Г.Б. Поднебесова

ТЕОРИЯ АЛГОРИТМОВ

Практикум

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Южно-Уральский государственный
гуманитарно-педагогический университет»

Г.Б. Поднебесова

ТЕОРИЯ АЛГОРИТМОВ
Практикум

Челябинск, 2017

УДК 001.8 (076)
ББК 73я7
П 44

Поднебесова, Г.Б. Теория алгоритмов [Текст]: практикум / Г.Б. Поднебесова. – Челябинск: Изд-во Южно-Урал. гос. гуман.-пед. ун-та, 2017. – 91 с.

ISBN 978-5-906908-75-9

Практикум содержит материалы для изучения курсов «Теория алгоритмов» и «Теоретические основы информатики и современных информационных технологий». Пособие предназначено для организации аудиторной и самостоятельной работы студентов, обучающихся по направлениям «Педагогическое образование» и «Информационные системы и технологии».

Структура пособия позволяет использовать модульно-рейтинговую систему оценивания учебных достижений студентов при изучении дисциплины «Теория алгоритмов». В работе имеется также банк тестовых заданий.

Практикум адресован преподавателям и учителям, для которых интересна данная предметная область.

Рецензенты: М.М. Кипнис, д-р физ.-мат. наук, профессор
С.А. Иванов, канд. физ.-мат. наук

ISBN 978-5-906908-75-9

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Учебный курс знакомит студентов с современными проблемами теоретической информатики. Основной акцент в курсе делается на методологические аспекты и математический аппарат информатики, составляющие ядро широкого спектра научно-технических и социально-экономических информационных технологий, которые реально используются современным мировым профессиональным сообществом в теоретических исследованиях и практической деятельности.

Учебная дисциплина «Теория алгоритмов» базируется на материале предшествующих ей дисциплин Математики (Математический анализ, Алгебра и теория чисел) и курса Абстрактной и компьютерной алгебры.

Программа курса предусматривает аудиторные занятия (лекции и практические занятия – лабораторные практикумы) и самостоятельную работу студентов. В самостоятельную работу студентов входит освоение теоретического материала, выполнение индивидуальных заданий, подготовка сообщений и написание кейса по разделам дисциплины.

В результате изучения дисциплины студент должен:

– знать понятие эффективной вычислимости в интуитивном смысле;

– иметь представление о модели вычисления;

– знать описание класса арифметических функций (т.е. функций, значения и аргументы которых – натуральные числа), инвариантных относительно общих моделей вычислений;

– знать о существовании арифметической функции, не принадлежащей этому классу;

– иметь представление о разрешимых и перечислимых множествах;

– знать понятия нумерации и теория нумераций;

– уметь строить алгоритмические модели;

– иметь представление об алгоритмически неразрешимых проблемах.

Цель данного курса – познакомить студентов с основными алгоритмическими моделями, используемыми для уточнения понятия «алгоритм».

Для демонстрации работы алгоритмов используются эмуляторы, разработанные студентами факультета информатики Южно-Уральского государственного гуманитарно-педагогического университета (далее ЮУрГГПУ).

Изучение данной дисциплины предусмотрено вариативной частью профессионального цикла основной образовательной программы ФГОС ВО.

Компетенцией, формируемой в результате освоения дисциплины, по направлению Педагогическое образование (ПО) профиль Информатика, является способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве (ОК-3). Планируемые результаты обучения представлены в таблице 1.

Таблица 1

Планируемые результаты обучения (ПО)

№ п/п	Компетенция (содержание и обозначение в соответствии с ФГОС ВО и ОПОП)	Конкретизированные цели освоения дисциплины		
		знать	уметь	владеть
1	Способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве (ОК-3)	З.1. свойства алгоритмов; З.2. понятие модели вычисления; З.3. понятие эффективной вычислимости в интуитивном смысле; З.4. понятие сложности вычисления	У.1. строить алгоритмические модели	В.1. методами разработки алгоритмов; В.2. методами нумерации алгоритмов

Компетенцией, формируемой в результате освоения дисциплины, по направлению Информационные системы и технологии (ИСиТ) профиль Информационные технологии в образовании, является готовность применять знания теоретической информатики, фундаментальной и прикладной математики для анализа и синтеза информационных систем и процессов (ОПК-2). Планируемые результаты обучения представлены в таблице 2.

Таблица 2

Планируемые результаты обучения (ИСиТ)

№ п/п	Компетенция (содержание и обозначение в соответствии с ФГОС ВО и ОПОП)	Конкретизированные цели освоения дисциплины		
		знать	уметь	владеть
1	Способность использовать основные законы естественнонаучных дисциплин в профессиональной деятельности, применять методы математического анализа и моделирования, теоретического и экспериментального исследования (ОПК-2)	З.1. понятие эффективной вычислимости в интуитивном смысле; З.2. понятие модели вычисления; З.3. понятие сложности вычисления	У.1. строить алгоритмические модели	В.1. методами создания алгоритмических конструкций

СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ

1. Формализация понятия «алгоритм».

Уточнение понятия «алгоритм». Требования к алгоритмам. Машина Тьюринга (МТ). Универсальная машина Тьюринга. Тезис Тьюринга. Проблема остановки. Машина Поста. Примитивно-рекурсивные функции. Частично-рекурсивные функции. Общерекурсивные функции. Нормальные алгоритмы Маркова (НАМ). Схема нормального алгоритма Маркова. Принцип нормализации Маркова.

2. Машина с неограниченными регистрами (МНР).

Необходимость простых моделей вычислений. Описание МНР – машины, выполняющей косвенную адресацию, проверку на равенство и вычисление функции следования. Программирование для МНР. Примеры. Функции, вычислимые на МНР. Примеры. Необходимость рассмотрения не всюду определенных функций. Тезис Черча. Построение эффективной нумерации программ для МНР. Существование универсальной МНР. Неразрешимость проблемы остановки для МНР. Алгоритмическая сводимость проблем. Неразрешимость исчисления предикатов. Пример функции, невычислимой на МНР. Сравнение МНР и ЭВМ.

3. Вычислимые функции. Разрешимые и перечислимые множества.

Понятие вычислимой функции. Примеры. Свойство пошагового выполнения алгоритма. Разрешимые множества и их свойства. Перечислимые множества и их свойства. Перечислимое множество как множество определения вычислимой функции. Перечислимое множество как множество значений вычислимой функции. Теорема Поста. Теорема о графике вычислимой функции.

4. Универсальные функции и неразрешимость.

Понятие универсальной функции. Существование вычислимой универсальной функции для класса вычислимых функций одной переменной. Диагональная конструкция. Отсутствие вычислимой всюду определенной функции двух переменных, универсальной для класса всех вычислимых всюду определенных функций одной переменной. Существование вычислимой функции, не имеющей всюду определенного вычислимого продолжения. Существование перечислимого множества с не перечислимым дополнением. Неразрешимость проблемы самоприменимости.

5. Нумерации.

Понятие нумерации. Главные универсальные функции. Существование главной универсальной функции. Теорема Успенского–Райса. Изоморфизм главных нумераций. Перечислимые свойства функций.

6. Элементы теории сложности.

Понятие сложности вычисления. Сигнализирующая функция (по времени). Аксиомы Блюма. Теорема об ускорении. Сложностные классы. Вычисления с оракулом. Описание классов P и NP. Примеры задач, принадлежащих этим классам. Отождествление класса P с классом реально вычислимых функций. Полиномиальная сводимость. NP-полные задачи. Теорема Кука. Примеры NP-полных задач. Проблема перебора ($P = NP$?). Применение теории NP-полноты для анализа сложности задач.

ТЕМЫ И ПЛАНЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Модуль 1. Машина Тьюринга.

1. Машины Тьюринга (4 часа):

– решение задач на обработку нечисловых данных;

– решение задач на обработку числовых данных (в унарной и десятичной системах счисления);

– создание циклических машин Тьюринга.

2. Эмулятор МТ (2 часа).

Модуль 2. Примитивно-рекурсивные функции. Нормальные алгоритмы Маркова.

3. Рекурсивные функции (4 часа):

– доказательство примитивной рекурсивности функций от одной и нескольких переменных;

– восстановление функций по схеме примитивной рекурсии.

4. Нормальные алгоритмы Маркова (2 часа):

– решение задач на работу с числовыми и нечисловыми объектами;

– доказательство нормальной вычислимости функций.

5. Эмулятор НАМ (2 часа).

Модуль 3. Машины с неограниченными регистрами.

6. Машина с неограниченными регистрами (2 часа):

– создание алгоритмов, работающих с нечисловыми объектами;

– создание алгоритмов, работающих с числовыми объектами.

7. Эмулятор МНР (2 часа).

Модуль 4. Вычислимость и разрешимость.

8. Нумерация программ (2 часа).

Модуль 5. Эффективные операции на вычислимых функциях. Сложность вычисления.

9. Машина Поста (2 часа).

10. Эмулятор МП (2 часа).

МОДУЛЬ 1. МАШИНЫ ТЬЮРИНГА

Практическое занятие 1

Машины Тьюринга (МТ)

Машина Тьюринга разработана Аланом Тьюрингом в 1936 г.

Машина Тьюринга состоит из:

1) управляющего устройства, которое может находиться в одном из состояний $Q = \{q_1, q_2, \ldots, q_n\}$;

2) ленты, бесконечной в обе стороны, разбитой на ячейки, в каждой из которых может быть записан один из символов конечного алфавита $A = \{a_1, a_2, \ldots, a_m\}$;

3) устройства обращения к ленте, считывающей и пишущей головки, которая в каждый момент времени обозревает ячейку ленты и в зависимости от символа в этой ячейке записывает новый символ, стирает этот или оставляет прежним и переходит в новое состояние или оставляет прежнее состояние.

Память МТ – конечное множество состояний (внутренняя память) и лента (внешняя память).

Данные МТ – слова в алфавите ленты.

Элементарные шаги МТ – считывание и запись символов, сдвиг головки вправо или влево, переход управляющего устройства в следующее состояние.

Детерминированность МТ – последовательность ее шагов определяется: для любого состояния q_i и символа a_j однозначно заданы:

1) q_i' – состояние;

2) символ a_j', который надо записать вместо a_j;

3) d_k – направление сдвига головки (может принимать значения R, L, E).

Машину Тьюринга можно записывать в виде:

– системы правил (команд);

– таблицы (строки – состояния; столбцы – символы; на пересечении – новое состояние, символ и направление сдвига головки);

– диаграммы (или графа).

Пример. Сложение. Во введенном раннее представлении чисел сложить числа a и b – это значит слово $1^a * 1^b$ переработать в слово 1^{a+b}, т.е. удалить разделитель * и сдвинуть одно из слагаемых, скажем, первое, к другому. Это осуществляет машина T_+ с четырьмя состояниями и следующей системой команд (первая команда введена для случая, когда а = 0 и исходное слово имеет вид $* 1^b$):

$$q_1* \to q_z \lambda R;$$
$$q_1 1 \to q_2 \lambda R;$$
$$q_2 1 \to q_2 1 R;$$
$$q_2 * \to q_3 1 L;$$
$$q_3 1 \to q_3 1 L;$$
$$q_3 \lambda \to q_z \lambda R.$$

Задания

1. Дан алфавит А = {1, 0} и состояния Q = $\{q_1\}$. Построить машину Тьюринга, удаляющую 0.
2. Построить машину Тьюринга T_+ для сложения двух натуральных чисел, записанных в унарной системе. Построить диаграмму машины Тьюринга T_+ , А = {1}.
3. Дан алфавит А={1}. Построить машину Тьюринга для получения следующего натурального числа.
4. Имеется машина Тьюринга с алфавитом А={1} и внутренними состояниями Q = $\{q_1\}$ со следующей системой команд:

$$q_1 0 \to q_z 1$$
$$q_1 1 \to q_1 1 R.$$

Определить, в какое слово переработает машина каждое из следующих слов, если она находится в начальном состоянии q_1:

a) $10q_1110011$;

b) $110q_111101$;

c) $1q_100111$.

Ответ изобразить схематически в виде последовательности конфигураций, возникающих на ленте на каждом такте работы машины.

5. Дано слово в алфавите A = {a, b, c, d}. Построить машину Тьюринга, удаляющую все буквы a.

6. Построить машину Тьюринга T_{++} для суммирования N натуральных подряд идущих чисел, записанных в унарной системе через «*». Пример: 111*11*1111* … *11.

7. Построить машину Тьюринга T_, которая вычисляет разность двух натуральных чисел, записанных в унарной системе через разделитель. Например: 11111-111.

Индивидуальное задание 1

Машины Тьюринга

1. Имеется машина Тьюринга с внешним алфавитом A = {0, 1} и внутренними состояниями Q = $\{q_1\}$ со следующей системой команд:

$$q_1 1 \to q_1 0R$$
$$q_1 0 \to q_1 1R$$
$$q_1 \lambda \to q_z 1.$$

Определить, в какое слово переработает машина слово α, если она находится в начальном состоянии q_1. Ответ изобразить схематически в виде последовательности конфигураций, возникающих на ленте на каждом такте работы машины.

1) $11q_101\lambda01$;	4) $11q_101\lambda01$;
2) $100q_11\lambda01$;	5) $100q_11\lambda01$;
3) $11\lambda01q_101$;	6) $11\lambda01q_101$;

7) $q_1 10\lambda 1100$;	16) $1q_1 111111$;
8) $1101q_1 001$;	17) $110q_1 1\lambda 01$;
9) $11q_1 01101$;	18) $0q_1 000\lambda 01$;
10) $1q_1 1010\lambda 1$;	19) $100q_1 1\lambda 01$;
11) $10011q_1 01$;	20) $0\lambda 1q_1 0101$;
12) $1\lambda 1q_1 0101$;	21) $11q_1 010\lambda 1$;
13) $110q_1 \lambda 101$;	22) $0111q_1 01\lambda$;
14) $\lambda 11q_1 0101$;	23) $1\lambda q_1 0110\lambda$;
15) $110q_1 1\lambda 01$;	24) $110q_1 1001$

2. Дана машина Тьюринга с внешним алфавитом A = {1, 0} и внутренними состояниями Q = {q_1, q_2, q_3, q_4}, записанная в виде таблицы:

A \ Q	q_1	q_2	q_3	q_4
λ	$q_2\lambda R$	$q_3\lambda R$	$q_4\lambda R$	$q_z\lambda R$
1	$q_1\lambda R$	$q_3 1R$	$q_4\lambda R$	$q_4 1L$
0	$q_1 1R$	$q_3 0R$	$q_4 1R$	$q_1 0L$

Изображая на каждом такте работы машины получающуюся конфигурацию, определите, в какое слово перерабатывает машина слово α, исходя из положения, когда машина находится в состоянии q_1.

1) $1101\lambda 0q_1 1\lambda 011$;	8) $110q_1 \lambda 01\lambda 1001$;
2) $10\lambda 0q_1 10\lambda\lambda 101$;	9) $11011q_1 \lambda 01\lambda 01$;
3) $11\lambda q_1 \lambda 01\lambda 0101$;	10) $10\lambda 0q_1 110\lambda\lambda 1$;
4) $1001\lambda q_1 011\lambda 0\lambda$;	11) $1001\lambda 01q_1 \lambda 101$;
5) $110\lambda 01\lambda q_1 \lambda 101$;	12) $1\lambda 10q_1 101\lambda 01\lambda$;
6) $110\lambda q_1 01\lambda 1\lambda 01$;	13) $11q_1 0\lambda 01\lambda\lambda 101$;
7) $10\lambda 11\lambda 0q_1 1\lambda 00$;	14) $\lambda 110q_1 \lambda 01\lambda 101$;

15) $110\lambda01\lambda q_1 1\lambda01$;	20) $0\lambda\lambda0q_1 1\lambda10101$;
16) $11\lambda01\lambda q_1 11111$;	21) $1\lambda1011\lambda1\lambda q_1 101$;
17) $11q_1 01\lambda01\lambda\lambda01$;	22) $00\lambda q_1 11\lambda1\lambda011$;
18) $00\lambda01\lambda q_1 00\lambda01$;	23) $\lambda110q_1\lambda01101\lambda$;
19) $1001\lambda\lambda01\lambda q_1 01$;	24) $1001\lambda10q_1\lambda\lambda11$

3. Постройте машину Тьюринга, которая бы в слове *cabdabda* выполнила указанные действия. Исполнить полученный алгоритм.

1) заменяла *ab* на *e*	13) добавляла *s* до *b*
2) добавляла *e* после *b*	14) сдвигала на 1 символ влево
3) добавляла *f* до *a*	15) сдвигала на 1 символ вправо
4) сдвигала на 2 символа влево	16) заменяла *da* на k
5) сдвигала на 2 символа вправо	17) добавляла *e* после *a*
6) заменяла *b* на *f*	18) добавляла *c* до *d*
7) добавляла *c* после *a*	19) переносила 2 символа слева в конец слова
8) меняла местами 1 символ и последний	20) заменяла *db* на *e*
9) переносила 2 символа справа в начало слова	21) добавляла *bc* после *b*
10) меняла местами 2-й символ и предпоследний	22) заменяла *a* на *p*
11) заменяла *d* на *a*	23) добавляла *e* после *b*
12) добавляла *f* после *c*	24) упорядочивала последовательность

Практическое занятие 2

Машины Тьюринга

1. Построить машину Тьюринга, которая правильно вычисляет функцию $f(x) = x + 1$ (число записано в двоичной системе счисления).

