МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Государственное образовательное учреждение

высшего профессионального образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра автоматизации обработки информации

Утверждаю:
Зав. каф. АОИ
профессор
___________Ю.П. Ехлаков
«__» _____________2015 г.

Методические указания по выполнению
самостоятельной работы
по дисциплине

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

для студентов специальности
230102 - «Автоматизированные системы обработки
информации и управления»

Разработчик:
доцент каф. АОИ
__________ Т.О. Перемитина

Томск – 2015

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1. ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ 3

2. СТРУКТУРА САМОСТОЯТЕЛЬНОЙ РАБОТЫ 3

3. МЕТОДИКА РЕАЛИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ТЕОРЕТИЧЕСКОМУ КУРСУ 4

4. СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ 4

4.2 Нормальные формы формул 8

4.3 Логические рассуждения 11

4.4 Логика предикатов 13

4.5 Основы теории алгоритмов 15

ОЦЕНКА ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТА 19

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА 19

ВВЕДЕНИЕ

Целью данного курса является изучение теоретических и алгоритмических основ базовых разделов математической логики и теории алгоритмов.

Задачи изучения дисциплины следующие:

- получить знания об основах логики высказываний, логики предикатов, нечеткой логики и теории алгоритмов;
- употреблять специальную математическую символику для выражения количественных и качественных отношений между объектами;
- знать основные методы и алгоритмы математической логики, связанные с моделированием и оптимизацией систем различной природы;
- уметь строить и анализировать алгоритмы для решения дискретных задач.

Данные методические указания предназначены для выполнения самостоятельной работы по дисциплине «Матема» подготовки специалистов 231002.65 «Автоматизированные системы обработки информации и управления».

1. ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ

Самостоятельная работа является важной составляющей в изучении дисциплины и состоит из следующих видов деятельности: самостоятельное изучение теоретического материала, подготовка к выполнению практических заданий.

Самостоятельная работа над теоретическим материалом направлена на изучение основных понятий математической логики и теории алгоритмов.

2. СТРУКТУРА САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа является важной составляющей в изучении дисциплины и состоит из следующих видов деятельности: самостоятельное изучение теоретического материала, выполнение и защита курсового проекта.

Объем и виды самостоятельной работы в структуре дисциплины приведены в табл.1.

Тематика самостоятельной работы (детализация)	Контроль выполнения работы
Подготовка к тестовому опросу на лекции	Тестовый опрос на лекции

Подготовка к контрольным работам	Контрольные работы
Подготовка к практическим занятиям	Оценка работы на практическом занятии

3. МЕТОДИКА РЕАЛИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ТЕОРЕТИЧЕСКОМУ КУРСУ

При самостоятельном изучении теоретического курса студентам необходимо:

– самостоятельно изучить темы теоретического курса в соответствии с учебной программой дисциплины;

– проработать тестовые вопросы;

– подготовить устные ответы на контрольные вопросы, приведенные после каждой темы.

Самостоятельную работу выполняют студенты на основе учебно-методческих материалов дисциплины.

Темы для самостоятельного изучения, тесты и контрольные задания преподаватель выдает на лекционных занятиях в соответствии с графиком. Некоторые задания выполняют по определенному варианту, номер которого определяет преподаватель.

4. СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

4.1 Алгебра высказываний

4.1.1 Содержание темы

Математическая логика и ее применение. Понятие высказывания. Логические операции. Формулы логики высказываний. Равносильные преобразования формул.

Литература: [1, 2, 3,5].

4.1.2 Методические указания

Логика изучает формальные законы мышления. Для изучения дисциплины “Математическая логика и теория алгоритмов” требуется, чтобы студент владел базовыми понятиями теории множеств (множество, подмножество, отношение, функция и т. д.). В основе стандартной (классической) логики лежат логика высказываний (пропозициональная логика) и логика предикатов.

Высказывание это повествовательное предложение, в отношении которого имеет смысл утверждение об его истинности или ложности. Пример истинного высказывания: "Земля вращается вокруг Солнца". Пример ложного высказывания: "3 > 5".

Не всякое предложение является высказыванием, к высказываниям не относятся вопросительные и восклицательные предложения. Не является высказыванием предложение: «Каша – вкусное блюдо», так как не может быть единого мнения о том, истинно оно или ложно. Предложение «Есть жизнь на Марсе» следует считать высказыванием, так как объективно оно либо истинно, либо ложно, хотя никто пока не знает, какое именно.

Поскольку предметом изучения логики являются только значения истинности высказываний, для них вводят буквенные обозначения А, В, С, X, Y...

