ЯФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»

Кафедра «Математика и теоретическая механика**»**

Методические рекомендации для самостоятельной работы обучающихся по дисциплине

Математическая логика и теория алгоритмов

Направление подготовки (специальность) 27.03.04 Управление в технических системах

Профиль образовательной программы «Системы и средства автоматизации технологических процессов»

Форма обучения очная

Содержание:

1. ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1.1. Организационно-методические данные дисциплины

№ п.п.	Наименование тем	Общий объем часов по видам самостоятельной работы (*из табл. 5.1 РПД*)				
		подготовка курсового проекта (работы)	подготовка реферата/ эссе	Индивидуальные домашние задания (ИДЗ)	самостоятельное изучение вопросов (СИВ)	подготовка к занятиям (ПкЗ)
1	2	3	4	5	6	7
1	Введение в математическую логику	-	-	-	-	3
2	Операции над высказываниями	-	-	-	3	3
3	Функции алгебры логики	-	-	-	-	6
4	Исчисление высказываний	-	-	-	3	6
5	Логика предикатов	-	-	-	-	6
6	Числовые теории	-	-	-	3	3
7	Геометрические теории	-	-	-	-	6
8	Основные понятия теории алгоритмов				3	6
9	Машинная реализация алгоритмов				-	3
	Итого				12	42

2. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ ВОПРОСОВ

2.1 Понятие алгебры Буля.

Алгебра Буля

Пусть М – непустое множество элементов любой природы $\{x, y, z\ldots\}$, в котором определены отношение равенства и три операции – сложение, умножение, отрицание, подчиняющиеся следующим аксиомам:

1) Коммутативные законы: а) $x+y=y+x$; б) $x\cdot y=y\cdot x$.
2) Ассоциативные законы: а) $x+(y+z)=(x+y)+z$; б) $x\cdot(y\cdot z)=(x\cdot y)\cdot z$.
3) Дистрибутивные законы: а) $(x+y)\,z=(x\cdot z)+(y\cdot z)$; б) $(x\cdot y)+z=(x+z)\cdot(y+z)$.
4) Законы идемпотентности: а) $x+x=x$; б) $x\cdot x=x$.
5) Закон двойного отрицания: $\overline{\overline{x}} = x$.
6) Законы де-Моргана: а) $\overline{x + y} = \overline{x} \cdot \overline{y}$; б) $\overline{x \cdot y} = \overline{x} + \overline{y}$.
7) Законы поглощения: а) $x + (y \cdot x) = x$; б) $x \cdot (y + x) = x$.

Множество М называется *булевой алгеброй*.

Замечание 1. Если под основными элементами x, y, $z\ldots$ множества М подразумевать высказывания, под операциями сложения, умножения, отрицания – дизъюнкцию, конъюнкцию и отрицание соответственно, а знак равенства рассматривать как знак равносильности, то, как следует из равносильностей групп 1–3, все аксиомы булевой алгебры выполняются. Алгебра логики является моделью булевой алгебры или ее интерпретацией.

Замечание 2. Если под основными элементами x, y, $z\ldots$ множества М подразумевать множества, под операциями сложения, умножения, отрицания – объединение, пересечение, дополнение множеств соответственно, а знак равенства рассматривать как знак равенства множеств, приходим к алгебре множеств. В алгебре множеств все аксиомы алгебры Буля выполняются.

Замечание 3. Таблица соответствия между понятиями теории множеств и математической логики имеет вид:

Теория множеств	Математическая логика
Множество	Высказывание
Объединение	Дизъюнкция
Пересечение	Конъюнкция
Дополнение	Отрицание
Основное множество, Е	Тавтология
Пустое множество, Ø	Противоречие

2.2. Проблемы аксиоматического исчисления высказываний

Теорема. Проблема разрешимости для исчисления высказываний разрешима.

Доказано существование алгоритма, который позволяет для любой заданной формулы исчисления высказываний определить, является ли она доказуемой или не является.

Определение. Логическое исчисление называется непротиворечивым, если в нем не доказуемы никакие две формулы, из которых одна является отрицанием другой.

