

Г.Б. Поднебесова

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ
ИНФОРМАТИКИ**

Часть 2

Челябинск

2025

УДК 001.8(021)

ББК 73я73

П 44

Поднебесова Г.Б. Теоретические основы информатики. В 2-х частях. Часть 2: учебно-практическое пособие / Г.Б. Поднебесова. – Челябинск: Изд-во ЗАО «Библиотека А. Миллера». 2025. – 97 с.

ISBN 978-5-93162-895-0 (общий)

ISBN 978-5-93162-925-4 (часть 2)

Учебное пособие содержит материал для изучения курса «Теоретические основы информатики». Пособие предназначено для организации аудиторной и самостоятельной работы студентов, обучающихся по направлению 04.03.05 «Педагогическое образование» с двумя профилями подготовки «Информатика. Иностранный язык» и «Математика. Информатика».

Учебное пособие адресовано преподавателям и учителям, которым интересна данная предметная область.

Рецензенты:

С.А. Загребина, д-р физ.-мат. наук, профессор ЮУрГУ

А.А. Рузаков, канд. пед. наук, доцент ЮУрГГПУ

ISBN 978-5-93162-895-0 (общий)

ISBN 978-5-93162-925-4 (часть 2)

© Г.Б. Поднебесова, 2025

Содержание

Введение.....	4
Содержание разделов дисциплины	7
Темы и планы лабораторных занятий	9
Модуль 1. Теория информации	11
Модуль 2. Сжатие информации. Помехоустойчивое кодирование.....	20
Модуль 3. Основы теории автоматов	41
Модуль 4. Динамическое программирование. Алгоритмы на строках.....	58
Заключение	79
Приложения	81
Приложение 1. Рабочая (модульная) программа по видам деятельности ...	81
Приложение 2. Содержание самостоятельной работы	84
Приложение 3. Рейтинг	86
Приложение 4. Оценочные материалы.....	87
Приложение 5. Пример текста для анализа	96

Введение

Для повышения качества подготовки будущего педагога по инициативе Министерства просвещения России разработаны и внедрены с 1 сентября 2022 года единые подходы к структуре и содержанию программ педагогических вузов – «Ядро высшего педагогического образования». Развитие профессиональных и цифровых компетенций будущих учителей является основной целью этого шага.

В программу дисциплины «Теоретические основы информатики» включены такие разделы, как динамическое программирование и поиск подстроки. Акцент также сделан на использовании программирования при изучении данных дисциплин. Учебный курс вводит студентов в современные проблемы 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 «Теоретические основы информатики» базируется на материале предшествующих курсов «Теория алгоритмов» и «Основы искусственного интеллекта».

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 разработка кейса по разделам дисциплины.

В задачи дисциплины входит:

- подготовка обучающихся к реализации трудовых функций, определенных профстандартом 01.001 Педагог (педагогическая

деятельность в сфере дошкольного, начального, общего, основного общего, среднего общего образования);

- знакомство с наиболее широко используемыми классами информационных моделей и основными математическими методами получения, хранения, обработки, передачи и использования информации;
- изучение математических основ для анализа и синтеза информационных систем;
- обучение применению методов программирования и навыков работы с математическими пакетами для решения практических задач хранения и обработки информации.

В данном случае оптимальной формой контроля усвоения материала становится рейтинговая система оценки учебных достижений студентов, а именно:

- текущий контроль: практические занятия и тесты;
- промежуточный контроль: индивидуальные задания;
- итоговый контроль: экзамен в виде теста.

Данная дисциплина относится к модулю обязательной части Блока 1 «Дисциплины/модули» основной профессиональной образовательной программы по направлению подготовки 44.03.05 «Педагогическое образование (с двумя профилями подготовки)» (уровень образования бакалавриат).

Основной компетенцией, формируемой в результате освоения дисциплины, является способность осваивать и использовать теоретические знания и практические умения и навыки в предметной области при решении профессиональных задач (ПК-1).

Данная компетенция предполагает знание структуры, состава и дидактических единиц предметной области (преподаваемого предмета) (ПК-1.1) и умение осуществлять отбор учебного содержания для его реализации в

различных формах обучения в соответствии с требованиями ФГОС ВО (ПК-1.2).

Планируемые результаты обучения

Таблица 1 – Образовательные результаты по дисциплине

<i>№ n/n</i>	<i>Код и наименование компетенции по ФГОС</i>	
	<i>Код и наименование индикатора достижения компетенции</i>	
1	<p>ПК-1 *способен осваивать и использовать теоретические знания и практические умения и навыки в предметной области при решении профессиональных задач</p> <p>ПК.1.1 знает структуру, состав и дидактические единицы предметной области (преподаваемого предмета)</p> <p>ПК.1.2 умеет осуществлять отбор учебного содержания для его реализации в различных формах обучения в соответствии с требованиями ФГОС ОО</p>	
<i>№ n/n</i>	<i>Код и наименование индикатора достижения компетенции</i>	<i>Образовательные результаты по дисциплине</i>
1	ПК.1.1 знает структуру, состав и дидактические единицы предметной области (преподаваемого предмета)	<p>3.1 иметь представление об общих проблемах и задачах теоретической информатики;</p> <p>У.1 применять алгоритмы для кодирования информации;</p> <p>В.1 методами кодирования информации;</p> <p>3.2 сущность теории автоматов;</p> <p>У.2 строить схемы автоматов;</p> <p>В.2 способами построения автоматов;</p> <p>3.3 принципы динамического программирования</p> <p>У.3 осуществлять поиск решения задачи с помощью динамического программирования;</p> <p>В.3 методами поиска подстроки.</p>
2	ПК.1.2 умеет осуществлять отбор учебного содержания для его реализации в различных формах обучения в соответствии с требованиями ФГОС ВО	<p>3.4 теоретическую информатику, фундаментальную и прикладную математику для обеспечения качества учебно-воспитательного процесса;</p> <p>У.4 структурировать информацию для формирования необходимых образовательных результатов;</p> <p>В.4 методами анализа для отбора содержания преподаваемых дисциплин;</p> <p>3.5 содержание преподаваемого предмета;</p> <p>У.5 выбирать и структурировать учебный материал;</p> <p>В.5 методами проведения различных видов занятий.</p>

Содержание разделов дисциплины

1. ОСНОВЫ ТЕОРИИ ИНФОРМАЦИИ

Основы теории информации. Исходные понятия информации. Понятие информации в теории Шеннона. Измерение информации. Информационная энтропия. Энтропия сложных систем. Канал связи как сложная система. Объемный подход.

2. СЖАТИЕ ДАННЫХ. ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЯ

Побуквенное кодирование. Сжатие информации. Алгоритмы Хаффмана, Шеннона-Фано, сжатие со словарем (LZW). Критерий однозначности декодирования. Неравенство МакМиллана для разделимых кодов. Оптимальные коды. Методы построения оптимальных кодов. Помехоустойчивое кодирование. Понятие о помехоустойчивом кодировании, самокорректирующие коды Хэмминга.

3. ОСНОВЫ ТЕОРИИ АВТОМАТОВ

Элементы теории автоматов. Представления о конечных автоматах. Абстрактный синтез автоматов. Автоматные функции. Состояния автомата. Эквивалентность и минимизация автоматов. Автоматные языки и распознавание. Проблема распознавания. Общая характеристика задач распознавания и их типы. Математическая теория распознавания образов. Постановка задачи распознавания. Алгебраический подход к задаче распознавания. Геометрические процедуры распознавания. Алгоритмы распознавания, основанные на вычислении оценок. Структурные методы распознавания. Типы задач распознавания изображений. Распознавание и обработка изображений.

4. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ. АЛГОРИТМЫ НА СТРОКАХ

Алгоритмический метод решения задачи оптимизации. Разбиения задач на подзадачи. Концепция динамического программирования. Одномерное динамическое программирование. Двумерное динамическое

программирование. Общая задача динамического программирования. Динамика «сверху вниз» и «снизу вверх». Мемоизация. Методы решения рекурсивных задач и использование кэширования для оптимизации. Алгоритмы на строках: поиск подстроки, вычисление расстояния Левенштейна, нахождение наибольшей общей подпоследовательности. Примеры реальных задач, решаемых динамическим программированием на строках.

Темы и планы лабораторных занятий

Модуль 1. Основы теории информации

1. Вычисление статистических характеристик текстовой информации (2 часа):

- определение количества информации;
- построение таблицы частот;
- разработка программы формирования псевдослучайных чисел и оценка их качества статистическими тестами.

2. Разработка приложения для подсчета количества информации разными методами (2 часа):

- вычисление количества информации методом Хартли;
- вычисление количества информации методом Шеннона;
- вычисление количества прагматической информации.

Модуль 2. Сжатие информации. Помехоустойчивое кодирование

3. Кодирование информации методами Шеннона–Фано и Хаффмана (2 часа):

- метод Шеннона–Фано;
- кодирование информации;
- метод Хаффмана.

4. Сжатие данных по методу Лемпеля–Зива (2 часа):

- LZ-метод;
- алгоритм LZ78.

5. Помехоустойчивое кодирование (2 часа):

- обнаружение одиночной ошибки;
- разработка и защита кейса (помехоустойчивые коды).

Модуль 3. Основы теории автоматов

6. Комбинационные схемы (2 часа):

- построение схем из логических элементов;
- упрощение логических уравнений;

- построение субтрактора.

7. Абстрактный синтез автоматов (2 часа):

- запись регулярных событий;
- применение методов абстрактного синтеза;
- работа с программой «Регулярные выражения».

8. Распознавание образов (2 часа):

- методы распознавания образов;
- работа с программой Prog.

Модуль 4. Динамическое программирование. Алгоритмы на строках

9. Поиск чисел Фибоначчи с помощью динамического программирования (2 часа):

- получение последовательность чисел Фибоначчи в Excel;
- разработка программы с использованием динамического программирования в системе программирования.

10. Алгоритмы поиска подстроки (2 часа):

- определение количества символов;
- замена символов;
- вставка символа в строку;
- анализ символа на принадлежность к группе.

Модуль 1. Теория информации

Практическое занятие № 1.1

Вычисление статистических характеристик текстовой информации

Теоретические сведения

Важными характеристиками текста являются повторяемость букв, пар букв (биграмм) и вообще m-ок (m-грамм), сочетаемость букв друг с другом, чередование гласных и согласных, и некоторые другие. Эти характеристики являются достаточно устойчивыми.

Для русского языка частоты знаков алфавита, в котором отождествлены Е с Ё, Ъ с Ъ, а также имеется знак пробела между словами, приведены в таблице 2.

Таблица 2 – Частоты знаков алфавита

Символ / Вероятность	Символ / Вероятность	Символ / Вероятность	Символ / Вероятность
пробел 0.175	О 0.090	Е, Ё 0.072	А 0.062
И 0.062	Т 0.053	Н 0.053	С 0.045
Р 0.040	В 0.038	Л 0.035	К 0.028
М 0.026	Д 0.025	П 0.023	У 0.021
Я 0.018	Ы 0.016	З 0.016	Ь, Ъ 0.014
Б 0.014	Г 0.013	Ч 0.012	Й 0.010
Х 0.009	Ж 0.007	Ю 0.006	Ш 0.006
Ц 0.004	Щ 0.003	Э 0.003	Ф 0.002

Некоторая разница значений частот в различных источниках объясняется тем, что частоты существенно зависят не только от длины текста, но и от его характера.

Если бы сообщения передавались с помощью равновероятных букв алфавита и между собой статистически независимых, то энтропия таких сообщений была бы максимальной. На самом деле реальные сообщения строятся из не равновероятных букв алфавита с наличием статистических связей между буквами. Поэтому энтропия реальных сообщений – H_p ,

оказывается много меньше оптимальных сообщений – H_o . Допустим, нужно передать сообщение, содержащее количество информации, равное I . Источнику, обладающему энтропией на букву, равной H_p , придется затратить некоторое число n_p , то есть $I = n_p H_p$.

Если энтропия источника была бы H_0 , то пришлось бы затратить меньше букв на передачу этого же количества информации $I = n_0 H_0$, т.е.

$$n_o = \frac{1}{H_o} < n_p . \quad (1)$$

Таким образом, часть букв $n_p - n_o$ является как бы лишними, избыточными. Мера удлинения реальных сообщений по сравнению с оптимально закодированными и представляет собой избыточность D .

$$D = 1 - \frac{H_p}{H_o} = 1 - \frac{n_o}{n_p} = \frac{n_p - n_o}{n_o} . \quad (2)$$

Но наличие избыточности нельзя рассматривать как признак несовершенства источника сообщений. Наличие избыточности (2) способствует повышению помехоустойчивости сообщений. Высокая избыточность естественных языков обеспечивает надежное общение между людьми.

Задания

Задание 1. Определить количество информации (по Хартли), содержащееся в заданном сообщении, при условии, что значениями являются буквы кириллицы.

«Информация в общем виде является свойством материальных объектов, существует вечно, никогда не возникала и никогда не исчезает».

Задание 2. Построить таблицу распределения частот символов, характерных для заданного сообщения (см. Приложение 5). Вычислить количество информации, приходящейся на один символ текста. Использовать формулу Шеннона. Сравнить с имеющимися данными в таблице 2.

Задание 3. На основании полученных данных определить среднее и полное количество информации, содержащееся в заданном сообщении.

Задания для самостоятельной работы

Оценить избыточность сообщения из задания 3.

	A	B	C	D	E	F	G	H	I
1	Буква	кол-во	вер1	log вер1	c*d	вер2	log вер2	f*g	разница вер-стей
2	a	208	0,059	-4,093	-0,240	0,081	-3,626	-0,294	-0,022
3	b	49	0,014	-6,179	-0,085	0,014	-6,158	-0,086	0,000
4	c	102	0,029	-5,121	-0,147	0,027	-5,211	-0,141	0,002
5	d	118	0,033	-4,911	-0,163	0,038	-4,718	-0,179	-0,005
6	e	324	0,091	-3,454	-0,315	0,13	-2,943	-0,383	-0,039
7	f	80	0,023	-5,472	-0,123	0,029	-5,108	-0,148	-0,006
8	g	57	0,016	-5,961	-0,096	0,02	-5,644	-0,113	-0,004
9	h	148	0,042	-4,584	-0,191	0,052	-4,265	-0,222	-0,010
10	i	256	0,072	-3,794	-0,274	0,063	-3,989	-0,251	0,009
11	j	2	0,001	-10,794	-0,006	0,001	-9,587	-0,012	-0,001
12	k	7	0,002	-8,986	-0,018	0,004	-7,966	-0,032	-0,002
13	l	100	0,028	-5,150	-0,145	0,034	-4,878	-0,166	-0,006
14	m	70	0,020	-5,664	-0,112	0,025	-5,322	-0,133	-0,005
15	n	283	0,080	-3,649	-0,291	0,071	-3,816	-0,271	0,009
16	o	270	0,076	-3,717	-0,283	0,079	-3,662	-0,289	-0,003
17	p	51	0,014	-6,121	-0,088	0,019	-5,718	-0,109	-0,005
18	q	7	0,002	-8,986	-0,018	0,001	-9,828	-0,011	0,001
19	r	186	0,052	-4,254	-0,223	0,068	-3,878	-0,264	-0,016
20	s	235	0,066	-3,917	-0,259	0,061	-4,035	-0,246	0,005
21	t	237	0,067	-3,905	-0,261	0,105	-3,252	-0,341	-0,038
22	u	69	0,019	-5,685	-0,110	0,024	-5,381	-0,129	-0,005
23	v	27	0,008	-7,039	-0,054	0,009	-6,796	-0,061	-0,001
24	w	33	0,009	-6,749	-0,063	0,015	-6,059	-0,091	-0,006
25	x	5	0,001	-9,472	-0,013	0,002	-9,381	-0,014	0,000
26	y	59	0,017	-5,911	-0,098	0,019	-5,718	-0,109	-0,002
27	z	1	0,000	-11,794	-0,003	7E-04	-10,480	-0,007	0,000
28	Пробел	566	0,159	-2,649	-0,422	0,18	-2,474	-0,445	-0,021
29		3550		4,101				4,547	
30									
31							Разница	0,006	

Рисунок 1 – Пример оформления таблицы

Практическое занятие № 1.2 Формирование псевдослучайных чисел и оценка их качества статистическими тестами

Теоретические сведения

1. Генератор случайных чисел (ГСЧ) должен выдавать близкие к следующим значениям статистических параметров, характерных для равномерного случайного закона:

$$m_r = \frac{\sum_{i=1}^n r_i}{n} \approx 0.5 \quad \text{– математическое ожидание}$$

$$D_r = \frac{\sum_{i=1}^n (r_i - m_r)^2}{n} \approx \quad \text{– дисперсия}$$

$$\sigma_r = \sqrt{D_r} \approx 0.2887 \quad \text{– среднеквадратичное отклонение}$$

2. Частотный тест.

Частотный тест позволяет выяснить, сколько чисел попало в интервал $(m_r - \sigma_r; m_r + \sigma_r)$, то есть $(0.5 - 0.2887; 0.5 + 0.2887)$ или, в конечном итоге, $(0.2113; 0.7887)$. Так как $0.7887 - 0.2113 = 0.5774$, заключаем, что в хорошем ГСЧ в этот интервал должно попадать около 57.7% из всех выпавших случайных чисел (см. рис. 2).

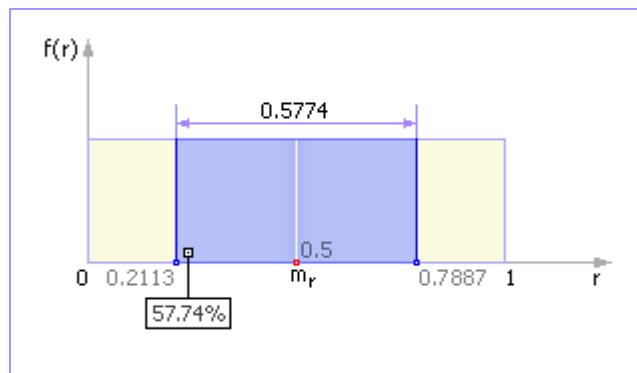


Рис. 2 – Частотная диаграмма идеального ГСЧ

в случае проверки его на частотный тест

Также необходимо учитывать, что количество чисел, попавших в интервал $(0; 0.5)$, должно быть примерно равно количеству чисел, попавших в интервал $(0.5; 1)$.

Таблица 3 – Пример оформления

№	случ1	случ2	случ3	$(r-m)^2$	случ4	$(r-m)^2$
1	0,497		1	15,707	0,014	0,231
2	0,005	0	5,169	0,139	0,432	0,009
3	0,085	0	10,097	0,085	0,543	0
4	0,525	0	3,89	0,022	0,865	0,116
5	0,625	0	19,255	0,061	0,123	0,161
6	0,067	1	3,937	0,096	0,142	0,146
7	0,178	1	1,106	0,04	0,655	0,017
8	0,07	1	6,722	0,095	0,543	0
9	0,402	1	16,681	0,001	0,695	0,029
10	0,28	0	4,518	0,01	0,958	0,188
11	0,528	1	17,048	0,023	0,764	0,058
12	0,282	1	19,336	0,009	0,362	0,026
13	0,8	1	13,2	0,179	0,764	0,058
14	0,14	0	9,414	0,056	0,154	0,137
15	0,842	0	19,154	0,216	0,874	0,122
16	0,586	1	3,542	0,043	0,433	0,008
17	0,482	1	16,36	0,011	0,436	0,008
18	0,207	1	14,188	0,029	0,976	0,204
19	0,833	1	19,727	0,207	0,217	0,094
20	0,118	0	18,788	0,068	0,318	0,043
Сумма	7,552	12	237,838	1,403	10,487	1,511
m				0,378		0,524
Дисперсия				0,07		0,076
Сигма				0,265		0,275
Разница						0,01

Задание 1. Получить случайные последовательности чисел в Excel (см. Таблица 3).

Используйте функции СЛЧИС и СЛУЧМЕЖДУ ((например, $\text{СЛЧИС}()*(100 - 1)+1$, СЛУЧМЕЖДУ(1;100)).

Пример 1. Выполнение проверки по критерию «хи-квадрат».

