или местностью.

Все выражения исчисления предикатов строятся из выше перечисленных символов алфавита. Среди таких выражений различают термы.

Термы – это выражения, которые являются аналогами имен объектов теории и соответствующих именных форм. По индуктивному определению терм - это:

1) предметные переменные и константы;
2) если f - n–местный функциональный символ и $t_1,t_2,...,t_n$ - термы, то $f(t_1,t_2,...,t_n)$ - терм;
3) других термов, кроме построенных по пп. 1) и 2) нет.

Из термов с помощью предикатных символов (имен конкретных предикатов) строятся атомы. Атомом (атомарной формулой, элементарной формулой) называется любая пропозициональная переменная (0-местная предикатная переменная), а также выражение вида $P(t_1,t_2,...,t_n)$, где: P - n–местный предикатный символ, а $t_1,t_2,...,t_n$ - термы.

2. Формулы.

1) атом есть формула;

2) если A и B – формулы, то выражения $\overline{A}$, $A \to B$ считаются формулами;

3) если A – формула, а x предметная переменная, то выражения $\forall xA$ и $\exists xA$, считаются формулами.

3. Аксиомы исчисления предикатов P включают в себя все аксиомы исчисления высказываний L, плюс

$$P_1 : \forall x A(x) \to A(t),$$

$$P_2 : A(t) \to \exists x A(x).$$

Здесь: t – терм свободный для переменной x в формуле A

4. Правила вывода:

1) $\dfrac{A, A \to B}{B}$ Modus ponens,

2) $\dfrac{B \to A(x)}{B \to \forall x A(x)} \forall^+$ $\forall$-правило,

3) $\dfrac{A(x) \to B}{\exists x A(x) \to B} \exists^+$ $\exists$-правило.

Исчисление предикатов, которое не содержат предметных постоянных, функторов, предикатов и собственных аксиом называется чистым.

Исчисление предикатов, которые содержат предметные постоянные, функторы предикаты и связывающие их собственные аксиомы называется прикладным.

Исчисление предикатов, в котором кванторы могут связать только предметные переменные, называется формальной теорией первого порядка.

Исчисление, в котором, кванторы могут связать не только переменные, но и сами предикаты, называется формальной теорией второго порядка. В формальных теориях более высокого порядка действие кванторов распределяется и на предикаты от предикатов и т.д.

Установлено что интерпретация исчисления высказываний в логику предикатов адекватно.

Можно доказать что исчисление предикатов не противоречиво.

Вопрос полноты (по Посту) в исчислении предикатов решается отрицательно. Существует формула $\exists x W(x) \to \forall x W(x)$, которая не является теоремой и добавления к которой не нарушает непротиворечивости исчисления предикатов.

2.9. АВТОМАТИЧЕСКОЕ ДОКАЗАТЕЛЬСТВО ТЕОРЕМ

2.9.1. Постановка задачи

Алгоритм, который проверяет отношение $\Gamma \mapsto S$ для формулы *S*, множества функций *Γ*, в формально теории *T* называется алгоритмом автоматического доказательства теорем.

В общем случае такой алгоритм невозможен т.к. не существует правила по которому для любых *S*, *Γ* и *T* выдавался бы ответ “да” если верно, что $\Gamma \mapsto S$ и ответ “нет” если не верно, что $\Gamma \mapsto S$. Более того, известно также, что нельзя построить автоматического доказательства теорем для большинства конкретных достаточно сложных формальных теорий. В некоторых случаях удается построить алгоритм автоматического доказательства теорем, который применим не ко всем формулам теории (частичный алгоритм).

Для простых формальных теорий (например, исчисление высказываний) и некоторых простых классов формул, в частности, например, прикладное исчисление предикатов с одним одноместным предикатом, алгоритмы автоматического доказательства теорем известны.

Так как для исчисления высказываний теоремами являются общезначимые формулы, поэтому можно воспользоваться таблицами истинности для проверки общезначимости. Для этого достаточно вычислить истинные значения формулы при всех возможных интерпретациях. Их число их конечно. Если во всех случаях получаются истинные значения, то проверяемая формула - тавтология и, следовательно, она является теоремой теории *L*. Если же, хотя бы в одном случае, получается ложное значение, то проверяемая формула не является тавтологией и, следовательно, не является теоремой теории *L*.

Классический алгоритм автоматического доказательства теорем называется методом резолюций. Для любого прикладного исчисления предикатов первого порядка *T*, любой формулы *S* и множества формул *Γ* теории *T* метод резолюций выдает ответ «да» если $\Gamma \mapsto S$ и выдает ответ «нет», или не выдает никакого ответа, (т.е. зацикливается) если неверно, что $\Gamma \mapsto S$.

В основе метода резолюции лежит идея доказательства «от противного».

Теорема: Если $\Gamma, \overline{S} \mapsto F$, где F – любое противоречие (тождественно ложная формула), то $\Gamma \mapsto S$.

При доказательстве от противного по методу резолюции в качестве формулы F принято использовать пустую формулу, которая обозначается $\square$. Пустая формула не имеет никакого значения, ни в одной из интерпретаций. В частности, она не является истинной ни в одной из интерпретаций и, поэтому, по определению, является противоречием.

2.9.2. Сведение формул к предложениям

В методе резолюции используется стандартная форма формул, которая называется предложением. Предложение – это бесквантороная дизъюнкция литералов. Любая формула исчисления предикатов может быть преобразована в множество предложений следующим образом (для обозначения способа преобразования формул используется знак $\Rightarrow$):

1. Элиминация импликации.

Преобразование $A \to B \Rightarrow \overline{A} \vee B$. После первого этапа формула содержит только $\overline{\ }, \vee, \wedge, \forall, \exists$.