Определение. Аксиоматическое исчисление называется непротиворечивым, если в нем не существует такая формула А, что доказуема А и доказуема $\overline{A}$, в противном случае (если такая формула существует), исчисление противоречиво.

Теорема. Исчисление высказываний непротиворечиво.

Определение. Аксиоматическое исчисление называется полным в узком смысле, если добавление к списку его аксиом любой недоказуемой в исчислении формулы в качестве новой аксиомы приводит к противоречивому исчислению.

Определение. Исчисление высказываний называется полным в широком смысле, если любая тавтология в нем доказуема.

Теорема. Исчисление высказываний полно в узком смысле.

Теорема. Исчисление высказываний полно в широком смысле.

Определение. Аксиома *А* называется независимой от всех остальных аксиом, если она не может быть выведена из остальных аксиом.

Определение. Система аксиом исчисления называется независимой, если каждая аксиома системы независима.

Теорема. Система аксиом исчисления высказываний независима**.**

2.3. Элементарная теория множеств

Множеством называется любая совокупность объектов произвольной природы, каждый из которых называется элементом множества.

Примеры множеств: множество студентов на лекции; множество точек на плоскости, лежащих внутри круга радиуса *r*; множество точек на числовой оси, расстояние от которой до точки *b* с абсциссой *a* не превышает *d*; множество натуральных чисел.

Множества обозначаются по-разному. Множество *M* натуральных чисел от 1 до 100 может быть записано как

$M=\{1,2,\ldots,100\}=\{i$ - целое; $1\le i\le 100\}=\{i=1,2,\ldots,100\}$.

Множество точек на числовой оси, расстояние от которой до точки *b* с абсциссой *a* не превышает *d*, можно записать в виде $S=\{|x-a|\le d\}$ или $S=\{x:|x-a|\le d\}$,
где *x*– абсцисса точки.

Множество точек плоскости, лежащих внутри или на границе круга радиуса *r* с центром в начале координат, $C=\{x^2+y^2\le r^2\}$ или $C=\{(x,y):x^2+y^2\le r^2\}$,
где *x*, *y*– декартовы координаты точки.

Еще одна запись этого множества $C=\{\rho\le r\}$,
где ρ – одна из полярных координат точки.

По числу элементов множества делятся на конечные и бесконечные. Множество $M=\{1,2,\ldots,100\}$ конечно и состоит из 100 элементов. Но множество может состоять и из одного элемента и даже вообще не содержать элементов.

Множество всех натуральных чисел $N_1=\{1,2,\ldots,n,\ldots\}$ бесконечно, также как бесконечно множество четных чисел $N_2=\{2,4,\ldots,2n,\ldots\}$.

Бесконечное множество называется счетным, если все его элементы можно расположить в какой-то последовательности и пронумеровать (оба множества, N_1 и N_2, являются счетными). Множества *S* и *C* бесконечны и несчетны (их элементы нельзя пронумеровать).

Два множества *A* и *B* <u>совпадают</u>, если они состоят из одних и тех же элементов: $\{1, 4\}$ и $\{4, 1\}$. Совпадение множеств обозначается знаком равенства :*A*=*B*. Запись $a \in A$ обозначает, что объект *a* является элементом множества *A* или "*a* принадлежит *A*". Другая запись $a \notin A$ означает , что " *a* не принадлежит *A*".

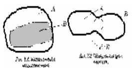
Множество, не содержащее ни одного элемента, называется <u>пустым</u>и обозначается символом $\varnothing$.

Множество *B* называется подмножеством (частью) множества *A*, если все элементы *B* содержатся и в *A*, и обозначается как $B \subseteq A$ или $A \supseteq B$. Например, $\{1, 2, 3\} \subseteq \{1, 2, \ldots, 100\}$.

Подмножество может быть равно самому множеству. Графически можно изобразить соотношение множества и подмножества, как показано на рис. 2.1, где каждая точка фигуры *B* принадлежит и фигуре *A*, т. е. $B \subseteq A$.