Считается, что каждое высказывание либо истинно, либо ложно. Для краткости, будем вместо значения истинно писать 1, а вместо значения ложно – 0. Например, X = "Земля вращается вокруг Солнца" и Y = "3 > 5", причем X =1 и Y = 0. Высказывание не может быть одновременно истинным и ложным.

Высказывания могут быть простыми и составными. Высказывания "Земля вращается вокруг Солнца" и "3 > 5" являются простыми. Составные высказывания образуются из простых с помощью связок естественного (русского) языка НЕ, И, ИЛИ, ЕСЛИ-ТО, ТОГДА-И-ТОЛЬКО-ТОГДА. При использовании буквенных обозначений для высказываний эти связки заменяются специальными математическими символами, которые можно рассматривать как символы логических операций.

Ниже, в таблице 1 приведены варианты символов для обозначения связок и названия соответствующих логических операций.

Таблица 1. Логические операции

Связка	Варианты символов	Наименование операции
не	$\rceil$ $\overline{\ }$	отрицание
и	& /\ ·	конъюнкция
или	\/	дизъюнкция
если-то	$\rightarrow$	импликация
тогда-и-только-тогда	$\leftrightarrow$ ~	эквивалентность

Отрицанием (инверсией) высказывания *X* называется высказывание, истинное тогда и только тогда, когда *X* ложно (обозначается $\neg X$ или $\overline{X}$, читается “не *X*” или “неверно, что *X*”).

Конъюнкцией $X \& Y$ двух высказываний называется высказывание, истинное тогда и только тогда, когда истинны оба высказывания *X* и *Y*. Эта логическая операция соответствует соединению высказываний союзом ”и”.

Дизъюнкцией $X \vee Y$ двух высказываний X и Y называется высказывание ложное в том и только в том случае, когда оба высказывания X и Y ложны. В разговорной речи этой логической операции соответствует союз “или” (неисключающее “или”).

Импликацией двух высказываний X и Y называется высказывание, ложное тогда и только тогда, когда X истинно, а Y – ложно (обозначается $X \to Y$; читается “X влечет Y”, “если X, то Y”). Операнды этой операции имеют специальные названия: X – посылка, Y – заключение.

Эквиваленцией двух высказываний X и Y называется высказывание, истинное тогда и только тогда, когда истинностные значения X и Y одинаковы (обозначение: $X \sim Y$, $X \leftrightarrow Y$).

Операнды логических операций могут принимать только два значения: 1 или 0. Поэтому каждую логическую операцию $\neg$, &, $\vee$, $\to$, $\leftrightarrow$ легко задать с помощью таблицы, указав значение результата операции в зависимости от значений операндов. Такая таблица называется ***таблицей истинности*** (табл. 2).

Таблица 2. Таблицы истинности логических операций

X	**Y**	$\neg X$	X&Y	$X\vee Y$	$X\to Y$	$X\leftrightarrow Y$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

С помощью логических операций, определенных выше, можно из простых высказываний строить ***формулы логики высказываний***, представляющие различные составные высказывания. Логическое значение составного высказывания зависит от структуры высказывания, выраженной формулой, и логических значений образующих его элементарных высказываний.

Для систематического изучения формул, выражающих высказывания, вводят переменные высказывания P, P_1, P_2, ..., P_N, принимающие значения из множества $\{0, 1\}$.

Формула логики высказываний F (P_1, P_2,..., P_N) называется тавтологией или ***тождественно истинной***, если ее значение для любых значений P_1, P_2,..., P_N есть 1 (истина). Формулы, принимающие значение “истина” хотя бы при одном наборе списка переменных, называются ***выполнимыми***. Формулы, принимающие значение “ложь” при любых значениях переменных, называются ***противоречиями*** (тождественно ложными, невыполнимыми).

Формулы F_1 и F_2 называются ***равносильными***, если они принимают одинаковые истинностные значения при любых значениях своих переменных.