2. Протаскивание отрицаний.

Преобразования $\overline{\forall x A} \Rightarrow \exists x \overline{A}$, $\overline{\exists x A} \Rightarrow \forall x \overline{A}$, $\overline{\overline{A}} \Rightarrow A$, $\overline{A \vee B} \Rightarrow \overline{A} \wedge \overline{B}$, $\overline{A \wedge B} \Rightarrow \overline{A} \vee \overline{B}$.

После второго этапа формула содержит отрицания только перед атомами.

3. Разделение связанных переменных.

Преобразование

$Q_1 x A(..Q_2 x B(..x...)...) \Rightarrow Q_1 x A(..Q_2 y B(..y...)...)$, где Q_1 Q_2 – любые кванторы. После третьего этапа формулы не должны содержать случайно совпадающих переменных.

4. Приведение к предваренной форме .

Преобразование

$Qx A \vee B \Rightarrow Qx(A \vee B)$, $Qx A \wedge B \Rightarrow Qx(A \wedge B)$, где Q – любой кван-

тор. После этого этапа формула находится в предваренной форме.

5. Элиминация кванторов существования.

Преобразование

$$\exists x_1 Q_2 x_2 ... Q_n x_n A(x_1, x_2 ... x_n) \Rightarrow Q_2 x_2 ... Q_n x_n A(a_1, x_2 ... x_n);$$

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

$$\Rightarrow \forall x_1 ... \forall x_{i-1} Q_{i+1} x_{i+1} ... Q_n x_n A(x_1, ... x_{i-1}, f(x_1, ..., x_{i-1}), x_{i+1}, ..., x_n),$$

где: a_1 – новая предметная константа, f – новый функтор, а $Q_1,...,Q_n$ – произвольные кванторы. После этого этапа формулы содержат только кванторы всеобщности.

6. Элиминация кванторов всеобщности.

Преобразование $\forall x A(x) \Rightarrow A(x)$.

После этого этапа формулы не содержат кванторов.

7. Приведение к конъюнктивной нормальной форме.

Преобразования:

$$A \vee (B \wedge C) \Rightarrow (A \vee B) \wedge (A \vee C),$$

$$(A \wedge B) \vee C \Rightarrow (A \vee C) \wedge (B \vee C).$$

После седьмого этапа формулы находятся в КНФ.

8. Элиминация конъюнкций.

Преобразование $A \wedge B \Rightarrow A, B$.

После восьмого этапа формула распадается на множество предложений.

Теорема: Если Γ - множество предложений полученных из формулы S, то S является противоречием, тогда и только тогда множество Γ не выполнимо. Множество формул Γ невыполнимо – это означает, что множество Γ не имеет модели, т.е. не существует такой интерпретации, в которой все формулы Γ имели бы истинное значение.

2.9.3. Правило резолюции для исчисления высказываний

Пусть C_1 и C_2 - два предложения в исчислении высказываний и пусть $C_1 = p \vee C_1'$, $C_2 = \overline{p} \vee C_2'$, где p – пропозицио-

нальная переменная C_1' и C_2' – любые предложения. В том числе они могут быть пустыми или содержать один литерал.

Правило вывода: $\dfrac{C_1, C_2}{C_1' \vee C_2'} R$

Это правило называется правилом резолюции, при этом C_1 и C_2 называются резольвируемыми (родительскими) предложениями, а $C_1' \vee C_2'$ - резольвентой. p и $\bar{p}$ - контрарными литералами.

Заметим, что многие правила вывода являются частными случаями правила резолюции:

$\dfrac{A, A \to B}{B}$	Modus ponens	$\dfrac{A, \bar{A} \vee B}{B} R$
$\dfrac{A \to B, B \to C}{A \to C}$	Транзитивность	$\dfrac{\bar{A} \vee B, \bar{B} \vee C}{\bar{A} \vee C} R$
$\dfrac{A \vee B, A \to B}{B}$	Слияние	$\dfrac{A \vee B, \bar{A} \vee B}{B} R$

Теорема: Правило резолюции логично, т.е. резольвента является логическим следствием резольвируемых предложений.

2.9.4. Правила резолюции для исчисления предикатов

Для применения правила резолюции нужны контрарные литералы в резольвируемых предложениях. Пусть C_1 и C_2 - два предложения в исчислении предикатов. Правило вывода $\dfrac{C_1, C_2}{(C_1' \vee C_2')\sigma} R$ называется правилом резолюции в исчислении предикатов, если в предложениях C_1 и C_2 существуют унифицируемые контрарные литералы p_1 и p_2, т.е. такие, что $C_1 = p_1 \vee C_1'$, $C_2 = \bar{p}_2 \vee C_2'$, причем атомарные формулы p_1 и

p_2 являются унифицируемыми наиболее общим унификатором σ. В этом случае резольвентой предложений C_1 и C_2 является предложение $(C_1' \vee C_2')\sigma$, полученное из предложения $C_1' \vee C_2'$ применение унификатора σ.

Понятие унификатора. Если в формулу A вместо переменных $x_1,\ldots,x_n$ подставить соответствующие формулы $B_1,\ldots,B_n$, то получится формула B, которая называется частным случаем формулы A; т.е. $B := A(x_1,...,x_n)\{B_i // x_i\}_{i=1}^n$. При этом формула B_i подставляется вместо всех вхождений переменной x_i. Набор подстановок $\{B_i // x_i\}_{i=1}^n$ называется унификатором.