Объединением (суммой) множеств *A* и *B* называется множество $C = A \cup B = A + B$, состоящее из всех элементов *A* и всех элементов *B*. Таким образом, объединение – это совокупность элементов, принадлежащих хотя бы одному из объединяемых множеств.

Например: $\{1, 2, \ldots, 100\} + \{50, 51, \ldots, 200\} = \{1, 2, \ldots, 200\}$.



Геометрическая интерпретация объединения двух множеств *A* и *B* показана на рис. Аналогично определяется объединение (сумма) нескольких множеств

$$A_1 + A_2 + \ldots + A_n = \sum_{i=1}^{n} A_i ,$$

где результирующее множество есть множество всех элементов, входящих хотя бы в одно из множеств: $A_1, A_2, \ldots, A_n$.

Пересечением (произведением) множеств *A* и *B* называется множество *D*, состоящее из элементов, входящих одновременно и в *A*, и в B :

$$D = A \cap B = A \times B = AB .$$

Геометрическая интерпретация пересечения представлена на рис. 2.3.

Аналогично определяется пересечение нескольких множеств

$$A_1 \cdot A_2 \cdot \ldots \cdot A_n = \prod_{i=1}^{n} A_i$$

как множество, состоящее из элементов, входящих одновременно во все множества.

Операции объединения (сложения) и пересечения (умножения) множеств обладают рядом свойств, которые аналогичны свойствам сложения и умножения чисел:

1. Переместительное свойство:

$$A + B = B + A, \quad AB = BA .$$

2. Сочетательное свойство:

$$(A + B) + C = A + (B + C), \quad (AB)C = A(BC) .$$

3. Распределительное свойство:

$$A(B + C) = AB + AC .$$

Прибавление пустого множества и умножение на пустое множество аналогичны соответствующим операциям над числами, если считать нуль за пустое множество:

$$A + \varnothing = A, \quad A \cdot \varnothing = \varnothing .$$

Некоторые операции над множествами не имеют аналогов в обычных операциях над числами, в частности

$$A + A = A, \quad A \cdot A = A .$$

2.4. Неразрешимые алгоритмические проблемы

Неразрешимые алгоритмические проблемы

Алгоритмическая проблема — это проблема, в которой требуется найти единый метод (алгоритм) для решения бесконечной серии однотипных единичных задач. Такие проблемы называют также массовыми проблемами. Они возникали и решались в различных областях математики на протяжении всей ее истории. Примеры таких проблем рассматривались ранее.

Математики в начале XX в. столкнулись с тем, что для некоторых массовых проблем не удается подобрать общий алгоритм для их решения. В связи с этим возникла необходимость дать точное определение самому понятию алгоритма. Мы познакомились с несколькими способами такого уточнения, и в настоящем параграфе приведем примеры алгоритмически неразрешимых массовых проблем. Сначала в качестве понятия, уточняющего понятие алгоритма, будем использовать понятие машины Тьюринга. Затем рассмотрим проблему алгоритмической разрешимости в рамках общей теории алгоритмов.

Нумерация алгоритмов

Понятие нумерации алгоритмов — важное средство для их исследования, в частности для доказательств несуществования единого алгоритма для решения той или иной массовой проблемы. Посмотрим сначала на это понятие в рамках нашей общей теории алгоритмов.

Поскольку любой алгоритм можно задать конечным описанием (словом) (например, в конечном алфавите знаков, используемых при наборе математических книг), а множество всех конечных слов в фиксированном конечном алфавите счетно, то множество всех алгоритмов счетно. Это означает наличие взаимно-однозначного соответствия между множеством N натуральных чисел и множеством всех алгоритмов, рассматриваемым как подмножество множества Al^* всех слов в алфавите Al, выбранном для описания алгоритмов $(\varphi\colon N \to Al^*)$. Такая функция называется нумерацией алгоритмов. Если $\varphi(n) = A$, то число n называется номером алгоритма A. Из взаимной однозначности отображения φ следует существование обратной функции φ^{-1}, восстанавливающей по описанию алгоритма A_n его номер в этой нумерации $\varphi^{-1}(A_n) = n$. Очевидно, что различных нумераций много.