Критерий «хи-квадрат» (χ^2 -критерий) – это один из самых известных статистических критериев; он является основным методом, используемым в сочетании с другими критериями. Критерий «хи-квадрат» был предложен в 1900 г. Карлом Пирсоном. Его замечательная работа рассматривается как фундамент современной математической статистики.

Для нашего случая проверка по критерию «хи-квадрат» позволит узнать, насколько созданный нами *реальный ГСЧ* близок к *эталону ГСЧ*, то есть удовлетворяет ли он требованию равномерного распределения или нет.

Частотная диаграмма *эталонного ГСЧ* представлена на рис. 3. Так как закон распределения *эталонного ГСЧ* равномерный, то (теоретическая) вероятность p_i попадания чисел в i -ый интервал (всего этих интервалов k) равна $p_i = 1/k$. И, таким образом, в каждый из k интервалов попадет *ровно* по $p_i \cdot N$ чисел (N – общее количество сгенерированных чисел).

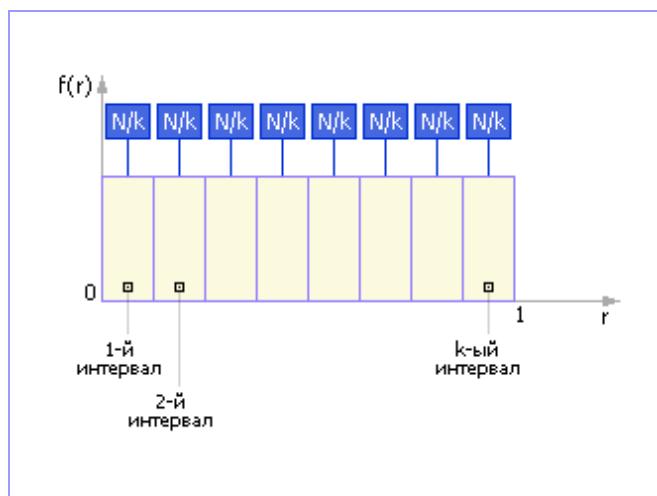


Рис. 3 – Частотная диаграмма эталонного ГСЧ

Пример 2. Выполнение проверки на частоту появления цифры в последовательности.

Случайное число 0.2463389991 состоит из цифр 2463389991, а число 0.5467766618 состоит из цифр 5467766618. Соединяя последовательности цифр, имеем: 24633899915467766618.

Теоретическая вероятность p_i выпадения i -ой цифры (от 0 до 9) равна 0.1. Далее следует вычислить частоту появления каждой цифры в выпавшей экспериментальной последовательности. Например, цифра 1 выпала 2 раза из 20, а цифра 6 выпала 5 раз из 20. После этого считают оценку и принимают решение по критерию «хи-квадрат».

Пример 3. Выполнение проверки появления серий из одинаковых цифр.

Обозначим через n_L число серий одинаковых подряд цифр длины L . Проверять надо все L от 1 до m , где m – это заданное пользователем число: максимально встречающееся число одинаковых цифр в серии.

В примере «24633899915467766618» обнаружены 2 серии длиной в 2 (33 и 77), то есть $n_2 = 2$ и 2 серии длиной в 3 (999 и 666), то есть $n_3 = 2$.

Вероятность появления серии длиной в L равна: $p_L = 9 \cdot 10^{-L}$ (теоретическая). То есть вероятность появления серии длиной в один символ равна: $p_1 = 0.9$ (теоретическая). Вероятность появления серии длиной в два символа равна: $p_2 = 0.09$ (теоретическая). Вероятность появления серии длиной в три символа равна: $p_3 = 0.009$ (теоретическая).

Например, вероятность появления серии длиной в один символ равна $p_L = 0.9$, так как всего может встретиться один символ из 10, а всего символов 9 (ноль не считается). А вероятность того, что подряд встретится два одинаковых символа «xx» равна $0.1 \cdot 0.1 \cdot 9$, то есть вероятность 0.1 того, что в первой позиции появится символ «x», умножается на вероятность 0.1 того, что во второй позиции появится такой же символ «x» и умножается на количество таких комбинаций 9.

Частотность появления серий подсчитывается по ранее разобранной формуле «хи-квадрат» с использованием значений p_L .

Задание для самостоятельной работы

Выполнить проверку на статистическую независимость двух случайных чисел, полученных с помощью генератора случайных чисел (см. пример 2 и пример 3).

Практическое занятие № 2

Разработка приложения для подсчета количества информации разными методами

Р. Хартли предложил определять информационную ёмкость системы, передающей или накапливающей информацию, и количества информации в сообщении. С именем Хартли связано структурное направление измерения количества информации, в основу которого положено понятие о дискретном строении массивов информации и их разнообразии без учёта условий использования источников информации. При структурном подходе принимается во внимание возможность получения от источника информации некоторого дискретного множества сообщений, которое в свою очередь обусловлено количеством информационных элементов и связей между ними.

Если количество возможных сообщений в этом множестве равно N , то, по Хартли, информация I_1 , приходящаяся на одно сообщение, определяется логарифмом общего числа возможных сообщений

$$I_1 = \log N.$$

В основу статистического подхода к определению количества информации положена необходимость учитывать при определении количества информации не только количество разнообразных сообщений, но и вероятность получения тех или иных сообщений, то есть вероятностные характеристики источника. Такой подход был предложен К. Шенноном в 1948 году и получил широкое распространение при определении среднего количества информации, которое содержится в сообщениях от источников самой разной природы. Количество информации, приходящейся на один символ сообщения равно:

$$I_1 = - \sum_{i=1}^n p_i \log p_i$$

Задание 1. Разработать приложение для подсчета количества информации методами Р. Хартли и К. Шеннона (см. рис. 1).

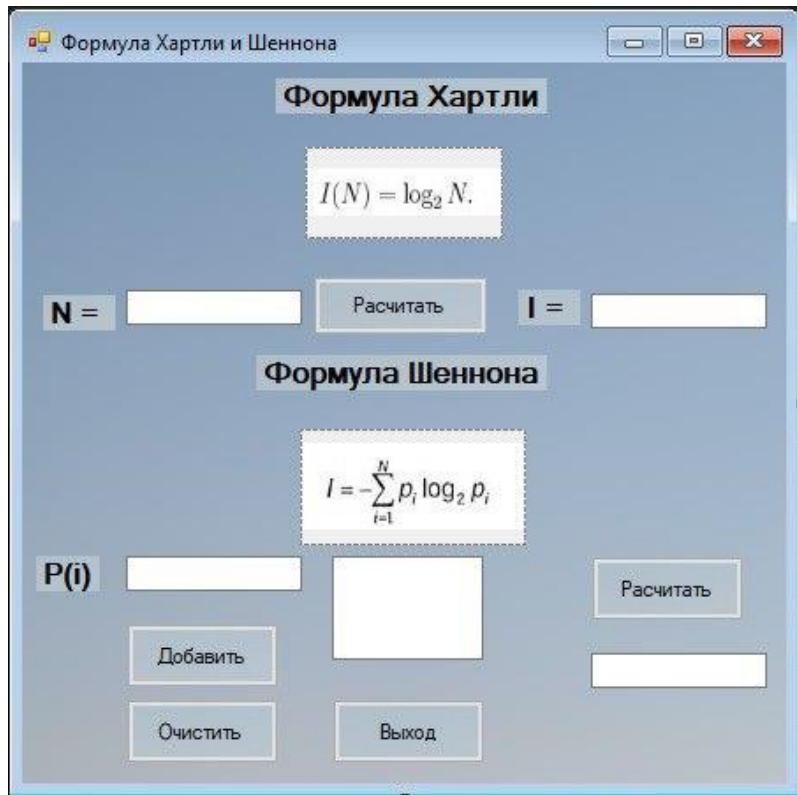


Рисунок 1 – Пример расчета количества информации

Процедуры для вычисления количества информации приведены ниже.

```
Procedure TForm1.Button1Click(Sender: TObject);
begin
if strtofloat(Edit1.Text) < 1 then
  ShowMessage ('Количество не может быть меньше одного')
else
  Edit2.Text:=floattostr(Ln(strtofloat(Edit1.Text))/Ln(2));
end;

Procedure TForm1.Button2Click(Sender: TObject);
var x: real;
begin
if ((strtofloat(Edit3.Text) > 1) or (strtofloat(Edit3.Text)<=0)) then
  ShowMessage ('Вероятность не может быть больше единицы или меньше либо
равна нулю!')
else
  begin
    x:=Ln(strtofloat(Edit3.Text))/Ln(2);
    Label9.Caption:=inttostr(strtoint(Label9.Caption)+1);
  end;
```

```
Memo1.Lines.Text:=Memo1.Lines.Text+Edit3.Text+':';
Edit4.Text:=floattostr(strtofloat(Edit4.Text)+(strtofloat(Edit3.Text)*(-x)));
end;
end;
```

Задание 2 (самостоятельно). Рассмотреть еще один метод вычисления количества информации.

Вопросы к модулю 1

1. Как связаны понятия неопределенности и вероятности?
2. Как влияет выбор вариантов решения на неопределенность?
3. Чему равно количество информации по Р. Хартли?
4. Как связана вероятность с информативностью?
5. Дать определение количества информации.
6. Как вычисляется количество информации по К. Шенону?
7. В каком случае формула Шеннона переходит в формулу Хартли?
8. К какому подходу относятся методы Р. Хартли и К. Шеннона?
9. Суть концепции разнообразия Эшби.
10. Как определяется количество информации по А.Н. Колмогорову?
11. Кто предложил связывать понятие «информации» с разнообразием?
12. Зависит ли информация от нашего сознания (по В.М. Глушкову)?
13. Чем занимается наука семиотика?
14. С каких позиций рассматриваются знаковые системы?
15. В чем состоит основная идея семантической концепции информации?
16. В чем заключается подход к измерению количества семантической информации Р. Карнапа и И. Бар-Хиллела?
17. Кому принадлежит идея учета «запаса знаний»?
18. В чем суть pragматического подхода к измерению количества информации?

Модуль 2. Сжатие информации. Помехоустойчивое кодирование

Практическое занятие № 3

Кодирование информации. Методы Шеннона-Фано и Хаффмена

Теоретические сведения

Метод Шеннона–Фано

Код строится следующим образом. После подсчета всех символов, встречающихся в файле, они вписываются в таблицу в порядке убывания частот их появления. На первом этапе она разбивается на две части так, чтобы суммы частот появления символов в них были приблизительно равны. Так, первое подмножество: $\alpha_2 = 0,5$; второе: $\alpha_4 = 0,25$; $\alpha_1 = 0,125$; $\alpha_3 = 0,125$ (в сумме 0,5). Для верхнего подмножества каждому символу приписывается 0, для нижнего – 1. Затем каждое из этих двух подмножеств делится пополам, и процедура повторяется для каждого из подмножеств. В конечном итоге в каждом подмножестве останется по одному символу.

Таблица 4 – Пример кодирования методом Шеннона-Фано

Буква	Вероятность	Цифры кода			Код
		1ая	2ая	3я	
α_2	0,5	0			0
α_4	0,25	1	0		10
α_1	0,125	1	1	0	110
α_3	0,125	1	1	1	111

Закодируем строку $\alpha_2\alpha_3\alpha_1\alpha_4$ полученным кодом – 011111010. Ее можно однозначно раскодировать.

Метод Хаффмена

Код строится следующим образом. Например, имеется строка символов «От топота копыт». Всего 15 букв. После подсчета всех символов, встречающихся в файле, они вписываются в таблицу в порядке убывания частот их появления. Каждый следующий столбец получаем сложением двух нижних чисел, и эту сумму помещаем на

соответствующее место, или ниже, если такое число есть в последовательности.

Таблица 5 – Пример кодирования методом Хаффмена

Символ	6	5	4	3	2	1	1	2	3	4	5	6
т	4	4	4	4	7	<u>8</u>	0	<u>1</u>	00	00	00	00
о	4	4	4	4	4	7	1	00	01	01	01	01
п	2	2	<u>3</u>	<u>4</u>	4			01	<u>10</u>	<u>11</u>	100	100
« »	2	2	2	3					11	100	101	101
к	1	<u>2</u>	2						101	<u>110</u>	111	
ы	1	1								111	1100	
а	1										1101	

В итоге на вершине должно остаться два числа. Их сумма должна быть равна общему количеству символов в строке. Верхнее число кодируется 0, нижнее кодируется 1. Подчеркнутое число выделяем. В каждом следующем столбце выделенный код записываем внизу дважды, к первому добавляем 0, ко второму – 1, остальные коды переписываем, сдвигая на одну позицию вверх.

В результате каждый из символов получает свой код (т – 00, о – 01, п – 100 и т. д.). При этом, чем чаще встречается символ в строке, тем код короче.

Примеры

1. Закодировать фразу «Тише, мыши, тише, кот на крыше», используя метод Шеннона–Фано.

Незакодированная фраза – 30×8 бит = 240 бит. Закодированная фраза – 106 бит.

2. Закодировать фразу «Тише, мыши, тише, кот на крыше», используя метод Хаффмена.

Задание 2. Закодировать фразу «во дворе трава на траве дрова». Подсчитать количество бит, которое занимает закодированная фраза. Использовать оба метода.

Задание для самостоятельной работы

Закодировать фразу «Сшит колпак да не по-колпаковски, надо колпак переколпаковать». Использовать метод Шеннона–Фано.

Таблица 6 – Пример кодирования методом Шеннона–Фано

Символ	Кол-во	1-я цифра	2-я цифра	3-я цифра	4-я цифра	5-я цифра	Код	Кол-во бит
пробел	5	0	0	0			000	15
ш	4	0	0	1			001	12
е	3	0	1	0			010	9
,	3	0	1	1			011	9
и	3	1	0	0			100	9
т	3	1	0	1	0		1010	12
ы	2	1	0	1	1		1011	8
к	2	1	1	1	0		1110	8
н	1	1	1	1	1		1111	4
о	1	1	1	0	0	0	11000	5
а	1	1	1	0	0	1	11001	5
м	1	1	1	0	1	0	11010	5
р	1	1	1	0	1	1	11011	5
								106

**Индивидуальные задания № 1–2
Кодирование методами Шеннона–Фано и Хаффмена**

Закодировать фразу методами Шеннона–Фано и Хаффмена в соответствии с вариантом (см. Таблица 8).

Таблица 7 – Пример кодирования методом Хаффмена

Таблица 8 – Индивидуальное задание по теме кодирование

1.	Раз, два, три, четыре – Меня грамоте учили: Не считать, не писать, Только по полу скакать.	13.	Тучи, тучи, тучи, тучи, Скачет конь большой, могучий. Через тучи скачет он, Кто не верит – выйди вон!
2.	Ахи, ахи, ахи, ох, Маша сеяла горох, Уродился он густой, Мы помчимся, ты постой!	14.	Ежик, ежик, чудачок, Сшил колючий пиджачок. Встал в кружок и ну считать – Нам водилку выбирать!
3.	Дора, Дора, помидора, Мы в саду поймали вора, Стали думать и гадать, Как нам вора наказать.	15.	Подогрела чайка чайник, Пригласила восемь чаек: - Приходите все на чай! - Сколько чаек? Отвечай!
4.	Наша Маша рано встала, Кукол всех пересчитала: Две матрешки на оконке, Две танюшки на подушке.	16.	Удивительная киска – Отказалась есть сосиску. Есть не стал он даже мяса, Мой котёнок... из пласти массы.
5.	Уронили мишку на пол, Оторвали мишке лапу, Все равно его не брошу, Потому что он хороший.	17.	Цветом мой зайчишка – белый, А ещё, он очень смелый! Не боится он лисицы, Льва он тоже не боится.
6.	Аты-баты, шли солдаты, Аты-баты, на базар. Аты-баты, что купили? Аты-баты, самовар.	18.	Ёжик должен быть колючим. Ну, а этот ёжик мой Не уколет моих ручек, Этот ёжик - шерстяной.
7.	Катилася торба С высокого горба; В этой торбе: хлеб, пшеница, С кем хочешь, с тем поделися.	19.	Должен ёжик быть фырчающим, Топотучим должен быть. Может ёжик настоящий Ночью всех нас разбудить.
8.	Кони-огони сидели на балконе, Чай пили, по-турецки говорили: Чоби, ачоби, Челябичи и чоби.	20.	Шла кукушка мимо сети, А за нею малы дети. Все кричали: «Кукумак!» Убирай один кулак!
9.	Кто-то шёл, шёл, шёл И корзиночку нашёл. В этой маленькой корзинке есть помада и духи, Ленты, кружево, ботинки...	21.	Эна, бена, раба, Квинтер, финтер, жаба. Эна, бена, рес, Квинтер, финтер, жес!
10.	Шла кукушка мимо сада, Поклевала всю рассаду. И кричала: ку-ку, мак - Отжимай один кулак!	22.	Раз, два, три, четыре, пять, Негде зайчику скакать, Всюду ходит волк, волк, Он зубами – щёлк, щёлк!
11.	Зуба зуба, зуба зуба, Зуба дони дони мэ, А шарли буба раз два три, А ми раз два три замри.	23.	Эне, бене, лики, паки, Цуль, буль-буль, Калики-цваки, Эус-беус, клик-мадеус, бокс
12.	Эни бэни рики паки Турбаурбасентибряки. Может выйдет – может нет, В общем – полный Интернет.	24.	Раз, два – упала гора; три, четыре – прицепило; пять, шесть – бьют шерсть; семь, восемь – сено косим.

Практическое занятие № 4

Сжатие данных по методу Лемпеля–Зива

Теоретические сведения

Лемель и Зив используют следующую идею: если в тексте сообщения появляется последовательность из двух ранее уже встречавшихся символов, то эта последовательность объявляется новым символом, для нее назначается код, который при определенных условиях может быть значительно короче исходной последовательности. В дальнейшем в сжатом сообщении вместо исходной последовательности записывается назначенный код. При декодировании повторяются аналогичные действия, и потому становятся известными последовательности символов для каждого кода. Одна из алгоритмических реализаций этой идеи включает следующие операции. Первоначально каждому символу алфавита присваивается определенный код (коды – порядковые номера, начиная с 0). При *кодировании*:

1. Выбирается первый символ сообщения и заменяется на его код.
2. Выбираются следующие два символа и заменяются своими кодами.

Одновременно этой комбинации двух символов присваивается свой код. Обычно это номер, равный числу уже использованных кодов. Так, если алфавит включает 8 символов, имеющих коды от 000 до 111, то первая двухсимвольная комбинация получит код 1000, следующая – код 1001 и т.д.

3. Выбираются из исходного текста очередные 2, 3, ..., N символов до тех пор, пока не образуется еще не встречавшаяся комбинация. Тогда этой комбинации присваивается очередной код, и поскольку совокупность A из первых N–1 символов уже встречалась, то она имеет свой код, который и записывается вместо этих N–1 символов. Каждый акт введения нового кода назовем шагом кодирования.

4. Процесс продолжается до исчерпания исходного текста. При *декодировании* код первого символа, а затем второго и третьего символов, заменяются на символы алфавита. При этом становится известным код

комбинации второго и третьего символов. В следующей позиции могут быть только коды уже известных символов и их комбинаций. Процесс декодирования продолжается до исчерпания сжатого текста.

Сколько двоичных разрядов нужно выделять для кодирования? Ответ может быть следующим: число разрядов R на каждом шаге кодирования равно числу разрядов в наиболее длинном из использованных кодов (т.е. числу разрядов в последнем использованном порядковом номере). Поэтому если последний использованный код (порядковый номер) равен $(13)_{10} = (1101)_2$, то коды А всех комбинаций должны быть четырехразрядными при кодировании вплоть до появления номера 16, после чего все коды символов начинают рассматриваться как пятиразрядные ($R = 5$).

Пример. Пусть исходный текст представляет собой двоичный код (первая строка таблицы 9), т.е. символами алфавита являются 0 и 1. Коды этих символов соответственно также 0 и 1. Образующийся по методу Лемпеля–Зива код (LZ-код) показан во второй строке таблицы 6. В третьей строке отмечены шаги кодирования, после которых происходит переход на представление кодов А увеличенным числом разрядов R . Так, на первом шаге вводится код 10 для комбинации 00, и поэтому на следующих двух шагах $R = 2$, после третьего шага $R = 3$, после седьмого шага $R = 4$, т.е. в общем случае $R = K$ после шага $2^{K-1} - 1$.