Формула C называется совместным частным случаем формул A и B, если C является частным случаем формулы A и одновременно частным случаем формулы B при одном и том же наборе подстановок, т.е. когда:

$\{X_i // x_i\}_{i=1}^n$

$$C = A(...x_i...)\{X_i // x_i\}_{i=1}^n \wedge C = B(...x_i...)\{X_i // x_i\}_{i=1}^n .$$

Формулы, которые имеют совместный частный случай, называются унифицируемыми, а набор подстановок $\{X_i // x_i\}_{i=1}^n$, с помощью которого получается совместный частный случай унифицируемых формул называется общим унификатором. Наименьший возможный унификатор называется наиболее общим унификатором.

2.9.5 Опровержение методом резолюции

Опровержение методом резолюции – это алгоритм автоматического доказательства теорем в прикладном исчислении предикатов, который сводится к следующему. Пусть установлена выводимость $S \mapsto G$.

Каждая формула множества S и формула $\overline{G}$ (это отрицание целевой теоремы) независимо преобразуются в множества предложений. В полученном совокупном множестве предложений отыскиваются резольвируемые предложения. К ним применяется правило резолюций и резольвента добавляется в

множество предложений до тех пор, пока не будет получено пустое предложение. При этом возможны 3 случая:

1) Среди текущего множества предложений нет резольвируемых. Это означает, что теорема опровергнута, т.е. формула *G* не выводима из множества формул *S*.

2) В результате очередного применения правила получено пустое предложение – это означает, что теорема доказана, т.е. $S \mapsto G$.

3) Процесс не заканчивается, т.е. множество предложений пополняется все новыми резольвентами, среди которых нет пустых. Это ничего не означает.

2.10. *k*-значная логика

Конечнозначная логика вводится как обобщение двузначной логики. Она используется для описания функционирования сложных управляющих систем. Компоненты этих систем могут находиться в конечном числе состояний.

2.10.1. Функции и формулы *k*-значной логики

Функция $f(x_1,...,x_n)$, аргументы и значения, которой определены на множестве истинностных значений $E_k = \{0,1,...,k-1\}$, называется функцией *k*-значной логики.

Каждую функцию *k*-значной логики от n – аргументов можно задать в виде таблицы содержащей K^n строк.

x_1 .... x_{n-1} x_n	$f(x_1,...,x_{n-1},x_n)$
0 .... 0 0	$f(0, ..., 0, 0)$
0 .... 0 1	$f(0, ..., 0, 1)$
. . . . .	. . . . .
0 .... 0 *k*-1	$f(0, ..., 0, k-1)$
0 .... 1 0	$f(0, ..., 1, 0)$
. . . . .	. . . .
0 .... *k*-1 *k*-1	$f(0, ...,k-1,k-1)$

Множество всех функций k-значной логики обозначается P_k ($P_k(n)$). Заметим, что $|P_k| = k^{k^n}$. В частности число функций от двух переменных в P_3 равно 3^{3^2} =19683, т.е. это множество практически не обозримо. Поэтому в P_3 так же как и в P_2, используется задание функций с помощью формул. В качестве «элементарных» в k-значной логике рассматриваются следующие функции:

1. $\bar{x} = x + 1 (\mathrm{mod} k)$.

Эта функция представляет собой отрицание в смысле «циклического сдвига значений».

2. $\tilde{x} = k - 1 - x$ – это обобщение отрицания в смысле «зеркального отображения значений». Оно носит название отрицание Лукашевича.

3. $I_i(x) = \begin{cases} k-1, & \text{при } x = i \\ 0, & \text{при } x \neq i \end{cases}$ $(i = 0,1,...,k-1)$.

Эта функция также является обобщением некоторых свойств отрицания.

4. $f_i(x) = \begin{cases} 1, & \text{при } x = i \\ 0, & \text{при } x \neq i \end{cases}$ $(i = 0,1,...,k-1)$.

Это характеристическая функция значения i, которая также обобщает отрицание.

5. $\min(x_1, x_2)$ – это обобщение конъюнкции.

6. $x_1 x_2 (\mathrm{mod} k)$ – это есть второе обобщение конъюнкции.

7. $\max(x_1, x_2)$ – обобщение дизъюнкции.

8. $x_1 + x_2 (\mathrm{mod} k)$ – второе обобщение дизъюнкции.

Используя переменные, допустимые значения которых являются элементы множества E_k и символы некоторых функций из P_k можно строить формулы, которые задают функции из P_k.

Любая функция из P_k может быть представлена в виде:

$$f(x_1, x_2, ..., x_n) = \bigvee_{\sigma_1, \sigma_2, ..., \sigma_n} I_{\sigma_1}(x_1) I_{\sigma_2}(x_2) .. I_{\sigma_n}(x_n) f(\sigma_1, \sigma_2, ..., \sigma_n),$$

где под конъюнкцией понимается $xy = \min(x_1, x_2)$, а под дизъ-

юнкцией $x \vee y = \max(x_1, x_2)$. В этом выражении дизъюнкция распространяется по всем наборам $\sigma_1, \sigma_2, ..., \sigma_n$ элементов E_k.

Данное представление является аналогом СДНФ.

2.10.2. Полнота и замкнутость функций *k*-значной логики

Система В функций из P_k ($f_1, f_2, ..., f_s, ...$) называется (функционально) полной, если любая функция из P_k может быть записана в виде формулы через функции этой системы.

Примерами полных систем является:

1.В= P_k.

2.В= $\{0,1,..k-1, I_0(x), ..., I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}$.

3.В= $\{\bar{x}, \max(x_1, x_2)\}$.

4.В= $\{V_k(x_1, x_2)\}$,где $V_k(x_1, x_2) = \max(x_1, x_2) + 1 (\mathrm{mod} k)$.

Функция $V_k(x_1, x_2)$ называется фуннцией Вебба представляет собой аналог функции Шеффера.