Нумерация всех алгоритмов является одновременно и нумерацией всех вычислимых функций в следующем смысле: номер функции f — это номер некоторого алгоритма, вычисляющего f. Ясно, что в любой нумерации всякая функция будет иметь бесконечное множество различных номеров.

Существование нумераций позволяет работать с алгоритмами как с числами. Это особенно удобно при исследовании алгоритмов над алгоритмами. Отсутствие именно таких алгоритмов часто приводит к алгоритмически неразрешимым проблемам.

Нумерация машин Тьюринга

Опишем теперь более конкретный процесс нумерации всех машин Тьюринга, который используем при построении примера невычислимой по Тьюрингу функции. Будем считать, что для обозначения внутренних состояний машин Тьюринга используются буквы бесконечной последовательности: $q_0, q_1, q_2, \ldots, q_r, \ldots$, а для обозначения букв внешних алфавитов используются буквы последовательности: $a_0, a_1, a_2, \ldots, a_s, \ldots$.

Выразим (или, как говорят, закодируем) все символы этих бесконечных последовательностей словами конечного стандартного алфавита $\{a_0, 1, q, C, П, Л\}$ по следующим правилам:

$q_i \ (i = 0, 1, \ldots)$ обозначим (закодируем) словом из $i+1$ букв q: $qq\ldots q$;
$a_j \ (j = 1, 2, \ldots)$ обозначим (закодируем) словом из у единиц: $11\ldots1$.

В стандартном алфавите программу машины Тьюринга можно записать в виде слова, руководствуясь следующим правилом. Сначала все команды программы переводятся на язык стандартного алфавита, для чего в записях этих команд $q_i a_j \to q_l a_m X$, где $X \in \{C, \Pi, L\}$, опускается символ " $\to$ ", а буквы q_i, a_j, q_l, a_m заменяются соответствующими словами стандартного алфавита. Затем полученные слова-команды записываются подряд в любом порядке в виде единого слова.

Например, программа машины Тьюринга, рассмотренной в Примере 32.1, в этих обозначениях имеет вид:

$$q_1 a_0 \to q_2 a_0 \Pi, \quad q_1 a_1 \to q_1 a_1 \Pi, \quad q_2 a_0 \to q_0 a_1 C, \quad q_2 a_1 \to q_2 a_1 \Pi.$$

Опускаем символ " $\to$ ", заменяем буквы словами стандартного алфавита и в результате полу- чаем следующие слова в стандартном алфавите, кодирующие соответствующие команды:

$$qqa_0qqqa_0\Pi, \quad qq1qq1\Pi, \quad qqqa_0q1C, \quad qqq1qqq1\Pi.$$

Выписываем эти слова подряд и получаем слово, кодирующее программу данной машины Тьюринга:

$$qqa_0qqqa_0\Pi qq1qq1\Pi qqqa_0q1Cqqq1qqq1\Pi.$$

Нетрудно указать алгоритм, позволяющий узнавать, является ли слово в стандартном алфавите программой некоторой машины Тьюринга. Такой алгоритм может, например, состоять в следующем. Нужно анализировать все подслова данного слова, заключенные между всевозможными парами букв из $\{C, \Pi, L\}$. Эти подслова должны иметь следующую структуру: сначала записаны несколько букв q, затем a_0 или несколько букв 1, затем снова несколько букв q и, наконец, снова a_0 или несколько единиц.

Таким образом, каждая машина Тьюринга вполне определяется некоторым конечным словом в конечном стандартном алфавите. Поскольку множество всех конечных слов в конечном алфавите счетно, то и всех мыслимых машин Тьюринга (отличающихся друг от друга по существу своей работы) имеется не более чем счетное множество.

