ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал ордена Трудового Красного Знамени
федерального государственного бюджетного образовательного учреждения
высшего образования
«Московский технический университет связи и информатики»

Е.А. РОМАНЕНКО

Методические указания для проведения практических занятий по дисциплине

«МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ»

Кафедра **«Информатика и вычислительная техника»**

Направление подготовки **09.03.01. Информатика и вычислительная техника**

Профиль **Программное обеспечение и интеллектуальные системы,**
Профиль **Вычислительный машины, комплексы, системы и сети**

Разработала:
Ст. преподаватель кафедры ИВТ Романенко Е. А.

Ростов-на-Дону
2019

Методические указания
для проведения практических занятий
по дисциплине
«Математическая логика и теория алгоритмов»

Составитель: Романенко Е.А., Ст. преподаватель каф. «ИВТ»

Рассмотрено и одобрено
на заседании кафедры «ИВТ»
Протокол от «26» августа 2019 г., № 1.

Практическое занятие №1.

Установление правил построения формул. Базис и индуктивный шаг в построении формул. Подформулы. Представление формулы в виде дерева Аксиомы - исходные тождественно истинные формулы. Проверка тождественной истинности аксиом: прямым вычислением значения формулы на каждом наборе; приведением аксиом к константе «1» путём эквивалентных преобразований.

Исчисление высказываний (ИВ), т. е. логика высказываний,— это формальная система, интерпретацией которой является алгебра высказываний. Под высказыванием понимается повествовательное предложение, о котором можно сказать, что оно истинно или ложно. Основной задачей исчисления высказываний является порождение общелогических законов — тождественно истинных высказываний, т. е. высказываний (в том числе составных), которые всегда истинны независимо от входящих в них элементарных высказываний.

Как и любая формальная система, исчисление высказываний строится на основе четырех основных процедур: задания алфавита, установления правил построения формул, аксиом и правил вывода. Определим все эти четыре компонента.

- 1. *Алфавит* состоит из символов трех категорий:
  - а) бесконечного счетного множества высказываний (или переменных высказываний), которые обычно обозначаются буквами: $x, y, z, a,$ *Ь*, $c, x_u\ x_2$ ит. д.;
  - б) логических операторов (или логических связок), которые обозначают символы логических операций (v , &, —> и т. д.);
  - в) открывающихся и закрывающихся скобок: ().

Других символов в ИВ нет.

2. *Правила построения формул.* Для обозначения формул обычно используют заглавные буквы латинского алфавита. Эти буквы не являются символами исчисления и служат для условного обозначения формул. Формулы в ИВ определяются следующим образом.

Формулы в исчислении высказываний однозначно получаются с помощью правил, которые описываются базисом и индуктивным шагом:

базис: всякое высказывание есть формула;

индуктивный шаг: если *X*и *Y*— формулы, то *X,* (I v К), *(X* —»F),

(X & *Y*) и т. д. — также формулы.

Никакая другая последовательность символов не является формулой.

Пример. Если *x,y, z* — формулы в соответствии с правилом базиса, то *(x* —> y), *(x & z)* — формулы в соответствии с правилом индуктивного шага. Очевидно, что не будут формулами: *Sex, (x* v *z,* так как они не удовлетворяют указанным правилам (в первом случае в бинарной операции используется один операнд, а во втором — отсутствует закрывающая скобка).

С введением понятия формулы вводится и понятие *подформулы* или части формулы, делается это следующим образом.

- 1. Подформулой элементарной формулы является только она сама.
- 2. Если *X* — формула, то ее подформулами будут: она сама, *X*и все подформулы *X*.
- 3. Обозначим *w* — любую из логических операций (v, &,-»), кроме отрицания. Тогда если (*XwY)* — формула, то ее подформулы: она сама, формулы *Xи У и* все подформулы *Xи У.*

Пример. Пусть задана формула (x v y) —& y|, определим ее подформулы и глубину их вложенности.

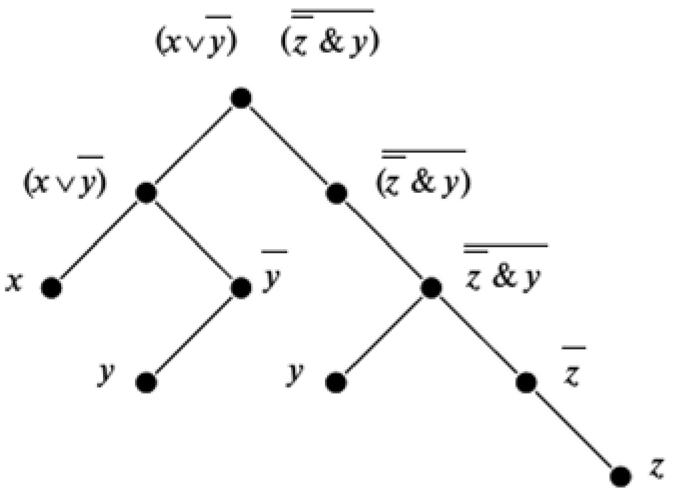
Представим решение в табличной форме

Подформула	Глубина
$\left(x \vee \bar{y}\right) \to \left(\overline{\bar{z} \,\&\, y}\right)$	0

$(x\vee\bar{y}),(\overline{\bar{z}\,\&\,y})$	1
$x,\bar{y},(\bar{z}\,\&\,y)$	2
$y,\bar{z}$	3
z	4

Кроме табличной формы каждая правильная формула может быть представлена в виде дерева, ветви которого — исходные и промежуточные формулы.