Таблица 3. Основные равносильности логики высказываний

№	Формула	Название
1	$X \vee \neg X \equiv$ И	Закон исключенного третьего
2	$X \& \neg X \equiv$ Л	Закон противоречия
3	$X \& Y \equiv Y \& X, X \vee Y \equiv Y \vee X$	Законы коммутативности
4	$(X \& Y) \& Z \equiv X \& (Y \& Z)$ $(X \vee Y) \vee Z \equiv X \vee (Y \vee Z)$	Законы ассоциативности
5	$X \& (Y \vee Z) \equiv (X \& Y) \vee (X \& Z)$ $X \vee (Y \& Z) \equiv (X \vee Y) \& (X \vee Z)$	Законы дистрибутивности
6	$\neg\neg X \equiv X$	Закон двойного отрицания
7	$X \& X \equiv X, X \vee X \equiv X$	Закон идемпотентности
8	$\neg(X \vee Y) \equiv \neg X \& \neg Y$ $\neg(X \& Y) \equiv \neg X \vee \neg Y$	Законы де Моргана
9	$X \vee (X \& Y) \equiv X,\ X \& (X \vee Y) \equiv X$	Законы поглощения
10	$(X \& Y) \vee (X \& \neg Y) \equiv X$ $(X \vee Y) \& (X \vee \neg Y) \equiv X$	Законы склеивания
11	$X \to Y \equiv \neg X \vee Y$	Замена импликации
12	$X \sim Y \equiv (X \& Y) \vee (\neg X \& \neg Y)$	Замена эквиваленции

Множество формул логики высказываний, операций {$\urcorner$, &, $\vee$}и констант {0, 1} образуют ***алгебру высказываний*** (булеву алгебру).

Вопросы для самопроверки:

1. Какие предложения являются высказываниями математической логики:
   - «Летайте самолетами аэрофлота!»
   - «Математическая логика – интересный предмет»
   - «Существуют инопланетные цивилизации»
   - «Москва – столица России»
2. Составьте таблицу истинности для формулы $F = (A \leftrightarrow B) \to \neg A$.
3. Какая формула логики высказываний называется тавтологией?

4.2 Нормальные формы формул

4.2.1 Содержание темы

Понятие элементарной дизъюнкции, элементарной конъюнкции. Дизъюнктивная и конъюнктивная нормальные формы. Теоремы о приведении формулы логического высказывания к ДНФ, к КНФ. Теорема о существовании КНФ (ДНФ) особого вида для тавтологии (противоречия). Понятие полной ЭК (ЭД) относительно данного списка переменных. Понятие совершенной ДНФ (КНФ). Теоремы о существовании СДНФ и СКНФ. Единственность представления в СКНФ (СДНФ). Критерий равносильности.

Литература: [1, 2, 3,5].

4.2.2 Методические указания

Элементарной конъюнкцией (ЭК) называется конъюнкция нескольких переменных, взятых с отрицанием или без отрицания, причем среди переменных могут быть одинаковые:

$$\neg X \,\&\, \mathrm{X};\quad \mathrm{X} \,\&\, \neg \mathrm{Z};\ X \,\&\, \mathrm{Y} \,\&\, \neg \mathrm{Z}.$$

Элементарной дизъюнкцией (ЭД) называется дизъюнкция нескольких переменных, взятых с отрицанием или без отрицания, причем среди переменных могут быть одинаковые:

$$\neg X \vee \mathrm{X};\quad \mathrm{X} \vee \neg \mathrm{Z};\ X \vee \mathrm{Y} \vee \neg \mathrm{Z}.$$

Элементарная конъюнкция (дизъюнкция) называется ***правильной***, если каждая переменная входит в нее не более одного раза, включая вхождения под знаком отрицания.

Правильная элементарная конъюнкция (дизъюнкция) называется ***полной*** относительно переменных $x_1, x_2, \ldots, x_n$ если каждая из переменных входит в нее один и только один раз.

Например, для набора переменных x_1, x_2, x_3, x_4, $x_1 \,\&\, x_2 \,\&\, x_3 \,\&\, \neg x_4$ – полная правильная ЭК, а $x_1 \,\&\, \neg x_1 \,\&\, x_3$ – неправильная ЭК.

Всякую дизъюнкцию элементарных конъюнкций назовем дизъюнктивной нормальной формой (ДНФ):

$$(X \,\&\, \neg Y) \vee (\neg X \,\&\, Z).$$

Всякую конъюнкцию элементарных дизъюнкций назовем конъюнктивной нормальной формой (КНФ):

$$(X \vee \neg Y) \vee (\neg X \vee Z).$$

Теорема 1. Любую формулу логики высказываний можно привести равносильными преобразованиями к ДНФ.

Теорема 2. Любую формулу логики высказываний можно привести равносильными преобразованиями к КНФ.

Доказательство:

Рассмотрим построение ДНФ для некоторой формулы F.

Шаг 1. Построить $F_1 \equiv F$ такую, что F_1 содержит только связки $\{\neg$, $\&$, $\vee\}$. Воспользоваться равносильностями 11, 12 (табл. 3).

Шаг 2. Преобразовать F_1 к $F_2 \equiv F_1$ такой, что в F_2 знак отрицания стоит только перед высказывательными переменными. Воспользоваться законом де-Моргана.