2. Построить машину Тьюринга, которая правильно вычисляет функцию *f(x)* = 0, A = {1, 0}.

3. Построить машины Тьюринга, которые в алфавите A = {1, 0} выполняют:

1) правый сдвиг: $q_1 01^x 0 \underset{Б^+}{\Rightarrow} 01^x q_z 0$;

2) левый сдвиг (самостоятельно): $01^x q_1 0 \underset{Б^-}{\Rightarrow} q_z 01^x 0$;

3) транспозицию: $01^x q_1 01^y 0 \underset{В}{\Rightarrow} 01^y q_z 01^x 0$;

4) удвоение: $q_1 01^x 0 \underset{Г}{\Rightarrow} q_z 01^x 01^x 0$.

Индивидуальное задание 2

Машины Тьюринга. Конструирование машин Тьюринга

1. Дан алфавит A = {a, b, c, d, e, f} и внутренние состояния Q = {q_1, q_2, …, q_z}. Построить машину Тьюринга для транспозиции элементов. Например, последовательность *abcd* заменить на *dcba*.

2. Построить машину Тьюринга, вычисляющую функцию:

$$f(x_1, x_2, \ldots, x_n) = x_m.$$

Пример: 1110110…011. Оставить *m*-й набор единиц, остальные стереть.

3. Дана конечная совокупность единиц, вписанных в ячейки без пропусков. Построить машину Тьюринга, которая записывала бы в десятичной системе счисления число этих единиц, т.е. пересчитывала набор этих единиц.

4. Построить машину Тьюринга, которая выполняет умножение двух чисел в унарной системе счисления.

5. Даны два набора единиц. Они разделены *. Построить машину Тьюринга, которая выбирала бы больший из этих наборов, а меньший стирала.

6. Дана строка из букв «a» и «b». Разработать машину Тьюринга, которая переместит все буквы «a» в левую, а буквы «b» – в правую части строки. Каретка находится над крайним левым символом строки.

7. На ленте машины Тьюринга находится целое положительное число, записанное в десятичной системе счисления. Найти произведение этого числа на число 11. Каретка обозревает крайнюю правую цифру числа.

8. На ленте машины Тьюринга находится десятичное число. Определить, делится ли это число на 5 без остатка. Если делится, то записать справа от числа слово «y», если нет – «n».

9. На ленте машины Тьюринга записано число в пятеричной системе счисления. Каретка находится над крайней правой цифрой. Записать цифры этого числа в обратном порядке.

10. Даны два натуральных числа n и m, представленные в унарной системе счисления. Между этими числами стоит знак «*». Построить машину Тьюринга, определяющую, равны эти числа или нет.

11. Даны два натуральных числа n и m, представленные в двоичной системе счисления. Между этими числами стоит знак «*». Найти разность этих чисел.

12. Дано число в двоичной системе счисления. Разработать машину Тьюринга, которая будет умножать это число на два (в двоичной системе счисления со сдвигом влево).

13. Сконструировать машину Тьюринга, которая выступит в качестве двоично-восьмеричного дешифратора.

14. Даны два натуральных числа n и m, заданных в унарной системе счисления. Числа n и m представлены наборами символов «1», разделенных «/». В конце набора стоит знак « = ». Разработать машину Тьюринга, которая будет производить деление нацело двух натуральных чисел n и m и находить остаток от деления.

15. Построить машину Тьюринга, которая выполняет деление на три в унарной системе счисления.

16. Дан алфавит A = {a, b, c, …, z} и внутренние состояния Q = {q_1, q_2, …, q_z}. Построить машину Тьюринга, которая подсчитывает количество букв *a* в заданной последовательности.

17. Даны два целых положительных числа в двоичной системе счисления. Разработать машину Тьюринга, которая будет находить сумму этих чисел.

18. На ленте машины Тьюринга записано число в двоичной системе счисления. Каретка находится над крайней правой цифрой. Записать цифры этого числа в обратном порядке.

19. Дано натуральное число *n*, представленное в унарной системе счисления. Выяснить, является это число четным или нечетным.

20. Сконструировать машину Тьюринга, которая выступит в качестве четверично-двоичного дешифратора.

21. На ленте машины Тьюринга в произвольном порядке записаны 4 буквы *k, l, m, n*. Каретка обозревает крайнюю левую букву. Необходимо построить машину Тьюринга, которая расположит эти буквы по алфавиту.

22. На ленте машины Тьюринга в трех секциях в произвольном порядке записаны 4 цифры 3, 5, 7, 9. Каретка обозревает крайнюю левую цифру. Необходимо построить машину Тьюринга, которая расположит эти цифры в порядке убывания.

23. Дан алфавит A = {+, =, 0, 1} и внутренние состояния Q = {q_1, q_2, …, q_z}. Построить машину Тьюринга, проверяющую, является ли исходная последовательность арифметическим выражением.

24. Дан алфавит A = {(,)} и внутренние состояния Q = {q_1, q_2, …, q_z}. Построить машину Тьюринга, проверяющую, верно ли в исходной последовательности расставлены скобки.

Практическое занятие 3

Работа с эмулятором МТ

Знакомство с эмулятором «Машина Тьюринга»

1. Запустите эмулятор «ТМ»: ЮУрГГПУ>Учебно-метод. материалы>IITiMOI>ISiT>3k>TA>Emuliytors>TM.swf.

2. Познакомьтесь с интерфейсной частью программы, рассмотрите возможности меню (рис. 1.1);

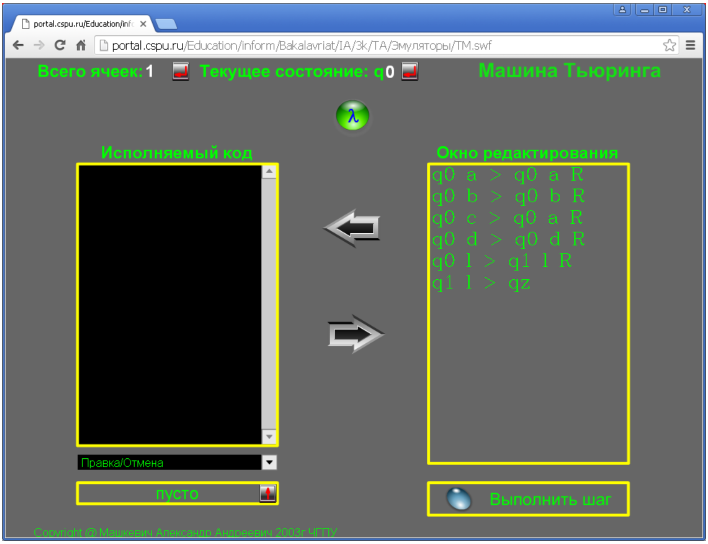


Рис. 1.1. Внешний вид эмулятора МТ

Задание 1. На ленте машины Тьюринга содержится последовательность символов “+”. Напишите программу для машины Тьюринга, которая каждый второй символ “+” заменит на “–”. Замена начинается с правого конца последовательности. Автомат в состоянии q_1 обозревает один из символов указанной последовательности.

В состоянии q_1 машина ищет правый конец числа, в состоянии q_2 – пропускает знак “+”, при достижении конца последовательности – останавливается. В состоянии q_3 машина знак “+”

заменяет на знак “–”, при достижении конца последовательности она останавливается.

Задание 2. Требуется построить машину Тьюринга, которая прибавляет единицу к числу на ленте. Входное слово состоит из цифр целого десятичного числа, записанных в последовательные ячейки на ленте. В начальный момент машина находится против самой правой цифры числа.

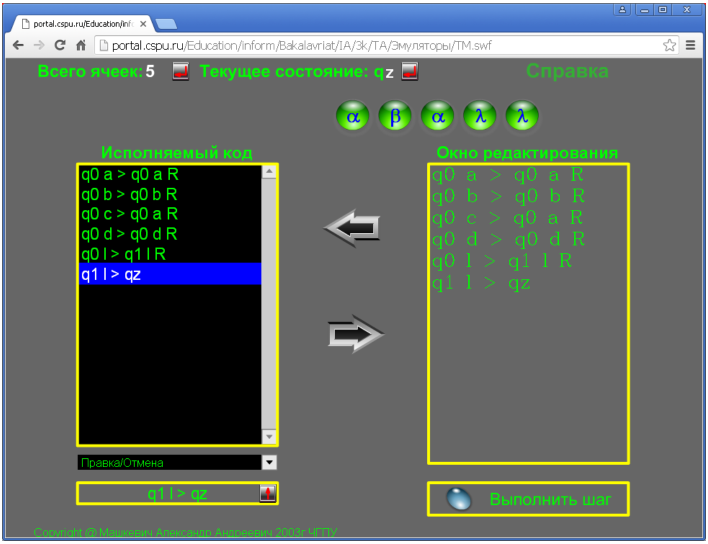


Рис. 1.2. Пример вычисления

Решение. Машина должна прибавить единицу к последней цифре числа. Если последняя цифра равна 9, то ее нужно заменить на 0 и прибавить единицу к предыдущей цифре. В этой машине Тьюринга q_1 – состояние изменения цифры, q_z – состояние останова. Если в состоянии q_1 автомат видит цифру 0..8, то он заменяет ее на 1..9 соответственно и переходит в состояние q_z, т.е. машина останавливается. Если же он видит цифру 9, то заменяет ее на 0, сдвигается влево, оставаясь в состоянии q_1. Так продолжается до тех пор, пока автомат не встретит цифру меньше 9. Если же все цифры были равны 9, то он заменит их нулями, запишет 0 на месте старшей цифры, сдвинется влево и в пустой клетке запишет 1. Затем перейдет в состояние q_z, т.е. остановится.

Задание 3. Дан массив из открывающих и закрывающих скобок. Построить машину Тьюринга, которая удаляла бы пары взаимных скобок, т.е. расположенных подряд “( )”.

Например, дано “) (( ) (( )”, надо получить “) (. . (”. Автомат в состоянии q_1 обозревает крайний левый символ строки.

Состояние q_1: если встретили “(”, то сдвиг вправо и переход в состояние q_2. Состояние q_2: анализ символа “(” на парность, в случае парности должны увидеть “)”. Если парная, то возврат влево и переход в состояние q_3. Состояние q_3: стираем сначала “(”, затем “)” и переходим в q_1.

Самостоятельно:

Задание 1. На ленте машины Тьюринга записаны два числа в унарной системе счисления, разделенные *. Построить машину Тьюринга, которая выполнит вычитание. Первое число больше второго.

Задание 2. Дан алфавит A = {0, 1}. Построить машину Тьюринга, которая подсчитывает количество 1.

Вопросы к модулю 1

1. Что такое «алгоритм»?
2. Перечислите и охарактеризуйте основные свойства алгоритма.
3. Какие существуют подходы к уточнению понятия алгоритма?
4. Почему возникла необходимость уточнить данное понятие?
5. Какая функция называется вычислимой, эффективно вычислимой?
6. Чем отличается описание алгоритма от механизма его реализации?
7. Что собой представляет блок-схема?

8. Как называются вершины в блок-схеме?

9. Можно ли записать несколько алгоритмов с помощью блок-схем?

10. Дайте характеристику блок-схем (назначение, свойства и др.).

11. Что называется декартовым произведением множеств?

12. Что называется областью определения функции? множеством значений?

13. Какая функция называется сюръективной? инъективной? биективной?

14. Что является ограничением функции?

15. Дать определение прообраза, композиции.

16. Какая пара объектов a и b называется неупорядоченной?

17. Какие отношения называются бинарными?

18. Какие бинарные отношения называются транзитивными, рефлексивными, симметричными, отношениями эквивалентности?

19. Что такое «класс эквивалентности»?

20. Какое бинарное отношение называется частичным порядком?

21. Перечислить свойства частичного порядка.

22. Сформулировать теорему Бома и Джакопини.

23. Охарактеризовать основные управляющие структуры.

24. Дайте определение машины Тьюринга.

25. Охарактеризовать каждую составляющую машины Тьюринга.

26. Из чего состоит память машины Тьюринга?

27. Бесконечна ли лента в одну сторону? в обе?

28. Сколько символов можно записать в одну ячейку?

29. Конечно ли множество ячеек, заполненных на ленте в любой момент времени?

30. Что обозначают следующие символы:

a) λ; b) q_1; c) q_z?

31. Что происходит с машиной Тьюринга, когда она попадает в состояние q_z?

32. Дать определение функции вычислимой по Тьюрингу.

33. Как выполняются требования к алгоритмам на примере МТ?

34. Сформулировать проблему остановки. Смысл?

35. Существует ли универсальная МТ?

36. Существует ли универсальная МТ с двумя состояниями и двумя символами на ленте?

МОДУЛЬ 2. ПРИМИТИВНО-РЕКУРСИВНЫЕ ФУНКЦИИ. НОРМАЛЬНЫЕ АЛГОРИТМЫ МАРКОВА

Практическое занятие 4

Примитивно-рекурсивные функции

Теория рекурсивных функций появилась в 30-х годах прошлого века. В этой теории, как и вообще в теории алгоритмов, принят конструктивный (финитный) подход, основной чертой которого является то, что все множество исследуемых объектов (в данном случае функций) строится из конечного числа конечных объектов – базиса с помощью простых операций, эффективная вычислимость которых достаточно очевидна. Операции над функциями будем в дальнейшем называть операторами.

В базис включим: константу 0; функцию следования $x'= x+1$; функцию проекции $U_m^n(x_1, x_2, ..., x_n) = x_m (m \le n)$.

Оператором суперпозиции S_m^n называется подстановка в функцию от m переменных m функций от n одних и тех же переменных:

$$S_m^n(h, g_1, ..., g_m) = h(g_1(x_1, ..., x_n), ..., g_m(x_1, ..., x_n)) = f(x_1, ..., x_n).$$

Суперпозиция дает новую функцию от n переменных.

Таким образом, если заданы функции U_m^n и операторы S_m^n, то можно считать заданными всевозможные операторы подстановки функций в функции, а такие переименования, перестановки и отождествления переменных.

Пример 1.

$$f(x_2, x_1, x_3, ..., x_n) = f(U_2^2(x_1, x_2), U_1^2(x_1, x_2), x_3, ..., x_n);$$

$$f(x_1, x_1, x_3, ..., x_n) = f(U_1^2(x_1, x_2), U_1^2(x_1, x_2), x_3, ..., x_n).$$

Это определение порождает семейство операторов суперпозиции $\{S_m^n\}$. Благодаря функциям проекции стандартизация суперпозиции не уменьшает ее возможностей: любую подстановку функций в функцию можно выразить через S_m^n, U_m^n.

Оператор примитивной рекурсии R_n определяет $(n+1)$-местную функцию f через n-местную функцию g и $(n+2)$-местную функцию h так:

$$\begin{cases} f(x_1, ..., x_n, 0) = g(x_1, ..., x_n); \\ f(x_1, ..., x_n, y+1) = h(x_1, ..., x_n, y, f(x_1, ..., x_n, y)). \end{cases}$$

Функция называется примитивно-рекурсивной, если она может быть получена из константы 0, функции x′ и функции   с помощью конечного числа применений операторов суперпозиции и примитивной рекурсии.

Пример 2. Сложение $f_+(x, y) = x + y$ примитивно-рекурсивно:

$$f\ (x,0) = x = U_1^1(x);$$

$$f\ (x, y+1) = f_+(x, y) + 1 = (f_+(x, y))'$$

Задание 1. Доказать, что если функция $f(x_1, x_2, ..., x_n)$ примитивно-рекурсивная, то следующие функции примитивно-рекурсивны:

а) $g(x_1, x_2, x_3, ..., x_n) = f(x_2, x_1, x_3, ..., x_n)$ – перестановка аргументов;

б) $\psi(x_1, x_2, \ldots, x_n) = f(x_2, \ldots, x_n, x_1,)$ – циклическая перестановка аргументов.

Задание 2. Доказать, что следующие функции примитивно-рекурсивны:

а) $f(x) = x + 2$;

б) $f(x) = 2^x$;

в) $f(x, y) = 10^{x \cdot y}$;

г) $f(x, y) = [x / y]$.

Задание 3. Какая функция получится с помощью схемы примитивной рекурсии:

а) $f(x, 0) = x$;

$f(x, y+1) = x^{f(x, y)}$;

б) $f(x, 0) = x$;

$f(x, y+1) = (f(x, y))^{x}$.

Задание 4. Доказать, что следующие функции примитивно-рекурсивны:

а) $f(x) = x \dot{-} 1 = \begin{cases} 0, \ \textit{если} \ x = 0; \\ x - 1, \ \textit{если} \ x > 0. \end{cases}$

б) $f(x, y) = x \dot{-} y = \begin{cases} 0, \ \textit{если} \ x \le y; \\ x - y, \ \textit{если} \ x > y. \end{cases}$

в) $f(x, y) = \begin{cases} x - y, \ \textit{если} \ x \ge y; \\ y - x, \ \textit{если} \ x < y. \end{cases}$.