Из примера видно, что на самом низком уровне находятся только элементарные формулы, однако они могут быть и на более высоких уровнях.



Система аксиом исчисления высказываний состоит из 11 аксиом, которые делятся на четыре группы.

Первая группа аксиом выражает импликацию:

I_1 $x(yx)$.

I_2 $(x(yz))((x\,y)(x\,z))$.

Вторая группа аксиом выражает конъюнкцию:

II_1 $x\wedge\, yx$.

II_2 $x\wedge\, y\,y$.

II_3 $(zx)((z\,y)(zx\wedge\, y))$.

Третья группа аксиом выражает дизъюнкцию:

III$_1$ $x x y$.

III$_2$ $y x y$,

III$_3$ $(x z)((y z)(x y z))$.

Четвертая группа аксиом выражает отрицание:

IV $_1$ $(x y)(\acute{y} \rightarrow \acute{x})$

IV$_2$ $\acute{x} \rightarrow x$.

IV$_3$ $x \rightarrow \acute{x}$.

Проверка тождественной истинности аксиом: прямым вычислением значения формулы на каждом наборе; приведением аксиом к константе «1» путём эквивалентных преобразований.Основной задачей, которую ставит перед собой математическая логика, является формализация мыслительных процессов в широком смысле - в смысле доказательства или поиска истины, а также расширения знаний. Для этого надо сформулировать "правильное", непротиворечивое логическое исчисление, включающее:

- Формализацию представления высказываний;
- Набор операций, позволяющий создавать формулы, как более сложные высказывания на базе более простых;
- Аксиомы и правила вывода, позволяющие выводить логические следствия из любой заданной системы высказываний и формально характеризовать тождественно истинные высказывания и формулы.

Тождественно истинными являются высказывания, которые не требуют задания каких-либо дополнительных условий для проверки своей истинности на протяжении всего процесса логического вывода. Например, высказывание "кислород является газом" имеет значение ИСТИНА (1) в тех условиях, в которых оно рассматривается. Ведь кислород может находиться не только в газообразном состоянии.

Высказывания, как постоянные, так и переменные, можно объединять в сложные высказывания, используя связки "и", "или", "если - то", "не" и т.п. Если в состав сложного высказывания входят переменные высказывания, представленные буквами, то при замене этих букв одними высказываниями сложное высказывание может оказаться истинным, а при замене другими - ложными. Например, сложное высказывание "А и В", где А и В - переменные высказывания, будет, очевидно, истинным в том и только том случае, если оба высказывания будут истинными.

Существуют и такие сложные высказывания, содержащие в своём составе переменные высказывания, которые остаются истинными при любых значениях переменных высказываний. Например, сложное высказывание "если неверно то, что высказывание А ложно, то высказывание А истинно" остаётся истинным, какое бы высказывание ни подставили на место переменного высказывания А. Такие высказывания называют тождественно истинными. Например, возьмём высказывание "извозчик Петров - водитель кобылы". Если неверно то, что Петров не водитель кобылы, значит он водитель кобылы. Сложное высказывание истинно. Если же неверно, что Петров водитель кобылы, значит он не водитель кобылы. Сложное высказывание снова истинно, ибо соответствует правде и не содержит противоречий.

Задача выделения тождественно истинных высказываний во множестве всех возможных высказываний является важнейшей задачей любого логического исчисления.

Например, рассмотрим формулу $\Omega = A \wedge B \to A$ и составим для неё таблицу истинности с учётом "промежуточного" результата $A \wedge B$

A	B	$A \wedge B$	$A \wedge B \to A$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

Видим, что при всех значениях переменных высказываний, сложное высказывание является истинным. Следовательно, рассмотренная формула Ω тождественно истинная.

Рассмотрим тот же пример:

$$A \wedge B \to A = \neg(A \wedge B) \vee A = \neg A \vee \neg B \vee A = (\neg A \vee A) \vee \neg B = 1 \vee \neg B = 1.$$

Эта цепочка доказывает тождественную истинность заданной формулы.

Задание.