Шаг 3. Преобразовать F_2 к $F_3 \equiv F_2$ пользуясь первым дистрибутивным законом $X \,\&\, (Y \vee Z) \equiv (X \,\&\, Y) \vee (X \,\&\, Z)$; при необходимости использовать законы идемпотентности, коммутативности, ассоциативности.

Формула $F_3 \equiv F$ в силу транзитивности отношения $\equiv$ и записана в ДНФ.

Для приведения формулы F к КНФ необходимо и достаточно выполнить *шаг 1* и *шаг 2* так же, как для построения ДНФ, а затем применить второй дистрибутивный закон $X \vee (Y \,\&\, Z) \equiv (X \vee Y) \,\&\, (X \vee Z)$ и сгруппировать в скобки.

Шаг 2. $F_1 = \neg X \vee \neg Y \vee \neg Y = F_2$.

Шаг 3. $F_2 = \neg X \vee \neg Y = F_3$.

Совершенной ДНФ называется ДНФ, в которой нет одинаковых элементарных конъюнкций и все конъюнкции состоят из одного и того же набора переменных, в который каждая переменная входит только один раз (возможно с отрицанием):

$$(X \,\&\, Y \,\&\, \neg Z) \vee (X \,\&\, Y \,\&\, Z).$$

Совершенной КНФ называется КНФ, в которой нет одинаковых элементарных дизъюнкций и все дизъюнкции состоят из одного и того же набора переменных, в который каждая переменная входит только один раз (возможно с отрицанием):

$$(X \vee \mathrm{Y} \vee \neg \mathrm{Z}) \vee (\mathrm{X} \vee \mathrm{Y} \vee \mathrm{Z}).$$

Теорема 3. Любую функцию, кроме констант 0 и 1, можно представить в виде как СДНФ, так и СКНФ.

Алгоритм получения СДНФ по таблице истинности

1. Отметить те строки ТИ, в последнем столбце которых стоят «1».
2. Выписать для каждой отмеченной строки конъюнкцию всех переменных следующим образом: если значение некоторой переменной в данной строке равно «1», то в конъюнкцию включают саму эту переменную, если «0», то ее отрицание.
3. Все полученные конъюнкции связать в дизъюнкцию.

Алгоритм получения СКНФ по таблице истинности

1. Отметить те строки ТИ, в последнем столбце которых стоят «0».
2. Выписать для каждой отмеченной строки дизъюнкцию всех переменных следующим образом: если значение некоторой переменной в данной строке равно «0», то в дизъюнкцию включают саму эту переменную, если «1», то ее отрицание.
3. Все полученные дизъюнкции связать в конъюнкцию.

Теорема 4. (о существовании СДНФ). Пусть формула F_1 не является тождественно ложной и зависит от списка переменных $(X_1, X_2, ..., X_n)$. Тогда существует равносильная ей формула F_2, находящаяся в СДНФ относительно этого списка переменных.

Теорема 5. Пусть формула $F_1 \neq И$ и зависит от списка переменных $(X_1, X_2, ..., X_n)$. Тогда существует равносильная ей формула F_2, находящаяся в СКНФ относительно этого списка переменных.

Теорема 6 (единственности). Если формулы F_1 и F_2 являются совершенными дизъюнктивными (конъюнктивными) нормальными формами формулы F относительно списка $(X_1, X_2, ..., X_n)$, то они могут отличаться лишь порядком ЭК (ЭД). (Без доказательства).

Теорема 7 (критерий равносильности). Две формулы F_1 и F_2, зависящие от списка $(X_1, X_2, ..., X_n)$, и не равные тождественно $Л(И)$, равносильны тогда и только тогда, когда они приводятся к СДНФ (СКНФ), отличающимся лишь порядком ЭК (ЭД). (Без доказательства).

Вопросы для самопроверки:

1. Дайте определение элементарной конъюнкции.
2. В чем отличие алгоритмов приведения к ДНФ и к КНФ?
3. Какая ДНФ называется совершенной?
4. В чем отличие алгоритмов приведения к СДНФ и к СКНФ?

4.3 Логические рассуждения

4.3.1 Содержание темы

Понятие логического следования, критерий логического следования. Схема рассуждения, правильность логического рассуждения. Способы проверки правильности схем. Способы косвенного доказательства теорем.

Литература: [1, 2, 3,5].

4.3.2 Методические указания

Пусть даны две формулы $A_1,....A_m, B$. Формула B является логическим следованием формул $A_1,....A_m$, если, придавая значения переменным $x_1,....x_n$, от которых зависят все рассматриваемые формулы, всякий раз, когда истинны одновременно все формулы $A_1,....A_m$, истинна и формула B.