Самостоятельно:

1) $f(x, y) = x \dot{-} (x \dot{-} y)$;

2) $f(x, y) = y + x \dot{-} y$;

3) $g(x, y) = \begin{cases} 0, \ \textit{если} \ x = y; \\ 1, \ \textit{если} \ x \ne y. \end{cases}$

4) $\overline{g}(x, y) = \begin{cases} 1, \ \textit{если} \ x = y; \\ 0, \ \textit{если} \ x \ne y. \end{cases}$

Практическое занятие 5

Примитивно-рекурсивные функции от *n*-переменных

1. Доказать, что если функция $f(x_1, x_2, \ldots, x_n)$ примитивно рекурсивная, то следующие функции примитивно рекурсивны:

а) $h(x_1, \ldots, x_n, x_{n+1}) = f(x_1, \ldots, x_n)$ – введение фиктивного аргумента;

б) $\varphi(x_1, \ldots, x_{n-1}) = f(x_1, x_1, x_2, \ldots, x_{n-1})$ – отождествление аргументов.

2. Доказать, что следующие функции примитивно рекурсивны:

а) $f(x) = x - 2$;

б) $f(x) = 2^{x-1}$.

3. Пусть g – примитивно рекурсивная функция. Доказать, что следующие функции примитивно рекурсивны:

а) $f(x_1, \ldots, x_n, x_{n+1}) = \sum_{i=0}^{x_{n+1}} g(x_1, \ldots, x_n, i)$;

б) $f(x_1, \ldots, x_n, y, z) = \begin{cases} \sum_{i=y}^{z} g(x_1, \ldots, x_n, i), & \text{если } y \le z; \\ 0, & \text{если } y > z; \end{cases}$

Самостоятельно:

1) $f(x_1, \ldots, x_n, x_{n+1}) = \prod_{i=0}^{x_{n+1}} g(x_1, \ldots, x_n, i)$;

2) $f(x_1, \ldots, x_n, y, z) = \begin{cases} \prod_{i=y}^{z} g(x_1, \ldots, x_n, i), & \text{если } y \le z; \\ 0, & \text{если } y > z. \end{cases}$

Индивидуальное задание 3

Примитивно-рекурсивные функции

1. Доказать, что следующие функции примитивно-рекурсивны:

1. $f(x) = x + n$; $f(x) = e^x$; $f(x, y) = (x \dot{-} y) + y$.

2. $f(x) = x!$; $f(x) = x\sqrt{2}$ ·; $f(x, y) = (x+y) \dot{-} y$.

3. $f(x) = 10^x$; $f(x) = a + b \cdot x$, a – const; $f(x, y) = (x+y) \dot{-} (y+x)$.

4. $f(x) = x + 2$; $f(x) = 10^{y-x}$; $f(x, y) = y \dot{-} (y \dot{-} x)$.

5. $f(x) = 2^x$; $f(x) = x - n$, n – const; $f(x, y) = (x \dot{-} y) + (y \dot{-} x)$.

6. $f(x) = x \dot{-} (x \dot{-} y)$; $f(x, y) = x^y$; $f(x) = 2 \cdot x$.

7. $sg(x) = \begin{cases} 0, x = 0, \\ 1, x > 0 \end{cases}$; $f(x) = 2 \cdot x - 1$; $f(x) = 100^{x-1}$.

8. $\overline{sg}(x) = \begin{cases} 1, x = 0, \\ 0, x > 0 \end{cases}$; $f(x) = x - 10$; $f(x, y) = 2^{x \cdot y}$.

9. $x \dot{-} 1 = \begin{cases} 0, x = 0, \\ x - 1, x > 0 \end{cases}$; $f(x) = c \cdot x - d$, c, d – const; $f(x) = x \cdot \sqrt{3}$.

10. $f(x)=3^x$; $f(x)=a - b \cdot x$; a, b – const; $f(x, y) = (x+y) \dot{-} (x \dot{-} y)$.

11. $f(x, y) =(y - x) + x$; $f(x, y)=x^{1-y}$; $f(x) = x / 2$;

12. $f(x) = n - x$; $f(x) = 2^{1-x}$; $f(x, y) = (y \dot{-} x) + x$.

* считать везде "–" урезанным вычитанием "$\dot{-}$".

2. Какая функция получится с помощью схемы примитивной рекурсии (по данной схеме примитивной рекурсии восстановить функцию):

1. $f(x, 0) = x$, $f(x, y+1) = f(x, y)^x$.
2. $f(x, 0) = x$, $f(x, y+1) = x^{f(x, y)}$.
3. $g(x, 0) = 2$, $g(x, y+1) = 2^{g(x, y)}$.
4. $g(x, 0) = 2$, $g(x, y+1) = g(x, y)^2$.
5. $f(x, 0) = x$, $f(x, y+1) = f(x, y) * x$.
6. $f(x, 0) = x$, $f(x, y+1) = f(x, y) + x$.
7. $g(x, 0) = 2$, $g(x, y+1) = g(x, y) * 2$.
8. $g(x, 0) = 2$, $g(x, y+1) = g(x, y) + 2$.
9. $f(x, 0) = x$, $f(x, y+1) = f(x, y)^2$.
10. $f(x, 0) = x$, $f(x, y+1) = 2^{f(x,y)}$.
11. $f(x, 0) = 10$, $f(x, y+1) = f(x, y) + 10$.
12. $g(x, 0) = 2^x$, $g(x,y+1) = g(x, y) * 2^x$.

Практическое занятие 6

Нормальные алгоритмы Маркова (НАМ)

Марковской подстановкой называется операция над словами, задаваемыми с помощью упорядоченной пары слов (*P, Q*), состоящая в следующем. В заданном слове *R* находят первое вхождение слова *P* (если оно есть) и, не изменяя остальных частей слова R, заменяют в нем это вхождение словом *Q*. Полученное слово называется результатом применения марковской подстановки (*P, Q*) к слову *R*. Если же первого вхождения *P* в слово *R* (и, следовательно, вообще нет ни одного вхождения *P* в *R*), то считается, что марковская подстановка (*P, Q*) не применима к слову *R*.

Запись $P \to Q$ называется формулой подстановки (*P,Q*). *P* называется левой частью, *Q* – правой частью в формуле подстановки. Некоторые подстановки называются заключительными. Для обозначения таких подстановок будем использовать запись $P \to \cdot Q$, называя ее формулой заключительной подстановки.

Упорядоченный конечный список формул подстановок в алфавите А

$$\begin{cases} P_1 \to (\cdot)Q_1 \\ P_2 \to (\cdot)Q_2 \\ \vdots \\ P_n \to (\cdot)Q_n \end{cases}$$

называется схемой нормального алгоритма в алфавите А. Запись точки в скобках означает, что она может стоять на этом месте, а может отсутствовать.

Пример. Построить нормальный алгоритм Маркова, заменив в алфавите А = {*a, b, c*} все буквы *a* на *c*. Используем символ α для расширения алфавита А. В = {α}∪А. Схема *Z* нормального алгоритма будет иметь следующий вид:

$$
Z: \begin{cases} \alpha a \longrightarrow c\alpha \\ \alpha b \longrightarrow b\alpha \\ \alpha c \longrightarrow c\alpha \\ \alpha \longrightarrow \cdot\Lambda \\ \Lambda \longrightarrow \alpha \end{cases}
$$

Например, $aacbab \Rightarrow \alpha aacbab \Rightarrow c\alpha acbab \Rightarrow cc\alpha cbab \Rightarrow ccc\alpha bab \Rightarrow cccb\alpha ab \Rightarrow cccbc\alpha b \Rightarrow cccbcb\alpha \Rightarrow cccbcb$.

Этот алгоритм может быть реализован так же следующей схемой:

$$
Z_1: \begin{cases} a \rightarrow c \\ \Lambda \rightarrow \cdot\Lambda \end{cases}.
$$

Задание 1. Что получится в результате следующих марковских подстановок в слово «апельсин»:

а) (Λ, к);

б) (пельс, спир);

в) (ль, Λ).

Задание 2. Построить нормальный алгоритм Маркова для вычисления функции $f(x) = x - 1$ $(x>1)$ в унарной системе счисления.

Задание 3. Исполнить алгоритм вычисления функции $f(x) = x + 1$ в десятичной системе счисления.

Задание 4. Построить нормальный алгоритм Маркова для вычисления функции $f(x) = x - 1$ в десятичной системе счисления.

Задание 5. Дано слово в алфавите A = $\{a, b, c\}$. Построить алгоритм Маркова, присоединяющий слово Q к данному слову.

Решение:

$$\begin{cases} \varepsilon a \to a\varepsilon \\ \varepsilon b \to b\varepsilon \\ \varepsilon c \to c\varepsilon \\ \varepsilon \to \cdot Q \\ \Lambda \to \varepsilon \end{cases}$$

Задание 6. Построить нормальный алгоритм Маркова, удваивающий слово в унарной системе счисления.

Задание 7. Построить нормальный алгоритм Маркова для перевода числа из двоичной системы счисления в четверичную систему счисления.

Самостоятельно:

1. Дан алфавит А = {*a, b, c*}. Заменить все *a* на *bc*.
2. Дан алфавит А = {*a, b, c*}. Удалить все *b*.
3. Построить нормальный алгоритм Маркова для вычисления функции $f(x) = x - 1$ в троичной системе счисления.
4. Построить нормальный алгоритм Маркова для перевода числа из четверичной системы счисления в двоичную систему счисления.

Практическое занятие 7

Нормальные алгоритмы Маркова (работа с эмулятором)

Рассмотрим примеры, в которых демонстрируются типичные приёмы составления нормальных алгоритмов Маркова (НАМ).

Как и в случае машины Тьюринга, для сокращения формулировки задач будем использовать следующие соглашения:

– буквой *P* будем обозначать входное слово;

– буквой *A* будем обозначать алфавит входного слова, т.е. набор тех символов, которые могут входить во входное слово *P* (но в процессе выполнения НАМ в обрабатываемых словах могут появляться и другие символы).

На рисунке 2.1 приведен пример работы эмулятора. Эмулятор находится на внутреннем портале ЮУрГГПУ (ЮУрГГПУ> Учебно-метод. материалы>IITiMOI>ISiT>3k>TA>Emuliytors) в архиве Марков и Нумерация. Рассмотрен пример добавления к слову в алфавите $A = \{a, b, c, d\}$ слова $Q = aaaa$.

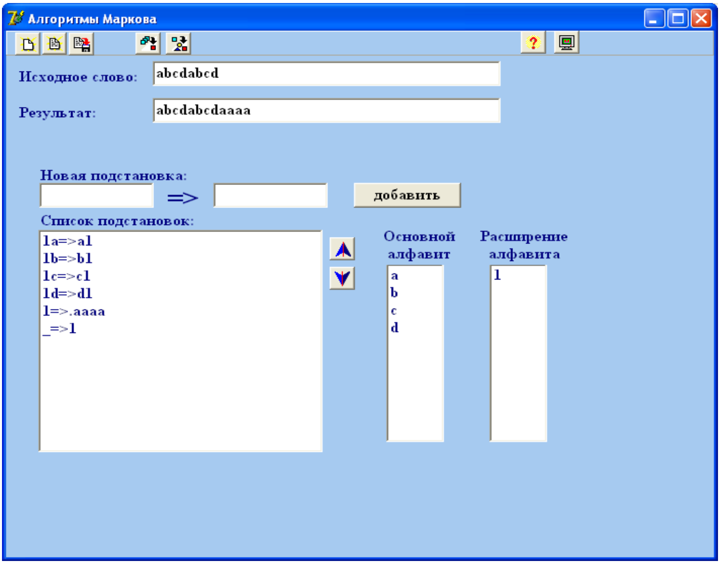


Рис. 2.1. Главное окно эмулятора НАМ

Задание 1. Вставка и удаление символов.

$A = \{a, b, c, d\}$. В слове P требуется заменить первое вхождение подслова bb на ddd и удалить все вхождения символа c. Исполнить по шагам.

Например: $abbcabbca \rightarrow adddabba$.

Задание 2. Перестановка символов.

$A = \{a, b\}$. Преобразовать слово P так, чтобы в начале оказались все символы a, а в конце – все символы b.

Например: $babba \rightarrow aabbb$.

Задание 3. Использование спецзнака.

$A = \{a, b\}$. Удалить из непустого слова P его первый символ. Пустое слово не менять.

Задание 4. Фиксация спецзнаком заменяемого символа.

$A = \{0, 1, 2, 3\}$. Пусть P – непустое слово. Трактуя его как запись неотрицательного целого числа в четверичной системе счисления, требуется получить запись этого же числа, но в двоичной системе.

Например: 0123 → 00011011.

Самостоятельно:

1. Составить НАМ, проверяющий деление десятичного числа на 5.
2. $A = \{a, b, c\}$. Составить НАМ, перемещающий все буквы a в конец слова.
3. Построить нормальный алгоритм Маркова для вычисления разности двух чисел, представленных в унарной системе счисления.

Индивидуальное задание 4

Нормальные алгоритмы Маркова

Задание 1.

1. Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все вхождения последовательности *abc* заменял на символ *f* и удалял первое вхождение пары *cf*.
2. Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ удалял все вхождения последовательности *bc* и удваивал гласные буквы.
3. Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все символы *a* заменял на *f*, а все *f* – на *af*.

4. Построить нормальный алгоритм Маркова, который бы в любом слове из алфавита $A = \{a, b, c, d, e, \dots , x , y, z\}$ удваивал все буквы, стоящие на четных местах в исходном слове.

5. Построить нормальный алгоритм Маркова, который упорядочивает любое слово в алфавите $A = \{a, b, c, d\}$.

6. Поменять местами первый и последний символы слова в алфавите $A = \{a, b, c, d, e\}$.

7. Даны два слова из букв, записанные через * в алфавите $A = \{a, b, c, d, e, f\}$. Вместо пробела поставить « = ».

8. Построить нормальный алгоритм Маркова, реализующий циклический сдвиг первого символа слова в алфавите $A = \{a, b, c, d, e, f, g\}$.

9. Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все вхождения последовательности *cde* заменял на символ *a* и удваивал согласные буквы.

10. Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ удалял все вхождения последовательности *fe* и заменял первое вхождение *da* на *b*.

11. Построить нормальный алгоритм Маркова, который бы переворачивал любое заданное слово в алфавите $A = \{a, b, c, d\}$.

12. Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все символы *e* заменял на *d*, а все *d* – на *de*.

13. Построить нормальный алгоритм Маркова, который бы в любом слове из алфавита $A = \{a, b, c, d, e, \dots, x, y, z\}$ удваивал все гласные буквы.

14. Поменять местами первый и последний символа слова в алфавите $A = \{0, 1, 2, 3, 4, 5\}$.

15. Построить нормальный алгоритм Маркова, реализующий циклический сдвиг последнего символа слова в алфавите $A = \{a, b, c, d, e, f\}$.

16. Построить нормальный алгоритм Маркова для вычисления функции $f(x) = x \bmod 3$ в унарной системе счисления.

17. Построить нормальный алгоритм Маркова для вычисления функции $f(x) = x\ div\ 3$ в унарной системе счисления.

18. Построить нормальный алгоритм Маркова для перевода числа из двоичной системы счисления в восьмеричную систему счисления.

19. Построить нормальный алгоритм Маркова для перевода числа из двоичной системы счисления в шестнадцатеричную систему счисления.

20. Построить нормальный алгоритм Маркова, который бы в слове из алфавита $A = \{a, b, c, d, e, f\}$ все символы e заменял на d, а все d – на de.

21. Построить нормальный алгоритм Маркова, который бы в любом слове из алфавита $A = \{a, b, c, d, e, ..., x, y, z\}$ удваивал все гласные буквы и удалял согласные буквы.

22. Построить нормальный алгоритм Маркова, который в любом слове в алфавите $A = \{a, b, c\}$ переносит все буквы a в конец слова, а буквы b – в конец слова.

23. Поменять местами первый и последний символы слова в алфавите $A = \{0, 1, 2, 3, 4, 5\}$.

24. Построить нормальный алгоритм Маркова для перевода числа из шестнадцатеричной системы счисления в двоичную систему счисления.

Задание 2 (+ 0,2 балла).

1. Даны два слова в унарной системе, записанные через *. Вместо звездочки поставить >, <, =.

2. Построить нормальный алгоритм Маркова для определения системы счисления (найти возможное минимальное основание), в которой записано натуральное число (предполагается, что основание не может превышать 16_{10}).

3. Считая непустое слово P записью двоичного числа в алфавите $A = \{0, 1\}$, определить, является ли это число степенью 2 (1, 2, 4, …). Ответ: слово 1, если является, или слово 0 иначе.

4. Считая непустое слово P в алфавите $A = \{0, 1, 2, 3\}$ записью четверичного числа, проверить, чётное оно или нет. Ответ: слово 0, если чётное, и слово 1 иначе.

5. Считая непустое слово P в алфавите $A = \{0, 1, 2, 3\}$ записью четверичного числа, получить остаток от деления этого числа на 4.