В приведенном примере LZ-код оказался длиннее исходного кода, так как обычно короткие тексты не дают эффекта сжатия. Эффект сжатия проявляется в достаточно длинных текстах и особенно заметен в графических файлах.

Таблица 9

Исходный текст	0.00. 000. 01. 11. 111. 1111. 110. 0000.00000.1101. 1110.									
LZ-код	0.00. 100. 001.011.1011.1101. 1010. 0000.10010. 10001.1110.									
R	2 3 4									
Вводимые коды	– 10 11 100 101 110 111 1000 1001 1010 1011 1100									

В другой известной реализации LZ-метода любая ранее встречавшаяся последовательность в сжатом тексте представляет собой совокупность следующих данных:

- номер первого символа в ранее встречавшейся последовательности;
- число символов в последовательности;
- следующий символ в текущей позиции кодируемого текста.

Алгоритм LZ78

Алгоритм LZ78 вносит все встреченные им последовательности в словарь.

Всякий раз, когда среди данных, которые надо сжать, встречается последовательность, программа обращается к словарю:

- если последовательность находится в словаре, то в выходной файл заносится код для этой записи;
- если последовательность представляет собой расширенный вариант последовательности из словаря, то она добавляется в таблицу.

Таблица 10 – Пример кодирования методом LZ78

Исходный символ	Словарь	Код
a	–	–
ab	257	a
bc	258	b
ca	259	c
ab (257)		
abc	260	257
ca (259)		
cab	261	259
bc (258)		258

Пример. Рассмотреть, как программа LZ78 читает каждый байт abcabcabc (см. Таблица 10).

Получен код: abc 257 259 258.

При раскодировании получаем: abc ab ca bc.

Задания

Задание 1. Сжать следующие данные, используя двухсимвольный алфавит, имеющий коды от 0 до 1, 1 01 10 101 1010 011. Выполнить обратную операцию.

Задание 2. Сжать последовательность 1) abcdabcd, 2) abaabcaabcdabcd, используя алгоритм LZ78. Привести пример распаковки.

Самостоятельно:

- использовать 4-х символьный алфавит (00, 01, 10, 11) для сжатия данных;
- сжать последовательность «по секрету по секрету всему свету».

Индивидуальное задание №3 Сжатие данных по методу Лемпеля–Зива

1. Используя двухсимвольный алфавит (0,1) закодировать следующую фразу:

Вариант 1

0100101010010000101

Вариант 3

1110100110111001101

Вариант 5

0100101010010000101

Вариант 7

0010100110010000001

Вариант 9

10101001101100111010

Вариант 11

0101101101100010001

Вариант 13

000101110110100111

Вариант 2

0001000010101001101

Вариант 4

0001000010101001101

Вариант 6

10110111100110001101

Вариант 8

01011011001010101011

Вариант 10

000100101100100010001

Вариант 12

110101011001100001001

Вариант 14

101000100101010001011

Вариант 15	Вариант 16
0100001000000100001	0100101010010000101
Вариант 17	Вариант 18
0100100010010000101	0001010010101001101
Вариант 19	Вариант 20
1110100110110001101	0001001010101001101

2. Закодировать следующую фразу, используя код LZ78:

Вариант 1	Вариант 2
кукуркукурекурекун	упупапекапекаупуп
Вариант 3	Вариант 4
лорлоралоранранлоран	пропронепронепрнепронас
Вариант 5	Вариант 6
какатанекатанекатата	менменаменаменатеп
Вариант 7	Вариант 8
долделдолдилдилдил	sarsalsarsanlasanl
Вариант 9	Вариант 10
kloklonkolonklonkl	tertrektekertektrek
Вариант 11	Вариант 12
bigbonebigborebigbo	commercommecommerce
Вариант 13	Вариант 14
webwerbwеберweberweb	porpoterpoterporter
Вариант 15	Вариант 16
mantopmentopomantomen	roporopoterropoterter
Вариант 17	Вариант 18
webwerbwеберweberweb	sionsinossionsinos
Вариант 19	Вариант 20
comconcomconacom	mantopmentopomantomen

Практическое занятие № 5.1

Помехоустойчивое кодирование

Теоретические сведения

1. Коды Хемминга. Обнаружение одиночной ошибки (1)

Наиболее известные из самоконтролирующихся и самокорректирующихся кодов – коды Хемминга. Построены они применительно к двоичной системе счисления.

Для построения самокорректирующегося кода достаточно иметь один контрольный разряд (код с проверкой на четность). Но при этом мы не получаем никаких указаний о том, в каком именно разряде произошла ошибка, и, следовательно, не имеем возможности исправить ее. Остаются незамеченными ошибки, возникшие в четном числе разрядов.

Коды Хемминга имеют большую относительную избыточность, чем коды с проверкой на четность, и предназначены либо для исправления одиночных ошибок (при $d = 3$), либо для исправления одиночных и обнаружения без исправления двойных ошибок ($d = 4$). В таком коде n -значное число имеет m информационных и k контрольных разрядов. Каждый из контрольных разрядов является знаком четности для определенной группы информационных знаков слова. При декодировании производится k групповых проверок на четность. В результате каждой проверки в соответствующий разряд регистра ошибки записывается 0, если проверка была успешной, или 1, если была обнаружена нечетность.

Группы для проверки образуются таким образом, что в регистре ошибки после окончания проверки получается K -разрядное двоичное число, показывающее номер позиции ошибочного двоичного разряда. Изменение этого разряда – исправление ошибки.

Первоначально эти коды предложены Хеммингом в таком виде, при котором контрольные знаки занимают особые позиции: позиция i -го знака имеет

номер 2^{i-1} . При этом каждый контрольный знак входит лишь в одну проверку на четность.

Рассмотрим код Хемминга, предназначенный для исправления одиночных ошибок, т.е. код с минимальным кодовым расстоянием $d = 3$.

Ошибка возможна и в одной из n позиций. Следовательно, число контрольных знаков, а значит, и число разрядов регистра ошибок должны удовлетворять условию

$$k \geq \log_2(n + 1). \quad (1)$$

Отсюда

$$m \leq n - \log_2(n + 1). \quad (2)$$

Значения m и k для некоторых коротких кодов, вычисленные по формулам (1) и (2) даны в табл. 11.

Таблица 11 – Значения m , n , k

n	3	4	5	6	7	8	9	10	11	12
m	1	1	2	3	4	4	5	6	7	8
k	2	3	3	3	3	4	4	4	4	4

Таблица 12 – Формирование контрольных групп

Номер проверки	Позиция контрольного знака	Проверяемые позиции
1	1	1, 3, 5, 7, 9, 11, 13, ...
2	2	2, 3, 6, 7, 10, 11, ...
3	4	4, 5, 6, 7, 12, 13, ...
4	8	8, 9, 10, 11, 12, 13, ...
5	16	16, 17, 18, 19, 20, 21

Чтобы число в регистре ошибок (РОШ) указывало номер позиции ошибочного разряда, группы для проверки выбираются по правилу:

I группа – все нечетные позиции, включая и позиции контрольного разряда, т.е. позиции, в первом младшем разряде которых стоит 1;

II группа – все позиции, номера которых в двоичном представлении имеют 1 во втором разряде справа (например, 2, 3, 6, 7, 10) и т. д.;

III гр – разряды, имеющие 1 в третьем разряде справа, и т. д.

Примечание: каждый контрольный знак входит только в одну проверяемую группу.

Таблица 13 – Семизначный код Хемминга

Десятичное число	Простой двоичный код				Код Хемминга						
					к			к		к	
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
2	0	0	1	0	0	0	1	1	0	0	1
3	0	0	1	1	0	0	1	1	1	1	0
4	0	1	0	0	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1	0	0	1	1
7	0	1	1	1	0	1	1	0	1	0	0
8	1	0	0	0	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0	1	1	0	0

Пример 1. Пусть $k = 5$ (см. Таблица 12). Перечислить проверяемые позиции.

Пример 2. Рассмотрим семизначный код Хемминга, служащий для изображения цифр от 0 до 9 (см. Таблица 13).

Пусть передан код цифры 6 в виде «0 1 1 0 0 1 1», а приняли в виде «0 1 0 0 0 1 1». Проверочные группы (см. Таблица 14):

I проверка: разряды 1, 3, 5, 7 – дает 1 в младший разряд РОШ.

II проверка: разряды 2, 3, 6, 7 – дает 0 во второй разряд РОШ.

III проверка: разряды 4, 5, 6, 7 – дает 1 в третий разряд РОШ.

Содержимое РОШ «101», значит ошибка в пятой позиции.

Примечание. В каждый из контрольных разрядов при построении кода Хемминга посыпается такое значение, чтобы общее число единиц в его

контрольной сумме было четным. Разряд ошибки (РОШ) записываем, начиная с младшего разряда.

Таблица 14 – Пример обнаружения одиночной ошибки

Отправлен		0	1	1	0	0	1	1		
Принят		0	1	0	0	0	1	1		
I проверка:		7	6	5	4	3	2	1		
1,3,5,7		0		0		0		1		1
II проверка:		7	6	5	4	3	2	1		
2,3,6,7		0	1			0	1			0
III проверка:		7	6	5	4	3	2	1		
4,5,6,7		0	1	0	0					1

Вывод. Рост кодового расстояния позволяет увеличить корректирующую способность кода. В то время как $d = 2$ у кода с проверкой на четность позволяет обнаруживать одиночную ошибку, код Хемминга с $d = 3$ исправляет ее.

2. Обнаружение одиночной ошибки (2)

В коде Хемминга вводится понятие кодового расстояния d (расстояния между двумя кодами), равного числу разрядов с неодинаковыми значениями. Возможности исправления ошибок связаны с минимальным кодовым расстоянием d_{\min} . Исправляются ошибки кратности $r = \text{ent} (d_{\min} - 1) / 2$ и обнаруживаются ошибки кратности $d_{\min} - 1$ (здесь ent означает «целая часть»). Так, при контроле на нечетность $d_{\min} = 2$ и обнаруживаются одиночные ошибки. В коде Хемминга $d_{\min} = 3$. Дополнительно к информационным разрядам вводится $L = \log_2 K$ избыточных контролирующих разрядов, где K – число информационных разрядов, L округляется до ближайшего большего целого значения. L -разрядный контролирующий код есть инвертированный результат поразрядного сложения (т.е. сложения по модулю 2) номеров тех информационных разрядов, значения которых равны 1.

Пример 1. Пусть имеем основной код 100110, т.е. $K = 6$. Следовательно, $L = 3$ и дополнительный код равен $010 \# 011 \# 110 = 111$, где $\#$ – символ операции поразрядного сложения. После инвертирования имеем 000. Теперь вместе с

основным кодом будет передан и дополнительный. На приемном конце вновь рассчитывается дополнительный код и сравнивается с переданным. Фиксируется код сравнения (поразрядная операция отрицания равнозначности), и если он отличен от нуля, то его значение есть номер ошибочно принятого разряда основного кода. Так, если принят код 100010, то рассчитанный в приемнике дополнительный код равен инверсии от $010 \# 110 = 100$, т.е. 011, что означает ошибку в 3-м разряде.

Пример 2. Основной код 1100000, дополнительный код 110 (результат инверсии кода $110 \# 111 = 001$). Пусть принятый код 1101000, его дополнительный код 010, код сравнения 100, т.е. ошибка в четвертом разряде.

Задания

Задание 1

1. Передан код цифры 5 в виде «0 1 0 1 1 0 1», а приняли в виде «0 1 0 1 0 0 1». Обнаружить позицию ошибки.
2. Передан код цифры 8 в виде «1 0 0 1 0 1 1», а приняли в виде «1 0 1 1 0 1». Обнаружить позицию ошибки.

Таблица 15 – Задание для самостоятельной работы

	Переданный код	Принятый код
1.	0101100	0111100
2.	0101101	0100101
3.	0101100	0101110
4.	0101101	0001101
5.	0101100	1101100
6.	0101101	0101001
7.	0101100	0101000
8.	0101101	0111101
9.	0101100	0101101
10.	0101101	1101101
11.	0101100	0100100
12.	0101101	0111101

Задание 2

1. Основной код 100110, принятый код 100100. Подсчитать дополнительный код. Обнаружить позицию ошибки.

2. Основной код 0101100, принятый код 0111100. Подсчитать дополнительный код. Обнаружить позицию ошибки.

Задание для самостоятельной работы

1. Передан код цифры (взять код цифры варианта – Таблица 13) в виде «...», приняли с ошибкой в позиции, равной количеству нулей в переданном коде. Обнаружить позицию ошибки.

2. Подсчитать дополнительный код. Обнаружить позицию ошибки (Таблица 15).

3. Познакомиться с программой Code (Папка «Учебник по помехоустойчивому кодированию» находится на портале университета).

Практическое занятие № 5.2

Помехоустойчивое кодирование (Разработка и защита кейса)

I. Разработка кейса.



Рис. 4 – Главное окно

1. На основе изученного теоретического материала подготовить кейс на тему.

2. Открыть шаблон кейса **кейс.pub** (рис.4).

3. Найти информацию, соответствующую теме кейса, заполнить раздел

Краткая аннотация (рис. 5):

- назначение рассматриваемого помехоустойчивого кода;
- алгоритм и примеры кодирования.

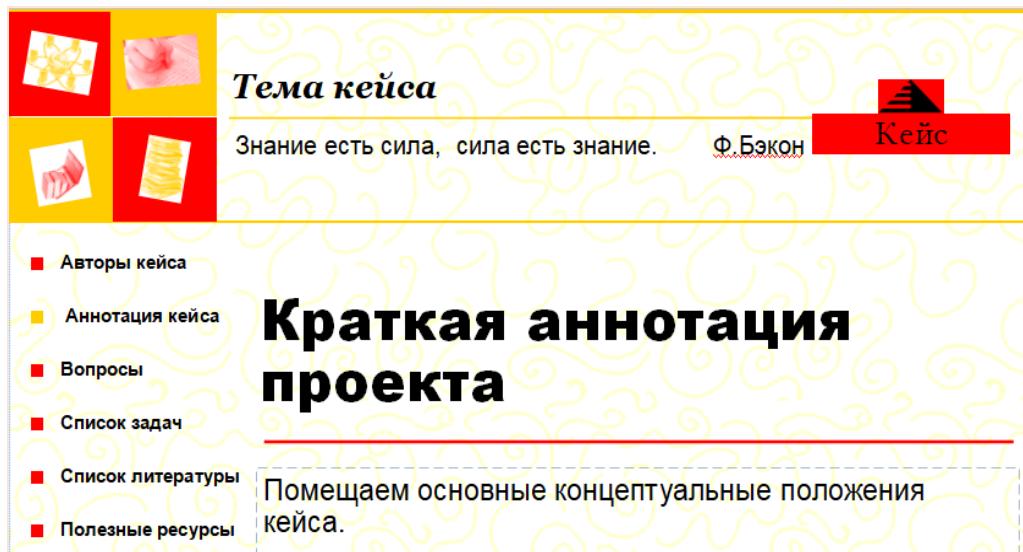


Рис. 5 – Краткая аннотация проекта

4. Сформулировать вопросы для осуждения данной темы (рис. 6).

Ответы на вопросы должны содержаться в краткой аннотации проекта.



Рис. 6 – Вопросы для проверки

5. Готовый кейс сохранить. Выбрать тип файла – Веб-страница в одном файле.

Коды Абрамсона
Знание есть сила, сила есть знание.

Примеры кодов Абрамсона

Пример 1: Код (7,4)

- * Количество битов данных: 4
- * Количество проверочных битов: 3
- * Общее количество битов: 7

Генерация кода:

1. Рассчитайте биты проверки на четность для каждого из 4 блоков по 3 бита данных.
2. Сохраните биты проверки на четность после каждого блока данных.

Рис. 7 – Примеры кодов Абрамсона

Самостоятельно:

Разработать кейс по теме из индивидуального задания 4.

Код Грэя
Знание есть сила, сила есть знание. Ф.Бэкон

Краткая аннотация проекта

Код Грэя – это система счисления, в которой два соседних значения различаются только в одном разряде. Наиболее часто на практике применяется рефлексивный двоичный код Грэя, хотя в общем случае существует бесконечное множество кодов Грэя для систем счисления с любым основанием. В большинстве случаев, под термином «код Грэя» понимают именно рефлексивный бинарный код Грэя.

Изначально предназначался для защиты от ложного срабатывания электромеханических переключателей.

Сегодня коды Грэя широко используются для упрощения выявления и исправления ошибок в системах связи, а также в формировании сигналов обратной связи в системах управления.

Рис. 8 – Пример кейса

II. Защита кейса.

Подготовить разработанный кейс к защите.

Индивидуальное задание №4

1. Разработать кейс, в котором описать предложенный метод кодирования, привести не менее двух примеров:

- 1) полиномиальный;
- 2) рекурентный;
- 3) БЧХ;
- 4) Грэя;
- 5) с пороговым декодированием;
- 6) Рида–Маллера;
- 7) Варшамова;
- 8) Цепной;
- 9) Сверточный;
- 10) Плоткина;
- 11) мажоритарный;
- 12) Файра;
- 13) Абрамсона;
- 14) Рида–Соломона;
- 15) с постоянным весом;
- 16) корреляционный;
- 17) итеративный;
- 18) с четным числом 1;
- 19) Миласа–Абрамсона;
- 20) по максимуму правдоподобия;
- 21) инверсный;
- 22) Голея;
- 23) компаундный;
- 24) Бергера.

2. Реализовать алгоритм (из задания 1) в среде программирования (на 0.2 – 0.5 балла).

Вопросы к модулю 2

1. Кем заложены теоретические основы сжатия информации?
2. Что лежит в основе алгоритма Лемпеля–Зива.
3. В чем заключается метод повторяющихся последовательностей?
4. В основе какого метода лежит идея замены часто встречающихся последовательностей символов в файле ссылками на образцы?
5. Какая система счисления должна использоваться в компьютере с целью минимизации количества элементов в устройствах хранения?
6. Какая схема кодирования называется разделимой?
7. Является ли префиксная схема кодирования разделимой?
8. Какие методы относятся к вероятностным методам?
9. Перечислить единицы измерения пропускной способности канала, энтропии, скорости.
10. Какие помехи вызывают внешние источники помех?
11. Данна схема кодирования $\langle \delta = a \rightarrow 01, b \rightarrow 110, c \rightarrow 101 \rangle$.

Удовлетворяет ли данная схема неравенству МакМиллана?

12. Является ли разделимая схема префиксной?
13. Задача оптимального кодирования. Метод Шеннона–Фано.
14. Суть метода Хаффмена.
15. Перечислить основные технические характеристики процессов сжатия.
16. Какое сжатие называется обратимым сжатием? Необратимым сжатием?
17. Что приводит к снижению объема выходного потока информации без изменения его информативности?
18. Как осуществляется передача дискретных сообщений по каналу с шумами.
19. В чем смысл помехоустойчивого кодирования?
20. Какие существуют помехоустойчивые коды?
21. Дать определение избыточности.
22. Перечислить виды избыточности.
23. Какие помехи вызывают внешние источники помех?

24. Что понимают под кодовым расстоянием?
25. Чему равно кодовое расстояние между двумя кодовыми комбинациями (например, 011 и 101)?
26. Какие искажения называются краевыми? Дробления?
27. Какие коды называются непрерывными? Разделимыми?
28. В чем смысл кода Хемминга?
29. Какой код является оптимальным?

Модуль 3. Основы теории автоматов

Практическое занятие № 6 Комбинационные схемы

Теоретические сведения

Автомат (лат. самодвижущийся) – техническое устройство, работающее без непосредственного участия человека.

В кибернетике в слово «автомат» вкладывается более узкое понятие так называемого синхронного дискретного устройства, обладающего следующими свойствами.