Пусть М – произвольное подмножество функции из P_k. Замыканием М называется множество [М] всех функций из P_k, представленных в виде формул через функции множества М

Класс М называется (функционально) замкнутым, если замыкание [М]=М.

Таким образом, в терминах замыкания можно определить полноту системы функций, а именно В является полной системой если замыкание [В]= P_k.

Справедлива теорема о функциональной полноте - теорема Кузнецова А.В.:

Можно построить систему замкнутых классов в P_k - M_1, M_2,…,M_s, каждый из которых не содержит целиком ни одного из остальных классов и такую, что подсистема функций из P_k полна тогда и только тогда, когда она целиком не содержится ни в одном из классов M_1, M_2,…,M_s. Это аналог теоремы Поста.

Теорема Кузнецова доказывает, что возможно выразить, условие полноты системы B в терминах принадлежности ее к специальным классам M_1, M_2,…,M_s, однако практическое построение классов даже при небольших k связано с трудоемкими вычислениями. Поэтому возникает вопрос о поиске других более эффективных критериев полноты. Эта цель достигается за счет введения ограничений , т.е. за счет знания дополнительной информации о системе B.

Существенными называются функции $f(x_1, x_2, ..., x_n)$ из P_k, если они существенно зависят не менее чем от двух переменных.

Теорема Яблонского

Пусть система B функций из P_k, где $k \geq 3$, содержит все функции одной переменной, принимающие не более k-1 значений. Тогда для полноты системы B необходимо и достаточно, чтобы B содержало существенную функцию $f(x_1, x_2, ..., x_n)$, принимающую все k значений.

Следствие (критерий Слупецкого):

Пусть система B функций из P_k, где к ≥ 3, содержит все функции одной переменной. Тогда для полноты системы B необходимо и достаточно, чтобы B содержало существенную функцию $f(x_1, x_2, ..., x_n)$, принимающую все k значений.

Непосредственное использование теоремы и ее следствия не всегда удобно, так как для этого необходимо установить наличие в B всех функций одной переменной, принимающих не более k-1 значения, т.е. $k^k - 1$ функций. С ростом k громоздкость вычислений возрастает. Поэтому это требование целесообразно заменить требованием, в котором система функций B порождало бы множество функций одной переменной.

Известно, что функции из B могут быть получены из конкретных систем функций одной переменной.

Теорема 1 (Пикара).

Все функции одной переменной из P_k могут быть порождены тремя функциями:

1) $f(x) = x - 1 (\mathrm{mod} k)$,

2) $g(x) = \begin{cases} x, & \text{если} \quad 0 \le x \le k-3 \\ k-1, & \text{если} \quad x = k-2 \\ k-2, & \text{если} \quad x = k-1 \end{cases}$,

3) $h(x) = \begin{cases} 1, & \text{если} \quad x = 0 \\ x, & \text{если} \quad x \ne 0 \end{cases}$.

Теорема 2

Все функции одной переменной из P_k могут быть порождены k функциями

$$f_i(x) = \begin{cases} i, & \text{при} \quad x = 0 \\ 0, & \text{при} \quad x = i \qquad (i = 0,1,..k-1) \\ x, & \text{в остальных случаях} \end{cases}$$

и функцией $h(x)$.

Теорема 3 (Мартина).

Функция $f(x_1, x_2, ..., x_n)$ из P_k при к ≥ 3 является функция Шеффера, тогда и только тогда, когда $f(x_1, x_2, ..., x_n)$ порождает все функции одной переменной принимающие не более k-1 значений.

2.10.3. Особенности k – значной логики

Во многом k – значная логика подобна двухзначной. В ней сохраняются многие результаты, имеющие место в двузначной логике. Однако ряд результатов, верных для функций алгебры логики, т.е. при k=2, уже не переносятся на случай, когда $k \ge 3$.

Например, как показал Пост, каждый замкнутый класс в P_2 имеет конечный базис и поэтому, число замкнутых классов в P_2 счетно. С другой стороны для $k \ge 3$ в P_k:

а) существует замкнутый класс не имеющий базиса;

б) существует замкнутый класс имеющий счетный базис;

в) имеется бесчисленные множества различных замкнутых классов.

3. ОСНОВНЫЕ ПОЛОЖЕНИЯ ТЕОРИИ АЛГОРИТМОВ

3.1. ПОНЯТИЯ АЛГОРИТМА И РЕКУРСИВНОЙ ФУНКЦИИ

3.1.1. Интуитивное понятие алгоритма

Алгоритмом называется совокупность точно предписанных правил, определяющих вычислительный процесс, который идет от варьируемых исходных данных к искомому результату.

Например:

1) алгоритм умножения двух многозначительных чисел;
2) алгоритм извлечения квадратного корня;
3) алгоритм определения истинностного значения некоторого высказывания.

Характерные свойства алгоритмов:

1. Дискретность. Алгоритм описывает процесс последовательного построения величин, идущий в дискретном времени. При этом интервал времени, необходимый для вычисления, разбивается на малые отрезки – такты. В результате осуществления элементарного шага алгоритма (определенной программы преобразований) из системы величин, имеющихся в начале такта, получается система величин приходящихся на конец такта.

2. Детерминированность. Это означает, что программа преобразований в каждом такте определена однозначно.

3. Результативность. Каждый алгоритм направлен на получение определенного результата. В частности, если вычисляемая функция в некоторой точке не определена, то совокупность правил должно точно указать, что нужно считать в этом случае результатом применения алгоритма.

4. Массовость. В определенных пределах исходные величины могут варьироваться. Это значит, что алгоритм служит для решения не только какой-либо одной конкретной задачи. Он предназначен для решения целого класса задач. Процедуру решения одной конкретной задачи никогда не называют алгоритмом.