6. Считая непустое слово P в алфавите $A = \{0, 1\}$ записью двоичного числа, получить это же число, но в четверичной системе. (*Замечание*: учесть, что в двоичном числе может быть нечётное количество цифр.)

7. Считая непустое слово P в алфавите $A = \{0, 1, 2\}$ записью троичного числа, увеличить это число на 3.

8. Считая непустое слово P в алфавите $A = \{0, 1, 2\}$ записью положительного троичного числа, уменьшить это число на 3.

9. Считая слово P записью числа в единичной системе счисления, получить запись этого числа в троичной системе. (*Рекомендация*: следует в цикле удалять из «единичного» числа по единице и каждый раз прибавлять 1 к троичному числу, которое вначале положить равным 0).

10. Считая непустое слово P в алфавите $A = \{0, 1, 2\}$ записью числа в троичной системе, получить запись этого числа в единичной системе.

11. Определить, входит ли первый символ непустого слова P в алфавите $A = \{a, b, c\}$ ещё раз в это слово. Ответ: слово a, если входит, или пустое слово иначе.

12. Если в непустом слове P в алфавите $A = \{a, b\}$ совпадают первый и последний символы, то удалить оба этих символа, а иначе слово не менять.

13. Определить, является ли слово P в алфавите $A = \{a, b\}$ палиндромом (перевёртышем, симметричным словом). Ответ: слово a, если является, или пустое слово иначе.

14. Пусть слово P в алфавите $A = \{a, b\}$ имеет нечётную длину. Удалить из него средний символ.

15. Пусть P имеет вид $Q = R$, где Q и R – любые слова из символов a и b. Выдать ответ a, если слова Q и R одинаковы, и пустое слово иначе.

16. Пусть P имеет вид $Q = R$, где Q и R – непустые слова из символов 0 и 1. Трактуя Q и R как записи двоичных чисел (возможно, с незначащими нулями), выдать в качестве ответа слово 1, если эти числа равны, и слово 0 иначе.

17. Пусть P имеет вид $Q > R$, где Q и R – непустые слова из символов 0 и 1. Трактуя Q и R как записи двоичных чисел (возможно, с незначащими нулями), выдать в качестве ответа слово 1, если число Q больше числа R, и слово 1 иначе.

18. Определить, сбалансировано ли слово P в алфавите $A = \{(,)\}$ по круглым скобкам. Ответ: y (да) или n (нет).

Вопросы к модулю 2

1. В чем заключается конструктивный подход, принятый в теории алгоритмов?
2. Как называются операции над функциями?
3. Что входит в базис?
4. Дать определение функции проекции, суперпозиции, примитивной рекурсии.
5. Дать определение примитивно-рекурсивной функции.
6. Как доказать примитивную рекурсию сложения? вычитания?
7. Какое отношение называется примитивно-рекурсивным?
8. Какой предикат называется примитивно-рекурсивным?
9. Какой оператор называется примитивно-рекурсивным?
10. Как задается циклическая перестановка аргументов, отождествление и др.?
11. Как определяется ограниченный оператор наименьшего числа (μ-оператор)?
12. Какой оператор называется оператором обращения?
13. Где определены примитивно-рекурсивные функции?

14. Как задается функция Аккермана?

15. Является ли функция Аккермана примитивно-рекурсивной? общерекурсивной?

16. Дать определение частично-рекурсивной функции.

17. Какая частично-рекурсивная функция называется общерекурсивной?

18. Как выполняются свойства детерминированности, массовости и дискретности для частично-рекурсивных функций?

19. Что собой представляют нормальные алгоритмы Маркова?

20. Дать определение алфавита, слова.

21. Какая операция называется «Марковской подстановкой»?

22. Как записывается формула подстановки? Конечной подстановки?

23. Как записывается схема нормального алгоритма Маркова?

24. Что означает, что алгоритм М применим к слову Р?

25. В чем заключается правило построения последовательности $\{P_i\}$ слов в алфавите А?

26. Что является результатом применения нормального алгоритма к слову Р?

27. Какая функция называется частично вычислимой по Маркову?

28. Сформулировать принцип нормализации Маркова.

29. Что называется расширением алфавита?

30. Какая функция называется вычислимой по Маркову?

31. В каком случае можно говорить о нормальном алгоритме, что он замкнут?

32. Что называется естественным распространением алгоритма Маркова?

33. В каком случае композиция алгоритмов называется нормальной композицией алгоритмов M_1 и M_2?

34. Сформулировать принцип нормализации Маркова.

МОДУЛЬ 3. МАШИНА С НЕОГРАНИЧЕННЫМИ РЕГИСТРАМИ

Практическое занятие 8

Машина с неограниченными регистрами (МНР)

Машина с неограниченными регистрами (МНР) состоит из:

1) ленты, содержащей бесконечное число регистров, обозначаемых через R_1, R_2, R_3, …, каждый из которых в любой момент времени содержит некоторое натуральное число. Число, содержащееся в R_n, мы будем обозначать через r_n;

2) программы P, состоящей из конечного списка команд. Команды бывают следующих четырех видов:

a) команда обнуления – Z(n) заменяет содержание R_n на 0; обозначается также $0 \to R_n$ или $r_n := 0$ (читается, как r_n присваивается 0);

b) команды прибавления единицы – S(n) увеличивает содержимое R_n на 1; обозначается также $r_n + 1 \to R_n$ или $r_n := r_n + 1$ (r_n присваивается $r_n + 1$);

c) команда переадресации – T(m, n) заменяет содержимое R_n числом r_m, содержащимся в R_m; обозначается также $r_m \to R_n$ или $r_n := r_m$ (r_n присваивается r_m);

d) команда условного перехода – J(m, n, q) сравнивает содержимое регистров R_m и R_n, далее, если $r_m = r_n$, то МНР переходит к выполнению q-й команды программы P, если $r_m \neq r_n$, то МНР переходит к выполнению следующей команды в P. Если условный переход невозможен ввиду того, что в P меньше, чем q команд, то МНР прекращает работу.

Команды обнуления, прибавления единицы и переадресации называются арифметическими.

Пример. $f(x, y) = x – y$. Мы получим $x – y$, прибавляя 1 к y (используя соответствующие команды) y раз. Вычисление $x – y$ начинается с начальной конфигурации x, y, 0, 0, …; наша программа продолжает прибавление 1 к r_2, используя R_3 как счетчик числа прибавлений 1 к r_2. Типичной конфигурацией в процессе вычисления является:

R_1	R_2	R_3	R_4	R_5	
x	y	k	0	0	

Программа спроектирована на останов, когда $x = y$, при этом в регистре R_1 оказывается число k, что и требовалось. Итак, нашу задачу решает МНР, программа которой следующая:

I_1 J (1,2,6)
I_2 S (2)
I_3 S (3)
I_4 J (1,2,6)
I_5 J (1,1,2)
I_6 T (3,1)

Таким образом, функция $f(x, y) = x - y$ оказывается вычислимой.

Задания

1. Построить МНР, вычисляющую функцию $f(x, y) = x + y$. Исполнить для чисел 3 и 2.

R_1	R_2	R_3	R_4	R_5	
x+k	y	k	0	0	

I_1 J(3,2,5)	3	2	0
I_2 S(1)	4	2	1
I_3 S(3)	5	2	2
I_4 J(1,1,1)			

2. Построить МНР, вычисляющую функцию

$$f(x) = \begin{cases} \frac{1}{2}x, & \text{если } x \text{ четно}, \\ \text{неопределена}, & \text{если } x \text{ нечетно}. \end{cases}$$

Исполнить алгоритм для числа 6.

R_1	R_2	R_3	R_4	R_5	
x	2k	k	0	0	…

I_1	J(1,2,6)	6	0	0	0
I_2	S(3)	6	1	1	0
I_3	S(2)	6	2	1	0
I_4	S(2)	6	3	2	0
I_5	J(1,1,1)	6	4	2	0
I_6	T(3,1)	6	5	3	0
		6	6	3	0
		3	6	3	0

3. Построить МНР, вычисляющую функцию $f(x, y) = x – 2$.
4. Построить МНР, вычисляющую функцию $f(x, y) = x * y$.
5. Построить МНР, вычисляющую функцию $f(x, y) = x / y$.

Самостоятельно:

1. Построить МНР, вычисляющую функцию

$$f(x) = \begin{cases} \frac{1}{3}x, & если \;\; x \;\; делится \;\; на \;\; 3, \\ неопределена, & в \;\; остальных \;\; случаях. \end{cases}$$

2. Построить МНР, вычисляющую функцию f(x) = 2 * x.

3. Построить МНР, вычисляющую функцию f(x) = x – 2, если x > 2.

4. Построить МНР, вычисляющую функцию

$$f(x, y) = (x + y) / 2.$$

Практическое занятие 9

Машина с неограниченными регистрами (работа с эмулятором)

Задание 1. Знакомство с эмулятором «Машина с неограниченными регистрами».

1. Запустить эмулятор «МНР» с внутреннего. Портала ЮУрГГПУ: ЮУрГГПУ>Учебно-метод. матер. >IITiMOI>ISiT> 3k>TA>Emuliytors>MNR.swf

2. Познакомиться с интерфейсной частью программы, рассмотреть возможности меню (рис. 3.1);

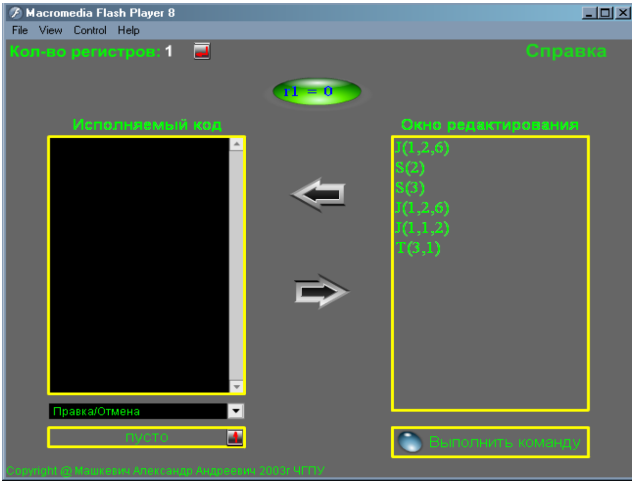


Рис. 3.1. Внешний вид эмулятора МНР

Задание 2. Реализация готовых алгоритмов в эмуляторе «МНР».

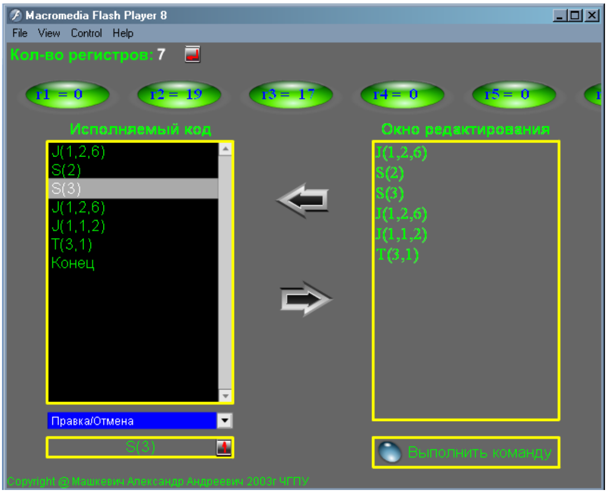


Рис. 3.2. Пример вычислений в эмуляторе «МНР»

1. Рассмотреть пошагово работу эмулятора для задачи, представленной на рисунке 3.2.

2. Какой алгоритм реализован с помощью данной МНР?

Задание 1. Построить МНР, вычисляющую функцию $f(x) = 4 \cdot x$.

Задание 2. Составить программу для вычисления функции

$$f(x) = \begin{cases} 0, & если \quad x = 0, \\ x - 1, & если x <> 0 \end{cases}$$

Самостоятельно:

1. Составить МНР-программу, вычисляющую f(x) = a · x + b.
2. Построить МНР, вычисляющую остаток от деления x на y.

Индивидуальное задание 5

Машина с неограниченными регистрами

Задание 1. Построить машину с неограниченными регистрами, вычисляющую функцию $y(x)$:

1. $y = a * x + b$.
2. $y = x\ div\ 2$.
3. $y = b - a * x$.
4. $y = x^2$.
5. $y = 2 * x - 1$.
6. $y = a * x$.
7. $y = a * x - b$.
8. $y = x\ mod\ 2$.
9. $y = a^x$.
10. Среднее 2-х чисел.
11. $y = 2^x$.
12. Проверка на четность.
13. $y = x^3$.
14. Проверка деления на 3.
15. $y = x / 3$.
16. Проверка на нечетность.

17. $y = x / 2$.

18. Проверка деления на 5.

19. $y = x\ div\ 3$.

20. $y = 3^x$.

21. $y = x\ mod\ 3$.

22. $y = x / a - d$.

23. $y = b - x / a$.

24. Проверка деления на 10.

Задание 2. Построить машину с неограниченными регистрами (не из списка) +0,2 балла.

Вопросы к модулю 3

1. Из чего состоит машина с неограниченными регистрами?
2. Охарактеризовать каждую составляющую МНР.
3. Из чего состоит память машины с неограниченными регистрами?
4. Бесконечна ли лента в одну сторону? в обе?
5. Сколько символов можно записать в одну ячейку?
6. Объяснить команды машины: Z, S, T, J.
7. Какие команды относятся к арифметическим?
8. Что является данными МНР?
9. Что собой представляет память, элементарные шаги, детерминированность МНР?
10. Что является результатом работы МНР?
11. Какая функция называется МНР-вычислимой?
12. Как выполняются требования к алгоритмам на примере МНР?
13. В каком случае вычисление $P(a_1, a_2, \ldots, a_n)$ сходится к b?
14. Что обозначает запись $P(a_1, a_2, a_3, \ldots)$?
15. Что является результатом работы МНР?

МОДУЛЬ 4. ВЫЧИСЛИМОСТЬ И РАЗРЕШИМОСТЬ

Практическое занятие 10

Нумерация программ

Биекция π: N*N→N определяется равенством $\pi(m, n) = 2^m(2 \cdot n + 1) - 1$.

Величину π^{-1} можно задать как $\pi^{-1}(x) = (\pi_1(x), \pi_2(x))$, где π_1 и π_2 – вычислимые функции, определяемые равенством: π_1 = минимальной степени 2 в двоичном разложении числа x;

$$\pi_2 = \frac{1}{2}\left(\frac{x+1}{2^{\pi_1(x)}} - 1\right).$$

Для явного задания биекции $\zeta : N^+ * N^+ * N^+ \to N$ используем функцию π:

$$\zeta(m, n, q) = \pi(\pi(m-1, n-1), q-1).$$

Тогда $\zeta^{-1}(x) = (\pi_1(\pi_1(x)) + 1, \pi_2(\pi_1(x)) + 1, \pi_2(x) + 1)$.

Биекция $\tau : \bigcup_{k>0} N^k \to N$ задается тождеством:

$$\tau(a_1, \ldots, a_к) = 2^{a_1} + 2^{a_1+a_2+1} + 2^{a_1+a_2+a_3+2} + \ldots + 2^{a_1+a_2+\ldots+a_k+k-1} - 1.$$

$\tau^{-1}(x) = (a_1, \ldots, a_k)$, где $a_1 = b_1$, а $a_{i+1} = b_{i+1} - b_i - 1$ $(1 <= i < k)$.

Определим биекцию β: $\mathcal{L} \to N$, которая отображает команды четырех типов в натуральные числа вида $4u$, $4u+1$, $4u+2$, $4u+3$ соответственно; используем функции π и ξ:

$\beta(Z(n)) = 4 \cdot (n - 1)$,

$\beta(S(n)) = 4 \cdot (n - 1) + 1$,

$\beta(T(m, n)) = 4 \cdot \pi(m - 1, n - 1) + 2$,

$\beta(J(m, n, q)) = 4 \cdot \xi(m, n, q) + 3$.

Для вычисления $\beta^{-1}(x)$ найдем сначала такие числа u, r, что $x = 4 \cdot u + r$ при $0 \le r < 4$. Значение r указывает тип команды $\beta^{-1}(x)$, а u позволяет найти конкретно команду данного типа, а именно:

– если $r = 0$, то $\beta^{-1}(x) = Z(u + 1)$;

– если $r = 1$, то $\beta^{-1}(x) = S(u + 1)$;

– если $r = 2$, то $\beta^{-1}(x) = T(\pi_1(u) + 1, \pi_2(u) + 1)$;

– если $r = 3$, то $\beta^{-1}(x) = I(m, n, q)$, где $(m,n,q) = \xi^{-1}(u)$.

Определим биекцию $\gamma : P \rightarrow \mathrm{N}$ следующим образом: если $\mathrm{P} = \mathrm{I}_1,\mathrm{I}_2,\ldots,\mathrm{I}_s$, то $\gamma(\mathrm{P}) = \tau(\beta(\mathrm{I}_1), \ldots, \beta(\mathrm{I}_s))$.

Примеры.

А. Пусть P является программой *T*(1,3), *S*(4), *Z*(6). Вычислим γ (P).