Такой автомат характеризуется:

- множеством входных сигналов $X = \{x_1, x_2, \dots, x_n\}$;
- множеством выходных сигналов $Y = \{y_1, y_2, \dots, y_n\}$;
- множеством состояний $Z = \{z_1, z_2, \dots, z_s\}$.

Изменение сигналов происходит только на границах тактов безразмерного времени Δt .

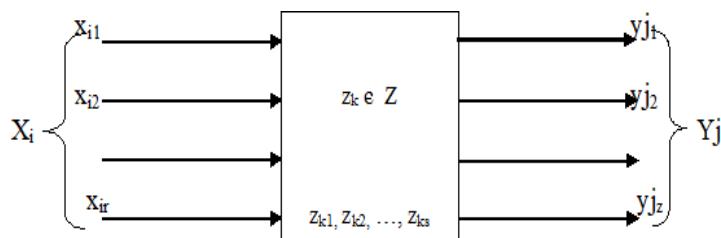


Рис. 9 – Пример автомата

Организация работы элементов памяти, при которой моменты их возможных переключений следуют друг за другом через одни и те же фиксированные промежутки времени называется тактом. Импульсы вырабатываются с помощью тактового синхронизатора.

Тактовая частота – количество тактовых импульсов, выдаваемых в течение 1 секунды (Гц, КГц, МГц, ГГц).

В каждый момент времени модель может находиться в одном из множества состояний $Z = \{z_1, z_2, \dots, z_s\}$. Таким образом, автомат можно представить в виде некоторого блока.

Функционирование автомата характеризуется его функцией переходов φ : $X \times Z \rightarrow Z$, определяющей изменение состояний под воздействием входных сигналов; и функцией выходов $\psi: X \times Z \rightarrow Y$, определяющей изменения выходов под влиянием входных сигналов.

Автомат, у которого множества X , Y , Z конечные, называется конечным автоматом.

Синтез логических и цифровых автоматов

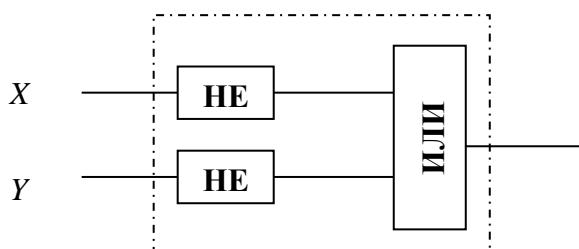
Задача синтеза автоматов может быть определена как процесс построения структур автоматов, реализующих некоторые заданные условия работы. В задачу синтеза входит оптимизация структуры автомата по тому или иному критерию, чаще всего по количеству элементов, а значит и стоимости, быстродействию и надежности. Наиболее просто решается задача синтеза автомата без памяти, то есть логических или комбинационных автоматов (схем). При этом успешно используется аппарат математической логики или булевой алгебры.

Пример. Построить схему из логических элементов.

Упростим логическое выражение:

$$\begin{aligned} X \cdot \bar{Y} \vee \bar{X} \cdot Y \vee \bar{X} \cdot \bar{Y} &= \bar{Y} \cdot (X \vee \bar{X}) \vee \bar{X} \cdot Y = \\ &= \bar{Y} \vee \bar{X} \cdot Y = \bar{Y} \vee \bar{X} \cdot \bar{Y} \vee Y = \bar{Y} \vee \bar{X} \end{aligned}$$

Схема будет выглядеть следующим образом:



Задания

Задание 1. Построить схемы из логических элементов, предварительно упростив логические выражения.

- 1) $\bar{X} \cdot \bar{Y} \vee X \cdot \bar{Y} \vee X \cdot Y;$
- 2) $X \cdot Y \cdot T \vee X \cdot Y \cdot \bar{T} \vee \bar{X} \cdot \bar{Y} \cdot \bar{T} \vee X \cdot \bar{Y} \cdot \bar{T};$
- 3) $X \cdot Y \cdot \bar{T} \vee X \cdot \bar{Y} \cdot T \vee \bar{X} \cdot Y \cdot T \vee X \cdot Y \cdot T;$

$$4) \bar{X} \cdot Y \cdot \bar{T} \vee \bar{X} \cdot \bar{Y} \cdot T \vee \bar{X} \cdot \bar{Y} \cdot \bar{T} \vee X \cdot Y \cdot \bar{T};$$

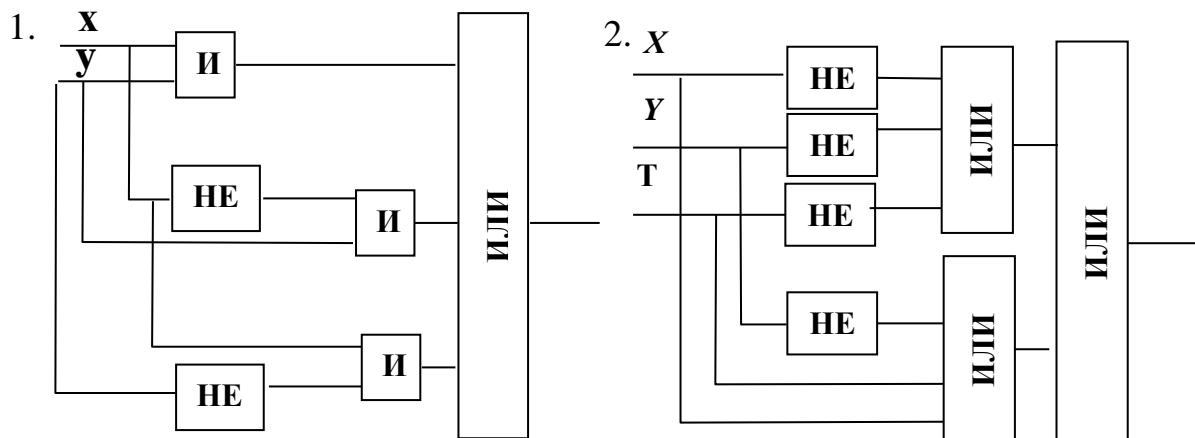
$$5) X \cdot \bar{Y} \cdot T \vee X \cdot \bar{Y} \cdot \bar{T} \vee \bar{X} \cdot \bar{Y} \cdot \bar{T} \vee X \cdot \bar{Y};$$

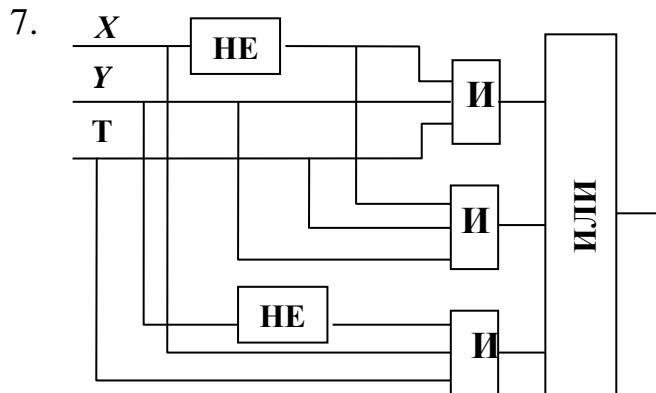
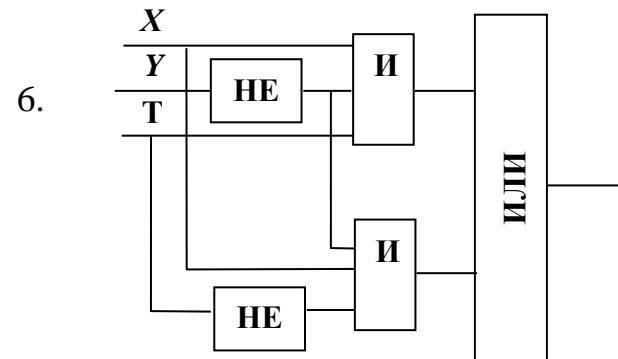
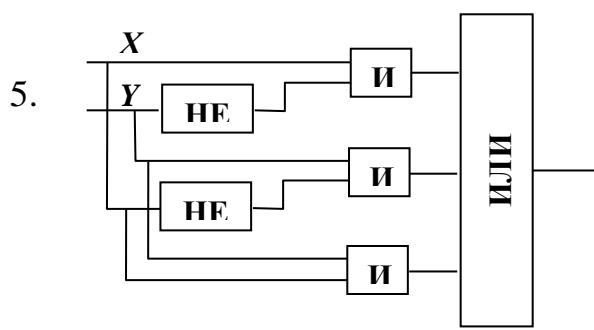
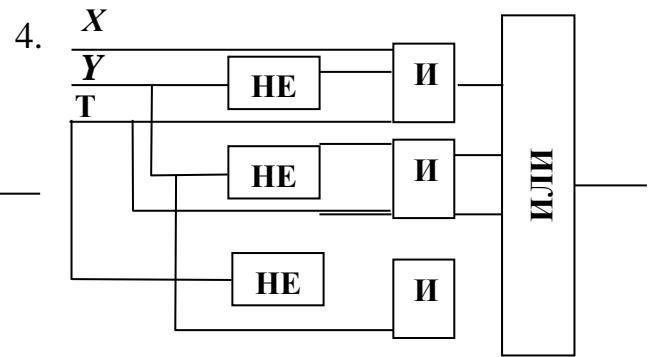
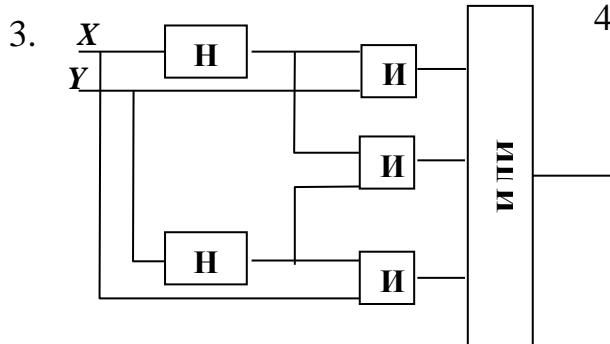
Таблица 16 – Основные законы алгебры логики

Закон	Для ИЛИ	Для И
Переместительный	$x \vee y = y \vee x$	$x \cdot y = y \cdot x$
Сочетательный	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Распределительный	$x \cdot (y \vee z) = x \cdot y \vee x \cdot z$	$x \vee (y \cdot z) = (x \vee y) \cdot (x \vee z)$
Правила де Моргана	$\overline{x \vee y} = \bar{y} \cdot \bar{x}$	$\overline{x \cdot y} = \bar{y} \vee \bar{x}$
Идемпотенции	$x \vee x = x$	$x \cdot x = x$
Поглощения	$x \vee (x \cdot y) = x$	$x \cdot (x \vee y) = x$
Склейивания	$(x \cdot y) \vee (\bar{x} \cdot y) = y$	$(x \vee y) \cdot (\bar{x} \vee y) = y$
Операция переменной с ее инверсией	$x \vee \bar{x} = 1$	$x \cdot \bar{x} = 0$
Операция с константами	$x \vee 0 = x; x \vee 1 = 1$	$x \cdot 1 = x; x \cdot 0 = 0$
Двойного отрицания	$\overline{\overline{x}} = x$	

Задание 2. По схеме восстановить функцию алгебры высказываний.

Оптимизировать представленную схему.





Задание 3. Построить схему одноразрядного субтрактора (от англ. *subtract* – вычитать).

Вычитание сводится к сложению **дополнительного кода вычитаемого**, то есть отрицательного числа.

Получить дополнительный код можно следующим путем:

- записать внутреннее представление **положительного** числа **X**;
- инвертируется двоичный код числа (заменяется 0 на 1 и 1 на 0);
- к полученному числу добавляем единицу.

Пример. Найти разность $47 - 23$

Таблица 17 – Пример выполнения действий

a) $(47)_{10}$	$(00101111)_2$
$(23)_{10}$	$(00010111)_2$
б) Инвертируем вычитаемое	11101000
в) Находим сумму (вычитаемое + 1)	11101000
	+
	1
	11101001
Результат сложения:	00101111
	11101001
	1 00011000
24	$11000_2 = 24_{10}$

Единица в девятом разряде свидетельствует о том, что получено положительное число, $(00011000)_2 = (24)_{10}$. Отсутствие в девятом разряде 1 свидетельствует о том, что получено отрицательное число. Для него следует получить дополнительный код.

Задание 4. Выполнить следующие действия: $37 - 17; 19 - 40$.

Задание для самостоятельной работы

Построить схему субтрактора.

Практическое занятие № 7 Абстрактный синтез автоматов

Теоретические сведения

1. Представление событий в автоматах

Для описания автоматов на языке регулярных событий вводят ряд операций над событиями, т.е. строят алгебру событий. Рассмотрим алгебру событий, введенную Клини и усовершенствованную академиком В.М. Глушковым.

Алгебра событий включает три операции:

- дизъюнкцию (объединение) событий;
- произведение событий;
- итерацию событий.

Дизъюнкцией событий S_1, S_2, \dots, S_k называют событие $S = S_1 \vee S_2 \vee \dots \vee S_k$, состоящее из всех слов, входящих в события S_1, S_2, \dots, S_k .

Примеры и задания

Пример. Событие S_1 содержит слова x_1, x_2x_1, x_1x_1 , т.е. $S_1 = (x_1, x_2x_1, x_1x_1)$, а $S_2 = (x_2, x_1x_2)$. Тогда $S = S_1 \vee S_2 = (x_1, x_2, x_1x_1, x_1x_2, x_2x_1)$.

Задание 1.

- $S_1 = (x_1, x_1x_1, x_2x_1)$, а $S_2 = (x_2, x_1x_2)$. Получить $S = S_1 \vee S_2$.
- $S_1 = (x_2, x_1x_2, x_2x_1)$, а $S_2 = (x_1, x_2x_1)$. Получить $S = S_1 \vee S_2$.

Произведением событий S_1, S_2, \dots, S_k называется событие $S = S_1^* S_2^* \dots S_k^*$, состоящее из всех слов, полученных приписыванием к каждому слову события S_1 каждого слова события S_2 , затем слова события S_3 и т.д.

Примеры и задания

Пример. S_1 и S_2 те же. $S = S_1^* S_2 = (x_1x_2, x_1x_1x_2, x_2x_1x_2, x_2x_1x_1x_2, x_1x_1x_2, x_1x_1x_1x_2)$.

Произведение событий не коммутативно, т.е. слова, входящие в события S_1S_2 и S_2S_1 различны, т.е. $S_1S_2 \neq S_2S_1$. Поскольку произведение не коммутативно, следует различать операции «умножение справа» и «умножение слева». Например, относительно произведения события S_1S_2 можно сказать, что событие S_2 умножено на событие S_1 справа, а событие S_1 на S_2 слева.

Задание 2. S_1 и S_2 те же, что в а) и б). Получить $S = S_1 * S_2$.

Третьей операцией, применяемой в алгебре событий, является одноместная операция **«итерация»**, которая применима только к одному событию. Для обозначения итерации вводят фигурные скобки, которые называются итерационными.

Итерацией события S называется событие $\{S\}$, состоящее из пустого слова e и всех слов вида S, SS, SSS и т.д. до бесконечности. Т.е. $\{S\} = e \vee S \vee SS \vee \dots$.

Примеры и задания

Пример. $S = (x_2, x_1x_2)$.

$$\{S\} = (e, x_2, x_2x_2, x_2x_2x_2, \dots, x_1x_2, x_1x_2x_1x_2, \dots, x_2x_1x_2, x_1x_2x_2, \dots).$$

Задание 3. $S = (x_1, x_2x_1)$. Получить $\{S\}$.

При синтезе конечных автоматов важнейшую роль играют регулярные события. Пусть дан конечный алфавит $X = (x_1, x_2, \dots, x_m)$.

Определение. Любое событие, которое можно получить из букв данного алфавита с помощью конечного числа операции дизъюнкции, произведения и итерации, называется *регулярным событием*, а выражение, составленное с помощью этих операций – *регулярным выражением*.

Рассмотрим пример составления регулярного выражения, определяющего закон функционирования конечного автомата.

Задание 4. Открыть программу FA, пример «*Составления регулярного выражения для автомата сравнения чисел*». Количество разрядов числа – произвольное.

Вопросы:

1. Как формируются выходные сигналы y_1, y_2, y_3 ?
2. Что поступает на входы автомата сравнения?
3. Какие буквы содержит выходной алфавит?
4. На пятом этапе ввести свои значения в соответствующие поля и просмотреть работу автомата.

Теоретические сведения

2. Методы абстрактного синтеза. Задача абстрактного синтеза заключается в составлении таблиц переходов и выходов автоматов по заданным условиям его функционирования, представленным в форме регулярных выражений.

Абстрактный синтез обычно выполняется в два этапа:

1. Первый этап заключается в получении таблиц переходов и выходов в некоторой исходной форме. Построенный по этим таблицам автомат обычно содержит «лишние» внутренние состояния.

2. На втором этапе производится минимизация количества внутренних состояний заданного автомата.

Синтезируемый автомат может быть задан либо как автомат Мура, либо как автомат Мили. Поскольку для автомата Мура всегда можно построить эквивалентный автомат Мили, то достаточно рассмотреть алгоритм синтеза автомата Мура, который проще автомата Мили.

Задание 5. Открыть программу FA **пример** «*Обучение. Абстрактный синтез конечных автоматов*». Рассмотреть пример абстрактного синтеза автомата для случая, когда регулярные отношения составлены без применения операции итерации, а также с применением операции итерации.

1) $S1=X_1X_1X_2VX_2X_2|Y_1$, $S2=X_1X_2VX_1X_2|Y_2$. $S3=S1VS2$. Выполнить абстрактный синтез автомата.

2) $\{X_1X_1VX_1X_2\}X_1X_1\{X_2\}X_1$. Выполнить абстрактный синтез автомата с применением операции итерации.

Задание 6. Запустить **систему** «*Конечные автоматы и регулярные выражения*». Добавить состояние или букву входного алфавита можно с помощью кнопок  и . На рис. 10 показан пример заполнения полей «Конечный алфавит» и «Регулярное выражение».

После того, как задан конечный алфавит и введено регулярное выражение,

 нажать кнопку . Установить входную цепочку можно по кнопке . Установка входной цепочки на ленту и автомата в начальное состояние осуществляется по кнопке .

Входная цепочка вводится в поле справа. В полях «Лента» и «Текущее состояние» отображается их содержимое на каждом шаге работы алгоритма.

Переход на следующий шаг – .

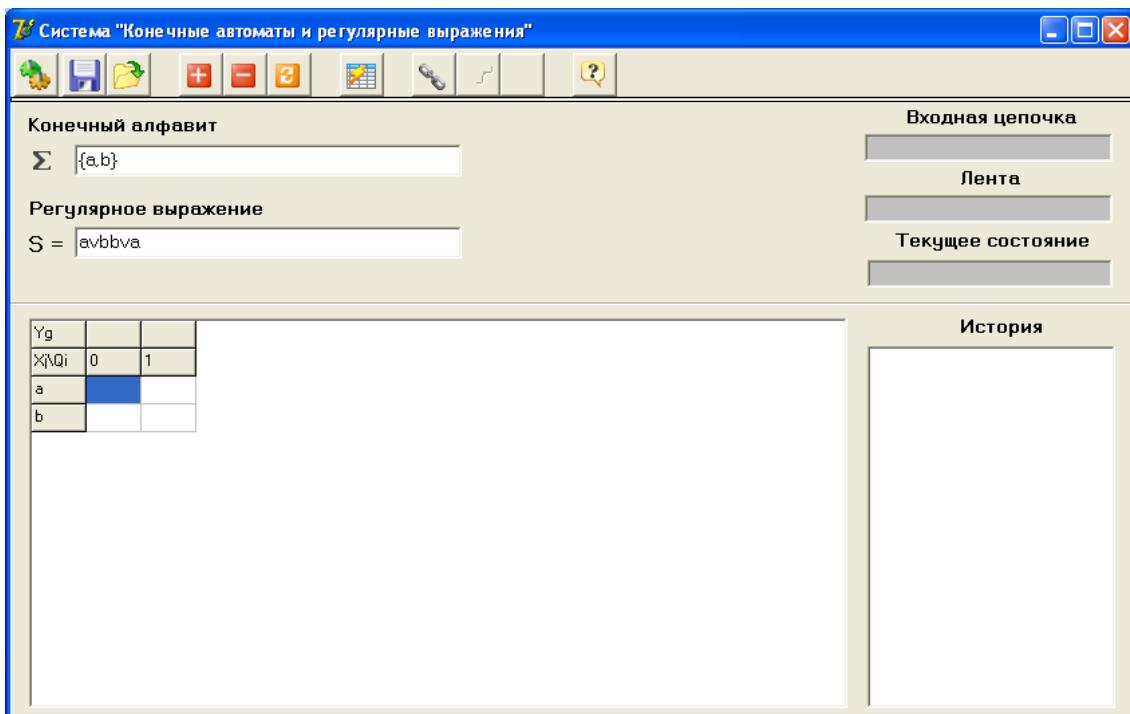


Рис. 10 – Пример заполнения полей «Конечный автомат» и «Регулярное выражение»

На рис. 11 в окне «История» отображается ход исполнения алгоритма и вывод о допустимости или недопустимости цепочки.