Доказать тождественную истинность следующих аксиом:

a. $x \to (y \to x)$;

b. $(x \to (y \to z)) \to ((x \to y) \to (x \to z))$;

c. $x \,\&\, y \to x$;

d. $(z \to x) \to ((z \to y) \to (z \to x \,\&\, y))$;

e. $y \to x \vee y$;

f. $(x \to z) \to ((y \to z) \to (x \vee y \to z))$;

g. $(x \to y) \to (\neg y \to \neg x)$;

h. $(z \to (y \to x)) \to ((z \to y) \to (z \to x))$;

i. $(z \to x) \to ((z \to y) \to (z \to x \,\&\, y))$;

j. $(y \to z) \to (\neg z \to \neg y)$.

Практическое занятие №2.
Термин и определение в исчислении высказываний. Определение выполнимости (невыполнимости) формул. Определение общезначимости и нейтральности формул.

Формула А логики предикатов называется *выполнимой в области М,* если существуют значения переменных входящих в эту формулу и отнесенных к области М (иначе – существует модель), при которых формула А принимает истинные значения.

Формула А логики предикатов называется *выполнимой*, если существует область, на которой эта формула выполнима.

Формула А логики предикатов называется *тождественно-истинной в области М* (выполнимой), если она принимает истинные значения для всех значений переменных, входящих в эту формулу и отнесенных к этой области.

Формула А логики предикатов называется *общезначимой*, если она тождественна истинна на всякой области (на любой модели).

Если две равносильные формулы логики предикатов соединить знаком эквиваленции $\Leftrightarrow$, то полученная формула будет принимать значение И для любого набора переменных в любой области, т.е. будет общезначимой. Это понятие является обобщением понятия тождественной истинности формулы логики высказываний. Все логические законы, представленный в логике высказываний формулами являются общезначимыми формулами логики предикатов и выражают, как и другие общезначимые формулы, законы логики на языке логике предикатов. Общезначимость формулы логики предикатов, например, F обозначается $\vdash$F. Все общезначимые формулы могут быть источниками новых $\vdash$ формул. Например, подставляя в закон исключенного третьего $x \vee \bar{x}$ – вместо x предикат $P(x_1,\ldots,x_n)$, получаем общезначимую формулу $P(x_1,\ldots,x_n) \vee \overline{P}(x_1,\ldots,x_n)$. При n=1 имеем общезначимую формулу

$P(x) \vee \overline{P}(x)$, и, таким образом, $\forall x[P(x) \vee \overline{P}(x)]$ - общезначимая формула логики предикатов.

Из тождественно истинной формулы логики высказываний $x \vee y \equiv y \vee x$ подстановкой вместо x предиката P(x, y), а вместо y- предиката Q(x,y) получаем общезначимую формулу $P(x, y) \vee Q(x, y) \equiv Q(x, y) \vee P(x, y)$ и т. д.

Формула А логики предикатов называется *тождественно ложной в области М,* если она принимает ложные значения для всех значений переменных, входящих в эту формулу и отнесенных к этой области (иными словами, на данной модели).

Формула А логики предикатов называется *тождественно ложной (невыполнимой),* если она тождественно ложна на всякой области (на всякой модели).

Например, формула $\forall x[P(x) \wedge \overline{P}(x)]$ является тождественно ложной (невыполнимой) формулой логики предикатов.

Из приведенных определений с очевидностью следует:

1. Если формула А общезначима, то она и выполнима на всякой области (модели).

2. Если формула А тождественно истинна в области М, то она и выполнима в этой области .

3. Если формула А тождественно ложна в области М , то она не выполнима в этой области .

4. Если формула А не выполнима, то она тождественно ложна на всякой области (на всякой модели).

5. Для того, чтобы формула А логики предикатов была общезначима, необходимо и достаточно, чтобы ее отрицание было не выполнимо.

6. Для того, чтобы формула А логики предикатов была выполнимой, необходимо и достаточно, чтобы формула $\overline{A}$ была не общезначима.

Пример 1.Доказать равносильность (логическое тождество):

$$(\exists x \forall y \overline{P}(x, y) \vee \exists u \forall v P(v, u)) \wedge (\forall v \exists u P(v, u) \wedge \exists y \forall x P(x, y)) \equiv \exists x \forall y P(y, x) \wedge \forall u \exists v P(u, v)$$

Заметив, что в каждой из кванторных подформул обе предметные переменные связаны и что, таким образом, они являются высказываниями, введем обозначения:

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

B= $\exists u \forall v P(v, u) = \exists y \forall x P(x, y) = \exists x \forall y P(y, x)$, или обозначив первую и вторую предметные переменные через n1 и n2, соответственно:

A= $\forall n_1 \exists n_2 P(n_1, n_2)$, B= $\exists n_2 \forall n_1 P(n_2, n_1)$.

В этих обозначениях заданное для рассмотрения тождество будет выглядеть так: $(\overline{A} \vee B) \wedge (A \wedge B) \equiv B \wedge A$.

Произведя равносильные преобразования, можем убедиться в справедливости этого тождества: $(\overline{A} \vee B)AB \equiv \overline{A}AB \vee BAB \equiv \text{Л}B \vee BA \equiv BA$

Если охарактеризовать рассматриваемое выражение в целом, то видим, что это общезначимая формула.