$\beta(T(1,3)) = 4\pi(0,2) + 2 = 4 \cdot (2^0 \cdot (2*2+1)) + 2 = 18$,

$\beta(S(4)) = 4 \cdot 3 + 1 = 13$,

$\beta(Z(6)) = 4 \cdot 5 = 20$.

Следовательно $\gamma(\mathrm{P}) = 2^{18} + 2^{32} + 2^{53} - 1 = 9007203549970431$.

В. Пусть $n = 4127$, найдем P_{4127}. $4127 = 2^5 + 2^{12} - 1$; следовательно, P_{4127} является программой с двумя командами I_1 и I_2, где

$\beta(\mathrm{I}_1) = 5 = 4 \cdot 1 + 1$.

$\beta(\mathrm{I}_2) = 12 - 5 - 1 = 6 = 4 \cdot 1 + 2 = 4\pi(1,0) + 2$.

Согласно определению β, $\mathrm{I}_1 = S(2)$, а $\mathrm{I}_2 = T(2,1)$, и, значит, P_{4127} есть *S*(2), *T*(2,1).

Задание 1.

1. Пусть P является программой Z(3), S(2), T(3,1). Вычислить γ (P).

2. Пусть P является программой S(2), S(3), T(3,1). Вычислить γ (P).

3. Пусть $n = 1089$, найти P_{1089}.

4. Пусть $n = 523$, найти P_{523}.

Задание 2.

Запустить эмулятор с внутреннего портала ЮУрГГПУ (ЮУрГГПУ>Учебно-методич. материалы>IITiMOI>

ISiT> 3k>TA>Emuliytors>), архив Марков и Нумерация. Ознакомиться с возможностями программы.

1. Вычислить номер программы:

1) Z(2), S(1),T(2,3).

2) J(2,3,4), S(2),T(3,1).

2. По номеру восстановить программу:

1) $n = 672$.

2) $n = 1076$.

Вопросы к модулю 4

1. Кто впервые сформулировал принцип, утверждающий пригодность некоторых конкретных уточнений понятия «алгоритм»?

2. Является ли разрешимой задача распознавания эквивалентности примитивно-рекурсивных описаний?

3. Пусть X – множество конечных объектов; в каком случае X называется эффективно счетным?

4. Существует ли невычислимая всюду определенная (или тотальная) функция?

5. Кто предложил метод диагональной конструкции для построения функции f?

6. Как можно найти индекс вычислимой функции?

7. Как обозначается область определения $\Phi_a^{(n)} = \{(x_1, \ldots, x_n) | P_a(x_1, \ldots, x_n)\downarrow\}$?

8. Как обозначается n-местная функция, вычислимая по программе $P_a = f_{P_a}^{(n)}$?

9. Как обозначается множество значений функции $\Phi_a^{(n)}$?

10. Для любого перечисления множества всюду определенных вычислимых функций существует ли общерекурсивная функция, входящая в это перечисление?

11. Что означает, что проблема «$x \in W_x$» неразрешима?

12. Что означает, что проблема «Φ_x всюду определена» неразрешима?

13. Что означает, что проблема «$\Phi_x(y)$ определена» неразрешима?

14. Что означает, что проблема «$\Phi_x = g$», где g – любая фиксированная вычислимая функция, неразрешима?

15. Можно ли проверить, вычисляют ли две программы одну и ту же одноместную функцию?

16. Как называется множество М, если существует общерекурсивная функция $\psi_M(x)$, такая, что $a \in M$ тогда и только тогда, когда для некоторого x $a = \psi_M(x)$?

17. Существует ли множество М, которое перечислимо, но не разрешимо?

18. В чем смысл теоремы Райса (с точки зрения программирования)?

19. Множество квадратов натуральных чисел $M = \{a| \ a = x^2\}$ перечислимо и разрешимо?

МОДУЛЬ 5. ЭФФЕКТИВНЫЕ ОПЕРАЦИИ НА ВЫЧИСЛИМЫХ ФУНКЦИЯХ. СЛОЖНОСТЬ ВЫЧИСЛЕНИЯ

Практическое занятие 11

Машина Поста[1]

Машина Поста – «абстрактная» машина. Она состоит из ленты и каретки. Лента, расположенная горизонтально, бесконечна и разделена на секции (или ячейки) одинакового размера (рис. 5.1):

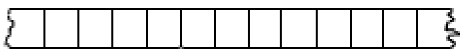


Рис. 5.1

Порядок, в котором расположены секции ленты, подобен порядку, в котором расположены все целые числа. Считая, что система координат жестко сопоставлена лентой, получим возможность указывать какую-либо секцию ленты, называя ее порядковый номер или координату (рис. 5.2):

Рис. 5.2

В каждой секции ленты может быть ничего не записано (такая секция называется пустой), либо записана метка V (тогда секция называется отмеченной) (рис. 5.3):

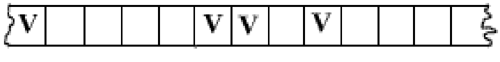


Рис. 5.3

1 Лабораторная работа составлена на основе и с использованием программных средств квалификационной работы выпускницы факультета информатики ЮУрГГПУ А.В. Бурлаковой

Информация о том, какие секции пусты, а какие отмечены, называется состоянием ленты. Каретка может передвигаться вдоль ленты влево и вправо. Когда она неподвижна, она стоит ровно против одной секции ленты; говорят, что каретка обозревает эту секцию или держит ее в поле зрения (рис. 5.4):

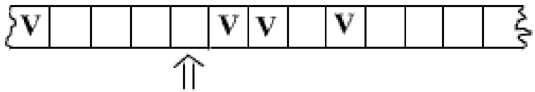


Рис. 5.4

Работа машины Поста состоит в том, что каретка передвигается вдоль ленты и печатает или стирает метки. Эта работа происходит по программе, заданной пользователем. Каждая программа машины Поста состоит из команд.

Команда машины Поста – выражение, имеющее один из следующих шести видов (буквы i и j означают натуральные числа 1, 2, 3, 4, 5, ..., где i – номер команды, j – номер команды для отсылки):

Таблица

Команды машины Поста

Вид команды	Действие машины Поста
i => j	двигаться влево (сдвиг каретки влево)
i <=	двигаться вправо (сдвиг каретки вправо)
i V j	печатать метку
i § j	стереть метку (если она имеется в ячейке)
i ? j_1 j_2	передать управление (если в ячейке есть метка, то выполняется команда j_1, иначе – j_2)
i стоп	остановиться

Программа машины Поста – конечный непустой (т.е. содержащий хотя бы одну команду) список команд машины Поста, обладающий двумя свойствами:

1. На первом месте в этом списке стоит команда с номером 1, на втором месте (если оно есть) – команда с номером 2 и т.д.; вообще, на k-м месте стоит команда с номером k.

2. Отсылка любой из команд списка совпадает с номером некоторой (другой или той же самой команды списка).

Например, следующий список команд будет программой машины Поста:

1стоп
2 ? $\begin{matrix} \nearrow 4 \\ \searrow 1 \end{matrix}$
3 § 3
4 стоп

Является ли ниже приведенные команды программой машины Поста? Почему?

1 стоп
2? $\begin{matrix} \nearrow 4 \\ \searrow 5 \end{matrix}$
3 § 3
4 стоп

Для работы машины Поста **необходимо**:

1. Иметь некоторую программу.

2. Знать состояние машины, т.е. расставить метки по секциям ленты (или оставить пустыми).

3. Поставить каретку против одной из секций. Мы будем предполагать, что в начальном состоянии (н.с.) машины каретка ставится всегда против секции с номером ноль.

Переход от выполнения одной команды к выполнению другой происходит по следующему **правилу**:

– пусть на k-м шаге выполнялась команда с номером i, тогда, если эта команда имеет единственную отсылку j, то на $k + 1$-м шаге выполняется команда с номером j;

– если эта команда имеет две отсылки j_1 и j_2, то на $k + 1$-м шаге выполняется одна из двух команд – с номером j_1 или с номером j_2;

– если эта команда вовсе не имеет отсылки, то на k+ 1-м шаге и на всех последующих шагах не выполняется никакая команда: машина останавливается.

Задание.

1. По начальному состоянию (н.с.) машины Поста и конечному (к.с.) сказать, какие команды были выполнены (см. рис. 5.5. – рис. 5.10):

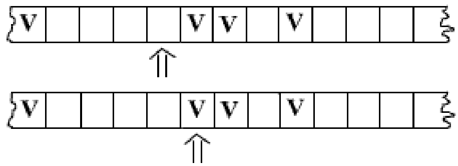


Рис. 5.5

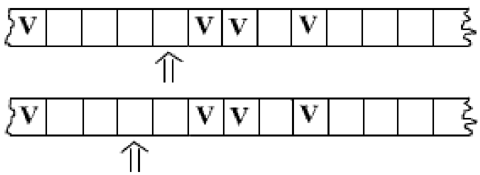


Рис. 5.6

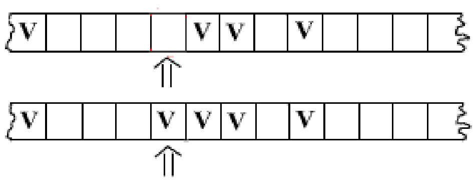


Рис. 5.7

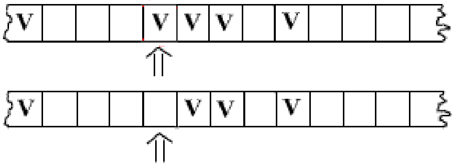


Рис. 5.8

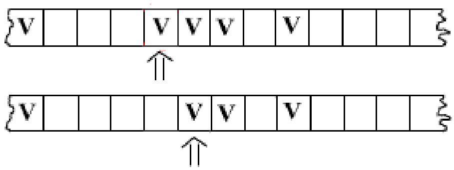


Рис. 5.9

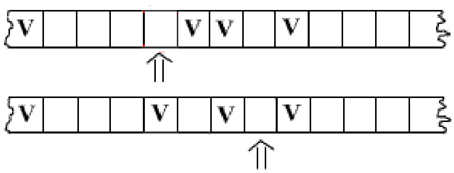


Рис. 5.10

В ходе выполнения программы машина может дойти до:

–выполнения невыполнимой команды (печатания метки в непустой секции или стирания метки в пустой секции). В этом случае выполнение программы прекращается, машина останавливается; происходит **безрезультатная остановка;**

–выполнения команды остановки; программа в этом случае считается выполненной, машина останавливается; происходит **результативная остановка;**

–невыполнения ни одной из команд, указанных в двух вариантах; выполнение программы при этом никогда не прекращается, машина никогда не останавливается; **процесс работы машины происходит бесконечно**.

Машина Поста может производить различные действия над числами. Для этого надо, прежде всего, договориться, как в машине Поста будут записываться числа. Речь всегда будет идти о целых неотрицательных числах 0, 1, 2, 3, 4 ... Рассмотрим конечную последовательность идущих подряд друг за другом отмеченных секций ленты, заключенную между двумя пустыми секциями. Такую последовательность отмеченных секций будем называть **массивом**, а число секций в ней – **длиной** массива. Так, на рис. 5.11 показан массив длины 3, а на рис. 5.12 – три массива: длины 5, длины 1 и длины 2.

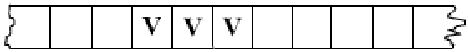


Рис. 5.11

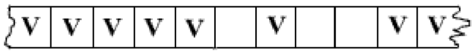


Рис. 5.12

2. Написать программу машины Поста, обладающую следующим свойством: каково бы ни было число **n**, если начальное состояние машины Поста таково, что на ленте имеется машинная запись числа **n** (а в остальном лента пуста) и каретка стоит против самой левой секции записи, то выполнение программы должно привести к результативной остановке, после чего на ленте должно быть записано число **n+1**, причем каретка может стоять где угодно.

Исходя из условия задачи, начальное состояние может быть следующим:

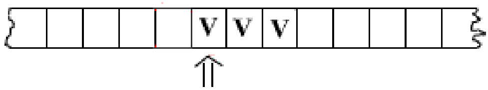


Рис. 5.13

Реализуйте начальное состояние и программу в эмуляторе.

1. Будет ли это решение задачи единственным?

2. Предложить другой вариант для решения данной задачи при конкретных начальных состояниях, но с требованием, чтобы в начальном состоянии каретка обозревала одну из секций

массива (например, рис. 5.14). Проверить работу программ, используя программную реализацию машины Поста.

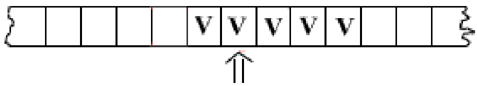


Рис. 5.14

Сложение чисел.

В качестве слагаемых будем использовать только целые неотрицательные числа. Осуществить на машине Поста сложение чисел – это, значит, составить такую программу, которая, будучи применена к ленте, содержащей записи чисел $m(1)$, $m(2)$, ..., $m(\mathrm{k})$ (k >= 2), приводила к результативной остановке, причем после этой остановки на ленте оказывалась бы запись числа $m(1) + m(2) + ... + m(\mathrm{k})$.

Расстоянием между двумя массивами естественно называть число секций, заключенных крайней правой секцией левого массива и крайней левой секцией правого массива.

Задание.

1. Рассмотреть наиболее простой случай вышеприведенной задачи: составить программу сложения двух чисел, записанных на расстоянии 1 друг от друга.

В силу сделанных предположений начальное состояние имеет вид:

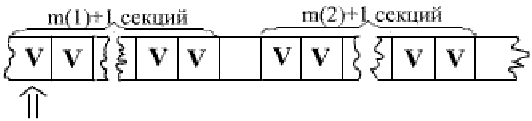


Рис. 5.15

Замечание. Проще всего решить эту задачу, «заполнив» отметкой пустую секцию между массивами. Тогда, если слагаемыми были числа $m(1)$ и $m(2)$, то на ленте возникнет $m(1) + m(2) + 3$ отмеченных секций, в то время как их должно быть $m(1) + m(2) + 1$. Поэтому надо стереть одну лишнюю метку.

Самостоятельно:

1. На ленте МП записаны несколько наборов меток. Составить программу для машины Поста, которая удаляет метку, если она стоит рядом и ставит метку через одну ячейку, если рядом метка отсутствует. Должна получиться последовательность меток и пустых ячеек.

Практическое занятие 12

Эмулятор машины Поста

Машина Поста имеет программную реализацию (расположенную на внутреннем портале ЮУрГГПУ).

1. Запустить эмулятор «Машина Поста»:
   – скопировать с портала папку Post;
   – запустить эмулятор Post.exe со значком .
2. Эмулятор представлен на рис. 5.16.

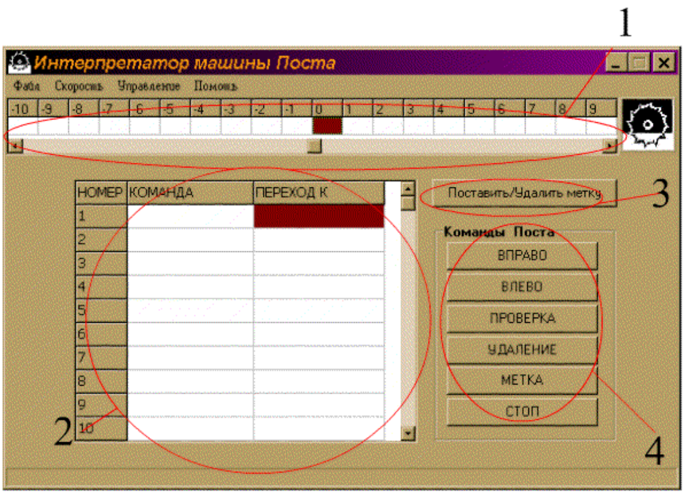


Рис. 5.16. Внешний вид эмулятора

В программной реализации:

1 – лента (от –1000 до 1000 ячеек); каретка имеет вид выделенной ячейки (на ячейке с номером 0);

2 – область ввода команд;

4 – команды машины Поста.

Задание.

1. Самостоятельно изучить назначение кнопки 3.
2. Изучить пункты меню эмулятора «Машина Поста».

Вопросы:

– Какое расширение имеют файлы эмулятора «Машина Поста»?

– Позволяет ли эмулятор выполнять программу пошагово?

– За что отвечает пункт меню «Скорость»?

– Как запустить программу на исполнение?

3. Задать начальное состояние, указанное на рисунке 5.17, и набрать следующую программу в эмуляторе «Машина Поста»:

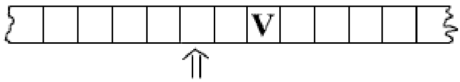


Рис. 5.17

1 V 4
2 <=> 3
3 <= 2
4 => 5
5 ? 4,3;
Запустить программу на пошаговое исполнение.

4. Используя начальное состояние машины, составить программы, реализующие процесс бесконечной работы программы, безрезультатной остановки, результативной остановки. Реализуйте их с помощью эмулятора.

Составленную программу сложения двух чисел реализовать с помощью эмулятора.

5. Рассмотреть более сложный случай задачи о сложении чисел: составить программу сложения произвольного количества чисел, записанных на ленте на расстоянии 1 друг от друга.