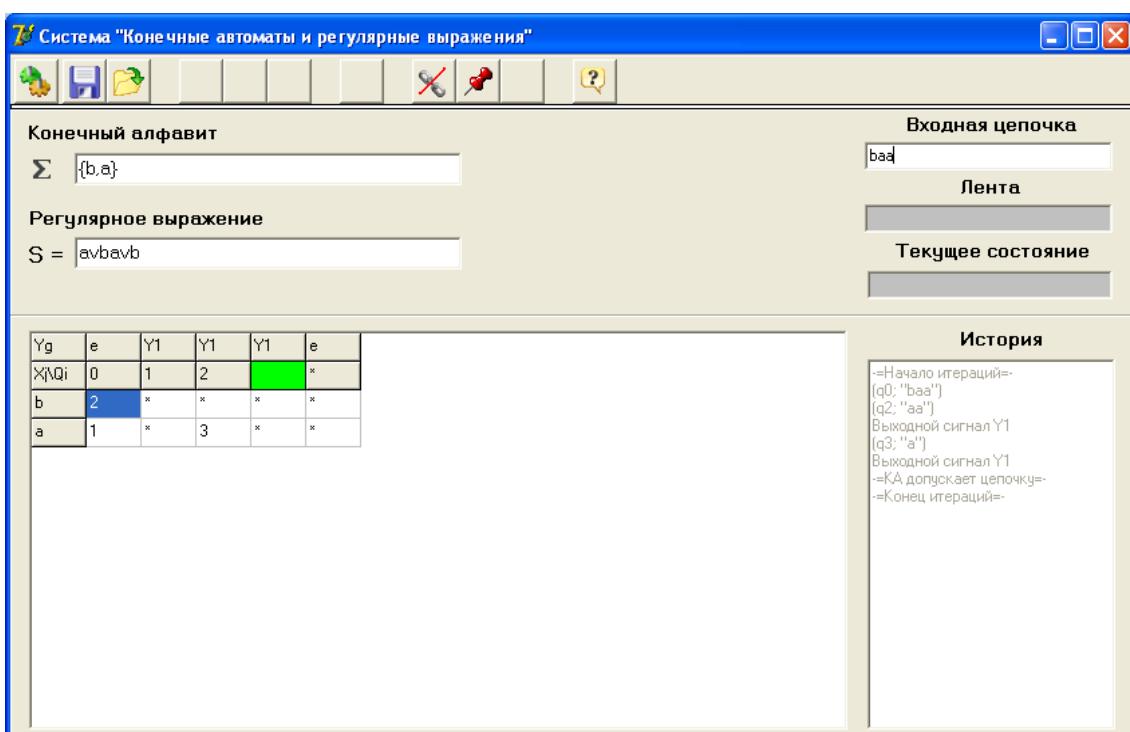


Рис. 11 – Ход выполнения алгоритма и вывод о допустимости или недопустимости цепочки

Задание 7. Задать конечный алфавит $\{a, b, c\}$ и регулярное выражение $a^*c^*a^*c^*b^*$. Просмотреть работу алгоритма для разных входных цепочек.

Задание для самостоятельной работы.

Привести два примера регулярных выражений.

Индивидуальное задание №5

Комбинационные схемы

1. Упростить логическое выражение. Построить схему из логических элементов $(X \supset Y = \bar{X} \vee Y)$.

Вариант 1

$$A \cdot B \vee \bar{A} \cdot B \vee A \cdot \bar{C} \vee C \cdot A \vee B$$

$$(\bar{A} \supset \bar{B}) \supset ((B \cdot C) \supset (A \cdot C))$$

Вариант 3

$$(X \vee Y) \cdot \bar{Z} \vee \bar{X} \cdot Z \vee Y$$

$$(C \supset A) \supset (\neg(B \vee C) \supset A)$$

Вариант 5

$$A \cdot B \vee ((A \vee B) \cdot (\bar{A} \vee \bar{B}))$$

$$(\neg(A \cdot (B \vee C)) \supset ((A \cdot B) \vee C))$$

Вариант 7

$$((A \cdot B) \vee (C \cdot B)) \cdot ((A \cdot B) \vee \bar{B})$$

$$(((A \supset B) \supset \bar{A}) \supset \bar{B}) \supset \bar{C} \supset C$$

Вариант 9

$$(X \vee Y) \cdot \bar{Z} \vee \bar{X} \cdot Y \cdot Z \vee Y \cdot \bar{X}$$

$$(P \supset Q) \vee (P \supset (Q \cdot P))$$

Вариант 11

$$X \cdot Y \vee \bar{X} \cdot Y \vee X \cdot \bar{Z} \vee Z \cdot X \vee \bar{Y}$$

Вариант 2

$$(A \vee B) \cdot A \cdot (\bar{C} \vee (A \cdot C))$$

$$((A \supset B) \supset \bar{A}) \supset (A \supset (B \cdot A))$$

Вариант 4

$$(X \vee \bar{Y}) \cdot \bar{Z} \vee \bar{Y} \cdot (Z \vee Y)$$

$$\neg((A \cdot B) \supset A) \vee (A \cdot (B \vee C))$$

Вариант 6

$$A \cdot \bar{B} \vee ((\bar{A} \vee B) \cdot (\bar{A} \vee \bar{B}))$$

$$((A \supset B) \supset (C \supset \bar{A})) \supset (\bar{B} \supset \bar{C})$$

Вариант 8

$$(((A \cdot B) \vee A) \vee C) \cdot (A \cdot \bar{B})$$

$$(A \supset (B \supset C)) \supset ((A \supset \bar{C}) \supset (A \supset \bar{B}))$$

Вариант 10

$$(Y \vee Z) \cdot \bar{Z} \vee \bar{X} \cdot (X \vee Y) \cdot \bar{Y}$$

$$\neg(P \supset \neg(Q \cdot P)) \supset (P \vee R)$$

Вариант 12

$$X \cdot Y \cdot Z \vee \bar{X} \cdot Y \cdot Z \vee X \cdot \bar{Y} \cdot \bar{Z} \vee Z \cdot X$$

$$P \cdot (Q \supset P) \supset \bar{P}$$

Вариант 13

$$X \cdot Y \vee \bar{X} \vee X \cdot \bar{Z} \vee Z \vee \neg(X \vee \bar{Y})$$

$$\left((P \cdot (Q \vee \bar{P})) \cdot (\bar{Q} \supset P) \vee Q \right)$$

Вариант 15

$$X \cdot Y \cdot Z \vee \bar{X} \cdot Y \cdot Z \vee X \cdot Y \cdot \bar{Z} \vee Z \cdot X \cdot \bar{Y}$$

$$\left((X \supset (Y \cdot Z)) \supset (\bar{Y} \supset \bar{X}) \right) \supset \bar{Y}$$

2. Выполнить следующие действия, построив дополнительный код (использовать 8-разрядный код):

Вариант 1

$$45 - 23; 12 - 34$$

Вариант 3

$$52 - 34; 21 - 33$$

Вариант 5

$$35 - 12; 22 - 45$$

Вариант 7

$$38 - 21; 25 - 32$$

Вариант 9

$$47 - 23; 22 - 45$$

Вариант 11

$$52 - 37; 25 - 67$$

Вариант 13

$$36 - 22; 24 - 51$$

Вариант 15

$$38 - 16; 30 - 47$$

$$\left((P \cdot \bar{Q}) \supset Q \right) \supset (P \supset Q)$$

Вариант 14

$$Y \cdot Z \vee \bar{Z} \vee Y \cdot Z \vee \bar{Y} \vee \bar{Z} \cdot Y$$

$$\left((Q \supset (P \cdot R)) \cdot \neg((P \vee R) \supset Q) \right)$$

2. Выполнить следующие действия, построив дополнительный код (использовать 8-разрядный код):

Вариант 2

$$56 - 33; 23 - 44$$

Вариант 4

$$42 - 31; 14 - 55$$

Вариант 6

$$55 - 32; 32 - 44$$

Вариант 8

$$52 - 29; 19 - 42$$

Вариант 10

$$51 - 34; 15 - 28$$

Вариант 12

$$33 - 26; 23 - 50$$

Вариант 14

$$54 - 12; 25 - 55$$

Практическое занятие № 8

Распознавание образов

Теоретические сведения

Распознаванием образов называются задачи построения и применения формальных операций над числовыми или символными отображениями объектов реального или идеального мира, результаты решения которых отражают отношения эквивалентности между этими объектами.

Распознавание образов (объектов, сигналов, ситуаций, явлений или процессов) – задача идентификации объекта или определения каких-либо его свойств по его изображению (оптическое распознавание) или аудиозаписи (акустическое распознавание) и другим характеристикам.

Образ – классификационная группировка в системе классификации, объединяющая (выделяющая) определенную группу объектов по некоторому признаку. Методика отнесения элемента к какому-либо образу называется решающим правилом. Метрика – способ определения расстояния между элементами универсального множества. Обучением обычно называют процесс выработки в некоторой системе той или иной реакции на группы внешних идентичных сигналов путем многократного воздействия на систему внешней корректировки. Самообучение отличается от обучения тем, что здесь дополнительная информация о верности реакции системе не сообщается.

Примеры задач распознавания образов: распознавание букв, распознавание штрих-кодов, распознавание автомобильных номеров, распознавание лиц и других биометрических данных, распознавание речи.

Задания

Задание 1. Запустить программу Prog. Познакомиться с вкладками Распознавание букв и цифр, введенных пользователем; Распознавание изображений; Распознавание и обучение.

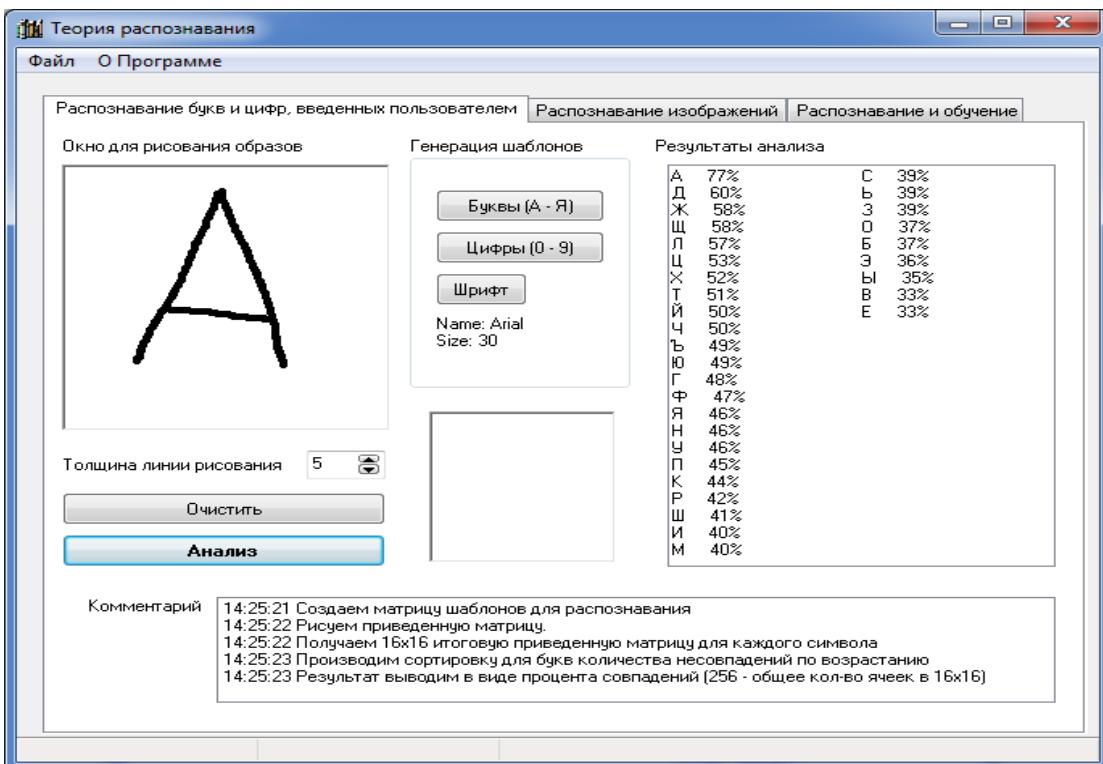


Рис. 12 – Распознавание букв и цифр, введенных пользователем

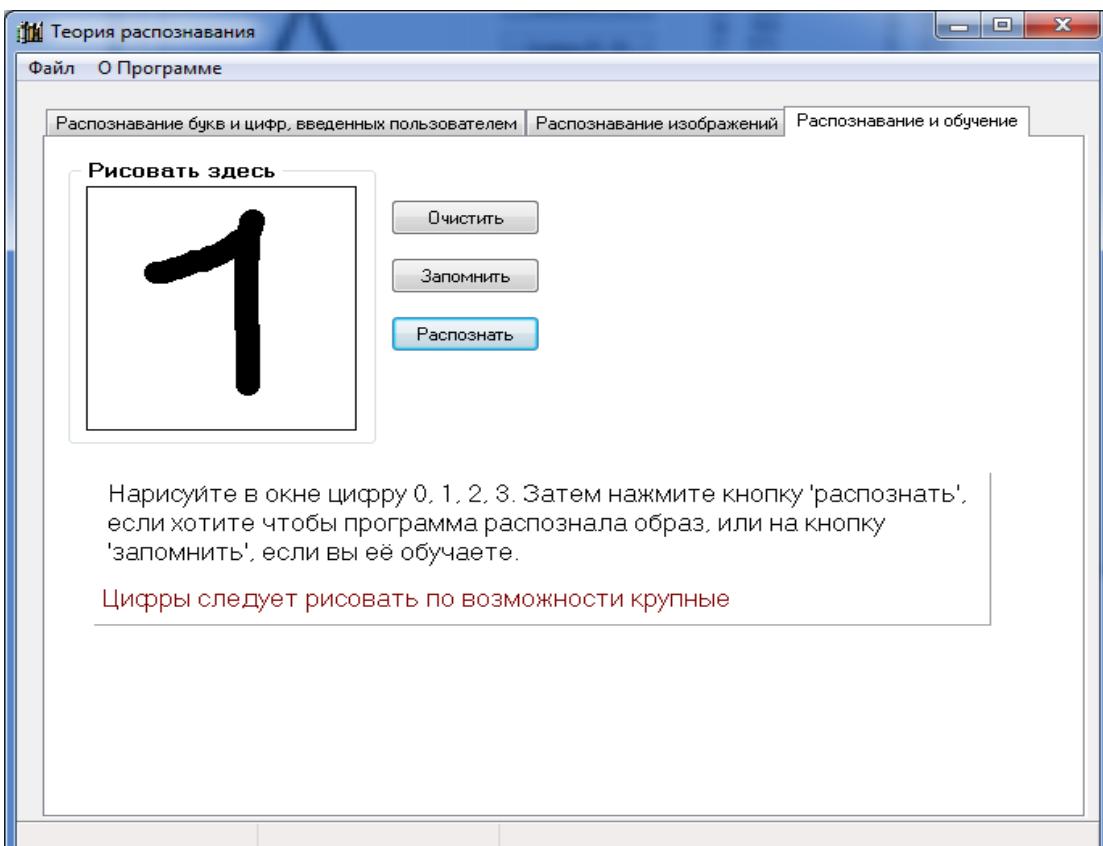


Рис. 13 – Распознавание и обучение

Задание 2. Распознать буквы А, Д, Я и цифры 1, 4, 8 на вкладке Распознавание букв и цифр, введенных пользователем. Определить процент соответствия (см. рис. 12).

Задание 3. Обучить систему распознавать цифры 2, 5, 9 на вкладке Распознавание и обучение (см. рис. 13).

Таблица 18 – Таблица соответствия признакам

x_1	x_2	x_3	x_4	цифры
4	2	0		0
				1
				2
				3
				4
				5
				6
				7
				8
				9
$\varepsilon = 1$	$\varepsilon = 1$	$\varepsilon = 1$	$\varepsilon = 1$	$\delta = 1$

Пример. Распознавание почтового индекса.

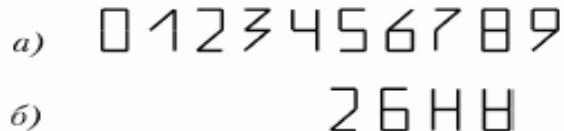


Рис. 14 – Пример задания для распознавания цифр почтового индекса

Признаки для распознавания: x_1 – количество вертикальных линий минимального размера; x_2 – количество горизонтальных линий; x_3 – количество наклонных линий; x_4 – количество горизонтальных линий снизу объекта.

Задание 4. Заполнить таблицу соответствия признакам для цифр индекса (Таблица 18).

Пример: Задача поиска геометрических аналогий.

Среди фигур второго ряда, представленных на рис. 9, требуется выбрать $X \in \{1, 2, 3, 4, 5\}$ такое, что А так соотносится с В, как С соотносится с X, и такое, которое лучше всего при этом подходит. В данном примере $X = 3$.

Задание 5. Описать один из методов распознавания образов (не менее 1000 знаков). По вариантам. Указать достоинства и недостатки. Привести примеры.

1. Методы, основанные на оценках плотностей распределения значений признаков.
2. Методы, основанные на предположениях о классе решающих функций.
3. Логические методы.
4. Лингвистические (структурные) методы.
5. Метод сравнения с прототипом.
6. Метод k-ближайших соседей.
7. Алгоритмы вычисления оценок (голосования).
8. Коллективы решающих правил.
9. Нейронные сети.
10. Метод глубокого обучения.
11. Эвристические методы.
12. Математические методы.
13. Дешифрование (как метод распознавания).

Пример: Задача поиска геометрических аналогий.

Среди фигур второго ряда, представленных на рис. 3, требуется выбрать $X \in \{1, 2, 3, 4, 5\}$ такое, что A так соотносится с B, как C соотносится с X, и такое, которое лучше всего при этом подходит. В данном примере $X = 3$.

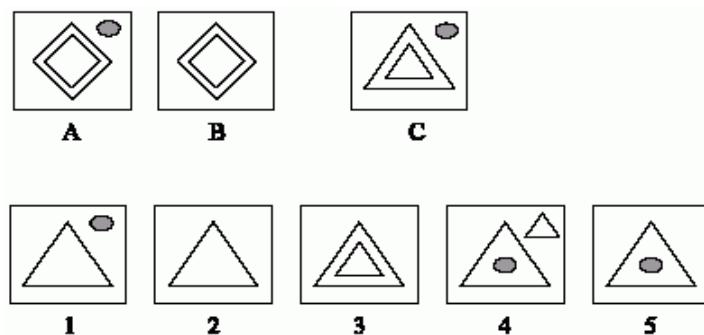


Рис. 15 – Пример задания на поиск аналогий

Правила преобразования в задаче поиска геометрических аналогий (рис. 4):

Правило 1 (исходное изображение): k выше m, k выше n, n внутри m.

Правило 2 (результатирующее изображение): n слева m.

Правило 3 (масштабирование, повороты):

- k исчезло;
- m изменение масштаба 1:1, вращение 00;
- n изменение масштаба 1:2, вращение 00.