Для логического следования используется запись:

$$A_1,....A_m \quad \vdash B.$$

Логически правильное рассуждение будем записывать в виде схемы рассуждения:

$$\left| \frac{A_1, A_2, \ldots, A_m}{B} \right.$$

Для проверки наличия логического следования достаточно построить таблицу истинности.

Три способа проверки правильности логического рассуждения:

I. Применить определение:
   а) записать все посылки и заключения в виде формул логики высказываний;
   б)составить конъюнкцию формализованных посылок $A_1 \& A_2 \& \ldots \& A_m$;
   в) проверить по таблице истинности, следует ли заключение B из формулы $A_1 \& A_2 \& \ldots \& A_m$.

II. Использовать Признак логического следования:

Формула B логически следует из формулы A тогда и только тогда, когда формула $A \to B$ является тавтологией.

Проверка правильности логического рассуждения сводится к ответу на вопрос: является ли формула $A \to B$ тавтологией? На этот вопрос можно ответить, построив таблицу истинности для формулы $A \to B$, или сведя эту формулу с помощью равносильных преобразований к известной тавтологии.

III. Применить сокращенный способ проверки правильности логического рассуждения.

Рассуждение строится «методом от противного»:

Рассуждение является неправильным, если найдется набор значений переменных $X_1^0, X_2^0, ..., X_n^0$ такой, что посылка $A(X_1^0, X_2^0, ..., X_n^0)=1$, а заключение $B(X_1^0, X_2^0, ..., X_n^0)=0$.

Сокращенный метод заключается в следующем.

Пусть требуется проверить правильность логического следования формулы B из посылок $A_1, ... A_m$.

Предположим, что существует набор $X_1^0, X_2^0, ..., X_n^0$, при котором все посылки истинны, а заключение ложно, и попытаемся найти этот набор. Если такой набор будет обнаружен, то наше предположение оправдалось, и рассуждение является логически неправильным. Если в процессе поисков набора придем к противоречию, то наше предположение ошибочно, а рассуждение является логически правильным.

Проверка является ли тавтологией формула $A \to B$ называется ***прямым методом*** доказательства теорем. Доказательство может быть и ***косвенным***, когда вместо формулы $A \to B$ мы рассматриваем другую, но равносильную ей формулу.

Для каждого предложения $A \to B$ можно составить три таких предложения:

- $B \to A$ – обратное данному предложение;
- $\neg A \to \neg B$ – противоположное данному;
- $\neg B \to \neg A$ –обратно-противоположное данному.

Закон контрапозиции: $A \to B \equiv \neg B \to \neg A$.

Согласно данному закону:

a) Два предложения вида $A \to B$ и $\neg B \to \neg A$ одновременно истинны либо одновременно ложны;

b) Предложение, обратно противоположное какой-либо теореме, также является теоремой;

c) Вместо данной теоремы можно доказывать обратно-противоположную теорему.

Вопросы для самопроверки:

1. Что означает предложение: ”Из формулы *P* логически следует формула *D*”?

2. Покажите по определению, что из формулы $P = A$ логически следует формула $D = A \vee B$.
3. Перечислите способы, которыми можно доказать правильность логического рассуждения.
4. Обоснуйте сокращенный способ проверки правильности логического рассуждения.
5. Приведите примеры косвенных доказательств теорем.

4.4 Логика предикатов

4.4.1 Содержание темы

Понятие предиката. Формулы логики предикатов. Интерпретация формул. Равносильность, общезначимость, проблема разрешимости. Доказательство равносильностей логики предикатов. Приведение формул к предваренной нормальной форме.

Литература: [1, 3, 6, 7].

4.4.2 Методические указания

Существуют такие виды логических рассуждений, которые не могут быть обоснованы в рамках исчисления высказываний.

Например:

- «Всякий друг Мартина есть друг Джона. Питер не есть друг Джона. Следовательно, Питер не есть друг Мартина»;
- «Все люди мыслят. Сократ – человек. Следовательно, Сократ мыслит».

Корректность этих умозаключений основывается не только на истинностно-функциональных отношениях между входящими в них предложениями, но и на внутренней структуре самих предложений, а также на понимании таких выражений, как «все», «всякий» и т.д.

Логика предикатов – это расширение возможностей логики высказываний, позволяющее строить высказывания с учетом свойств изучаемых объектов или отношений между ними.

Одноместным предикатом $P(x)$ называется функция переменного x, определенная на множестве M и принимающая значения из множества $\{1, 0\}$.