Перенумеруем теперь все машины Тьюринга, для чего все слова стандартного алфавита, представляющие собой программы всевозможных машин Тьюринга, расположим в виде фиксированной счетно-бесконечной последовательности, которую составим по такому правилу: сначала выписываются в какой-нибудь фиксированной последовательности все однобуквенные слова: $\alpha_0, \alpha_1, \ldots, \alpha_\xi$, представляющие программы машин Тьюринга (множество таких слов конечно, потому что конечен стандартный алфавит, из букв которого строятся слова), затем выписываются все двухбуквенные слова $\alpha_{\xi+1}, \ldots, \alpha_\eta$, представляющие программы машин Тьюринга (множество таких слов также конечно, потому что конечен стандартный алфавит), затем выписываются трехбуквенные слова и т.д. Получится последовательность программ $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha, \ldots$ всех мыслимых машин Тьюринга. Число k будем называть номером машины Тьюринга, если программа этой машины записывается словом α_k.

Существование невычислимых по Тьюрингу функций

Теорема 36.1. *Существует функция, не вычислимая по Тьюрингу, т.е. не вычислимая ни на одной машине Тьюринга.*

Доказательство. Функции, о которых идет речь, представляют собой функции, заданные и принимающие значения в множестве слов в алфавите $A_1 = \{1\}$. Ясно, что множество слов в алфавите $A_1 = \{1\}$ счетно. Следовательно, рассматривается множество всех функций, заданных на счетном множестве и принимающих значения в счетном же множестве. Как известно, это множество имеет мощность континуума. С другой стороны, поскольку множество всевозможных машин Тьюринга, как мы установили в предыдущем пункте, перенумеровав их, счетно, это и множество функций, вычислимых по Тьюрингу, также счетно. Континуальная мощность строго больше счетной. Следовательно, существуют функции, не вычислимые по Тьюрингу.

Доказанная теорема есть чистая теорема существования. Интересно получить пример конкретной функции, не вычислимой по Тьюрингу.

Пример 36.2. Укажем конкретную функцию, которую нельзя вычислить ни на какой машине Тьюринга. На основании тезиса Тьюринга это будет означать, что не существует вообще никакого алгоритма для вычисления значений такой функции.

Рассмотрим следующую функцию $\psi(\alpha)$ на словах в алфавите $A_1 = \{1\}$. Для произвольного слова α длиной n в алфавите $A_1 = \{1\}$ положим: $\psi(\alpha) = \beta_n 1$, если слово α перерабатывается машиной Тьюринга с номером n (см. предыдущий пункт) в слово β_n алфавита $A_1 = \{1\}$; $\psi(\alpha) = 1$ в противном случае. Докажем, что функция $\psi(\alpha)$ не вычислима по Тьюрингу.

Допустим противное. Это означает, что существует машина Тьюринга T со стандартным алфавитом $\{a_0, 1, q, \Pi, L\}$, вычисляющая эту функцию. Пусть k — номер этой машины в нумерации, описанной в предыдущем пункте. Посмотрим, чему равно слово $\psi(1^k)$ (напомним, что $1^k = 11\ldots1$ — слово из k единиц), являющееся значением функции $\psi(\alpha)$ при $\alpha = 1^k$. Предположим, что машина t перерабатывает слово 1^k в слово β_k в том же алфавите $A_1 = \{1\}$. Тогда по определению вычислимости функции $\psi(\alpha)$ на машине T это означает, что $\psi(1^k) = \beta_k$. Но с другой стороны, по самому определению функции $\psi(\alpha)$ это означает, что $\psi(1^k) = \beta_k 1$. Полученное противоречие доказывает, что машины Тьюринга, вычисляющей функцию $\psi(\alpha)$, не существует.

Принимая во внимание тезис Тьюринга, заключаем, что не существует вообще никакого алгоритма для вычисления значений функции $\psi(\alpha)$. Это означает, что массовая проблема нахождения значений функции $\psi(\alpha)$ для всевозможных значений аргумента алгоритмически не разрешима.

Проблемы распознавания самоприменимости и применимости

Это еще два примера алгоритмически не разрешимых проблем. Сначала о первой. Предположим, что на ленте машины Тьюринга записана ее собственная функциональная схема в алфавите машины. Если машина применима к такой конфигурации, то будем называть ее самоприменяемой, в противном случае — несамоприме-Няемой. Возникает массовая проблема распознавания самоприме-Кяемых машин Тьюринга, состоящая в следующем. По заданной Функциональной схеме (программе) машины Тьюринга устано-Вить, к какому классу относится машина: к классу самопримени-Mbix машин или к классу несамоприменимых машин.