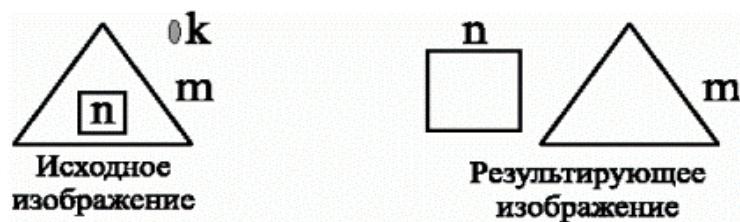


Рис. 16 – Пример правил преобразования

Пример:

Среди фигур, представленных на рисунке 5, требуется выбрать X

$\in \{1, 2, 3\}$ такое, что A так соотносится с B, как C соотносится с X, и такое, которое лучше всего при этом подходит.

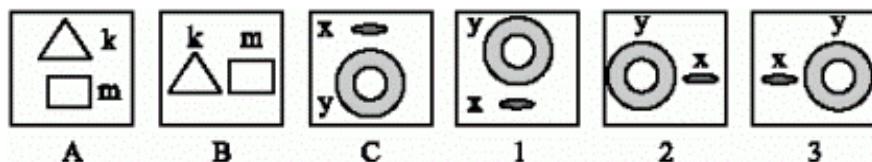


Рис. 17 – Пример на распознавание образов с использованием правил преобразования

Задание 6. Сформулировать правила преобразования в задаче поиска геометрических аналогий для рис. 5.

Задание для самостоятельной работы

1. Используя рисунок 3, сформулировать задание на распознавание образов по аналогии.
2. Используя рисунок 5, сформулировать задание на распознавание образов с использованием правил преобразования.

Вопросы к модулю 3

1. В чем заключается синтез автомата?
2. Для чего необходимо оптимизировать структуру автомата?
3. В каких единицах измеряется тактовая частота?
4. Какая логическая функция дает на выходе 0, только когда оба входа соответствуют 0?
5. В каком автомате функция ψ определяет значение выходного сигнала только по одному аргументу?
6. Как называется функция, определяющая изменения состояний под воздействием входных сигналов?
7. Как задается автомат?
8. Что собой представляет функция переходов?
9. Что собой представляет функция выходов?
10. Как называется автомат, если функции переходов и выходов однозначно определены?
11. А чем отличие математической модели автомата Мили и схемы рекуррентных соотношений отличаются от математической модели и схемы рекуррентных соотношений абстрактного автомата?
12. Какой режим работы сети автоматов называют асинхронным?
13. Какой режим работы сети автоматов называют синхронным?
14. Что собой представляет автомат без памяти?
15. В чем заключается блочный синтез?
16. В чем заключается абстрактный синтез?
17. В чем заключается структурный синтез?
18. Какие понятия в теории распознавания образов являются базовыми?
19. В чем состоит задача распознавание образов?
20. Дать определение решающего правила?
21. Что такое метрика?
22. Дать определение ситуации, состояния.

23. Что используется для распознавания сложных процессов и явлений?
24. Какие задачи относится к задачам распознавания?
25. Дать характеристику задач распознавания.
26. В чем заключается алгоритм распознавания, основанный на вычислении оценок?
27. К какому этапу относится оценка погрешности в общей схеме преобразования данных?
28. Перечислить методы распознавания образов.
29. Какие методы распознавания образов относятся к интенсиональным?
30. Какие методы распознавания образов относятся к экстенсиональным?

Модуль 4. Динамическое программирование. Алгоритмы на строках

Практическое занятие № 9 Поиск чисел Фибоначчи с помощью динамического программирования

Теоретические сведения

Для использования динамического программирования в Excel необходимо следовать нескольким шагам:

1. Определить целевую функцию. Это функция, которую нужно оптимизировать. Например, если необходимо найти путь с минимальной стоимостью в графе, функцией цели может быть сумма весов ребер на этом пути.
2. Определить ограничения. Это условия, которые должны быть удовлетворены в результате оптимизации. Например, в задаче о поиске пути с минимальной стоимостью в графе ограничением может быть не превышение заданного бюджета.
3. Создать таблицу решений. В Excel создать таблицу, где каждая ячейка представляет одну подзадачу. В этой таблице будут сохраняться результаты решения каждой подзадачи.

4. Заполнить начальные значения. Ячейки таблицы решений заполнить значениями для начальной подзадачи.

5. Применить рекурсивную формулу. В Excel можно использовать функции и формулы для вычисления значений ячеек таблицы решений на основе значений предыдущих подзадач. Это позволит решить каждую подзадачу и сохранить результаты для дальнейшего использования.

6. Найти оптимальное решение. После заполнения таблицы решений найти оптимальное решение, сравнивая значения в конечных ячейках с условиями оптимизации. Выбрать путь или решение, которое соответствует оптимальным значениям функции цели и ограничений.

Задание 1. Получить последовательность чисел Фибоначчи в Excel.

№	Числа Фибоначчи
1	0
2	1
3	1
4	2
5	3
6	5
7	8
8	13
9	21
10	34
11	55
12	89
13	144
14	233
15	377
16	610
17	987
18	1597
19	2584
20	4181

i	a	b	r
0	144	89	55
1	89	55	34
2	55	34	21
3	34	21	13
4	21	13	8
5	13	8	5
6	8	5	3
7	5	3	2
8	3	2	1
9	2	1	0

i	a	b	r
0	2584	1597	987
1	1597	987	610
2	987	610	377
3	610	377	233
4	377	233	144
5	233	144	89
6	144	89	55
7	89	55	34
8	55	34	21
9	34	21	13
10	21	13	8
11	13	8	5
12	8	5	3
13	5	3	2
14	3	2	1
15	2	1	0

Рисунок 18 – Пример вычислений в Excel

Задание 2. Вычислить НОД пары соседних чисел Фибоначчи. Сравнить количество итераций с оценкой Ламе.

Число итераций по оценке Ламе равно $n \leq 5(\lceil \log_{10} b \rceil + 1)$.

Задание 3. Реализовать в визуальной среде программирования.

Пример. Кузнечик сидит в клетке № 1 и может прыгать вправо на 1 или 3 клетки. Подсчитайте количество разных способов, которыми он может добраться до клетки N.

Программный код (фрагмент основного алгоритма):

```

for i := 1 to n do
  f[i] := 0;
  f[1] := 1;
  f[2] := 1;
  f[3] := 1;
for i := 4 to n do
  f[i] := f[i-1] + f[i-3];
  writeln F[n];

```

Задание 4. Реализовать алгоритм в Excel.

Таблица 19 – Пример реализации алгоритма

1	2	3	4	5	6	7	8	9	10	11
1	1	1	2	3	4	6	9	13	19	28

Задание 5. Получить n-е число трибоначчи. Реализовать в любой системе программирования.

Числа трибоначчи – это числовой ряд, начинающийся с трёх чисел 0, 0, 1, где каждое последующее число вычисляется как сумма трёх чисел перед ним:

0, 0, 1
 0, 0, 1, 1
 0, 0, 1, 1, 2
 0, 0, 1, 1, 2, 4
 0, 0, 1, 1, 2, 4, 7
 ...

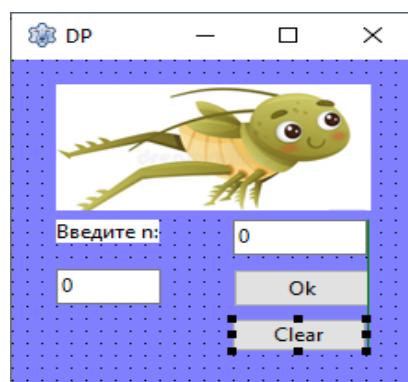


Рис. 19 – Форма

Пример кода

```

var
  Form1: TForm1;
  f:array[1..100] of integer;
  i,n:integer;
  a:string;
implementation
{$R *.lfm}
{ TForm1 }
procedure TForm1.Button1Click(Sender: TObject);

```

```

begin
a:=Edit2.Text;
n:=StrToInt(a);
for i:=1 to n do f[i]:=0;
f[1]:=1;
f[2]:=1;
f[3]:=1;
for i:=4 to n do
  f[i]:=f[i-1]+f[i-3];
  Edit1.Text:= IntToStr(f[n]);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Text:='0';
  Edit2.Text:='0';
end;

```

Самостоятельно: реализовать в Excel задачу о рюкзаке.

Практическое занятие №10 Поиск подстроки в строке

1. Работа со строками

Строка в Python – это объект класса str. В темах ранее, для того чтобы работать с числовыми данными, применялись функции приведения типов, например, int, как это продемонстрировано в следующем примере:

```
ind=int(input(" \n Введите индекс элемента "))
```

При обработке строки к ней можно обратиться непосредственно в цикле, как это показано в листинге ниже.

Пример.

```

stroka="строка"
for i in stroka:
    print(i, end=" ")

```

1) Определение количества символов

Пример. Ввести исходную строку stroka и символ simvol. Подсчитать, сколько раз заданный символ встречается в исходной строке.

```

k=0
stroka=input("\n Введите строку: ")
stroka1=stroka.lower()
spisok=list(stroka1)
simvol=input("\n Введите символ для поиска его повторений в списке: ")
n=len(spisok)

```

```

for i in range(0, n):
    if spisok[i]==simvol:
        k+=1
    print("\n", k)

```

2) Замена символов

Пример. Ввести исходную строку stroka и символ simvol. Заменить пробелы в исходной строке указанным символом.

```

stroka=input("\n Введите строку: ")
stroka1=stroka.lower()
spisok=list(stroka1)
simvol=input("\n Введите символ для замены им пробелов: ")
n=len(stroka1)
for i in range(0, n):
    if spisok[i]==" ":
        spisok[i]=simvol
stroka1="".join(spisok)
print("\n", stroka1)

```

Задание 1. Ввести исходную строку stroka и символ simvol. Удалить в исходной строке указанный символ (stroka=stroka.replace(simvol, "")).

3) Вставка символа в строку

```

*666.py - C:/Users/1/AppData/Local/Programs/Python/Python311/666.py (3.11.1)*
File Edit Format Run Options Window Help
stroka=input("\n Введите строку: ")
print(stroka)
spisok=list(stroka)
simvol=input("\n Введите первый символ, количество которого необходимо найти: ")
simvol1=input("\n Введите второй символ, количество которого необходимо найти: ")
k1=0
k2=0
k=0
n=len(stroka)
for i in range(0, n):
    if spisok[i]==simvol:
        k1+=1
    if spisok[i]==simvol1:
        k2+=1
print("\n Всего найденных символов", simvol, " = ", k1)
print("\n Всего найденных символов", simvol1, " = ", k2)
print("\n Всего найденных символов", simvol, " и ", simvol1, " = ", k)

```

Рисунок 20 – Определение количества символов simvol и simvol1 в исходной строке

Задание 2. Ввести исходную строку stroka. В исходную строку необходимо добавить символ simvol1 после символа simvol, которые пользователь вводит с клавиатуры.

```

n=len(stroka)
k=0
for i in range(0, n):
    if spisok[i]==simvol:
        k+=1
for i in range(0, n):

```

```

if spisok[i]==simvol:
    spisok.insert(i+1, simvol1)
stroka="".join(spisok)
print("\n", stroka)

```

4) Анализ символа на принадлежность к группе

На рисунках 20, 21, 22, 23 представлены примеры кода на обращение строки, срезы строк и др.

```

777.py - C:/Users/1/AppData/Local...
File Edit Format Run Options Window Help
stroka=input("\n Введите строку: ")
print("\n", stroka)
tmp=""
n=len(stroka)
for i in range(0, n):
    tmp=stroka[i]+tmp
print("\n Стока наоборот: ", tmp)

```

Ln: 6 Col: 4

Рисунок 21 – Пример записи символов в обратном порядке

```

888.py - C:/Users/1/AppData/Local/Programs/Python/Python311/888.py (3.11.1)
File Edit Format Run Options Window Help
stroka=input("\n Введите строку: ")
print("\n", stroka)
flag=None
while flag!="":
    flag=input("\n Введите начало среза или для выхода введите ENTER:")
    if flag:
        flag=int(flag)
        kon=input("\n Введите конец среза: ")
        kon=int(kon)
        print(stroka[flag:kon])
input("\n Введите ENTER для выхода ")

```

Ln: 12 Col: 0

Рисунок 22 – Пример получения символов, расположенных между заданными начальным и конечным значениями.

```

999.py - C:/Users/1/AppData/Local/Pr...
File Edit Format Run Options Window Help
ALFAVIT="print"
stroka=input("\n Введите строку: ")
print("\n", stroka)
tmp=""
for letter in stroka:
    if letter in ALFAVIT:
        tmp+=letter
print(tmp)

```

Ln: 1 Col: 14

Рисунок 23 – Пример выбора из строки символов, используемых в заранее заданном алфавите

Индивидуальное задание №6

Задание 1.

1. В строке заменить все пробелы на «_», посчитать количество замененных символов.
2. В строке заменить все пробелы на «*», посчитать количество замененных символов.
3. Данна строка. Определить, сколько в ней «,», заменить их на «;».
4. Данна строка. Определить, сколько в ней «;», удалить их.
5. Удалить часть символьной строки в скобках (вместе со скобками).
6. Данна строка, содержащая кавычки. Выдать сообщение, правильно ли расставлены кавычки (количество открывающихся соответствует количеству закрывающихся).
7. Удалить часть символьной строки в кавычках (вместе с кавычками).
8. Данна строка. Найти количество слов, начинающихся с буквы А.
9. Данна строка. Заменить первую букву слов, начинающихся с буквы А, на малую букву.
10. Данна строка. Заменить последнюю букву слов, заканчивающихся на е, на большую букву.

11. Данна строка. Найти количество букв «к» и «р».

12. Данна строка. Определить, сколько раз в нее входит группа букв «ель».

Задание 2.

1. Даны целые положительные числа n1 и n2 и строки S1 и S2. Получить из этих строк новую строку, содержащую первые n1 символов S1 и последние n2 символов строки S2.
2. Даны целое положительное число n и строка S. Преобразовать строку S в строку длины n следующим образом: если длина S больше n, то отбросить первые символы, если меньше, то в начало добавить «.».
3. Даны строки S1 и S2. Проверить, содержится ли S1 в строке S2.
4. Даны строки S1 и S2. Удалить из строки S1 подстроку S2, если S2 не содержится в S1, вывести S1 без изменений.

5. Даны строки S1 и S2. Удалить из строки S1 первую подстроку S2, если S2 не содержится в S1, вывести S1 без изменений.

6. Даны строки S1 и S2. Удалить из строки S1 последнюю подстроку S2, если S2 не содержится в S1, вывести S1 без изменений.

7. Даны строки S1 и S2. Удалить из строки S1 все подстроки S2, если S2 не содержится в S1, вывести S1 без изменений.

8. Даны строки S, S1 и S2. Заменить в строке S последнюю подстроку S1 на S2, если S1 не содержится в S, вывести S без изменений.

9. Даны строки S, S1 и S2. Заменить в строке S первую подстроку S1 на S2, если S1 не содержится в S, вывести S без изменений.

10. Даны строки S, S1 и S2. Заменить в строке S все подстроки S1 на S2, если S1 не содержится в S, вывести S без изменений.

11. Данна строка, содержащая круглые скобки. Выдать сообщение, правильно ли расставлены скобки (количество открывающихся соответствует количеству закрывающихся).

12. Стока содержит одно слово. Проверить, будет ли оно читаться одинаково слева направо и наоборот.

Вопросы к модулю 4

1. Что собой представляет динамическое программирование?

2. Какие задачи можно решать с помощью динамического программирования?

3. Как взаимодействуют элементы в условиях динамического программирования?

4. Чем отличается одномерное динамическое программирование от двумерного динамического программирования?

5. В какой последовательности осуществляется поиск решения задачи с помощью динамического программирования?

6. Как называется подход в динамическом программировании, при котором решения подзадач итеративно заносятся в таблицу?

7. Как называется подход в динамическом программировании, при котором решения подзадач рекурсивно вычисляются и сохраняются для использования в будущем?
8. В чем особенность алгоритма Беллмана-Форда?
9. Чем отличается алгоритм Беллмана-Форда от алгоритма Флойда-Уоршелла?
10. Какие существуют методы решения рекурсивных задач?
11. Сформулировать задачу о рюкзаке.
12. Как можно работать со строками?
13. Как найти подстроку в строке?
14. Как проверить, содержится ли подстрока в строке?
15. Какой метод можно использовать для поиска подстроки в строке?
16. Как извлечь подстроку из строки?

Вопросы для итогового тестирования

1. Среди перечисленных определений одно не относится к определению понятия «информация»:

- 1) сообщение о состоянии и свойствах объекта, явления, процесса;
- 2) последовательность действий, направленных на достижение цели;
- 3) содержание сигналов, поступающих в кибернетическую систему из окружающей среды, которое может быть использовано для целей управления системой.

2. Для преобразования сигналов из непрерывной формы в дискретную используются преобразователи:

- 1) код-аналог;
- 2) аналого-цифровые;
- 3) цифро-аналоговые.

3. Любой непрерывный сигнал, имеющий ограниченный спектр частот, полностью определяется последовательностью своих мгновенных значений, отсчитанных через интервалы времени $\Delta t = \frac{1}{2f_c}$, где f_c – верхняя граничная частота спектра непрерывного сигнала. Это теорема:

- 1) А. Хемминга;
- 2) К. Шеннона;
- 3) В. Котельникова.

4. Определять информационную емкость системы предложил:

- 1) К. Шенон;
- 2) А. Харкевич;
- 3) Р. Хартли;
- 4) Ю. Шрейдер.

5. Полезность или ценность информации для выполнения функций управления предложил определять:

- 1) К. Шенон;
- 2) Ю. Шрейдер;

- 3) Р. Хартли;
- 4) А. Харкевич.

6. Пусть алфавит содержит четыре символа и их вероятности равны соответственно $p_1 = p_2 = p_3 = p_4 = 0,25$. Чему будет равна неопределенность (энтропия) H ?

- 1) 1;
- 2) 2;
- 3) 4.

7. Единица измерения степени неопределенности, содержащаяся в одном опыте, имеющем два равновероятных исхода, называется:

- 1) бит;
- 2) байт;
- 3) бит/буква;
- 4) бит/сек.

8. Чему будет равно количество информации, если событие имеет два равновероятных исхода:

- 1) $\frac{1}{2}$;
- 2) 1;
- 3) 2.

9. Поставить в соответствие подходу его характеристику:

- 1) структурный;
 - 2) статистический;
 - 3) прагматический;
 - 4) семантический.
- a) учет вероятностных характеристик источника сообщений;
 - b) оценка сообщения с точки зрения получения лучшего управленческого решения;
 - c) опирается на тезаурусный подход;
 - d) определение меры количества информации;
 - e) определение информационной емкости сообщения.

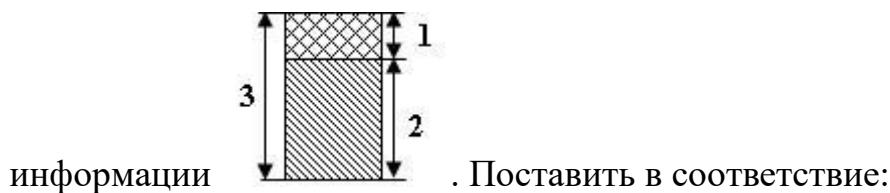
10. Опыт γ состоит в выполнении опытов α и β . Поставить в соответствие:

- 1) H_γ ;
- 2) H_α ;
- 3) $I_{\alpha,\beta}$.
 - a) условная энтропия;
 - b) безусловная энтропия;
 - c) энтропия сложного опыта;
 - d) количество информации.

11. Если опыт заключается в одновременной реализации опытов α и β , независимых друг от друга, то энтропия:

- 1) $H_\gamma > H_\alpha + H_\beta$;
- 2) $H_\gamma = H_\alpha + H_\beta$;
- 3) $H_\gamma \leq H_\alpha + H_\beta$.

12. На рисунке показана зависимость энтропии и количества



- 1) 1;
- 2) 2;
- 3) 3.
 - a) H_β ;
 - b) $H_{\beta/\alpha}$;
 - c) $H_{\beta/\beta}$;
 - d) $I_{\alpha,\beta}$.