Пример 2.Определить тип формулы $\exists x \exists y [P(x) \wedge \overline{P}(y)] = F$. .

Пусть P(x) : “ Число x - четно –” предикат, определенный в $M=N^2$.

Таким образом, рассматриваемая формула на данной модели представляет собой следующее утверждение: “ Среди натуральных чисел существуют как четные, так и нечетные ”. Очевидно, что это высказывание истинно и, таким образом, на данной модели формула F тождественно истинна.

Однако, если этот же предикат задать на множестве M=NxN,где N – множество четных чисел, то формула F на такой модели окажется тождественно ложной.

Учитывая изложенное, заключаем, что рассматриваемая формула F выполнима (но не общезначима).

Пример 3.Для формулы $\exists y P(x, y, z)$ подобрать модель, на которой она является тождественно истинной (и, таким образом, в целом выполнимой).

Пусть P(x, x, y): “x·x=y”, или иначе “x^2=y” – предикат, определенный на множестве натуральных чисел, т.е. M=N. Тогда рассматриваемая формула выражает утверждение о существовании натурального квадрата натурального числа, что, очевидно, является истиной, т.е. на данной модели формула тождественно истинна, что и требовалось доказать.

Пример 4.Рассмотрим формулу $\forall x P(x, y, x)$. Это выполнимая формула. Действительно, если P(x, y, x): “x+y=x” – предикат суммы, то на M=N существует подстановка вместо y соответствующего значения, дающего значение истинности данной формуле. Очевидно, это y=0, поскольку в этом случае получаем “x=x”.

Если же P(x, y, x): “xy=x” – предикат произведения, то таким значением y является y=1, так как при нем получаем истинное высказывание $\forall x(x \cdot 1 = x)$.

Но это единственные подстановки, приводящие к верным утверждениям, что и говорит именно о выполнимости данной формулы (но не об ее общезначимости).

Пример 5.Является ли общезначимой формула: $\exists x P(x, y) \to \forall x P(x, y)$?

Пусть P(x, y) – предикат порядка (бинарного отношения) “ $x \le y$ ”, определенный на конечном множестве натуральных чисел M_1. Тогда при подстановке в формулу вместо свободной переменной y величины $y = a_{max} \in M_1$ мы получим истинное утверждение, а при подстановке любой другой константы из множества M_1 – ложное. Таким образом, рассматриваемая формула не является общезначимой.

Пример 6.Рассмотрим формулу $\overline{A(y)} \wedge \forall z A(z)$. Покажем, что она невыполнима.

Допустим противное, т.е. что она выполнима. Это означает, что существует такое множество M и такой конкретный предикат A^0 в нем, что когда $y, z \in M$, то данная формула превращается в такой конкретный предикат $B(y) = \overline{A^{\diamond}(y)} \wedge \forall z A^{\diamond}(z)$, который, в свою очередь, превращается в истинное высказывание при всякой подстановке вместо у элементов из множества М. Возьмем любое $y^{\diamond} \in M$. Тогда высказывание $B(y^{\diamond}) = \overline{A^{\diamond}(y^{\diamond})} \wedge \forall z A^{\diamond}(z)$ истинно, как мы только что установили. Следовательно, истинны высказывания $\overline{A^{\diamond}(y^{\diamond})}$ и $\forall z A^{\diamond}(z)$.

Из истинности второго высказывания заключаем, что высказывание $A^{\diamond}(y^{\diamond})$ истинно (поскольку “для всех предметных переменных”, как бы они ни обозначались). Но это противоречит истинности первого высказывания $\overline{A^{\diamond}(y^{\diamond})}$.

Таким образом, наше предположение о выполнимости формулы неверно.

Проблема разрешимости в логике предикатов ставится так же, как и в алгебре логики: существуют ли алгоритмы, позволяющие для любой формулы А логики предикатов установить, к какому типу (классу) она относится, т.е. является ли она общезначимой, выполнимой или тождественно ложной (невыполнимой). Если бы такой алгоритм существовал, то, как и в алгебре высказываний, он сводился бы к критерию тождественной истинности любой формулы логики предикатов. Отметим, что, в отличие от алгебры логики, в логике предикатов не применим метод перебора всех вариантов значений переменных, входящих в формулу, так как таких вариантов может быть бесконечное множество.

Задания.

Требуется доказать общезначимость заданной формулы:

1. $(x \to y) \to ((y \to z) \to (x \to z))$;
2. $((p \,\&\, q) \to r) \to (p \to (q \to r))$;
3. $(a \to b) \to (a \vee c \to b \vee c)$;
4. $(a \to b) \,\&\, (b \to c) \to (a \to c)$;
5. (

$a \to (b \to c)) \to (a \& b \to c)$.

Практическое занятие №3.

Построение резольвенты. Обновление множества дизъюнктов.