Множество M, на котором определен предикат $P(x)$, называется ***предметной областью*** или областью определения предиката. Множество всех $x \in M$, при которых $P(x) = 1$, называется ***множеством истинности*** предиката.

Многоместным предикатом называется всякая функция n переменных $Q(x_1, x_2, ..., x_n)$, определенная на множестве $M = M_1 \times M_2 \times ..... \times M_n$ (декартово произведение) и принимающая на этом множестве одно из двух значений $\{1, 0\}$.

В общем случае n-местным предикатом $P(x_1, x_2, \ldots, x_n)$ называется функция, аргументы которой являются элементами произвольного множества M, а значения принадлежат множеству $\{1, 0\}$, или. $P(x_1, x_2, \ldots, x_n): M^n \to \{1, 0\}$. Элементы множества М называются ***предметными переменными***. Количество предметных переменных есть порядок (местность) предиката.

Чтобы сделать более прозрачной структуру сложных высказываний, удобно ввести специальные обозначения для некоторых часто встречающихся выражений - ***кванторы***. Для их обозначения используются символы:

$\forall$ - квантор всеобщности;

$\exists$ - квантор существования.

Пусть $P(x)$ – одноместный предикат, определенный на множестве М. Тогда под выражением $\forall x P(x)$ будем понимать высказывание, которое принимает значение истина тогда и только тогда, когда $P(x)$ истинно для каждого элемента х множества M. Это высказывание уже не зависит от x. Переменную x в предикате $P(x)$ называют ***свободной***, а в высказывании $\forall x P(x)$ – ***связанной*** квантором всеобщности.

Аналогично, под выражением $\exists x P(x)$ понимают высказывание, которое является истинным, если найдется хотя бы один элемент x множества M, для которого $P(x)$ истинно, и ложным, если ни одного такого элемента во множестве М нет. Высказывание $\exists x P(x)$ не зависит от x, в нем переменная x связана квантором существования.

Из предикатных символов с помощью знаков логических операций и кванторов строятся формулы логики предикатов, которые используются в информационных задачах для описания предметной области. При этом определяется содержание множества предметных переменных M, а каждому предикатному символу придается смысл – задается свойство, которое описывает этот предикат. Таким образом, формулам придается некоторая интерпретация. Одна и та же формула в разных интерпретациях может иметь разные значения.

Если формула F истинна при любых значениях своих аргументов в некоторой интерпретации, то она называется истинной в данной интерпретации. Формула, истинная в любой интерпретации, называется общезначимой. Две формулы логики предикатов называются ***равносильными***, если они принимают одинаковые истинностные значения при любых значениях переменной в любой интерпретации. Все равносильности логики высказываний (табл. 3 п.3.1.2) справедливы в логике предикатов. Кроме этого, в логике предикатов есть равносильности, связанные с преобразованиями формул, содержащих кванторы (табл. 4).

Таблица 4. Основные равносильности логики предикатов

№	Формула
1	$\neg(\exists x P(x)) \equiv \forall x(\neg P(x))$
2	$\neg(\forall x P(x)) \equiv \exists x(\neg P(x))$
3	$(\forall x P(x)) \,\&\, (\forall x Q(x)) \equiv \forall x(P(x) \,\&\, Q(x))$
4	$(\exists x P(x)) \vee (\exists x Q(x)) \equiv \exists x(P(x) \vee Q(x))$
5	$(\forall x P(x)) \vee (\forall x Q(x)) \equiv \forall x \forall y(P(x) \vee Q(y))$
6	$(\exists x P(x)) \,\&\, (\exists x Q(x)) \equiv \exists x \exists y(P(x) \,\&\, Q(y))$

Специальную форму записи формулы логики предикатов называют предваренной нормальной формой (ПНФ).

Алгоритм получения формулы ***в предваренной нормальной форме***:

1) перейти от символов $\rightarrow$ и $\sim$ к символам &, $\vee$, $\neg$;
2) внести все отрицания внутрь формулы, “приклеив” их к предикатным символам;
3) вынести все кванторы в начало формулы.

Вопросы для самопроверки:

1. Приведите примеры одноместных предикатов.
2. Дайте определение n-местного предиката.
3. Что такое предметные переменные?
4. Что такое порядок (местность) предиката?
5. Что такое множество истинности предиката?
6. Как привести формулу логики предикатов к предваренной нормальной форме?