13. Наибольшую неопределенность среди всех опытов, имеющих n исходов, имеет опыт, у которого исходы:

- 1) зависят от предыдущего опыта;
- 2) неравновероятны;

3) равновероятны.

14. Количество какой информации может быть вычислено по формуле

$$I = \log \frac{p_1}{p_0}, \text{ если } p_1 - \text{вероятность достижения цели после получения информации}$$

о событии, p_0 – вероятность достижения цели до получения информации о событии?

1) семантической;

2) синтаксической;

3) прагматической.

15. Восстановить правильную последовательность системы связи:

1) канал связи;

2) получатель сообщения;

3) источник сообщения;

4) передатчик;

5) приемник.

16. В избыточных кодах имеет место соотношение:

1) $d = r + s - 1, r \geq s$;

2) $d = 1 + r + s, r \geq s$;

3) $d = 1 + r + s, r \leq s$.

17. Кодовое расстояние $d = 2$. Выбрать свойство кода.

1) обнаруживает одну ошибку;

2) обнаруживает две ошибки;

3) отличает одну кодовую комбинацию от другой.

18. Если из $\beta_{i1}\beta_{i2}... \beta_{ik} = \beta_{j1}\beta_{j2}... \beta_{jl}$ следует, что $k = l, \forall t \in \{1, 2, ..., k\}$ и $i_t = j_t$, то схема кодирования называется:

1) префиксной;

2) постфиксной;

3) разделимой.

19. Префиксная схема кодирования является разделимой?

1) да;

2) нет.

20. Пусть $A = \{a, b\}$, $B = \{0, 1\}$ и $\delta = \langle a \rightarrow 010, b \rightarrow 01 \rangle$. $I_\delta(P) = \sum_{i=1}^n P_i l_i$. Чему будет равна средняя длина кодирования δ , если $P = \langle 0.2, 0.8 \rangle$?

- 1) 1;
- 2) 1.4;
- 3) 2;
- 4) 2.2.

21. Алгоритм Фано строит схему кодирования:

- 1) оптимальную префиксную;
- 2) разделимую префиксную;
- 3) оптимальную постфиксную.

22. Распределение вероятностей и код Хаффмена представлены в таблице:

P_i	код
0.1	10
0.2	11
0.18	00
0.17	001
0.15	010
0.15	0110
0.01	0111

Чему равно $I_\delta(P)$?

- 1) 2.56;
- 2) 2.76;
- 3) 2.94;
- 4) 3.

23. Поставить в соответствие.

- 1) С бит/с;
- 2) Н бит/букв;
- 3) В букв/с.
 - a) пропускная способность канала;
 - b) энтропия источника информации;
 - c) скорость передачи букв по каналу;

d) средняя длина кода сообщения.

24. Передача букв по каналу со скоростью $V < \frac{C}{H}$ возможна?

1) да;

2) нет.

25. Пусть вероятности появления букв $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ в сообщении равны $p_1 = 0,125$, $p_2 = 0,5$, $p_3 = 0,125$, $p_4 = 0.25$, $C = 10000$ бит/с. Чему равна максимальная скорость передачи сообщения?

1) 2500 букв/с;

2) 5000 букв/с;

3) 5714 букв/с.

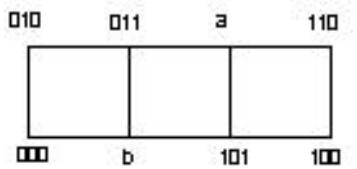
26. Число разрядов, на которые отличаются любые две кодовые комбинации, называется:

1) оптимальным кодом;

2) коэффициентом сжатия;

3) кодовым расстоянием.

27. Трехэлементный двоичный код представлен в виде графа:



Чему равны a и b?

1) a - 011, b - 001;

2) a - 111, b - 001;

3) a - 001, b - 111.

28. Чему равно кодовое расстояние между двумя кодовыми комбинациями 111 и 100?

1) 1;

2) 2;

3).

29. Пусть $A = \{c, d\}$, $B = \{0, 1\}$ и $\delta = \langle c \rightarrow 01, d \rightarrow 0 \rangle$. Схема δ :

- 1) разделима;
- 2) неразделимая;
- 3) префиксная.

30. Данна схема $\delta = \langle a \rightarrow 01, b \rightarrow 110, c \rightarrow 101 \rangle$. Удовлетворяет ли данная схема неравенству МакМиллана?

- 1) да;
- 2) нет.

31. Данна схема $\delta = \langle a \rightarrow 01, b \rightarrow 10, c \rightarrow 101 \rangle$. Чему равно значение $\sum_{i=1}^n \left(\frac{1}{2}\right)^{l_i}$?

- 1) $\frac{1}{2}$;
- 2) $\frac{3}{8}$;
- 3) $\frac{5}{8}$;
- 4) 1.

32. Является ли префиксная схема неразделимой?

- 1) да;
- 2) нет.

33. Выбрать последовательность операций, записанных в порядке уменьшения их приоритета:

- 1) $\neg, ., \&;$
- 2) $\neg, \&, ;$
- 3) $\&, \neg, ;$
- 4) $\&, ., \neg.$

34. Двоичные цифры или булевы величины представляются в виде:

- 1) непрерывных кодов;
- 2) физических сигналов;
- 3) последовательностей сигналов.

35. Какая логическая функция дает на выходе 1, только когда оба входа соответствуют 1?

- 1) дизъюнкция;
- 2) конъюнкция;

3) импликация.

36. Какая логическая функция дает на выходе 0, только когда оба входа соответствуют 0?

- 1) дизъюнкция;
- 2) конъюнкция;
- 3) логическое следование.

37. Комбинационная схема может обучаться?

- 1) да;
- 2) нет.

38. Тактовая частота измеряется в:

- 1) байтах;
- 2) бодах;
- 3) бит/с;
- 4) герцах;
- 5) кулонах.

39. Автоматом называется система $S = \langle A, Q, V, \delta, \lambda \rangle$. Поставить в соответствие:

- 1) A ;
a) алфавит сообщений;
- 2) Q ;
b) алфавит состояний;
- 3) V ;
c) входной алфавит;
- 4) δ ;
d) выходной алфавит;
- 5) λ .
e) функция входов;
f) функция переходов;
g) функция выходов.

40. Конечный автомат, имеющий только одно внутреннее состояние – это:

- 1) автомат с памятью;
- 2) автомат без памяти;
- 3) детерминированный автомат.

41. Базовым в теории распознавания является понятие:

- 1) алгебры;
- 2) матрицы;
- 3) множества;
- 4) многочлена.

41. Методика отнесения элемента к какому-либо образу называется:

- 1) метрикой;
- 2) распознаванием;
- 3) решающим правилом.

42. Состояние – это:

- 1) некоторая совокупность состояний сложного объекта, каждая из которых характеризуется одними и теми же или схожими характеристиками объекта;
- 2) определенная форма отображения измеряемых текущих (или мгновенных) характеристик наблюдаемого объекта;
- 3) каждое отображение какого-либо объекта на воспринимающие органы распознающей системы, независимо от его положения относительно этих органов.

43. Некоторую совокупность состояний сложного объекта, каждая из которых характеризуется одними и теми же или схожими характеристиками объекта, принято называть:

- 1) изображением;
- 2) ситуацией;
- 3) состоянием.

44. Для распознавания зрительных и слуховых образов используются:

- 1) состояние и ситуации;
- 2) изображения объекта и образы;

3) нет верного ответа.

45. Среди перечисленных задач одна не относится к задачам распознавания:

- 1) для этих задач трудно строить формальные теории и применять классические математические методы;
- 2) в этих задачах возможна «плохая» информация (информация с пропусками, разнородная и др.);
- 3) в этих задачах нельзя вводить понятие аналогии или подобия объектов.

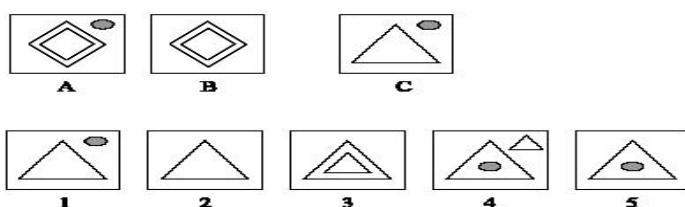
46. Поставить в соответствие типу задачи распознавания ее характеристику:

- 1) задача распознавания;
- 2) задача автоматической классификации.
 - a) отнесение предъявленного объекта по его описанию к одному из заданных классов (обучение с учителем);
 - b) разбиение множества объектов (ситуаций) по их описаниям на систему непересекающихся классов (таксономия, кластерный анализ, обучение без учителя);
 - c) задача, в которой решение должно относиться к некоторому моменту в будущем.

47. Восстановить алгоритм распознавания, основанный на вычислении оценок:

- 1) удалим произвольный набор признаков из строк $\omega_1, \omega_2, \dots, \omega_m$;
- 2) обозначим полученные строки через $S\omega_1, S\omega_2, \dots, S\omega_m, S\omega'$;
- 3) зададим полный набор признаков x_1, \dots, x_n ;
- 4) Выделим систему подмножеств множества признаков S_1, \dots, S_k .

48. Среди фигур второго ряда, представленных на рисунке, требуется выбрать $X \in \{1, 2, 3, 4, 5\}$ такое, что А так соотносится с В, как С соотносится с X,



и такое, которое лучше всего при этом подходит.

- 1) $X = 2$;
- 2) $X = 3$;
- 3) $X = 5$.

49. В общей схеме преобразования данных оценка погрешности относится к этапу:

- 1) корректировки;
- 2) обработки.

50. Алгоритмический метод решения задачи оптимизации путем разбиения ее на более простые подзадачи и использования того факта, что оптимальное решение общей проблемы зависит от оптимального решения ее подзадач – это:

- а) динамическое программирование;
- б) модульное программирование;
- в) объектно-ориентированное программирование.

51. Как взаимодействуют элементы в условиях динамического программирования?

- а) должны находиться в зависимости друг от друга;
- б) должны не зависеть друг от друга;
- в) допустимы оба случая.

52. Восстановить правильную последовательность поиска решения задачи с помощью динамического программирования:

- а) определение класса подзадач;
- б) (указание рекуррентного соотношения, основанного на решении каждой подзадачи с точки зрения более простых подзадач;
- в) указание алгоритма вычисления рекуррентного соотношения.

53. Поставить в соответствие технике реализации ДП ее описание:

- 1) табуляция;
- 2) мемоизация.

а) это подход «снизу вверх» в динамическом программировании, при котором решения подзадач итеративно заносятся в таблицу;

б) это подход «сверху вниз» в динамическом программировании, при котором решения подзадач рекурсивно вычисляются и сохраняются для использования в будущем;

в) это подход «сверху вниз» в, при котором решения подзадач итеративно заносятся в таблицу.

54. Алгоритм позволяет найти кратчайшие пути из одной вершины графа до всех остальных, даже с отрицательными весами ребер. Это –

- а) алгоритм Беллмана-Форда;
- б) алгоритм Флойда-Уоршелла?

55. Среди перечисленных задач выбрать задачу, относящуюся к одномерному динамическому программированию.

- а) задача о кузнецике;
- б) задача о рюкзаке;
- в) задача о максимальном потоке.

56. Определение префикс-функции является основой алгоритма:

- а) Кнута – Морриса – Пратта;
- б) Бойера – Мура;
- в) Рабина – Карпа.

57. Суть алгоритма состоит в последовательном переборе с последующим сравнением символов строки и образца. Это –

- а) алгоритм Бойера – Мура;
- б) переборный алгоритм;
- в) простой поиск?

58. Метод `find()` в Python:

- а) возвращает позицию первого вхождения подстроки в другую строку или `false`, если подстрока не найдена;
- б) возвращает индекс первого вхождения подстроки в строку;
- в) позволяет проверить, содержит ли строка определённую подстроку.

Заключение

В пособии представлены материалы для изучения второй части курса «Теоретические основы информатики».

Концепция ядра высшего педагогического образования направлена на повышение качества предметной подготовки будущих учителей, а также ее практикоориентированность. В связи с этим, в представленном пособии уделено внимание собственно программированию. В качестве языка программирования выбраны Object Pascal и Python.

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 дисциплин «Теоретические основы информатики» и «Теория информации, данные, знания» используется модульно-рейтинговая система контроля знаний студентов. В приложении приведен пример оценки сформированности компетенций по указанным дисциплинам.

Библиографический список

1. Глушков В.М. Синтез цифровых автоматов / В.М. Глушков. – М.: Физматгиз, 1962. – 476 с.
2. Колмогоров А.Н. Три подхода к определению понятия «Количество информации» / А.Н. Колмогоров // Новое в жизни, науке, технике. – Сер. «Математика, кибернетика», № 1, 1991. – С. 24–29.
3. Кормен Т. Глава 15. Динамическое программирование / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн // Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова. – 2-е изд. – М.: Вильямс, 2005. — 1296 с.
4. Лидовский В.В. Теория информации: уч. пособие / В.В. Лидовский. – М.: Компания Спутник+, 2004. – 111 с.
5. Общероссийский портал Math-Net.Ru. – URL: <http://www.mathnet.ru/> (дата обращения 16.06.2025).
6. Окулов С.М. Динамическое программирование / С.М. Окулов. – М.: Бином. Лаборатория знаний, 2017. – 296 с.
7. Поднебесова, Г. Б. Теоретические основы информатики : Практикум / Г. Б. Поднебесова. – Челябинск : Челябинский государственный педагогический университет, 2015. – 110 с. – ISBN 978-5-906777-56-0. – EDN VUFGIJ.
8. Поднебесова, Г. Б. Теоретические основы информатики / Г. Б. Поднебесова. – Челябинск : Южно-Уральский государственный гуманитарно-педагогический университет, 2022. – 196 с. – ISBN 978-5-907611-09-2. – EDN CGHBSN.
9. Подчукаев В.А. Теория информационных процессов и систем: учеб. пособие для вузов / В.А. Подчукаев. – М.: Гардарики, 2007. – 209 с. – ISBN 5-8297-0297-5
10. Современная теория автоматов. – Калининград: Изд-во БФУ им. И. Канта, 2013. – 261 с.

Приложения

Приложение 1. Рабочая (модульная) программа по видам деятельности

Лекции

Наименование раздела дисциплины (модуля)/ Тема и содержание	Трудоемкость (кол-во часов)
1. Основы теории информации	4
Формируемые компетенции, образовательные результаты:	
ПК-1: ПК.1.1	
1.1. Информация. Канал связи <ul style="list-style-type: none">1. Понятие об информации. Понятие информации в теории Шеннона.2. Ансамбли источника и приемника3. Различные определения информации.4. Формы представления информации.5. Канал связи как сложная система.6. Объемный подход.7. Теорема Котельникова-Найквиста. Учебно-методическая литература: 2, 4, 7, 8 Профессиональные базы данных и информационные справочные системы: 1	2
1.2. Подходы к измерению количества информации <ul style="list-style-type: none">1. Измерение количества информации.2. Подходы к измерению количества информации3. Информационная энтропия. Безусловная энтропия.4. Условная энтропия.5. Энтропия сложных систем. Учебно-методическая литература: 2, 4, 7, 8	2
2. Сжатие информации. Помехоустойчивое кодирование	4
Формируемые компетенции, образовательные результаты:	
ПК-1: ПК.1.1	
2.1. Передача дискретных сообщений по каналу без помех и с помехами <ul style="list-style-type: none">1. Оптимальное кодирование.2. Первая теорема Шеннона.3. Алгоритмы Хаффмана, Шеннона-Фано.4. Понятие о помехоустойчивом кодировании.5. Вторая теорема Шеннона.6. Кодовое расстояние.7. Самокорректирующие коды Хэмминга. Учебно-методическая литература: 4, 8, 9	2
2.2. Алфавитное кодирование. Сжатие данных <ul style="list-style-type: none">1. Основные принципы.2. Алфавитное кодирование.3. Минимизация длины кода сообщения.4. Неравенство Макмиллана.5. Сжатие данных. Сжатие со словарем (LZW). Учебно-методическая литература: 4, 8, 9	2
3. Основы теории автоматов	4
Формируемые компетенции, образовательные результаты:	
ПК-1: ПК.1.1	

3.1. Представления о конечных автоматах 1. Логические элементы. 2. Комбинационные схемы. 3. Последовательные схемы. Учебно-методическая литература: 1, 10 Профессиональные базы данных и информационные справочные системы: 5	2
Наименование раздела дисциплины (модуля)/ Тема и содержание	Трудоемкость (кол-во часов)
3.2. Распознавание образов и анализ сцен 1. Понятие образа. 2. Проблема обучения распознаванию образов (ОРО). 3. Геометрический и структурный подходы. 4. Обучение и самообучение. 5. Автоматные языки и распознавание. Учебно-методическая литература: 1, 10 Профессиональные базы данных и информационные справочные системы: 5	2
4. Динамическое программирование. Алгоритмы на строках	4
Формируемые компетенции, образовательные результаты: ПК-1: ПК.1.1	
4.1. Динамическое программирование 1. Одномерное динамическое программирование. 2. Концепция динамического программирования. 3. Двумерное динамическое программирование. 4. Общая задача динамического программирования. Учебно-методическая литература: 1, 3, 6 Профессиональные базы данных и информационные справочные системы: 5	2
4.2. Алгоритмы на строках 1. Префикс-функция. 2. Алгоритм Кнута-Морриса-Пратта. 3. Z-функция строки. 4. Поиск подстроки в строке. Учебно-методическая литература: 1, 3, 7 Профессиональные базы данных и информационные справочные системы: 5	2

Лабораторные занятия

Наименование раздела дисциплины (модуля)/ Тема и содержание	Трудоемкость (кол-во часов)
1. Основы теории информации	2
Формируемые компетенции, образовательные результаты: ПК-1: ПК.1.1	
1.1. Вычисление статистических характеристик текстовой информации - определение количества информации; - построение таблицы частот. - анализ частоты появления букв русского алфавита в тексте с помощью Excel. Учебно-методическая литература: 2, 5, 7	2
2. Сжатие информации. II кодирование омехоустойчивое	8
Формируемые компетенции, образовательные результаты: ПК-1: ПК.1.1	
2.1. Кодирование информации методами Шеннона-Фано и Хаффмена - кодирование информации. - методом Шеннона-Фано закодировать фразу "Сшил колпак да не поколпаковски". - методом Хаффмана закодировать фразу "На дворе трава, на траве дрова". Учебно-методическая литература: 4, 5, 9	2

2.2. Сжатие данных методом Лемпеля-Зива - закодировать фразу "прямпрямпрямопрямо" LZ-методом. - применить алгоритм LZ78 для кодирования последовательности 0100011101010. Учебно-методическая литература: 4, 5, 9	2
2.3. Помехоустойчивое кодирование - обнаружение одиночной ошибки (1-ый метод) - применить три проверки. - выполнение сложения по модулю 2. - обнаружение одиночной ошибки (2-ой метод). Учебно-методическая литература: 4, 5, 7, 9	4
3. Основы теории автоматов	6
Формируемые компетенции, образовательные результаты:	
ПК-1: ПК.1.1	
3.1. Комбинационные схемы - построение схем из логических элементов. - упрощение логических уравнений. - построение субтрактора. Учебно-методическая литература: 8, 5, 10 Профессиональные базы данных и информационные справочные системы: 5	2
3.2. Абстрактный синтез автоматов - абстрактный синтез конечных автоматов. - минимизация и детерминация конечных автоматов. - автоматы Мили и Мура. Учебно-методическая литература: 8, 5, 10 Профессиональные базы данных и информационные справочные системы: 5	2
3.3. Автоматные языки и распознавание - распознавание символов; - обучение и распознавание с помощью программы Prog. Учебно-методическая литература: 4, 6, 7 Профессиональные базы данных и информационные справочные системы: 5	2
4. Динамическое программирование. Алгоритмы на строках	4
Формируемые компетенции, образовательные результаты:	
ПК-1: ПК.1.1	
4.1. Поиск чисел Фибоначчи с помощью динамического программирования - постановка задачи; - рекурсивное решение; - решение с помощью динамического программирования. Учебно-методическая литература: 3, 6 Профессиональные базы данных и информационные справочные системы: 5	2
4.2. Алгоритмы поиска подстроки - алгоритм последовательного (прямого) поиска; - алгоритм Рабина - Карпа; - анализ алгоритмов; - реализация программного кода. Учебно-методическая литература: 3, 6 Профессиональные базы данных и информационные справочные системы: 5	2