Понятие алгоритма, задаваемое его эмпирическими свойствами, не является строго математическим, поэтому оно называется интуитивным.

3.1.2. Проблема уточнения понятия алгоритма

Интуитивное понятие алгоритма допустимо при положительном решении вопроса о существовании алгоритма. Теоремы о не существовании алгоритмов не могут быть доказаны ввиду нечеткости интуитивного понятия алгоритма. В этом случае требуется его уточнение. Существуют несколько подходов уточнения понятия алгоритма. Основаниями для них являются:

1) теория рекурсивных функций;
2) машины Тьюринга;
3) нормальный алгоритм Маркова.

В алгоритмических проблемах обычно рассматриваются алгоритмы, имеющие дело только с натуральными числами. Можно доказать, что это не является потерей общности, т. к. объекты другой природы можно закодировать натуральными числами. Для пользователей компьютеров такое утверждение является очевидным.

Пусть N – множество натуральных чисел. *k*-местными частичными функциями называются функции с областью определения $D_f \subset N^k$ (k – целое положительное число) и областью значений $\mathrm{Im}_f \subset N$. Слово «частичная» означает, что функция определена на подмножестве N^k.

Если $D_f = N^k$, то функция называется всюду определенной.

k–местная частичная функция $f: N^k \to N$ называется вычислимой, если существует алгоритм, вычисляющий *f*. Этот алгоритм должен удовлетворять следующим условиям:

1. Если на выход алгоритма поступает вектор $\bar{x} = (x_1, \ldots, x_k) \in D_f$, то вычисление должно закончиться после конечного числа шагов и выдать значение $f(\bar{x})$.

2. Если на выход алгоритма поступает $\bar{x} \notin D_f$, то алгоритм никогда не заканчивается.

Заметим, что множество вычислимых функций не совпадает с множеством "практически вычислимых" функций, которые связаны с ограничениями современных вычислительных машин.

Определение вычислимой функции не является формальным. Формализацией этого понятия служит частично-рекурсивная функция.

3.1.3. Частично-рекурсивные функции

Простейшими (базисными) называются следующие числовые функции:

1) постоянная функция $\varphi(x_1, \ldots, x_k) = m$, где $m \in N$;

2) одноместная функция следования $S(x) = x' = x + 1$;

3) функция проекции $\Psi(x_1, \ldots, x_k) = x_m \ \ (1 \le m \le k)$.

Преобразование функций называется оператором. При построении рекурсивно частичной функции используются следующие операторы:

1. Оператор суперпозиции. *k*-местная функция $f(x_1, \ldots, x_k)$ получена с помощью суперпозиции из *m*-местной функции $\varphi(y_1, \ldots, y_m)$ и *k*-местных функций $g_1(x_1, \ldots, x_k)$, $g_2(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k)$, если

$$f(x_1, \ldots, x_k) = \varphi(g_1(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k)).$$

2. Оператор примитивной рекурсии. При $k \ge 0$ из *k*-местной функции *f* и *(k+2)*-местной функции *g* строится *(k+1)*-местная функция *h* по следующей схеме:

$$h(x_1, \ldots, x_k, 0) = f(x_1, \ldots, x_k);$$

$$h(x_1, \ldots, x_k, y+1) = g(x_1, \ldots, x_k, y, h(x_1, \ldots, x_k, y)).$$

3. Оператор минимизации. Этот оператор ставит в соответствие частичной функции $f: N^{k+1} \to N$ частичную функцию $h: N^k \to N$, которая определяется следующим образом:

1) область определения

$D_h = \{(x_1,\ldots,x_k) \mid \exists\, x_{k+1} \geq 0 : f(x_1,\ldots,x_k,x_{k+1}) = 0$ и

$(x_1,\ldots,x_k,y) \in D_f \ \forall\, y \leq x_{k+1}\}$;

2) $h(x_1,\ldots,x_k) =$ наименьшее значение y, при котором $f(x_1,\ldots,x_k,y) = 0$.

Естественный путь вычисления $h(x_1,\ldots,x_k)$ состоит в подсчете значения $f(x_1,\ldots,x_k,y)$ последовательно для y=0,1,2,… до тех пор, пока не найдется значение y, обращающее $f(x_1,\ldots,x_k,y)$ в 0. Этот алгоритм может не остановиться, если $f(x_1,\ldots,x_k,y)$ нигде не обращается в 0.

Все функции, которые можно получить из базисных с помощью трех указанных операторов называются частично-рекурсивными. Если при этом функция, полученная с помощью этих операторов, всюду определена, то она называется общерекурсивной. Если при этом функция полученная из базисных функций без механизма минимизации, всюду определена, то она называется примитивно-рекурсивной.

Можно показать, что введение фиктивных переменных, а также перестановка и отождествление переменных не выводят функцию за пределы класса частично-рекурсивных.

В частности:

Введение фиктивных переменных. Если $g(x_1,x_2)$ - примитивно-рекурсивная функция и $f(x_1,x_2,x_3) = g(x_1,x_2)$, то $f(x_1,x_2,x_3)$ - примитивно-рекурсивная функция.

Перестановка переменных. Если $g(x_1,x_2)$ - примитивно-рекурсивная функция и $f(x_2,x_1) = g(x_1,x_2)$, то f есть также примитивно-рекурсивная функция.

Отождествление переменных. Если $g(x_1, x_2, x_3)$ - примитивно-рекурсивная функция и $f(x_1, x_2) = g(x_1, x_2, x_1)$, то $f(x_1, x_2)$ есть также примитивно-рекурсивная функция.