Существует эффективный алгоритм логического вывода - алгоритм резолюции. Этот алгоритм основан на том, что выводимость формулы B из множества посылок F_1; F_2; F_3; . . . F_n равносильна доказательству теоремы

$\vdash (F_1 \& F_2 \& F_3 \& \ldots \& F_n \to B)$,

формулу которой можно преобразовать так:

$\vdash (F_1 \& F_2 \& F_3 \& \ldots \& F_n \to B) =$

$\vdash (\neg(F_1 \& F_2 \& F_3 \& \ldots \& F_n) \vee B) =$

$\vdash \neg(F_1 \& F_2 \& F_3 \& \ldots \& F_n \& (F_2 \neg B))$.

Следовательно, заключение B истинно тогда и только тогда, когда формула $(F_1 \& F_2 \& F_3 \& \ldots \& F_n \& (\neg B))$=л. Это возможно при значении “л” хотя бы одной из подформул F_i или $\neg B$.

Для анализа этой формулы все подформулы F_i и $\neg B$ должны быть приведены в конъюнктивную нормальную форму и сформировано множество дизъюнктов, на которые распадаются все подформулы. Два дизъюнкта этого множества, содержащие пропозициональные переменные с противоположными знаками (*контрарные атомы)* формируют третий дизъюнкт - *резольвенту*, в которой будут исключены контрарные пропозициональные переменные. Неоднократно применяя это правило к множеству дизъюнктов и резольвент, стремятся получить пустой дизъюнкт. Наличие пустого дизъюнкта свидетельствует о выполнении условия $F_1 \& F_2 \& F_3 \& \ldots \& F_n \& \neg B$=л.

Алгоритм вывода по принципу резолюции

Шаг 1. принять отрицание заключения, т.е. $\neg B$;

Шаг 2. привести все формулы посылок и отрицания заключения к конъюнктивной нормальной форме ;

Шаг 3. выписать множество дизъюнктов всех посылок и отрицания заключения: $K = \{D_1; D_2; \ldots D_k\}$;

Шаг 4. выполнить анализ пар множества K по правилу:

“если существуют дизъюнкты D_i и D_j, один из которых (D_i) содержит литеру A, а другой (D_j) - контрарную литеру ⌉A, то соединить эту пару логической связкой дизъюнкции ($D_i \vee D_j$) и сформировать новый дизъюнкт - резольвенту, исключив контрарные литеры A и ⌉A;

Шаг5. если в результате соединения дизъюнктов, содержащих контрарные литеры, будет получена пустая резольвента - □, то конец (доказательство подтвердило противоречие), в противном случае включить резольвенту в множество дизъюнктов K и перейти к шагу 4.

<u>Пример:</u> Работа автоматического устройства, имеющего три клапана A, B и C, удовлетворяет следующим условиям: если не срабатывают клапаны A или B или оба вместе, то срабатывает клапан C; если срабатывают клапаны A или B или оба вместе, то не срабатывает клапан C. Следовательно, если срабатывает клапан C, то не срабатывает клапан A .

$$\frac{((\acute{A}\acute{B}\acute{A}\acute{B})C);((ABAB)\acute{C})}{(C\acute{A})}$$

1) $F_1 = (\acute{A}\acute{B}\acute{A}\acute{B})C$¿$= (A \vee C) \& (B \vee C)$ - посылка;

2) $F_2 = (ABAB)\acute{C} = (\acute{A}\acute{C})(\acute{B}\acute{C})$ -посылка;

3) $F_3 = (C\acute{A}$¿) ¿$= C \& A$ –отрицание заключения;

1) множество дизъюнктов: $K = \{(A \vee C); (B \vee C); (\acute{A}\acute{C}); (\acute{B}\acute{C}); C; A\}$;

2) $C(\acute{A}\acute{C}) = \acute{A}$ – резольвента из 2) и 3);

3) $K_1 = \{(A \vee C); (B \vee C); (⌉A \vee ⌉C); (⌉B \vee ⌉C); C; A; ⌉A\}$;

4) $⌉A \vee (A \vee C) = C$ – резольвента из 1) и 5);

5) $K_2 = \{(A \vee C); (B \vee C); (⌉A \vee ⌉C); (⌉B \vee ⌉C); C; A; ⌉A\}$;

6) $C \vee (⌉B \vee ⌉C) = ⌉B$ –резольвента из 2) и 5);

10) $K_3 = \{(A \vee C); (B \vee C); (⌉A \vee ⌉C); (⌉B \vee ⌉C); C; A; ⌉A; ⌉B\}$;

11) $⌉B \vee (B \vee C) = C$ – резольвента из 1) и 9);

12) $C \vee \neg A = (C \vee \neg A)$ – резольвента из 5) и 11);

13) $K_4 = \{(A \vee C); (B \vee C); (\neg A \vee \neg C); (\neg B \vee \neg C); C; A; \neg A; \neg B; (C \vee \neg A)\}$;

14) $(C \vee \neg A) \vee (\neg A \vee \neg C) = \neg A$ – резольвента из 2) и 12);

15) $K_5 = \{(A \vee C); (B \vee C); (\neg A \vee \neg C); (\neg B \vee \neg C); C; A; \neg A; \neg B; (C \vee \neg A)\}$;

16) $\neg A \vee A = \square$ - пустая резольвента.