В силу сделанных предположений начальное состояние будет следующим:

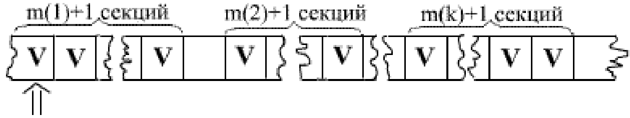


Рис. 5.18

Следовательно, необходимо составить такую программу, которая для любого числа k и любых чисел m(i), m(1), m(2), ..., m(k) давала бы результативную остановку в состоянии, в котором на ленте записано число m(1) + m(2) + ... + m(k).

Подсказка: идея состоит в том, что мы последовательно отщепляем по две метки слева, помещая взамен одну метку в ближайшую пустую секцию справа; если следующая за ближайшей справа пустой секцией также пуста, то это значит, что пора заканчивать.

Самостоятельно:

1. Составить программу для машины Поста, расставляющую пять пар меток, начиная от центра ленты.

2. Составить программу для машины Поста, вычисляющую разность между двумя массивами меток.

Индивидуальное задание 6

1. На информационной ленте машины Поста расположено N массивов меток, отделенных друг от друга свободной ячейкой. Каретка находится над крайней левой меткой первого массива. Определить количество массивов.

2. На ленте машины Поста расположен массив из n меток. Составить программу, действуя по которой, машина выяснит, делится ли число n на 3. Если да, то после массива через одну пустую секцию поставить метку V.

3. На ленту машины Поста нанесены 2 массива меток на некотором расстоянии друг от друга. Соединить эти два массива в один. Каретка находится над крайней левой меткой левого массива.

4. Дан массив меток. Каретка обозревает 1-ю пустую секцию перед началом массива. Раздвинуть массив так, чтобы после каждой метки была пустая секция.

5. Написать для машины Поста программу, проверяющую, делится ли записанное метками число на 5.

6. На информационной ленте машины Поста помечены $2n - 1$ секция. Составить программу отыскания средней помеченной секции и стирания из нее метки.

7. На информационной ленте машины Поста расположены два массива помеченных секций. Написать программу стирания меток, расположенных в большем массиве.

8. На ленте расположен массив в n отмеченных секций. Необходимо справа от данного массива через одну пустую секцию разместить массив, вдвое больше первого (он должен состоять из $2n$ меток). Предложите программу для решения данной задачи. Исходный массив может быть стерт.

9. На ленте дан массив, содержащий $2n$ меток. Составить программу, действуя по которой, машина раздвинет на расстояние в одну метку две половины данного массива.

10. На ленте два массива расположены на расстоянии в одну секцию. Левый массив содержит m меток, правый – n меток. Левый больше правого. Найти разность. По окончании работы машины Поста на ленте должно остаться $m - n$ меток.

11. На ленте два массива в m и n меток. Составить программу выяснения, одинаковы ли массивы по длине.

12. На ленте машины Поста расположен массив из n меток (метки расположены через пробел). Нужно сжать массив так, чтобы все n меток занимали n расположенных подряд секций.

Вопросы к модулю 5

1. Что собой представляют рекурсивные операторы?
2. 1-я и 2-я теоремы о рекурсии. Смысл теоремы.
3. Вычислительный метод.
4. Метод неподвижной точки.
5. Что является примером меры вычислительной сложности?
6. Определение сложности; что такое $t_p^{(n)}$? Примеры.
7. Теорема Блюма.
8. Почему теорема Блюма называется теоремой о пробелах?
9. Машинно-независимость определений.
10. Языки и грамматики. Определения (строка, алфавит, замыкание, язык).
11. Задачи теории формальных языков.
12. Определение грамматики с фразовой структурой, сентенциальной формы, предложения.
13. Задачи распознавания. Определения задач.
14. Задача о коммивояжере.
15. Полиномиальная эквивалентность.
16. Какая схема кодирования называется разумной?
17. Детерминированные машины Тьюринга.
18. Недерминированные машины Тьюринга.
19. Полиномиальная сводимость и *NP*-полные задачи.

ТЕСТОВЫЕ ЗАДАНИЯ

1. Множество $\{x \mid f(x)\,\text{определено}\}$ – это:
1) ограничение функции *f*;
2) область определения функции *f*;
3) множество значений функции *f*.

2. Если из $x, y \in Dom(f)$ и $x \neq y$ следует $f(x) \neq f(y)$, то функция называется:
1) биективной;
2) инъективной;
3) сюръективной.

3. Функция *f* из *A* в *B* называется сюръективной, если:
1) *Dom(f)=A*;
2) *Dom(f)=Ram(f)*;
3) *Ran(f)=B.*

4. Поставить в соответствие. Функция *f* называется:

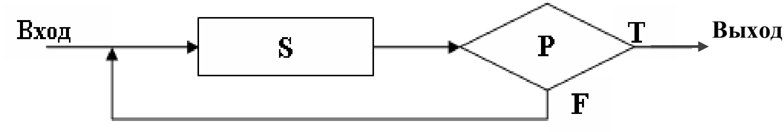
1) инъективной	a) если из $x, y \in Dom(f)$ и $x \neq y$ следует $f(x) \neq f(y)$
2) сюръективной	b) если *Ran(f)=B*
	c) если *Dom(f)=Ram(f)* и *g(f(x))=x*

5. Множество $f^{-1}(Y) = \{x \mid f(x) \in Y\}$ называется:
1) продолжением функции *f*;
2) прообразом Y относительно *f*;
3) ограничением функции *f* на множестве *Y*.

6. Поставить в соответствие:

1) следование	a) управление передается от одного функционального блока к следующему
2) развилка	b) служит для выбора одной из двух альтернатив
3) повторение	c) используется для многократно повторяющегося действия
	d) управление передается от одного функционального блока к другому и обратно

7. Какая структура изображена на схеме?



1) развилка;
2) цикл-пока;
3) цикл-до;
4) структура, эквивалентная структуре цикл-пока.

8. Машина Тьюринга – это:
1) ее полное состояние;
2) устройство, представленное в виде бесконечной ленты, управляющего устройства и головки;
3) набор команд, определяющих ее состояние в каждый конкретный момент.

9. Следующая система правил $q_i a_j \rightarrow q_i a_j d_k$ описывает:
1) конфигурацию МТ;
2) последовательность шагов МТ;
3) полное состояние МТ.

10. Натуральные числа в машине Тьюринга представляются:
1) в виде 0 и 1;
2) в унарном коде;
3) в двоичном коде.

11. Можно ли построить универсальную МТ с двумя символами на ленте?
1) да;
2) нет.

12. Смысл проблемы остановки (с точки зрения программирования) заключается в следующем:
1) существует алгоритм воспроизведения работы МТ по заданному алгоритму;
2) не существует общего алгоритма для отладки программ;

3) не существует алгоритма, который бы по номеру алгоритма определял результат;

4) нет верного ответа.

13. Какой алгоритм реализован следующей машиной Тьюринга (в унарной системе):

$q_1 * \rightarrow q_z * E$

$q_1 1 \rightarrow q_2 \lambda\ R$

$q_2 1 \rightarrow q_2 1\ R$

$q_2 * \rightarrow q_3 1\ L$

$q_3 1 \rightarrow q_3 1\ L$

$q_3 \lambda \rightarrow q_z \lambda\ R$?

1) вычисление функции $f(x) = x + 1$;

2) вычисление функции $\mathrm{f}(x) = x - 1$;

3) вычитание двух чисел;

4) сложение двух чисел.

14. Какая из машин Тьюринга вычисляет функцию $f(x) = x + 1$ (в унарной системе)?

1) $q_1 1 \rightarrow q_z \lambda R$;

2) $q_1 1 \rightarrow q_1 1L;\ q_1 \lambda \rightarrow q_z 1R$;

3) $q_1 1 \rightarrow q_2 1L;\ q_2 \lambda \rightarrow q_1 1R$.

15. Внутренняя память машины Тьюринга – это:

1) лента;

2) конечное множество состояний;

3) нет верного ответа.

16. Внешняя память машины Тьюринга – это:

1) конечное множество состояний;

2) конечное множество команд;

3) нет верного ответа.

17. Машинное слово, обозначаемое $\alpha_1 q_i \alpha_2$, называется:

1) системой команд МТ;

2) конфигурацией или полным состоянием МТ;

3) нет верного ответа.

18. Поставить в соответствие:

1) a_j	a) направление сдвига головки
2) q_i	b) текущая конфигурация
3) d_k	c) символ из алфавита А
4) q_1	d) машинное слово
5) q_z	e) состояние управляющего устройства
	f) начальное состояние управляющего устройства
	g) конечное состояние управляющего устройства

19. Что получится в результате выполнения следующей последовательности команд: $q_1 1 \to q_1 1R;\ q_1 \lambda \to q_1 1R$?

1) МТ остановится, все стирая;
2) МТ остановится, поставив вторую 1;
3) МТ не остановится, заполняя ленту 1.

20. В текущей конфигурации $\alpha_1 a_k q_i a_j \alpha_2$ выполнена команда $q_i a_j \to q_i{}' a_j{}' L$. Какая конфигурация получится в результате выполнения данной команды?

1) $\alpha_1 a_k a_i{}' q_i{}' \alpha_2$;
2) $\alpha_1 a_k q_i{}'{}_2$;
3) $\alpha_1 q_i{}' a_k a_i{}' \alpha_2$.

21. Поставить в соответствие:

1) слова в алфавите ленты	a) элементарные шаги машины
2) конечное множество состояний и лента	b) детерминированность машины
3) считывание и запись символов, сдвиг на ячейку влево или вправо, а также переход управляющего устройства в следующее состояние	c) данные машины Тьюринга
	d) память машины Тьюринга

22. Числовая функция $f(x_1, \dots, x_n)$ правильно вычислима по Тьюрингу, если машина Т работает бесконечно, начиная с $q_1 1^{x_1} * 1^{x_2} * \dots * 1^{x_n}$, когда

1) $f(x_1, \dots, x_n) = y$;
2) $f(x_1, \dots, x_n) = 0$;
3) $f(x_1, \dots, x_n)$ не определена.

23. Среди требований к алгоритмам одно лишнее:

1) детерминированность;

2) дискретность;

3) простота;

4) результативность.

24. Можно ли построить универсальную МТ с двумя символами на ленте и двумя состояниями?

1) да;

2) нет.

25. Можно ли построить универсальную машину Тьюринга?

1) да;

2) нет.

26. Смысл проблемы остановки (с точки зрения программирования) заключается в следующем:

1) не существует алгоритма, который бы по номеру алгоритма определял результат;

2) не существует общего алгоритма для отладки программ;

3) существует алгоритм воспроизведения работы по заданному алгоритму;

4) нет верного ответа.

27. Функция *f* называется вычислимой по Тьюрингу, если:

1) для *f* существует правильная система команд;

2) для *f* существует машина Т, которая ее вычисляет;

3) для *f* не существует машины Т, которая ее вычисляет.

28. Числовая функция $f(x_1, \ldots, x_n)$ вычислима по Тьюрингу, если существует машина Т, такая, что $q_1\ 1^{x_1} * 1^{x_2} * \ldots * 1^{x_n} \Rightarrow q_z\ 1^y$, когда:

1) $f(x_1, \ldots, x_n) = y$;

2) $f(x_1, \ldots, x_n) = 0$;

3) $f(x_1, \ldots, x_n)$ не определена.

29. Что является исходными данными машины T_2 для композиции машин $T_2(T_1)$?

1) система команд машины T_1;

2) система команд машины T_2;

3) результаты работы машины T_2;

4) результаты работы машины T_1.

30. Пусть функция $f(\alpha)$ задана описанием: «если $P(\alpha)$ истинно, то $f(\alpha) = g_1(\alpha)$, иначе $f(\alpha) = g_2(\alpha)$». Тогда $f(\alpha)$ называется?

1) композицией;

2) развилкой;

3) повторением.

31. Пусть функция $f(\alpha)$ задана описанием: «пока истинно $P(\alpha)$ вычислять $g_1(\alpha)$, иначе $f(\alpha)=g_2(\alpha)$». Тогда $f(\alpha)$ называется?

1) композицией;

2) развилкой;

3) повторением.

32. Среди требований к алгоритмам одно лишнее:

1) детерминированность;

2) дискретность;

3) результативность;

4) простота.

33. Среди перечисленных средств описания примитивно-рекурсивных функций одно лишнее.

1) константа 0;

2) функция следования;

3) функция проекции;

4) оператор минимизации;

5) оператор суперпозиции;

6) оператор примитивной рекурсии.

34. Частично-рекурсивные функции называется общерекурсивной, если она:

1) может быть получена с помощью константы 0, функции следования и оператора проекции;

2) всюду определена;

3) все ответы верные.

35. Чему равно значение функции проекции $U_2^5(x_1, x_2, x_3, x_4, x_5)$?

1) 2;

2) 5;

3) x_2;

4) x_5.

36. Примитивно-рекурсивная функция $f_o(x) = x - 1$ определена схемой:

1) $f(0) = 0, f(x + 1) = f_o(x) - 1$;

2) $f(0) = 1$, $f(1) = 0, f(x+1) = x$;

3) $f(0) = 0, f(1) = 0, f(x+1) = f_o(x) + 1$.

37. Всякая эффективно вычислимая функция частично рекурсивна. Это высказывание принадлежит:

1) Райсу;

2) Черчу;

3) Тьюрингу.

38. Смысл теоремы Райса заключается в том, что:

1) не существует общего алгоритма для отладки программ;

2) по синтаксису программы ничего нельзя узнать о ее семантике;

3) все ответы верные.

39. Поставить в соответствие:

1) μ	a) оператор минимизации
2) S_m^n	b) оператор суперпозиции
3) R_n	c) оператор примитивной рекурсии
4) x'	d) функция следования
5) U_m^n	e) функция проекции
	f) оператор тождества

40. Оператор примитивной рекурсии R_n определяет (n)-местную функцию f через (n+1)-местную функцию g и (n+2)-местную функцию h?

1) да; 2) нет.

41.Как можно записать функцию $f(x_1, x_2) = h(g_1(x_1, x_2, x_3), g_2(x_1, x_2, x_3))$ с помощью оператора суперпозиции?

1) R_2; 2) S_2^3; 3) S_3^2.

42. Поставить в соответствие:

1) $h_1(x_1, x_2, x_3) = f(x_3, x_2, x_1)$	a) перестановка аргументов
2) $h_2(x) = f(x, x)$	b) циклическая перестановка аргументов
3) $h_1(x_1, x_2, x_3) = f(x_2, x_3)$	c) отождествление
	d) добавление фиктивных переменных

43. Числа натурального ряда можно получить с помощью:

1) константы 0, функции следования и оператора проекции;
2) константы 0 и оператора минимизации;
3) функции следования и оператора проекции;
4) константы 0 и функции следования.

44. Чему будет равно значение функции $f(x) = 2^{x-1}$ в нуле?

1) ½; 2) $0''$; 3) $0'$; 4) 1; 5) 2.

45.Примитивно-рекурсивная функция $f_o(x, y) = x^y$ определена схемой:

1) $f(x, 0) = 0'$, $f(x, y + 1) = f_+(f_o(x, y), U_1^1(x))$;
2) $f(x, 0) = 0$, $f(x, y + 1) = f_*(f_o(x, y), U_1^1(x))$;
3) $f(x, 0) = 0'$, $f(x, y + 1) = f_*(f_o(x, y), U_1^1(x))$;
4) $f(x, 0) = 0'$, $f(x, y + 1) = f_*(f_o(x, y), 0')$.

46. Какой из трех типов основных алгоритмических моделей занимается переработкой слов в произвольных алфавитах?

1) рекурсивные функции;
2) нормальные алгоритмы Маркова;
3) машины Тьюринга.

47. Функция вычислима машиной Тьюринга тогда и только тогда, когда она частично-рекурсивна. Это высказывание:

1) Черча;
2) Тьюринга–Черча;

3) Тьюринга.

48. Что получится в результате Марковской подстановки (рама, пано) в слово «панорама»?

1) рама;
2) панопано;
3) панорама;
4) рамарама.

49. Восстановить правильную последовательность команд в схеме алгоритма, определяющего, делится число в унарной системе на 3 или не делится.

1) $\Lambda \to \bullet 1$;
2) $1 \to \bullet\Lambda$;
3) $111 \to \Lambda$;
4) $11 \to \bullet\Lambda$.

50. Если A и B два алфавита, причем $B \subseteq A$, то

1) алфавит B называется расширением алфавита A;
2) алфавит B содержится в A;
3) алфавит A называется расширением алфавита B;
4) нет верного ответа.

51. Поставить в соответствие:

	a) заключительная подстановка
1) (P, Q)	b) формула подстановки
	c) упорядоченная пара слов
2) P → •Q	d) схема нормального алгоритма

52. После применения алгоритма получилась следующая последовательность: $babca \Rightarrow bbca \Rightarrow bbc \Rightarrow bc \Rightarrow c \Rightarrow \Lambda \Rightarrow \bullet\Lambda$. Выбрать соответствующую схему.