3.2. МАШИНЫ ТЬЮРИНГА

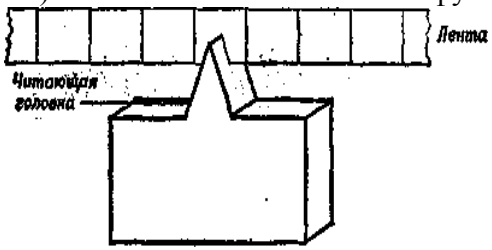
3.2.1. Понятие и формальное определение машины Тьюринга

Понятие машины Тьюринга впервые было введено в 1936 году логиком Тьюрингом. Тьюринг рассматривал гипотетическую «машину», имеющую конечное множество *S* внутренних состояний и одну бесконечно длинную ленту, разделенную на ячейки. Машина за один такт может передвигать на одну ячейку вправо или влево. В каждой ячейке машина записывает символ из конечного алфавита *A*. Первоначально лента должна быть пустой, за исключением конечного числа ячеек, заполненных заранее. Эти заранее заполненные ячейки представляют собой программу запуска машины.

Формальное определение машины Тьюринга состоит в следующем. Машиной Тьюринга называется пятерка объектов $[A, S, \nu, \xi, \delta]$ следующего типа: A - это конечный алфавит символов, которые могут быть записаны в ячейках и являются одновременно входными и выходными, т. е. $A = \{a_0, a_1, \ldots, a_n\}$. S есть конечное множество внутренних состояний, т. е. $S = \{s_0, s_1, \ldots s_r\}$; ν - функция из $S \times A$ в S; ξ - функция из $S \times A$ в A; δ - функция из $S \times A$ в множество {П, Л, ОСТАНОВ}. Интуитивный смысл данных выражений станет ясен из дальнейшего.

Машина Тьюринга работает следующим образом. Она начинает работу, находясь в начальном состоянии s^0. После считывания первого символа она переходит в новое внутреннее состояние, определяемое функцией ν, записывает в ячейке символ, являющийся значением функции ξ, перемещает ленту

направо (П), налево (Л), или остается на месте и прекращает работу (ОСТАНОВ) в зависимости от значения функции δ.



На рисунке схематически изображена лента машины Тьюринга и считывающе-записывающая головка.

Таким образом работа машины состоит в повторении следующего цикла: считывание символа из ячейки, печатанье нового символа в эту ячейку, выбор которого определяет функция ξ (может оказаться, что это тот же символ), сдвиг ленты налево или направо либо остановка. Лента бесконечна в обоих направлениях, однако вначале (и, значит, после конечного числа тактов) заполнено лишь конечное число ячеек.

3.2.2. Примеры машин Тьюринга

Пример 1. Машина Тьюринга считывает входную последовательность нулей и единиц, печатает Ч, если число единиц четное, и Н, если нечетно. Строке из нулей и единиц предшествуют и последуют пустые ячейки, обозначаемые #. Символы Ч или Н печатаются в первой пустой ячейке вслед за входной строкой. Таким образом, алфавит данной машины Тьюринга имеет вид:

$$A = \{\#, 0, 1, Ч, Н\}.$$

Внутренние состояния: $S = \{s_0, s_1, s_2\}$; s_0 - начальное состояние. Машина останавливается по сигналу ОСТАНОВ.

Функции ν, ξ, δ могут быть представлены табличным способом в следующем виде

v:	ξ:	δ:
$(s_0,0) \mapsto s_1$	$(s_0,0) \mapsto 0$	$(s_0,0) \mapsto Л$
$(s_0,1) \mapsto s_2$	$(s_0,1) \mapsto 1$	$(s_0,1) \mapsto Л$
$(s_1,0) \mapsto s_1$	$(s_1,0) \mapsto 0$	$(s_1,0) \mapsto Л$
$(s_1,1) \mapsto s_2$	$(s_1,1) \mapsto 1$	$(s_1,1) \mapsto Л$
$(s_2,0) \mapsto s_2$	$(s_2,0) \mapsto 0$	$(s_2,0) \mapsto Л$
$(s_2,1) \mapsto s_1$	$(s_2,1) \mapsto 1$	$(s_2,1) \mapsto Л$
$(s_0,\#) \mapsto s_0$	$(s_0,\#) \mapsto \#$	$(s_0,\#) \mapsto Л$
$(s_1,\#) \mapsto s_1$	$(s_1,\#) \mapsto Ч$	$(s_1,\#) \mapsto ОСТАНОВ$
$(s_2,\#) \mapsto s_2$	$(s_2,\#) \mapsto Н$	$(s_2,\#) \mapsto ОСТАНОВ$

Удобнее задавать функции v, ξ, δ, пользуясь обозначениями Тьюринга. В этом варианте машина Тьюринга задается конечным множеством пятерок $[s_i, a_j, s_r, z_l, t_n]$. В каждой такой пятерке

s_i - текущее состояние машины;

a_j - символ, считываемый из ячейки;

s_r - следующее состояние машины, $s_r = v(s_i, a_j)$;

z_l - символ, печатаемый в ячейке, $z_l = \xi(s_i, a_j)$;

t_n - одна из команд П, Л, ОСТАНОВ.

В этих обозначениях описанная выше машина задается так:

s_0	#	s_0	#	Л
s_0	0	s_1	0	Л
s_0	1	s_2	1	Л
s_1	0	s_1	0	Л

s_1	1	s_2	1	Л
s_2	0	s_2	0	Л
s_2	1	s_1	1	Л
s_1	#	s_1	Ч	ОСТАНОВ
s_2	#	s_2	Н	ОСТЕНОВ

Машины Тьюринга, могут вычислять разнообразные функции от чисел, подаваемых на вход. Стандартным представлением неотрицательного числа n в машине Тьюринга является последовательность из n+1 единиц, стоящих подряд. Два таких числа отделяются нулем. Так, строка ...##111011##... представляет упорядоченную пару (2,1). Строка ...##111101101011# #... представляет последовательность (3, 1, 0, 1).