Теорема 36.3. *Проблема распознавания самоприменимых машин Тьюринга алгоритмически не разрешима.*

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЗАНЯТИЯМ

3.1 Практические занятия по теме «Введение в математическую логику»

1. История развития математической логики.
2. Логические парадоксы. Апории Зенона.
3. Виды множеств.
4. Диаграммы Венна.
5. Взаимно-однозначные соответствия.
6. Изоморфизм.

При подготовке акцентировать внимание необходимо на следующем:

1. Особенности становления и развития математической логики.

2. Отношения порядка, эквивалентности и др. на различных множествах.

3.2. Практические занятия по теме «Множества и функции»

1. Истинные и ложные высказывания.
2. Составление таблиц истинности для высказываний.
3. Элементарные и составные высказывания.

При подготовке акцентировать внимание необходимо на следующем:

1. Логические связки, полные системы логических связок.

3.3. Практические занятия по теме « Функции алгебры логики»

1. Доказательство равносильности.
2.Доказтельство истинности или ложности формул.
3. Нахождение формулы по заданной таблице истинности.
4. Нахождение СКНФ и СДНФ.
5. Релейно-контактные схемы.
6. Решение логических задач.
7. Контрольная работа по теме «Семантика»

При подготовке акцентировать внимание необходимо на следующем:

1. Законы логики, их применение в равносильных преобразованиях.
2. Проблема разрешимости в АВ.
3. Задачи анализа и синтеза РКС.

3.4. Практические занятия по теме «Исчисление высказываний»

1. Формулы и подформулы.
2. Доказательство формул.
3. Доказуемость формул.
4. Применение производных правил вывода.

При подготовке акцентировать внимание необходимо на следующем:

1. Построение формальных аксиоматических теорий (ФАТ).
2. Свойства ФАТ.
3. Исчисление высказываний как ФАТ.

3.5. Практические занятия по теме «Логика предикатов»
1. Нахождение области истинности предиката.
2. Использование кванторных операций.
3. Запись математических предложений в виде формул логики предикатов.
4. Построение противоположных утверждений.
5. Прямая, обратная и противоположная теоремы.
6. Необходимые и достаточные условия.
7. Контрольная работа № 2 «Логика предикатов»

При подготовке акцентировать внимание необходимо на следующем:
1. Теоретико-множественный смысл операций над предикатами.
2. Исчисление предикатов как ФАТ.
3. Интерпретация формулы в ИП или семейства формул.

3.6. Практические занятия по теме «Числовые теории»
1. Понятие группы.
2. Метод математической индукции.
3. Теория колец.
4. Поле рациональных чисел.
5.. Поле действительных чисел.
6. Поле комплексных чисел.

При подготовке акцентировать внимание необходимо на следующем:
1. Числовые структуры современной математики.
2. Алгебраические структуры в моделировании инженерных процессов.

3.7. Практические занятия по теме «Геометрические теории»
1. Аксиомы Тарского.
2. Евклидова геометрия.
3. Аксиомы Гильберта.
4. Аксиома о параллельных.
5. Теоремы геометрии Лобачевского.
При подготовке акцентировать внимание необходимо на следующем:
1.Аксиоматическое строение формальных математических теорий.
2. Аксиоматика планиметрии Гильберта и Тарского (сравнительный анализ).

3.8. Практические занятия по теме «Основные понятия теории алгоритмов»
1. Частично рекурсивные функции.
2. Общерекурсивные функции.

При подготовке акцентировать внимание необходимо на следующем:
1. Математический подход к понятию алгоритма.
2. Нормальные алгоритмы Маркова.

3.9. Практические занятия по теме «Машинная реализация алгоритмов»
1. Устройство машины.
2. Определение функции по программе команд.
3. Построение машины Тьюринга для алгоритма.

При подготовке акцентировать внимание необходимо на следующем:
1. Применение машины Тьюринга в инженерных приложениях.