1) $\begin{array}{l} a \to \Lambda \\ b \to \Lambda \\ c \to \Lambda \\ d \to \Lambda \\ \Lambda \to \bullet\Lambda \end{array}$
2) $\begin{array}{l} a \to a \\ b \to b \\ c \to c \\ d \to d \\ \Lambda \to \bullet\Lambda \end{array}$
3) $\begin{array}{l} a \to \Lambda \\ b \to \Lambda \\ c \to \bullet\Lambda \\ d \to \Lambda \end{array}$

53. Нормальный алгоритм определяет схема:

$$\alpha 1 \to 11\alpha$$
$$\alpha \to \bullet\Lambda$$
$$\Lambda \to \alpha$$

Что получится в результате применения данного алгоритма к слову 111?

1) 111111;
2) 11111;
3) 1111;
4) 111.

54. Что получится в результате подстановки (Λ, па) в слово «памама»?

1) мама;
2) папа;
3) папамама;
4) мапамапа.

55.Что получится в результате подстановки (пап, Λ) в слово «папамама»?

1) амама;
2) паппапа;
3) папама;
4) не применима.

56.Нормальный алгоритм определяет схема: $\begin{cases} a \to \bullet\Lambda \\ b \to b \end{cases}$. Что получится в результате применения данного алгоритма к слову $bbbb$?

1) алгоритм не применим к данному слову;
2) $abbbb$;
3) $aaaa$;
4) $bbbb$.

57. Операция над словами, задаваемыми с помощью упорядоченной пары слов (P, Q) называется:

1) результатом применения Марковской подстановки;

2) схемой нормального алгоритма;

3) нет верного ответа.

58. Функция f, заданная на некотором множестве слов алфавита A, называется нормально вычислимой, если:

1) существует схема нормального алгоритма, вычисляющая эту функцию;

2) найдется такое расширение B данного алфавита A ($A \subseteq B$) и такой нормальный алгоритм в B, что каждое слово P (в алфавите A) из области определения функции f этот алгоритм перерабатывает в слово $f(P)$;

3) найдется такое расширение A данного алфавита B ($B \subseteq A$) и такой нормальный алгоритм в B, что каждое слово P (в алфавите A) из области определения функции f этот алгоритм перерабатывает в слово $f(P)$.

59. Функции $f(x)=x+1$ и $\varphi_3(x)$ нормально вычислимы?

1) да; 2) нет.

60. Невычислимая всюду определенная функция

1) существует;

2) не существует;

3) является счетной;

4) называется гёделевой.

61. Какое из высказываний является истинным?

1) проблема определения общерекурсивности алгоритмов разрешима;

2) универсальный алгоритм существует;

3) частный случай алгоритмически неразрешимой проблемы не разрешим;

4) существует общий алгоритм для отладки программ.

62. Какую из команд не имеет машина с неограниченными регистрами (МНР):

1) условного перехода;

2) прибавления единицы;

3) переадресации;

4) цикла.

63. Поставить в соответствие командам МНР их назначение:

1) Z_n	a) заменяет содержимое R_m на число r_n из R_n
2) S_n	b) заменяет содержимое R_n на число r_m из R_m
3) $T(m, n)$	c) заменяет содержимое R_n на 0
4) $J(m, n, q)$	d) увеличивает содержимое R_n на 1
	e) сравнивает содержимое R_n и R_m и осуществляет переход на команду с номером q

64. Если $r_m = r_n$ для команды $J(m, n, q)$, то

1) будет осуществлен переход на следующую команду;

2) будет осуществлено вычисление команды с номером q;

3) нет верного ответа.

65. Какая команда МНР не относится к арифметическим командам?

1) обнуления;

2) переадресации;

3) прибавления единицы;

4) условного перехода.

66. Что получится в результате выполнения следующей МНР-программы:

I_1 J (3, 2, 5)
I_2 S (1)
I_3 S (3)
I_4 J (1, 1, 1),

если в R_1 содержится число 4, а в R_2 число 2:

1) 2; 2) 6; 3) 8.

67. Внутренняя память МНР – это:

1) лента;

2) конечное множество состояний;

3) нет верного ответа.

68. Запись $P(a_1, a_2, \ldots, a_n)$ обозначает, что:

1) вычисление P останавливается;

2) вычисление P никогда не останавливается;

3) нет верного ответа.

69. Поставить в соответствие:

1) функция f называется МНР-вычислимой;

2) вычисление P ($a_1, a_2, ..., a_n$) сходится к b;

3) МНР (машина с неограниченными регистрами) с программой P МНР-вычисляет f;

a) если существует такая МНР, которая МНР-вычисляет f;

b) если P ($a_1, a_2, ..., a_n$) ↓ и в заключительной конфигурации в регистре R_1 находится b;

c) если для всех $a_1, a_2, ..., a_n$, b имеет место P ($a_1, a_2, ..., a_n$) ↓ b.

70. Какую из команд имеет машина с неограниченными регистрами (МНР):

1) цикла;

2) присваивания;

3) условного перехода;

4) все ответы верные.

71. Какую из команд не имеет машина с неограниченными регистрами (МНР):

1) прибавления единицы;

2) условного перехода;

3) переадресации;

4) нет верного ответа.

72. Какая команда МНР выполняет замену содержимого R_n на число r_m из R_m:

1) Z_n;

2) S_n;

3) $J(m,n,q)$;

4) нет верного ответа.

73. Вычисление команды с номером q будет осуществлено (для команды J (m, n, q)), если:

1) $r_m = r_n$; 2) $r_m \neq r_n$; 3)нет верного ответа.

74. Какая из команд МНР относится к арифметическим командам?

1) условного перехода;

2) прибавления единицы;

3) все ответы верные.

75. Что получится в результате выполнения следующей МНР-программы:

I_1 T (1, 3)
I_2 S (5)
I_3 J (2, 5, 9)
I_4 Z (4)
I_5 J (1, 4, 2)
I_6 S (3)
I_7 S (4)
I_8 J (1, 1, 5)
I_9 T (3, 1)

если в R_1 содержится число 6, в R_2 – число 3, в R_3, R_4, R_5 – нули:

1) 2; 2) 3; 3) 9; 4) 18.

76. Внешняя память МНР – это:

1) лента;

2) конечное множество команд;

3) конечное множество состояний.

77. Внутренняя память МНР – это:

1) лента;

2) конечное множество команд;

3) конечное множество состояний.

78. Запись P $(a_1, a_2, a_3, \ldots)$ обозначает, что:

1) вычисление P останавливается;

2) вычисление P никогда не останавливается;

3) нет верного ответа.

79. Вычисление P $(a_1, a_2, \ldots, a_n)$ сходится к b, если:

1) существует такая МНР, которая МНР-вычисляет f;

2) для всех $a_1, a_2, \ldots, a_n, b$ имеет место P $(a_1, a_2, \ldots, a_n) \downarrow b$;

3) P $(a_1, a_2, \ldots, a_n) \downarrow$ и в заключительной конфигурации в регистре R_1 находится b.

80. Для любой команды $I_к$ и содержимого регистров r_1, r_2, r_3, … однозначно заданы:

a) следующая к выполнению команда;

b) последовательность $r_1^{'}$, $r_2^{'}$, $r_3^{'}$, … , которую нужно записать на ленту вместо r_1, r_2, r_3, … . Это –

1) дискретность машины;
2) детерминированность МНР;
3) элементарные шаги машины;
4) нет верного ответа.

81. Что является результатом работы МНР?

1) последовательность ее шагов;
2) заключительная конфигурация после остановки вычисления;
3) какая-то команда и переход к выполнению следующей команды;
4) нет верного ответа.

82. Поставить в соответствие:

1) данные МНР	a) последовательность натуральных чисел, записанная на ленте
2) память МНР	b) последовательность шагов машины
3) элементарные шаги машины	c) лента и конечное множество команд
	d) какая-то арифметическая команда и переход к выполнению следующей команды

83. Новые функции можно получить перестановкой и отождествлением переменных или добавлением новых фиктивных переменных. Поставить в соответствие функции ее название:

1) $h_2(x) = f(x, x)$	a) рекурсия
2) $h_1(x_1, x_2) = f(x_2, x_1)$	b) подстановка
3) $h_1(x_1, x_2, x_3) = f(x_2, x_3)$	c) отождествление
	d) перестановка
	e) добавление фиктивных переменных

84. Функция $g\ (x, z) = \mu y\ (y + z - x = 0)$ получена с помощью оператора минимизации. Чему равно $g\ (7, 2)$?

1) 0;
2) 2;
3) 5;
4) 7;
5) 9.

85. Всякий алгоритм может быть реализован МНР или интуитивно и неформально определенный класс вычислимых частичных функций совпадает с классом b МНР-вычислимых функций. Это тезис:

1) Райса;
2) Тьюринга;
3) Черча;
4) Тьюринга-Черча.

86. Поставить в соответствие. Для каждого $a \in N$ и $n \geq 1$:

1) $\Phi_a^{(n)}$	a) область определения $\Phi_a^{(n)} = \{(x_1, \ldots, x_n) \mid P_a(x_1, \ldots, x_n) \downarrow \}$
2) $W_a^{(n)}$	b) множество значений функции $\Phi_a^{(n)}$
3) $E_a^{(n)}$	c) n-местная функция, вычислимая по программе $P_a = f_{P_a}^{(n)}$
	d) одноместная вычислимая функция, представленная в перечислении $\Phi_0, \Phi_1, \Phi_2, \ldots$

87. Для любого перечисления множества всюду определенных вычислимых функций:

1) существует общерекурсивная функция, входящая в это перечисление;

2) существует общерекурсивная функция, не входящая в это перечисление;

3) не существует общерекурсивной функции.

88. Поставить в соответствие.

1) проблема определения общерекурсивности алгоритмов неразрешима;

2) проблема самоприменимости алгоритмов неразрешима;

3) проблема определения результативности алгоритма неразрешима;

a) т.е. проблема «$\Phi_x(y)$ определена» неразрешима;

b) т.е. проблема «Φ_x всюду определена» неразрешима;

c) т.е. проблема «$\Phi_x = g$», где g – любая фиксированная вычислимая функция, неразрешима;

d) т.е. проблема «$x \in W_x$» неразрешима.

89. Выбрать верное объяснение теоремы: проблема «$\Phi_x = g$», где g – любая фиксированная вычислимая функция, неразрешима. Не существует общего метода осуществить проверку того:

1) будет ли программа вычислять нулевую функцию;

2) вычисляет ли программа конкретную функцию;

3) вычисляют ли две программы одну и ту же одноместную функцию.

90. Выбрать неверное утверждение. Из теоремы Райса следует, что:

1) по номеру вычислимой функции f ничего нельзя узнать о свойствах этой функции;

2) по синтаксису программы ничего нельзя узнать о её семантике;

3) не существует общего алгоритма для отладки программ.

ЗАКЛЮЧЕНИЕ

В настоящее время уделяется большое внимание вопросам фундаментализации обучения информатике. В учебных планах вузов предусмотрено изучение теоретической информатики, как по направлениям подготовки бакалавров, так и по направлениям подготовки магистров.

В практикуме представлены материалы для изучения курсов «Теория алгоритмов» и «Теоретические основы информатики и современных информационных технологий».

Для организации самостоятельной работы студентов, обучающихся по направлениям «Педагогическое образование» и «Информационные системы и технологии», в пособии представлены индивидуальные задания и вопросы к тестам.

Пособие может быть использовано при обучении по программам бакалавриата и магистратуры. При изучении дисциплины «Теория алгоритмов» используется модульно-рейтинговая система контроля знаний студентов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Задачи и упражнения по математической логике, дискретным функциям и теории алгоритмов [Текст]: учебное пособие для вузов / М.М. Глухов, О.А. Козлитин, В.А. Шапошников, А.Б. Шишков. – СПб.: Лань, 2008.
2. Игошин, В.И. Математическая логика и теория алгоритмов [Текст] / В.И. Игошин. – Саратов, 1991.
3. Клини, С.К. Введение в метаматематику [Текст] / С.К. Клини. – М.: Изд-во иностр. лит., 1957.
4. Крупский, В.Н. Математическая логика и теория алгоритмов [Текст]: учебное пособие для бакалавров / В.Н. Крупский, В.Е. Плиско. – М.: Академия, 2013.
5. Колмогоров, А.Н. К определению алгоритма [Текст] / А.Н. Колмогоров, В.А. Успенский // Успехи математических наук. – № 4. – 1958. – С. 3–28.
6. Кузнецов, О.П. Дискретная математика для инженера [Текст] / О.П. Кузнецов, Г.Н. Адельсон-Вельский. – М.: Энергоатомиздат, 1988.
7. Кук, С.А. Сложность процедур вывода теорем [Текст] / С.А. Кук // Кибернетический сборник. – М.: Мир, 1975. – Вып. 12. – С. 5–15.
8. Мальцев, И.А. Дискретная математика [Текст]: учебное пособие / И.А. Мальцев. – Изд. 2-е, испр. – СПб.: Лань, 2011.
9. Марков, А.А. Теория алгорифмов [Текст] / А.А. Марков // Тр. матем. ин-та АН СССР. – Т. 42. – 1954.
10. Марков, А.А. Теория алгорифмов [Текст] / А.А. Марков, Н.И. Нагорный. – М.: Наука, 1984.
11. Матрос, Д.Ш. Теория алгоритмов [Текст]: учеб. для вузов / Д.Ш. Матрос, Г.Б. Поднебесова. – М.: Бином, Лаборатория знаний, 2008.
12. Мендельсон, Э. Введение в математическую логику [Текст] / Э. Мендельсон. – М., 1976.
13. Минский, М. Вычисления и автоматы [Текст] / М. Минский. – М.: Мир, 1971.

14. Теория рекурсивных функций и эффективная вычислимость [Текст] / Х. Роджерс. – М.: Мир, 1972.
15. Теория алгоритмов. Примерная программа дисциплины [Электронный ресурс] / В.А. Стеценко // Режим доступа: http://www.edu.ru/db/portal/spe/progs/030100_pp.04.htm.
16. Трахтенброт, Б.А. Алгоритмы и вычислительные автоматы [Текст] / Б.А. Трахтенброт. – М.: Советское радио, 1974.
17. Хьюз, Дж. Структурный подход к программированию [Текст] / Дж. Хьюз, Дж. Мичтом. – М.: Мир, 1980.

ПРИЛОЖЕНИЯ

Приложение 1

Рабочая (модульная) программа

<table>
<tr><td colspan="4">Модуль 1. Машины Тьюринга
Цель – знать основные понятия и уметь разрабатывать машины Тьюринга</td></tr>
<tr><td colspan="4">Содержание раздела</td></tr>
<tr><td>План лекций</td><td colspan="2">План лабораторных работ</td><td>План семинарских занятий</td></tr>
<tr><td>Лекция 1. Основные определения
1. Определение машины Тьюринга.
2. Конфигурация.
3. Вычислимость по Тьюрингу.
Лекция 2. Тезис Тьюринга
1. Действия над машинами Тьюринга.
2. Тезис Тьюринга.

Методическое обеспечение лекций
Презентации лекций хранятся на портале университета.

Список литературы [1, 2, 4, 6, 11, 15]</td><td colspan="2">Машины Тьюринга (4 часа)
– решение задач на обработку нечисловых данных
– решение задач на обработку числовых данных (в унарной и десятичной системах счисления);
– создание циклических Машин Тьюринга.
Эмулятор МТ (2 часа)</td><td>Не предусмотрены учебным планом</td></tr>
<tr><td colspan="4">Самостоятельная работа</td></tr>
<tr><td colspan="2">Инвариантная часть</td><td colspan="2">Вариативная часть</td></tr>
<tr><td colspan="2">Темы, по которым осуществляется самостоятельная работа студентов
– Универсальная машина Тьюринга
– Проблема остановки и другие алгоритмически неразрешимые проблемы</td><td colspan="2">Сдача индивидуальных заданий досрочно +0,2</td></tr>
</table>

<table>
<tr><td colspan="2">Самостоятельная работа</td></tr>
<tr><td>Инвариантная часть</td><td>Вариативная часть</td></tr>
<tr><td>Типы заданий, предлагаемых студентам
– Индивидуальное задание

Форма отчетности
– Сдача и защита индивидуального задания в течение 12 учебных дней со дня выдачи.