Так доказано, что если срабатывает клапан С, то не срабатывает клапан А.

Задания.

Докажите выводимость заключения

1)методом дедукции

2) методом резолюций.

1) $\dfrac{(A B);(A C);(B D)}{C D}$

2) $\dfrac{(\acute{A} B);(C \acute{B})}{A \acute{C}}$

3) $\dfrac{(A B)(C D);(D E) F}{A F}$

4) $A B ¿ C; C D M; M N; \acute{D} \acute{N} \dfrac{¿}{\acute{A}}$

5) $\dfrac{(A B)(C D); D B E; \acute{E}}{\acute{C} \acute{A}}$

6) $\dfrac{A; B; A C \acute{B}}{\acute{C}}$

7) $\dfrac{(A B);(A B);(B A)}{A B}$

Практическое занятие №4.
Примеры построения предикатных выражений.

Высказывания в алгебре логики рассматриваются как единый объект с точки зрения истинности или ложности. Структура и содержание высказываний не рассматриваются. Однако на практике для построения полноценного логического вывода важно иметь представление о структуре и содержании используемых в выводе высказываний. Поэтому логика предикатов является расширением логики высказываний, которую включает в себя в качестве составной части.

Одноместным предикатом P(x) называется произвольная функция переменной x , определённая на множестве M и принимающая значение из множества {0, 1}. $P(x_1, x_2, \ldots, x_n)$ можно трактовать, что переменные $x_1, x_2, \ldots, x_n$ связаны отношением P. n – местный предикат – это двузначная функция от n аргументов, определённая на произвольном множестве M , принимающая значение 0 или 1.

Пример. Предикат P(x) – «x – простое число» определён на множестве N. Предикат Q(x) – «sin x = 0» определён на множестве R.

Предикаты, так же как высказывания, принимают два значения (1, 0) истина и ложь. Кроме логических операций в логике предикатов вводятся кванторные операции, что позволяет повысить выразительную мощность предикатных предложений. Как известно из курса «Дискретная математика» кванторов всего два($\forall, \exists$).

Пример. Пусть предикат P(x, y) – «$x \le y$ » определён на множестве N.

Предикатное выражение $\exists x \forall y\, P(x, y)$ означает - Существует x, который меньше любого y – значение истинности 1.

Предикатное выражение $\exists x \exists y\, P(x, y)$ означает - Существуют такие x и y, что $x \le y$ - значение истинности 1.

Предикатное выражение $\forall x \forall y\, P(x, y)$ означает - Для любого x и любого y имеет место $x \le y$ – значение истинности 0.

Логику предикатов можно рассматривать как формальную систему. О логическом значении формулы логики предикатов можно говорить лишь тогда, когда задано множество M, на котором определены входящие в формулу предикаты. При задании аргументам предикатов конкретных значений предикаты становятся высказываниями и можно говорить об их истинности или ложности.

Задание.

Доказать тождественную истинность или тождественную ложность предикатных выражений, преобразовав их по правилам, регламентирующим преобразование выражений в исчислении предикатов, содержащих кванторы.

(a) $\forall x(P(x) \rightarrow (P(x) \vee P(y)));$

(b) $\exists x \exists y((F(x) \rightarrow F(y)) \& (F(x) \rightarrow \neg F(y)) \& F(x));$

(c) $\forall x(q \rightarrow p_1(x)) \leftrightarrow (q \rightarrow \forall x p_1(x));$

(d) $\forall x(F1(x) \rightarrow F2(x)) \rightarrow (\forall x F1(x) \rightarrow \forall x F2(x));$

(e) $\forall x(p_1(x) \rightarrow p_2(x)) \leftrightarrow (\exists x p_1(x) \rightarrow \forall x p_2(x));$

(f) $\exists x R(x) \vee \exists x Q(x) \leftrightarrow \exists x(R(x) \vee Q(x));$

При выполнении решения задач показать, обосновать все шаги доказательства и привести результат.

Практическое занятие №5.

Машина Тьюринга. Решение задач на применимость машины Тьюринга (работающей по заданной программе Р) к заданному слову S.

Лента

^	0	1	0	0	1	1	0	1	^

ЛЮБОЙ АЛФАВИТ ДОЛЖЕН СОДЕРЖАТЬ ПУСТОЙ СИМВОЛ $\wedge$
УУ – НАХОДИТСЯ В ОДНОМ ИЗ СОСТОЯНИЙ Q
ВЫДЕЛЯЕТСЯ НАЧАЛЬНОЕ СОСТОЯНИЕ $\mathbf{q_1}$
ЗАКЛЮЧИТЕЛЬНОЕ СОСТОЯНИЕ $\mathbf{q_0}$
НАЧАЛЬНЫЙ МОМЕНТ – ОБОЗРЕВАЕТСЯ КРАЙНЯЯ ЛЕВАЯ НЕПУСТАЯ ЯЧЕЙКА ЛЕНТЫ И **УУ** НАХОДИТЬСЯ В СОСТОЯНИИ $\mathbf{q_1}$

Правила преобразования, которые выполняет МТ, определяются системой команд. Работа МТ может быть описана следующим образом: системой команд, таблицей, диаграммой (графом) переходов.