Приложение 2. Содержание самостоятельной работы

<i>Наименование раздела дисциплины (модуля)/ Тема и содержание</i>	<i>Трудоемкость (кол-во часов)</i>
1. Основы теории информации	8
Формируемые компетенции, образовательные результаты: ПК-1: ПК.1.1	
1.1. Информация. Канал связи Задание для самостоятельного выполнения студентом: Подготовить эссе по одной из тем: 1. Вклад отечественных и зарубежных ученых в создание и развитие теории информации. 2. Понятие информации, как нового предмета труда. Истоки информационных технологий. 3. Информационный взрыв. Информационные ресурсы. Этапы становления промышленной обработки данных. 4. Объекты и явления. Система и процесс. 5. Сигналы и сообщения. Дискретные и непрерывные сообщения. 6. Источники информации: комбинаторные и вероятностные; конечные и бесконечные. Способы получения информации. Учебно-методическая литература: 2, 5, 8 Профессиональные базы данных и информационные справочные системы: 5	8
2. Сжатие информации. Помехоустойчивое кодирование	8
Формируемые компетенции, образовательные результаты: ПК-1: ПК.1.1	
2.1. Передача дискретных сообщений по каналу без помех и с помехами Задание для самостоятельного выполнения студентом: Описать предложенный метод помехоустойчивого кодирования, привести 2 примера: 1) инверсный; 2) Голея; 3) БЧХ; 4) Грэя. Учебно-методическая литература: 4, 8, 9 Профессиональные базы данных и информационные справочные системы: 5	8
3. Основы теории автоматов	12
Формируемые компетенции, образовательные результаты: ПК-1: ПК.1.1	
3.1. Представления о конечных автоматах Задание для самостоятельного выполнения студентом: Задание 1. Построить автомат для грамматики целых чисел без знака. Задание 2. Построить автомат для грамматики простейших арифметических формул. Учебно-методическая литература: 4, 9 Профессиональные базы данных и информационные справочные системы: 5	4
3.2. Распознавание образов и анализ сцен Задание для самостоятельного выполнения студентом: Задание 1. Сформулировать задачу поиска геометрических аналогий. Задание 2. Использовать правила преобразования в задаче поиска геометрических аналогий. Учебно-методическая литература: 4, 6, 7 Профессиональные базы данных и информационные справочные системы: 5	8
4. Динамическое программирование. Алгоритмы на строках	8
Формируемые компетенции, образовательные результаты: ПК-1: ПК.1.1	

<p>4.1. Динамическое программирование</p> <p><i>Задание для самостоятельного выполнения студентом:</i></p> <p>Темы сообщений:</p> <ul style="list-style-type: none"> 1. Наибольшие подпоследовательности. 2. Строки и шаблоны. 3. Расстояние Левенштейна. 4. Алгоритмы замощения. 5. Динамическое программирование по подмножествам. Задача коммивояжёра. 6. Динамическое программирование по дереву. Сбор информации о поддеревьях. 7. Алгоритм Флойда-Уоршелла. 8. Алгоритм Беллмана-Форда. <p>Учебно-методическая литература: 3, 6</p> <p>Профессиональные базы данных и информационные справочные системы: 5</p>	8
--	---

Приложение 3. Рейтинг

№	Фамилия Имя	Тест 1	Текст_ инф лаб	ГСЧ лб	Кол. инф.	Доп.	Модуль1	Тест2	Метод Шеннона- Фено	Код Хафф- мена	Метод Лемпеля- Зива	Коды Хемминг а лаб	Доп.	Модуль2
Тест3	Доп.	Комб_сх емы	Абстр_ синтез лаб	Распозн авание образов лаб	Модуль3	Тест4	Числа Фиб-чи	Алгорит мы поиска	Доп.	Модуль4	Итого	Оценка		

Приложение 4. Оценочные материалы

Задание 1

Время на выполнение задания 1 мин.

Среди перечисленных определений одно не относится к определению понятия «информация»:

- 1) сообщение о состоянии и свойствах объекта, явления, процесса;
- 2) последовательность действий, направленных на достижение цели;
- 3) содержание сигналов, поступающих в кибернетическую систему из окружающей среды, которое может быть использовано для целей управления системой.

Задание 2

Время на выполнение задания 1 мин.

Для преобразования сигналов из непрерывной формы в дискретную используются преобразователи:

- 1) код-аналог;
- 2) аналого-цифровые;
- 3) цифро-аналоговые.

Задание 3

Время на выполнение задания 1 мин.

Определять информационную емкость системы предложил:

- 1) К. Шенон;
- 2) А. Харкевич;
- 3) Р. Хартли;
- 4) Ю. Шрейдер.

Задание 4

Время на выполнение задания 1 мин.

Учитывать вероятностные характеристики источников сообщений предложил:

- 1) К. Шенон;
- 2) А. Харкевич;
- 3) Р. Хартли;
- 4) Ю. Шрейдер.

Задание 5

Время на выполнение задания 1 мин.

Поставить в соответствие подходу его характеристику:

- 1) структурный;
 - 2) статистический;
 - 3) прагматический;
 - 4) семантический.
- a) учет вероятностных характеристик источника сообщений;
 - b) оценка сообщения с точки зрения получения лучшего управленческого решения;
 - c) опирается на тезаурусный подход;
 - d) определение меры количества информации;
 - e) определение информационной емкости сообщения.

Задание 6

Время на выполнение задания 1 мин.

Наибольшую неопределенность среди всех опытов, имеющих n исходов, имеет опыт, у которого исходы:

- 1) зависят от предыдущего опыта;
- 2) неравновероятны;
- 3) равновероятны.

Задание 7

Время на выполнение задания 1 мин.

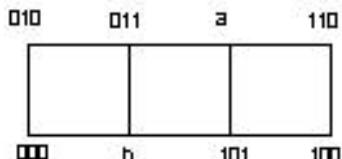
Число разрядов, на которые отличаются любые две кодовые комбинации, называется:

- 1) оптимальным кодом;
- 2) коэффициентом сжатия;
- 3) кодовым расстоянием.

Задание 8

Время на выполнение задания 1 мин.

Трехэлементный двоичный код представлен в виде графа:



Чему равны a и b ?

- 1) $a = 011, b = 001$;
- 2) $a = 111, b = 001$;
- 3) $a = 001, b = 111$.

Задание 9

Время на выполнение задания 1 мин.

Является ли префиксная схема разделимой?

- 1) да;
- 2) нет.

Задание 10

Время на выполнение задания 1 мин.

Автоматом называется система $S = \langle A, B, Q, \delta, \lambda \rangle$. Поставить в соответствие:

- 1) A ;
 - 2) Q ;
 - 3) B ;
 - 4) δ ;
 - 5) λ .
- a) алфавит сообщений;
 - b) алфавит состояний;
 - c) входной алфавит;
 - d) выходной алфавит;
 - e) функция входов;
 - f) функция переходов;
 - g) функция выходов.

Задание 11

Время на выполнение задания 1 мин.

Конечный автомат, имеющий только одно внутреннее состояние – это:

- 1) автомат с памятью;
- 2) автомат без памяти;
- 3) детерминированный автомат.

Задание 12

Время на выполнение задания 1 мин.

Базовым в теории распознавания является понятие:

- 1) алгебры;
- 2) матрицы;
- 3) множества;
- 4) многочлена.

Задание 13

Время на выполнение задания 1 мин.

Методика отнесения элемента к какому-либо образу называется:

- 1) метрикой;
- 2) распознаванием;
- 3) решающим правилом.

Задание 14

Время на выполнение задания 1 мин.

Состояние – это:

- 1) некоторая совокупность состояний сложного объекта, каждая из которых характеризуется одними и теми же или схожими характеристиками объекта;
- 2) определенная форма отображения измеряемых текущих (или мгновенных) характеристик наблюдаемого объекта;
- 3) каждое отображение какого-либо объекта на воспринимающие органы распознающей системы, независимо от его положения относительно этих органов.

Задание 15

Время на выполнение задания 1 мин.

Для распознавания зрительных и слуховых образов используются:

- 1) состояние и ситуации;
- 2) изображения объекта и образы;
- 3) нет верного ответа.

Задание 16

Время на выполнение задания 1 мин.

Среди перечисленных задач одна не относится к задачам распознавания:

- 1) для этих задач трудно строить формальные теории и применять классические математические методы;
- 2) в этих задачах возможна «плохая» информация (информация с пропусками, разнородная и др.);
- 3) в этих задачах нельзя вводить понятие аналогии или подобия объектов.

Задание 17

Поставить в соответствие типу задачи распознавания ее характеристику:

- 1) задача распознавания;
 - 2) задача автоматической классификации.
- а) отнесение предъявленного объекта по его описанию к одному из заданных классов (обучение с учителем);

- b) разбиение множества объектов (ситуаций) по их описаниям на систему непересекающихся классов (таксономия, кластерный анализ, обучение без учителя);
 с) задача, в которой решение должно относиться к некоторому моменту в будущем.

Задание 18

Время на выполнение задания 1 мин.

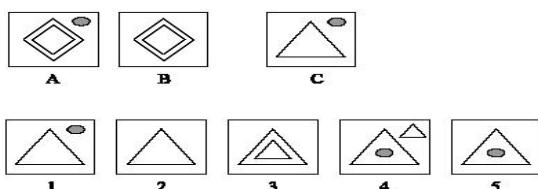
Некоторую совокупность состояний сложного объекта, каждая из которых характеризуется одними и теми же или схожими характеристиками объекта, принято называть:

- 1) изображением;
- 2) ситуацией;
- 3) состоянием.

Задание 19

Время на выполнение задания 1 мин.

Среди фигур второго ряда, представленных на рисунке, требуется выбрать $X \in \{1, 2, 3, 4, 5\}$ такое, что A так относится к B, как C относится к X, и такое, которое лучше всего при этом подходит.



- 1) $X = 2$;
- 2) $X = 3$;
- 3) $X = 5$.

Задание 20

Время на выполнение задания 1 мин.

Алгоритмический метод решения задачи оптимизации путем разбиения ее на более простые подзадачи и использования того факта, что оптимальное решение общей проблемы зависит от оптимального решения ее подзадач – это:

- 1) динамическое программирование;
- 2) модульное программирование;
- 3) объектно-ориентированное программирование.

Задание 21

Время на выполнение задания 1 мин.

Как взаимодействуют элементы в условиях динамического программирования?

- 1) должны находиться в зависимости друг от друга;
- 2) должны не зависеть друг от друга;
- 3) допустимы оба случая.

Задание 22

Время на выполнение задания 1 мин.

Восстановить правильную последовательность поиска решения задачи с помощью динамического программирования:

- 1) определение класса подзадач;
- 2) (указание рекуррентного соотношения, основанного на решении каждой подзадачи с точки зрения более простых подзадач;

3) указание алгоритма вычисления рекуррентного соотношения.

Задание 23

Время на выполнение задания 1 мин.

Поставить в соответствие технике реализации ДП ее описание:

- 1) табуляция;
- 2) мемоизация.

а) это подход «снизу вверх» в динамическом программировании, при котором решения подзадач итеративно заносятся в таблицу;
б) это подход «сверху вниз» в динамическом программировании, при котором решения подзадач рекурсивно вычисляются и сохраняются для использования в будущем;
в) это подход «сверху вниз» в, при котором решения подзадач итеративно заносятся в таблицу.

Задание 24

Время на выполнение задания 1 мин.

Алгоритм позволяет найти кратчайшие пути из одной вершины графа до всех остальных, даже с отрицательными весами ребер. Это –

- 1) алгоритм Флойда-Уоршелла;
- 2) алгоритм Беллмана-Форда.

Задание 25

Время на выполнение задания 1 мин.

Среди перечисленных задач выбрать задачу, относящуюся к одномерному динамическому программированию.

- 1) задача о рюкзаке;
- 2) задача о кузнечике;
- 3) задача о максимальном потоке.

Задания открытого типа

Задание 26

Время на выполнение задания 5 мин.

Как вычисляется количество информации по К. Шеннону?

Задание 27

Время на выполнение задания 5 мин.

Дать определение понятия «информация» с точки зрения кибернетики.

Задание 28

Время на выполнение задания 5 мин.

Кому принадлежит идея учета «запаса знаний»?

Задание 29

Время на выполнение задания 10 мин.

Пусть алфавит содержит четыре символа и их вероятности равны соответственно $p_1 = p_2 = p_3 = p_4 = 0,25$. Чему будет равна неопределенность (энтропия) H ?

Задание 30

Время на выполнение задания 5 мин.

Количество какой информации может быть вычислено по формуле $I = \log \frac{p_1}{p_0}$, если p_1 – вероятность достижения цели после получения информации о событии, p_0 – вероятность достижения цели до получения информации о событии?

Задание 31

Время на выполнение задания 5 мин.

Какое соотношение относительно кодового расстояния имеет место избыточных кодах?

Задание 32

Время на выполнение задания 5 мин.

Кодовое расстояние $d = 3$. Выбрать свойство кода.

Задание 33

Время на выполнение задания 15 мин.

Пусть $A = \{a, b\}$, $B = \{0, 1\}$ и $\delta = \langle a \rightarrow 010, b \rightarrow 01 \rangle$. $I_\delta(P) = \sum_{i=1}^n P_i l_i$. Чему будет равна средняя длина кодирования δ , если $P = \langle 0.2, 0.8 \rangle$?

Задание 34

Время на выполнение задания 15 мин.

Распределение вероятностей и код Хаффмена представлены в таблице:

P_i	код
0.1	10
0.2	11
0.18	00
0.17	001
0.15	010
0.15	0110
0.01	0111

Чему равно $I_\delta(P)$?

Задание 35

Время на выполнение задания 15 мин.

Пусть вероятности появления букв в сообщении равны $p_1 = 0.125$, $p_2 = 0.5$, $p_3 = 0.125$, $p_4 = 0.25$, $C = 10000$ бит/с. Чему равна максимальная скорость передачи сообщения?

Задание 36

Время на выполнение задания 10 мин.

Сжать последовательность $abca bca bc$ методом LZ78.

Задание 37

Время на выполнение задания 10 мин.

Передан код числа 8 в виде «1 0 0 1 0 1 1», а приняли в виде «1 0 1 1 0 1 1». Обнаружить позицию ошибки.

Задание 38

Время на выполнение задания 10 мин.

Передан код 0101100, принят код 0111100. Подсчитать дополнительный код. Обнаружить позицию ошибки.

Задание 39

Время на выполнение задания 15 мин.

Передан код 1100000, принят код 1101000. Подсчитать дополнительный код. Обнаружить позицию ошибки.

Задание 40

Время на выполнение задания 15 мин.

Упростить логическое выражение $(X \vee \bar{Y}) \cdot \bar{Z} \vee \bar{Y} \cdot (Z \vee Y)$. Построить схему из логических элементов.

Задание 41

Время на выполнение задания 25 мин.

Упростить логическое выражение $(C \supset A) \supset (\neg(B \vee C) \supset A)$. Построить схему из логических элементов.

Задание 42

Время на выполнение задания 25 мин.

Выполнить следующие действия, построив дополнительный код (использовать 8-разрядный код): 47 – 23; 23 – 47.

Задание 43

Время на выполнение задания 25 мин.

Выполнить следующие действия, построив дополнительный код (использовать 8-разрядный код): 42 – 31; 31 – 42.

Задание 44

Время на выполнение задания 25 мин.

Привести пример задания на распознавание образов по аналогии.

Задание 45

Время на выполнение задания 5 мин.

Динамическое программирование – это ...

Задание 46

Время на выполнение задания 5 мин.

В чем отличие динамичного программирования от использования рекурсивных функций?

Задание 47

Время на выполнение задания 5 мин.

Дать определение одномерного динамического программирования.

Задание 48

Время на выполнение задания 5 мин.

Дать определение двумерного динамического программирования.

Задание 49

Время на выполнение задания 25 мин.

Какие существуют техники реализации динамического программирования.

Задание 50

Время на выполнение задания 25 мин.

Перечислить методы решения рекурсивных задач.

Критерии шкалы оценивания

Код	Содержание компетенции		
Уровни освоения компетенции	Основные признаки выделения уровня (критерии оценки)	Пятибалльная шкала (академическая) оценка	% освоения
Высокий (продвинутый)	<p>Знает: общие проблемы и задачи теоретической информатики; сущность теории автоматов; принципы динамического программирования.</p> <p>Свободно демонстрирует умение: применять алгоритмы для кодирования информации; строить схемы автоматов; осуществлять поиск решения задачи с помощью динамического программирования.</p> <p>Свободно владеет: методами кодирования информации; способами построения автоматов; методами поиска подстроки.</p>	Отлично	86-100
Средний (оптимальный)	<p>Знает: общие проблемы и задачи теоретической информатики; сущность теории автоматов; принципы динамического программирования.</p> <p><i>И допускает незначительные ошибки.</i></p> <p>Демонстрирует умение: применять алгоритмы для кодирования информации; строить схемы автоматов; осуществлять поиск решения задачи с помощью динамического программирования.</p> <p><i>Уверенно владеет:</i> методами кодирования информации; способами построения автоматов; методами поиска подстроки.</p> <p><i>Но допускает незначительные ошибки</i></p>	Хорошо	61-85
Пороговый	<p>Знает: общие проблемы и задачи теоретической информатики; сущность теории автоматов; принципы</p>	Удовлетворительно	41-60

	<p>динамического программирования. <i>Не демонстрирует глубокого понимания материала.</i> <i>В основном демонстрирует</i> умение: применять алгоритмы для кодирования информации; строить схемы автоматов; осуществлять поиск решения задачи с помощью динамического программирования. <i>Владеет:</i> методами кодирования информации; способами построения автоматов; методами поиска подстроки. <i>Но допускает ошибки.</i></p>		
Недостаточный	Отсутствие признаков удовлетворительного уровня	Неудовлетворительно	40 и менее

Приложение 5. Пример текста для анализа

Незаменимость ручного анализа

Современные технологии позволяют накапливать значительные объемы информации. Но если для работы со структурированной информацией есть достаточно известных и доступных инструментов, то обработку текстовой информации приходится осуществлять вручную. Да есть масса поисковых систем, но все они способны только искать заданный набор символов и не проводить смысловой анализ. И это вполне понятно – слишком велико количество возможных вариантов и оттенков, которые дает человеческая речь, в том числе и письменная. Поэтому основным инструментом при анализе текстовой информации является голова аналитика. А важной составляющей – способность анализировать и знание технологий анализа. Тем технологиям, которые требуются в первую очередь для работы подразделений бизнес разведки на современных предприятиях. В настоящее время в Интернете есть достаточно информации о программном обеспечении для аналитика. Сама сущность Интернета – это работа с информацией, а его поисковые системы ни что иное как софт по поиску нужной информации. Например, на ресурсе www.it2b.ru можно найти описание современного софта для обработки информации, на форуме этого же сайта можно пообщаться с практиками и теоретиками бизнес разведки, в новостном разделе можно почерпнуть информацию о последних событиях в бизнес разведке. Еще один интересный ресурс посвящен собственно анализу информации www.prologistic.ru – здесь много полезной информации о самом анализе. И везде указывается на центральную роль человека в процессе анализа. А посему способности и знания аналитика являются первостепенным и главным условием успешного анализа информации.

Учебное издание

Поднебесова Галина Борисовна

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

Часть 2

Учебно-практическое пособие

ISBN 978-5-93162-895-0 (общий)

ISBN 978-5-93162-925-4 (часть 2)

Издательство ЗАО «Библиотека А. Миллера»

454091, г. Челябинск, ул. Свободы, 159

Подписано в печать 01. 12. 2025. Формат 60x84/16 Бумага офсетная.

Объем 6,00 усл. печ. л. Тираж 100 экз. Заказ № 538.

Отпечатано с готового оригинал-макета в типографии ЮУрГГПУ

454080, Челябинск, пр. Ленина, 69