4.5 Основы теории алгоритмов

4.5.1 Содержание темы

Понятие алгоритма. Вычислимые, частично рекурсивные и общерекурсивные функции. Примитивная рекурсия. Операция минимизации. Вычисление функций на машине Тьюринга-Поста. Тезис Тьюринга. Универсальная машина Тьюринга-Поста. Эффективные алгоритмы. Алгоритмически неразрешимые проблемы.

Некоторые алгоритмические массовые проблемы. Нумерация Геделя. Самоприменимые программы. Проблема останова.

Литература: [2, 3,4, 6].

4.5.2 Методические указания

Алгоритм – это общий, единообразный, точно установленный способ решения любой задачи из данной массовой проблемы.

Приведенное понятие нестрогое, но оно позволяет выделить некоторые характерные черты алгоритма:

1. Дискретность. Каждая последующая величина получается из значений предыдущих по определенному закону. Все величины получаются последовательно друг за другом.
2. Детерминированность. Между всеми величинами, получаемыми алгоритмом, существует жесткая причинная связь. Последующие значения зависят от предыдущих.
3. Элементарность шагов алгоритма. Закон получения последующей системы величин из предшествующей должен быть простым.
4. Массовость. Начальная система величин выбирается из некоторого множества. Начальные условия могут варьироваться в бесконечных пределах.
5. Результативность. Конечный результат всегда должен быть получен.

Интуитивное определение алгоритма приемлемо, когда речь о найденном алгоритме решения конкретной задачи. В случае, когда возникает алгоритмическая проблема, решение которой не найдено, необходимо доказать либо существование алгоритма решения, либо его отсутствие. Для доказательства несуществования алгоритма необходимо точное формальное определение.

В дальнейшем будут рассмотрены два основных типа алгоритмических моделей, различающиеся исходными трактовками, что такое алгоритм:

1. Первый тип связывает понятие алгоритма с традиционным представлением – процедурами вычисления значений числовых функций. Основной теоретической моделью этого типа являются рекурсивные функции – исторически первая формализация понятия алгоритма. А. Черчем и К. Геделем выделен класс частично рекурсивных функций, имеющих строгое математическое определение.
2. Второй тип трактует алгоритм как некоторое детерминированное устройство, способное выполнять в каждый момент лишь строго фиксированное множество операций. Основной теоретической моделью такого типа является машина Тьюринга, предложенная им в 30-х годах и оказавшая существенное влияние на понимание логической природы разрабатываемых ЭВМ. Другой теоретической моделью данного типа является машина произвольного доступа (МПД) – введенная достаточно недавно (в 70-х годах) с целью моделирования реальных вычислительных машин и получения оценок сложности вычислений.

Рассмотрим алгоритмическую модель, представляющую собой идеализированную ЭВМ и предложенную в 70-х годах с целью моделирования реальных вычислительных машин и анализа сложности вычислений. В результате попыток разложить интуитивно известные нам вычислительные процедуры на элементарные операции построена математическая модель, называемая ***машиной Тьюринга***. Повторение элементарных операций, определенных в этой машине, достаточно для проведения любого возможного вычисления.

Машина Тьюринга включает:

1. Внешний алфавит $A=\{a_0, a_1,..., a_n\}$, т.е. конечное множество символов. В этом алфавите (в символах этого алфавита) информация вводится в машину. Машина преобразует введенную информацию в новую.
2. Внутренний алфавит $Q=\{q_0, q_1,..., q_m, R,\ L,\ C,\ P,\ E\}$. Символы $q_0, q_1,..., q_m$ выражают конечное число состояний машины, причем q_1 – начальное состояние, q_0 – стоп-состояние.
3. Бесконечную в обе стороны ленту, представляющую память машины. Эта память разбита на клетки. В каждую клетку может быть записана Толька одна буква. a_0 – пустая клетка, она всегда может появиться при движении вправо или влево, если закончится слово информации $a_0a_1a_2...a_n$.
4. Считывающее устройство. Оно передвигается вдоль ленты и может останавливаться напротив какой-либо клетки и воспринимать записанный там символ исходного слова. В одном такте работы машины считывающее устройство может сдвигаться на одну клетку или оставаться на месте.

Работа машины складывается из тактов, по ходу которых происходит преобразование начальной информации в промежуточную. В качестве начальной информации на ленту можно подать любую конечную систему знаков внешнего алфавита, расставленного произвольным образом по ячейкам. При этом работа машины

Тьюринга может заканчиваться так:

- После конечного числа тактов машина останавливается в q_0 состоянии. При этом на ленте оказывается преобразованная информация. В этом случае говорят, что машина применима к начальной информации I_1 и преобразует ее в результирующую информацию I_2;
- Машина никогда не останавливается (не переходит в q_0 состояние). В этом случае машина не применима к начальной информации I_1.