Пример реализации МТ с помощью системы команд.

Выяснить, применима ли машина Тьюринга, заданная программой P к слову S и, если применима, то указать результат применения машины Тьюринга к заданному слову.

При решении задачи следует учесть, что в начальный момент времени головка машины обозревает самую левую ячейку на ленте и устройство управления (УУ) находится в начальном состоянии q_1.

$$
P = \begin{array}{|l}
q_1 0 \rightarrow 0 q_3 \\
q_1 1 \rightarrow 1 R q_1 \\
q_2 0 \rightarrow 0 L q_3 \\
q_2 1 \rightarrow 1 R q_3 \\
q_3 0 \rightarrow 0 R q_0 \\
q_3 1 \rightarrow 1 L q_3
\end{array}
\qquad S = 111101
$$

Решение.

Машина Тьюринга, заданная программой P, применима к слову, если она закончит работу за конечное число шагов. Предполагается, что слово S записывается с левого конца ленты, начиная с первой ячейки. Все правые ячейки после окончания слова заполняются пустыми символами. Начальное состояние машины, при котором головкой обозревается первая ячейка и УУ находится в начальном состоянии q1 изображено далее в виде двух строк: на первой строке – лента с содержимым в её ячейках, а на второй строке – головка с текущим состояниям УУ.

$\wedge$	1	1	1	1	0	1	$\wedge$	$\wedge$	$\wedge$
	q_1								

Из приведённого рисунка первый шаг работы машины определяет адрес $q_1 1$. По этому адресу следует выполнить команду $1Rq_1$. В обозреваемую ячейку записывается 1, головка сдвигается на одну ячейку вправо и УУ переходит в состояние q_1 .

Результаты выполнения этой и последующих команд отражены ниже.

$\wedge$	1	1	1	1	0	1	$\wedge$	$\wedge$	$\wedge$
		q_1							
$\wedge$	1	1	1	1	0	1	$\wedge$	$\wedge$	$\wedge$
			q_1						
$\wedge$	1	1	1	1	0	1	$\wedge$	$\wedge$	$\wedge$
				q_1					
$\wedge$	1	1	1	1	0	1	$\wedge$	$\wedge$	$\wedge$
					q_1				
$\wedge$	1	1	1	1	0	1	$\wedge$	$\wedge$	$\wedge$
					q_3				
$\wedge$	1	1	1	1	0	1	$\wedge$	$\wedge$	$\wedge$
						q_0			

$q_3 0 \rightarrow 0Rq_0$. В текущую ячейку записывается 0, головка сдвигается на одну ячейку вправо. УУ переходит в состояние q_0 (заключительное состояние машины).

Машина применима к данному слову и результат её работы 111101

Контрольные вопросы к выполнению заданий практического занятия.

1. Определение алгоритма и основные требования, применяемые к алгоритму?
2. Уточнение понятия алгоритма с помощью машины Тьюринга

(МТ)?

3. Элементарные операторы – сингулярный и бинарный?
4. Способы описания алгоритмов (словесное, линейная форма записи, в виде структурной схемы)?
5. Рекурсивные функции?
6. Частично рекурсивная функция?
7. Управляющее устройство машины Тьюринга, лента, устройство обращения к ленте?
8. Два варианта работы МТ?
9. Правила преобразования, выполняемые МТ?
10. Описание работы МТ таблицей?
11. Описание работы МТ диаграммой (графом) переходов?

Задания.

Выяснить, применима ли машина Тьюринга, заданная программой P к слову S и, если применима, то указать результат применения машины Тьюринга к заданному слову.

0.

$$P = \left| \begin{array}{l} q_1 0 \to 0 R q_2 \\ q_1 1 \to 1 R q_1 \\ q_2 0 \to 0 R q_2 \\ q_2 1 \to 1 R q_3 \\ q_3 0 \to 0 L q_0 \\ q_3 1 \to 1 R q_3 \end{array} \right.$$

S= 110101

1.

$$P = \left| \begin{array}{l} q_1 0 \to 1 q_2 \\ q_1 1 \to 1 R q_1 \\ q_2 0 \to 0 L q_3 \\ q_2 1 \to 0 R q_3 \\ q_3 0 \to 0 R q_0 \\ q_3 1 \to 1 L q_3 \end{array} \right.$$

S= 111101

2.

$$\left| \begin{array}{l} q_1 0 \to 1 R q_2 \\ q_1 1 \to 1 R q_1 \\ q_2 0 \to 0 L q_3 \\ q_2 1 \to 0 q_2 \\ q_3 0 \to 0 R q_2 \\ q_3 1 \to 1 L q_0 \end{array} \right.$$

S= 101111

Практическое занятие №6.