Пример 2. Следующая машина Тьюринга складывает два неотрицательных целых числа, поданных на вход:

s_0	#	s_0	#	Л
s_0	1	s_1	1	Л
s_1	1	s_1	1	Л
s_1	0	s_2	1	Л
s_2	1	s_2	1	Л
s_2	#	s_3	#	П
s_3	1	s_4	#	П
s_4	1	s_5	#	ОСТАНОВ

Она превращает две последовательности единиц, разделенные нулем, в последовательность единиц, представляющую число, равное сумме чисел на входе. Можно описать машину

Тьюринга, способную сложить три целых числа, четыре целых числа или даже любое не ограниченное наперед количество целых чисел, но эта задача несколько сложнее.

3.3. СЛОЖНОСТЬ АЛГОРИТМОВ (АНАЛИЗ АЛГОРИТМОВ)

3.3.1. Основные понятия

Сложность алгоритмов – это величина, характеризующая длину описания алгоритма или громоздкость его применения к исходным данным.

Изучение сложностей алгоритмов началось в 60-х годах 20 века. Этот раздел теории алгоритмов имеет важное теоретическое и практическое значение. Разрабатываемые методы оценки сложности алгоритмов позволяют находить наиболее эффективные способы вычисления функций. В терминах сложностей алгоритмов можно уточнить и исследовать многие проблемы, связанные с нахождением оптимального алгоритма для решения конкретных задач.

Класс однородных вычислительных задач называется проблемой (массовой задачей).

Индивидуальные случаи проблемы T называются частными случаями проблемы T.

Таким образом, T это множество всех своих частных случаев. Например, можно говорить о проблеме умножения матриц, частным случаем которой является нахождение произведения двух конкретных матриц.

Другим примером является задачи коммивояжера. Параметры этой массовой задачи - это множество «городов» $C=\{c_1,c_2,...,c_m\}$ и расстояний $d(c_i,c_j)$ между каждый парой городов c_i и c_j из C. Решение – это упорядоченный набор $c_{k_1},c_{k_2},...,c_{k_m}$ заданных городов, который минимизирует величину

$$\sum_{i=1}^{m-1} d(c_{k_i},c_{k_{i+1}}) + d(c_{k_m},c_{k_1})$$

Это выражение дает длину маршрута, начинающегося в городе c_{k_1}, проходящего последовательно через все города и возвращающегося в c_{k_1} непосредственно из последнего города c_{k_m}. Индивидуальная задача о коммивояжере задается любыми конкретными множествами $\{c_1, c_2, ..., c_m\}$ и $\{d(c_i, c_j)\}$.

С каждым частным случаем I проблемы T ($I \in T$) связывается размер $|I|$. Эта функция не единственна. Ее выбор определяется теоретическими и практическими соображениями, связанными с точками зрения на эту проблему.

Например, для пары квадратных матриц $I = (A, B)$, размерности которых - $n \times n$, разумной мерой их умножения может служить величина $|I| = n$. Однако, если изучается объем памяти, необходимый для алгоритма умножение матриц, то мера $|I| = n^2$.

В задаче о коммивояжере $|I|$ можно определить как количество данных городов m.

Пусть T – проблема и A – алгоритм, решающий ее. При решении частного случая $I \in T$ алгоритм A выполняет некоторую последовательность вычислений S_I.

При этом важны следующие характеристики:

1. длина S_I, которая характеризует время вычисления,
2. глубина S_I, то есть число уровней параллельных шагов, на которые S_I может быть разложено,
3. объем памяти, требуемый для вычисления S_I,
4. число арифметических операций при алгебраических вычислениях или число обращений к памяти.

Размер $|I|$ выбирается так, чтобы все частные случаи I одинакового размера n решались при помощи одной и той же схемы C_n. Сложность схемы C_n определяется как и I, так и другими мерами сложности.

После выбора меры μ вычисления S, функция сложности вычисления F_A определяется несколькими способами. Главные из них – сложность в наихудшем случае и сложность поведения в среднем.

Сложность в наихудшем случае

$$F_A(n) = \max\{\mu(S_I) \mid I \in T, |I| = n\}.$$

Сложность поведения в среднем

$$M_A(n) = \sum_{I \in T_n} p(I)\mu(S_I),$$

где: $T_n = \{I \mid I \in T, |I| = n\}$ и $p(I)$ – вероятность появления I среди других случаев размера n.

3.3.2. Классификация задач по степени сложности

Сложность задачи – это сложность наилучшего алгоритма известного для ее решения.

Для оценки сложности важно следующее определение. Функция $f(n)$ есть $O(g(n))$, если существует константа C, такая, что $\forall n \ |f(n)| \le C|g(n)|$.

Основной вопрос теории сложности – насколько успешно или с какой стоимостью может быть решена заданная проблема T? Решение вопроса заключается в рассмотрении всех возможных алгоритмов проблем T и попытке формулировки утверждения о вычислительной сложности внутренне присущей T. Любой алгоритм A для T дает верхнюю оценку величины F_A сложности T. Однако математический интерес представляет знание нижней оценки. В этом случае она дает руководство в поиске хороших алгоритмов, указывая заведомо ложные попытки.

Быстрыми являются линейные алгоритмы, которые обладают сложностью порядка n и называются алгоритмами порядка $O(n)$, где n – размерность входных данных.

К линейным алгоритмам относится алгоритм нахождения суммы двух десятичных чисел. Его сложность – $O(n_1 + n_2)$.

Существуют алгоритмы быстрее линейных. Например, алгоритм двоичного поиска в линейном упорядоченном массиве. Его сложность – $O(\log_2 n)$, где: n – длина массива.