В каждом такте работы машина Тьюринга действует по единой ***функциональной схеме***:

$$a_i q_j \Rightarrow a_l \begin{Bmatrix} R \\ L \\ C \\ P \\ E \end{Bmatrix} q_s ,$$

где $a_i, a_l \in A$, $q_j, q_s \in Q$ и a_i – буква на ленте, обозреваемая считывающим устройством на данном такте, q_j – текущее состояние машины на данном такте.

На каждом такте функциональной схемой вырабатывается команда, состоящая из трех элементов (правая часть формулы):

1. Буква внешнего алфавита a_l, на которую заменяется обозреваемая буква a_i.
2. Адрес внешней памяти и дополнительные действия для выполнения на следующем такте R, L, C, P или E.
3. Следующее состояние машины.

Формирование правой части функциональной схемы происходит по командам, совокупность которых образует программу машины Тьюринга. Программа представляется в виде двумерной таблицы (табл. 5), называемой ***тьюринговой функциональной схемой***, в каждой клетке которой записываются отдельные команды. Работа машины Тьюринга полностью определяется ее программой.

Таблица 5. Функциональная схема Тьюринга

Состояния	Символы внешнего алфавита			
	a_0	a_1	$\cdots$	a_n
q_1	$a_2 L q_3$	$a_1 R q_2$	$\cdots$	$a_2 L q_1$
q_2	$a_1 C q_0$	$a_2 C q_1$	$\cdots$	$a_1 C q_2$
…	…	…	…	…
q_m	$a_1 P q_3$	$a_0 R q_{m-1}$	$\cdots$	$a_{n-1} R q_1$

Говорят, что непустое слово α в алфавите A воспринимается машиной в стандартном положении, если оно записано в последовательных клетках ленты, все другие клетки пусты, и машина обозревает крайнюю клетку справа из тех, в которых записано слово α.

Если данное состояние описывается машинным словом M, то машинное слово, описывающее следующее состояние машины, будет обозначаться через $M^{(1)}$. Далее аналогично $M^{(i+1)} = (M^{(i)})^{(1)}$, $i = 0,1,2,\ldots$ Переход машины Тьюринга из начального в последующие состояния изображается в виде цепочки слов $M \vdash M^{(1)} \vdash M^{(2)} \vdash \ldots$

Чтобы описывать работу машины Тьюринга более удобным образом, текущее состояния машины пишут не внизу алфавита, а перед обозреваемой ячейкой. Например, пусть $A = \{0,1\}, Q = \{q_0, q_1, q_2\},\ q_0$ - символ остановки. Начальная информация: $q_1 11$. Тогда программа строится следующим образом:

$$q_1 0 \to q_2 R,\ q_2 0 \to q_0 1,\ \ q_1 1 \to q_1 R,\ q_2 1 \to q_2 R.$$

Если первое направление уточняет понятие алгоритма через класс рекурсивных функций, то второе, связанное с машинной арифметикой, сначала уточняет понятие алгоритма, а затем определяет класс вычислимых функций. Основная идея этого направления заключается в том, что алгоритмические процессы – это процессы, которые могут имитироваться на специально построенных машинах, которые описываются в точных математических терминах. В результате оказывается, что сложные процессы можно моделировать на простых устройствах. Всякий алгоритм может быть задан некоторой функциональной схемой и реализован в соответствующей машине Тьюринга. Эта гипотеза называется ***тезисом Тьюринга***.

Вопросы для самопроверки:

1. Существует точное формальное определение алгоритма?
2. Дайте определение класса рекурсивных функций.
3. Что включает в себя машина Тьюринга?
4. Какие существуют варианты завершения работы машины Тьюринга?

ОЦЕНКА ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТА

Самостоятельная работа студента оценивается преподавателем по результатам выполнения:

- тестовых опросов;
- контрольных работ;
- защиты лабораторных работ;
- ответов на экзамене.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Игошин В.И. Математическая логика и теория алгоритмов: Учебное пособие для вузов. - М.: Академия, 2004. - 446 с.
2. Шапорев С. Д. Математическая логика. Курс лекций и практических занятий: Учебное пособие для вузов. - СПб.: БХВ-Петербург, 2005. - 410 с.
3. Шевелев Ю. П. Математическая логика и теория алгоритмов: учебное пособие. - Томск: Дельтаплан, 2007. - 219 с.
4. Шелупанов А.А.Математическая логика и теория алгоритмов: Учебное пособие. - Томск : STT, 2001. - 176 с.