Нормальные алгоритмы А. А. Маркова. Определение. Примеры реализации алгоритма Маркова.

Основная операция при работе нормальных алгоритмов или алгоритмов Маркова – это переработка слов в некотором алфавите А.

Определение 5.11. Алфавитом называется всякое непустое конечное множество символов, а сами символы – буквами. Словом в алфавите А называется всякая конечная последовательность букв данного алфавита.
Эта переработка заключается в производстве некоторого количества замен определенных последовательностей символов. Эти замены совершаются в строго определенном порядке, а именно: после каждой замены алгоритм читается с самого начала, а слово анализируется с самого первого символа. В отличие от машин Тьюринга, алгоритмы Маркова выполняются без какого – либо устройства, осуществляющего движения и имеющего внутреннюю память.
Алгоритм представляет собой совокупность строк определенного вида, причем порядок строк имеет важнейшее значение. Формат строки следующий:

$\{a_i\} \rightarrow \{b_j\}$ [•], где

$\{a_i\}$ – последовательность символов, которая ищется в слове

$\rightarrow$ – знак перехода к операции записи

$\{b_j\}$ – последовательность символов, которая записывается вместо найденной

[•] – знак принудительного окончания алгоритма (необязательный параметр).

Окончание работы алгоритма происходит в тот момент, когда выполняется строка, содержащая знак принудительной остановки, либо тогда, когда более ни одна строка не может быть выполнена (в слове нет ни одной из искомых последовательностей символов).
Например, алгоритм, состоящий из одной строки, вида 0 → * будучи примененным к слову в алфавите {0,1}, заменит все нули на звездочки.
В свою очередь алгоритм0 → * • будучи примененным к слову в алфавите {0,1}, заменит на звездочку первый встреченный ноль.

Например, дано произвольное двоичное слово. Надо убрать из него два первых знака. Рассмотрим алгоритм вида:

00 → •

01 → •

10 → •

$11 \rightarrow \bullet$

Если в слове случайно первыми двумя символами были нули (например, «001011»), то алгоритм действительно выполнит указанную задачу. Также работа закончится успешно, если в слове ни разу не встретилось два нуля подряд, а первыми символами оказалась пара 01 (например, «01011101»). Но в слове 1100101 выбросятся два нуля, которые вовсе не являются первыми символами слова. В этом случае существующий алфавит надо расширить вспомогательными буквами, которых нет в начальном слове, и которые появляются в ходе вычисления. По сути, это некоторые аналоги внутренней памяти (состояний машин Тьюринга). Они вводятся с помощью формулы $\lambda \rightarrow \alpha$ (α – вспомогательная буква) или, что более корректно, пары формул

$\lambda\, 0 \rightarrow \alpha\, 0$

$\lambda\, 1 \rightarrow \alpha\, 1$

Применив такие правила к слову λ 1100101 λ, получим: $\lambda\ \alpha$ 1100101 λ.

Дальше мы перемещаем эту букву по слову вправо, стирая и отсчитывая символы

(стерли первый):

$\alpha\, 0 \rightarrow \beta$

$\alpha\, 1 \rightarrow \beta$

(стерли второй):

$\beta\, 0 \rightarrow \bullet$

$\beta\, 1 \rightarrow \bullet$

Однако если мы расположим эти строки в обычном порядке, а именно:

$\lambda\, 0 \rightarrow \alpha\, 0$

$\lambda\, 1 \rightarrow \alpha\, 1$

$\alpha\, 0 \rightarrow \beta$

$\alpha\, 1 \rightarrow \beta$

$\beta\, 0 \rightarrow \bullet$

$\beta\, 1 \rightarrow \bullet$

алгоритм будет работать совсем не так, как хотелось бы. В частности вся его деятельность будет сводиться к созданию бесконечно большого числа символов α поочередно со стиранием символов слова. Это связано с тем, что после выполнения каждой замены управление передается снова первой строке, а слово анализируется с левого символа. Поэтому чаще всего алгоритм пишется как бы «снизу-вверх», т.е. в самом начале ставятся строки, относящиеся к группе «окончание алгоритма», далее «тело программы» и в самом низу блок «инициализация», которая будет выполняться только однажды, а затем управление перейдет к более ранним строкам.

Правильный алгоритм выглядит следующим образом.

β 0 → •

β 1 → •

α 0 → β

α 1 → β

λ 0 → α 0

λ 1 → α 1

Задания.

Схема алгоритма U: $a \to b$, $b \to b$, $c \to \cdot a$. Определить результат действия алгоритма U на слово w.

abbbbcca
abbabbca
baabcca
abacbcca

Пример.

Действие алгоритма U на слово W=acabbbccc:

U (acabbbccc) = bU(cabbbccc) = ba← заключительная подстановка $c \to \cdot a$