Полиномиальным алгоритмом (алгоритм класса P) называется алгоритм, у которого временная сложность равна $O(n^k)$, где k – целое число >0. алгоритмы для временной сложности которых не существует такой оценки называются экспоненциальными (алгоритмы класса E).

Задача называется труднорешаемой, если для нее не существует разрешающего полиномиального алгоритма.

Задача называется «хорошей», если для нее существует полипомиальный алгоритм.

Примеры алгоритмов класса P:

1. Задача Прима – Краскала. Дана плоская страна и в ней n – городов. Нужно соединить все города телефонной связью так, чтобы общая длина телефонных линий была минимальной. Эта задача решается с помощью жадного алгоритма сложности $O(n\log n)$.

2. Кратчайший путь на графе, состоящем из n вершин и m ребер. Сложность – $O(mn)$.

3. Быстрое преобразование Фурье - $O(n\log n)$, обычное преобразование Фурье – $O(n^2)$.

4. Умножение чисел. Алгоритм Шенхаге–Штрассена - $O(n\log n\log\log n)$. Школьный метод умножения $O(n^2)$.

5. Умножение матриц размерности $(n\times n)$. Алгоритм Штрассена $O(n^{\log_2 7})$. Обычный алгоритм $O(n^3)$.

Примеры алгоритмов класса E:

К экспоненциальным задачам относятся задачи, в которых требуется построить множество всех подмножеств данного множества, все полные подграфы некоторого графа или все поддеревья некоторого графа.

3.3.3. Недетерминированные алгоритмы и классы NP алгоритмов

Недетерминированным называется алгоритм, который может находится одновременно во многих состояниях. Он не является ни вероятностным, ни случайным. Недетерминированный алгоритм исследует все альтернативные варианты одновременно. Если в одном из них обнаружен неправильный выбор, этот вариант прекращает выполняться. Если в какомлибо варианте найдено решение, то это решение фиксируется, а все остальные варианты прекращают свою работу.

Класс NP алгоритмов – это все задачи, которые можно решить недетерминированными алгоритмами, работающими в течении полиномиального времени.

К этому классу относятся задачи:

1 Задачи о выполнимости - существует ли для данной булевой формулы, заданной в КНФ, такое распределение истинностных значений, в которых она принимает истинное значение?

2 Задача коммивояжера.

3 Решение систем уравнений с целыми переменными.

4 Составление расписаний, учитывающих определенные условия.

5 Размещение обслуживающих центров (телефон, телевидение, срочные службы) для максимального числа клиентов при минимальном числе центров.

6 Оптимальная загрузка емкости (рюкзак, поезд, корабль, самолет) при наименьшей стоимости.

7 Оптимизация раскроя (бумаги, картона, стального проката), маршрутов, инвестиций, станочного парка.

8 Задача распознавание простого числа. Самый лучший в настоящее время алгоритм имеет сложность порядка $O(L(n)^{L(L(L(n)))})$, где $L(n)$ – количество цифр в числе n.

Хотя в данных задачах не содержатся экспоненциальных вычислений, тем не менее для них до сих пор не разработаны эффективные (то есть полиномиальные) алгоритмы.

ЗАКЛЮЧЕНИЕ

Данное пособие восполняет имеющиеся пробелы в учебной литературе по математической логике и теории алгоритмов.

Издание рекомендуется для работы студентов при самостоятельном изучении теоретического материала, на практических занятиях в качестве справочного пособия, а также при выполнении типовых расчетов и подготовки к зачетам и экзаменам, предусмотренных учебными планами по указанной дисциплине.

Пособие поможет более глубокому и полному усвоению студентами ученого материала по математической логике и теории алгоритмов и будет способствовать эффективной организации учебного процесса по этой дисциплине.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Яблонский С.В. Введение в дискретную математику: учеб. пособие для вузов /С.В.Яблонский .-3-е изд. - М.: Высш. шк. 2002. - 384 с.

2. Судоплатов С.В. Элементы дискретной математики: учебник / С.В.Судоплатов, Е.В. Овчинникова. - М.: ИНФРА-М, Новосибирск, Изд-во НГТУ, 2002. - 280 с.

3. Новиков Ф.А. Дискретная математика для программистов: учебник /Ф.А. Новиков. - СПб.: Питер, 2002.- 304с.

4. Шелупанов А.А. Математическая логика и теория алгоритмов: учеб. пособие /А.А. Шелупанов, В.М.Зюльков. - Томск: STT, 2001. - 176 с.

5. Столл Р. Множества, логика, аксиоматические теории /Р. Стол. - М.: Просвещение, 1968.- 231с.

6. Криницкий Н.А. Алгоритмы вокруг нас /Н.А. Криницкий. – Изд. 2-е. - М.: Наука, 1984. – 224с.

7. Биркгоф Г. Современная прикладная алгебра /Г. Биркгоф, Т. Барти. - М.: Мир, 1976. - 400с.

8. Карпов Ю.Г. Теория автоматов: учебник для вузов /Ю.Г. Карпов. - СПб.: Питер, 2003.- 208с.

ОГЛАВЛЕНИЕ

Учебное издание

Ююкин Николай Алексеевич
Моисеев Сергей Игоревич
Федотенко Галина Федоровна

**МАТЕМАТИЧЕСКАЯ ЛОГИКА
И ТЕОРИЯ АЛГОРИТМОВ**

В авторской редакции

Подписано в печать 06.07.2007.
Формат 60×84/16. Бумага для множительных аппаратов.
Усл. печ. л. 5,6. Уч. изд. л. 4,2. Тираж 250 экз.
Заказ №

ГОУВПО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14

91