Список литературы [1; 2; 4; 6, 11, 15]

Текст заданий, предлагаемых студентам, хранится на портале университета</td><td></td></tr>
<tr><td colspan="2">Паспорт оценочных средств по разделу</td></tr>
<tr><td>Код контролируемой компетенции (или её части) и ее формулировка</td><td>Наименование оценочного средства</td></tr>
<tr><td>Способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве (ОК-3)</td><td>Тест, ИДЗ</td></tr>
<tr><td colspan="2">Текущий контроль за выполнением лабораторных работ
Вопросы для контроля
1. Дайте определение машины Тьюринга.
2. Охарактеризуйте каждую составляющую машины Тьюринга.
3. Из чего состоит память машины Тьюринга?
4. Бесконечна ли лента в одну сторону? в обе?
5. Сколько символов можно записать в одну ячейку?
6. Конечно ли множество ячеек, заполненных на ленте в любой момент времени?
7. Что обозначают следующие символы:
a) λ; b) q_1; c) q_z?
8. Что происходит с машиной Тьюринга, когда она попадает в состояние q_z?
9. Дать определение правильно вычислимой функции по Тьюрингу</td></tr>
</table>

<table>
<tr><td colspan="6">Раздел 2. Рекурсивные функции
Цель – знать основные понятия и уметь работать с рекурсивными функциями</td></tr>
<tr><td colspan="6">Содержание раздела</td></tr>
<tr><td colspan="2">План лекций</td><td colspan="2">План лабораторных работ</td><td colspan="2">План семинарских занятий</td></tr>
<tr><td colspan="2">Лекция 1. Основные определения.
1. Примитивно-рекурсивные функции.
2. Частично-рекурсивные функции.
3. Общерекурсивные функции.
Лекция 2. Тезис Черча.
1. Действия над частично-рекурсивными функциями.
2. Тезис Черча.
Методическое обеспечение лекций
Презентации лекций хранятся на портале университета.
Список литературы [1; 2; 3; 6; 11; 12; 13, 14]</td><td colspan="2">Рекурсивные функции (4 часа)
– доказательство примитивной рекурсивности функций от одной и нескольких переменных;
– восстановление функций по схеме примитивной рекурсии</td><td colspan="2">Не предусмотрены учебным планом</td></tr>
<tr><td colspan="6">Самостоятельная работа</td></tr>
<tr><td colspan="3">Инвариантная часть</td><td colspan="3">Вариативная часть</td></tr>
<tr><td colspan="3">Темы, по которым осуществляется самостоятельная работа студентов
– Общерекурсивные функции
– Связь с понятием рекурсии в программировании
Типы заданий, предлагаемых студентам
– Индивидуальное задание

Форма отчетности
– Сдача и защита индивидуального задания в течение 12 учебных дней со дня выдачи.</td><td colspan="3">Сдача индивидуальных заданий досрочно +0,2</td></tr>
</table>

<table>
<tr><td colspan="2">Список литературы [1; 2; 3; 6; 11, 14]

Текст заданий, предлагаемых студентам, хранится на портале университета</td><td></td></tr>
<tr><td colspan="3">Паспорт оценочных средств по разделу</td></tr>
<tr><td colspan="2">Код контролируемой компетенции (или её части) и ее формулировка</td><td>Наименование оценочного средства</td></tr>
<tr><td colspan="2">Способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве (ОК-3)</td><td>Тест, ИДЗ</td></tr>
<tr><td colspan="3">Текущий контроль за выполнением лабораторных работ
Вопросы для контроля
1. В чем заключается конструктивный подход, принятый в теории алгоритмов?
2. Что входит в базис?
3. Где определены примитивно-рекурсивные функции?
4. Как задается функция Аккермана?
5. Дать определение частично-рекурсивной функции.
6. Какая частично-рекурсивная функция называется общерекурсивной?
7. Как выполняются свойства детерминированности, массовости и дискретности для частично-рекурсивных функций?</td></tr>
<tr><td colspan="3">Раздел 3. Нормальные алгоритмы Маркова
Цель – знать основные понятия и уметь разрабатывать нормальные алгоритмы Маркова</td></tr>
<tr><td colspan="3">Содержание раздела</td></tr>
<tr><td>План лекций</td><td>План лабораторных работ</td><td>План семинарских занятий</td></tr>
<tr><td>Лекция 1. Нормальные алгоритмы Маркова.
1. Схема Маркова.
2. Нормальный алгоритм.
3. Тезис нормализации Маркова.
Методическое обеспечение лекции
Презентации лекций хранятся на портале университета</td><td>Нормальные алгоритмы Маркова (2 часа)
– решение задач на работу с числовыми и нечисловыми объектами;
– доказательство нормальной вычислимости функций.
Эмулятор НАМ
(2 часа)</td><td>Не предусмотрены учебным планом</td></tr>
</table>

<table>
<tr><td>Список литературы
[1; 2; 4; 6; 9, 10, 11]</td><td colspan="3">Методическое обеспечение лабораторной работы
Разработки лабораторных работ хранятся на портале университета</td><td></td></tr>
<tr><td colspan="5">Самостоятельная работа</td></tr>
<tr><td colspan="3">Инвариантная часть</td><td colspan="2">Вариативная часть</td></tr>
<tr><td colspan="3">Не предусмотрена в данном модуле

Текст заданий, предлагаемых студентам, хранится на портале университета</td><td colspan="2">Сдача индивидуальных заданий досрочно +0,2</td></tr>
<tr><td colspan="5">Паспорт оценочных средств по разделу</td></tr>
<tr><td colspan="3">Код контролируемой компетенции (или её части) и ее формулировка</td><td colspan="2">Наименование оценочного средства</td></tr>
<tr><td colspan="3">Способность использовать естественно-научные и математические знания для ориентирования в современном информационном пространстве (ОК-3)</td><td colspan="2">Тест, ИДЗ</td></tr>
<tr><td colspan="5">Текущий контроль за выполнением лабораторных работ
Вопросы для контроля
1. Что собой представляют нормальные алгоритмы Маркова?
2. Дать определение алфавита, слова.
3. Какая операция называется «Марковской подстановкой»?
4. Как записывается формула подстановки? Конечной подстановки?
5. Как записывается схема нормального алгоритма Маркова?
6. Какая функция называется нормально вычислимой?
7. Сформулировать принцип нормализации Маркова</td></tr>
<tr><td colspan="5">Раздел 4. Машина с неограниченными регистрами
Цель – знать основные понятия курса и уметь разрабатывать МНР</td></tr>
<tr><td colspan="5">Содержание раздела</td></tr>
<tr><td colspan="2">План лекций</td><td colspan="2">План лабораторных работ</td><td>План семинарских занятий</td></tr>
<tr><td colspan="2">Лекция 1. Понятие МНР.
1. Определение МНР.
2. МНР – вычислимые функции.
Лекция 2. Порождение вычислимых функций.
1. Соединение программ.
2. Подстановка.
3. Рекурсия</td><td colspan="2">Машина с неограниченными регистрами
(2 часа)
– создание алгоритмов, работающих с нечисловыми объектами;
– создание алгоритмов, работающих с числовыми объектами</td><td>Не предусмотрены учебным планом</td></tr>
</table>

4. Тезис Черча. ***Методическое обеспечение лекций*** Презентации лекций хранятся на портале университета. Список литературы [1; 4; 6; 11, 16, 17]	***Эмулятор МНР*** (2 часа) ***Методическое обеспечение лабораторных работ*** Разработки лабораторных работ хранятся на портале университета	

Самостоятельная работа	
Инвариантная часть	*Вариативная часть*
Темы, по которым осуществляется самостоятельная работа студентов – Программирование для РАМ – Функции, вычислимые на РАМ ***Типы заданий, предлагаемых студентам*** – Индивидуальное задание ***Форма отчетности*** – Сдача и защита индивидуального задания в течение 12 учебных дней со дня выдачи. Список литературы [1; 2; 4; 6; 11, 16, 17] Текст заданий, предлагаемых студентам, хранится на портале университета	Сдача индивидуальных заданий досрочно +0,2

Паспорт оценочных средств по разделу	
Код контролируемой компетенции (или её части) и ее формулировка	Наименование оценочного средства
Способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве (ОК-3)	Тест, ИДЗ

Текущий контроль за выполнением лабораторных работ

Вопросы для контроля

1. Что понимается под алгоритмом?
2. Перечислите и охарактеризуйте основные свойства алгоритма.
3. Какие существуют подходы к уточнению понятия алгоритма?
4. Почему возникла необходимость уточнить данное понятие?
5. Из чего состоит МНР?

<table>
<tr><td colspan="4">6. Объяснить команды машины: Z, S, T, J.
7. Какие команды относятся к арифметическим?
8. Что является данными МНР?
9. Что собой представляет память, элементарные шаги, детерминированность МНР?
10. Что является результатом работы МНР?
11. Какая функция называется МНР-вычислимой?</td></tr>
<tr><td colspan="4">Раздел 5. Вычислимость и разрешимость
Цель – познакомить с основными алгоритмически неразрешимыми проблемами</td></tr>
<tr><td colspan="4">Содержание раздела</td></tr>
<tr><td>План лекций</td><td colspan="2">План лабораторных работ</td><td>План семинарских занятий</td></tr>
<tr><td>Лекция 1. Нумерация вычислимых функций.
1. Нумерация команд.
2. Нумерация программ.
3. Нумерация функций
Лекция 2. Алгоритмически неразрешимые проблемы.
1. Универсальный алгоритм.
2. Проблема остановки.
3. Самоприменимость.
4. Теорема Райса.
Методическое обеспечение лекции
Презентации лекций хранятся на портале университета.

Список литературы [1; 5; 7; 8; 11; 16]</td><td colspan="2">Нумерация программ (2 часа)
– вычисление номера по программе МНР
– восстановление программы МНР по номеру</td><td>Не предусмотрены учебным планом</td></tr>
<tr><td colspan="4">Самостоятельная работа</td></tr>
<tr><td colspan="2">Инвариантная часть</td><td colspan="2">Вариативная часть</td></tr>
<tr><td colspan="2">Темы, по которым осуществляется самостоятельная работа студентов
1. Алгебра разрешимых множеств.
2. Алгебра перечислимых множеств.
3. Примеры алгоритмически неразрешимых проблем.
4. Модели вычислений, отличные от РАМ.</td><td colspan="2">Сдача индивидуальных заданий досрочно +0,2</td></tr>
</table>

<table>
<tr><td colspan="2">5. Доказательство равносильности любых двух различных моделей вычислений.

Список литературы [1; 5; 7; 8; 11; 16]
Текст заданий, предлагаемых студентам, хранится на портале университета</td><td colspan="2"></td></tr>
<tr><td colspan="4">Паспорт оценочных средств по разделу</td></tr>
<tr><td colspan="2">Код контролируемой компетенции (или её части) и ее формулировка</td><td colspan="2">Наименование оценочного средства</td></tr>
<tr><td colspan="2">Способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве (ОК-3)</td><td colspan="2">Тест</td></tr>
<tr><td colspan="4">Текущий контроль за выполнением лабораторных работ</td></tr>
<tr><td colspan="4">Эффективные операции на вычислимых функциях.
Сложность вычисления
Цель – познакомить студентов с основными вопросами теории сложности</td></tr>
<tr><td colspan="4">Содержание раздела</td></tr>
<tr><td>План лекций</td><td colspan="2">План лабораторных работ</td><td>План семинарских занятий</td></tr>
<tr><td>Лекция 1. Понятие сложности алгоритмов.
1. Определения сложности.
2. Машинно-независимость определений.
3. Теорема Блюма об ускорении. Теорема о пробелах.
4. NP-проблемы
Методическое обеспечение лекции
Презентации лекций хранятся на портале университета.

Список литературы [4; 5; 6, 7, 8, 11, 13]</td><td colspan="2">Машина Поста
(2 часа)
Эмулятор машины Поста (2 часа)</td><td>Не предусмотрены учебным планом</td></tr>
</table>

<table>
<tr><td colspan="2">Самостоятельная работа</td></tr>
<tr><td>Инвариантная часть</td><td>Вариативная часть</td></tr>
<tr><td>Темы, по которым осуществляется самостоятельная работа студентов
– Примеры задач, принадлежащих классам P и NP.
– Примеры NP-полных задач.

Список литературы [4; 5; 6, 7, 8, 11, 13]

Текст заданий, предлагаемых студентам, хранится на портале университета</td><td></td></tr>
<tr><td colspan="2">Паспорт оценочных средств по разделу</td></tr>
<tr><td>Код контролируемой компетенции (или её части) и ее формулировка</td><td>Наименование оценочного средства</td></tr>
<tr><td>Способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве (ОК-3)</td><td>Тест</td></tr>
<tr><td colspan="2">Текущий контроль за выполнением лабораторных работ</td></tr>
</table>

Приложение 2

Содержание самостоятельной работы

Темы для самостоятельного изучения	Изучаемые вопросы	Кол-во часов	Формы самостоятельной работы	Методическое обеспечение	Форма отчетности
Предварительные подходы. Машина Тьюринга (МТ). Универсальная машина Тьюринга. Проблема остановки и другие алгоритмически неразрешимые проблемы	Определение машины Тьюринга. Тезис Тьюринга	6	Выполнение индивидуальных заданий по теме: Машина Тьюринга. Подготовка к тесту	ЮУрГГПУ (внутренний портал) > Учебно-методические материалы > IITiMOI > ISiT > 3k > TA	2 ИДЗ ТЕСТ
Частично-рекурсивные функции. Связь с понятием рекурсии в программировании	Частично-рекурсивные функции. Тезис Черча	4	Выполнение индивидуальных заданий по теме: Доказательство примитивной рекурсии функций. Подготовка к тесту	ЮУрГГПУ (внутренний портал) > Учебно-методические материалы > IITiMOI > ISiT > 3k > TA	2 ИДЗ ТЕСТ
Нормальные алгоритмы Маркова (НАМ)	Тезис Маркова. Действия над нормальными алгоритмами	4	Выполнение индивидуальных заданий по теме: НАМ. Подготовка к тесту	ЮУрГГПУ (внутренний портал) > Учебно-методические материалы > IITiMOI > ISiT > 3k > TA	ИДЗ ТЕСТ СИНКВЕЙН

Темы для самостоятельного изучения	Изучаемые вопросы	Кол-во часов	Формы самостоятельной работы	Методическое обеспечение	Форма отчетности
Машина с неограниченными регистрами (МНР). Программирование для РАМ. Функции, вычислимые на РАМ	Понятие МНР. Порождение вычислимых функций. Тезис Черча.	6	Выполнение индивидуальных заданий по теме: МНР. Подготовка к тесту	ЮУрГГПУ (внутренний портал) > Учебно-методические материалы > IITiMOI > ISiT > 3k > TA	ИДЗ ТЕСТ
Вычислимость и разрешимость. Алгебра разрешимых, перечислимых множеств. Примеры алгоритмически неразрешимых проблем. Модели вычислений, отличные от РАМ. Доказательство равносильности любых двух различных моделей вычислений	Эквивалентность различных подходов к определению понятия «алгоритм». Нумерация вычислимых функций. Алгоритмически неразрешимые проблемы	4	Подготовка к тесту	ЮУрГГПУ (внутренний портал) > Учебно-методические материалы > IITiMOI > ISiT > 3k > TA	ТЕСТ ЭССЕ
Сложность вычислений. Примеры задач, принадлежащих классам P и NP. Примеры NP-полных задач	Понятие сложности вычислений. Теорема Блюма. NP-проблема	4	Подготовка к тесту	ЮУрГГПУ (внутренний портал) > Учебно-методические материалы > IITiMOI > ISiT > 3k > TA	ТЕСТ ИНТЕЛЛЕКТ-КАРТА

Приложение 3

Рейтинг

Для контроля знаний студентов используется модульно-рейтинговая система. Учебный материал разбит на модули (таблицы 3.1 и 3.2). Каждый модуль содержит индивидуальные или практические задания и тест. Модуль может содержать баллы за дополнительные задания.

Таблица 3.1

№	Фамилия, имя	Модуль 1						Модуль 2				
		Доп.	1	2	Эмул	Тест 1	Итог	3	4	Эмул	Тест2	Итог
1	Семенов Андрей	0,2	1,2	1	1	0,74	98	1	1	1	0,31	72,4

Таблица 3.2

Модуль 3				Модуль 4				Модуль 5			Итог
5	Эмул	Тест 3	Итог	Доп.	Лаб.	Тест 4	Итог	6	Тест 5	Итог	
1	1	0,69	81,4	0,1	1	0,69	87,4	1	0,75	85	84,84

Для вычисления рейтинга по каждому модулю используется следующее соотношение: тест*60+индивидуальные задания*40. Если индивидуальных заданий несколько, то 40 делится на их количество. В таблице 3.3 в столбце Итог коэффициенты для двух индивидуальных заданий и практического занятия, соответственно равны 13, 13 и 14.

Таблица 3.3

		C8	D8	E8	F8	G8	
№	Фамилия, имя	Модуль 1					
		Доп.	1	2	Эмул	Тест 1	Итог
1	Семенов Андрей	0,2	1,2	1	1	0,74	=(G8+C8)*60+13*(D8+E8)+14*F8

Учебное издание

Поднебесова Галина Борисовна

ТЕОРИЯ АЛГОРИТМОВ

Практикум

ISBN 978-5-906908-75-9

Работа рекомендована РИСом ЮУрГГПУ
Протокол № 13, пункт 19 от 19.12.2016 г.

Издательство ЮУрГГПУ
454080, г. Челябинск, пр. Ленина, 69

Редактор Е.М. Сапегина
Компьютерная верстка Т.Н. Никитенко
Эксперт А.А. Рузаков

Подписано в печать 22.08.2017. Объем 2,4 уч.-изд. л.
Формат 60×84/16. Бумага типографская
Тираж 100 экз. Заказ ________

Отпечатано с готового оригинал-макета в типографии ЮУрГГПУ
454080, г. Челябинск, пр. Ленина, 69