Теоретические Основы Информатики

Лабораторный практикум

Министерство образования и науки РФ
ФГБОУ ВО «Уральский государственный
педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информационно-коммуникационных технологий в образовании

Б.Е. Стариченко

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по курсу «Теоретические основы информатики»

Екатеринбург 2018

УДК 378.016:004(075.8)
ББК 3973я7
С77

рекомендовано Ученым советом федерального государственного бюджетного образовательного учреждения высшего образования «Уральский государственный педагогический университет» в качестве *учебного* издания (Решение № 5 от 09.02.2018)

РЕЦЕНЗЕНТЫ:
доктор физико-математических наук, профессор А.Н. Сесекин
кандидат педагогических наук, доцент И.Н. Слинкина

Стариченко, Б. Е.
С77 Лабораторный практикум по курсу «Теоретические основы информатики» [Электронный ресурс] : учебное пособие / Б. Е. Стариченко ; Урал. гос. пед. ун-т. – Электрон. дан. – Екатеринбург : [б. и.], 2018. – 1 электрон. опт. диск (CD-ROM).

ISBN 978-5-7186-0984-4

Пособие является частью учебно-методического комплекса для решения задач по дисциплине «Теоретические основы информатики», входящей в план подготовки студентов информационных специальностей педагогического вуза. Помимо описаний лабораторных работ, представленных в данном пособии, комплекс включает рабочие файлы, содержащие необходимые программы моделирования, а также экранные формы отчетов с генерацией индивидуальных заданий и проверкой корректности выполнения работ (в формате MS Excel); для выполнения работ требуется пакет MS Office.

УДК 378.016:004(075.8)
ББК 3973я

ISBN 978-5-7186-0984-4

Оглавление

Вводные замечания

Основной целью изучения дисциплины «Теоретические основы информатики» в педагогическом вузе является освоение будущими учителями информатики и специалистами IT-сферы базовых положений теории информации как теоретической и методологической основы других дисциплин информационно-технологической подготовки. Помимо знакомства с теоретическими положениями программа дисциплины предусматривает также освоение методов решения задач, связанных с представлением и обработкой дискретной информации. В настоящем пособии решение этих задач предлагается выполнять в форме интерактивных компьютерных лабораторных работ.

В содержательном отношении практикум базируется на учебнике «Теоретические основы информатики»[1]. Он включает работы по всем основным разделам курса: теория информация К. Шеннона, кодирование символьной информации, кодирование и представление в компьютере числовой информации, передача информации, конечные автоматы, элементы криптографии. Перечень и последовательность выполняемых работ, а также конкретное содержание учебных заданий определяет преподаватель в зависимости от профиля специальности и объема учебных часов, выделенных учебным планом на изучение дисциплины.

Описания лабораторных работ, приведенные в данном пособии, включают формулировку учебного задания, рекомендации по его выполнению, указания по форме отчета, перечень заданий для самостоятельной работы и контрольные вопросы. Помимо описания для выполнения заданий требуются рабочие файлы – их перечень указан в начале каждого описания; перед выполнением работы студент должен скопировать нужные файлы в свою рабочую папку.

Предполагается, что часть заданий студент выполняет самостоятельно. Отчет о работе делается по предложенной экранной форме и представляется преподавателю в виде файла на электронном носителе (через электронную почту или облачную образовательную среду). В ряде работ предусмотрена автоматическая проверка корректности вычислений, что обеспечивает студенту дополнительные удобства в процессе самостоятельного выполнения заданий.

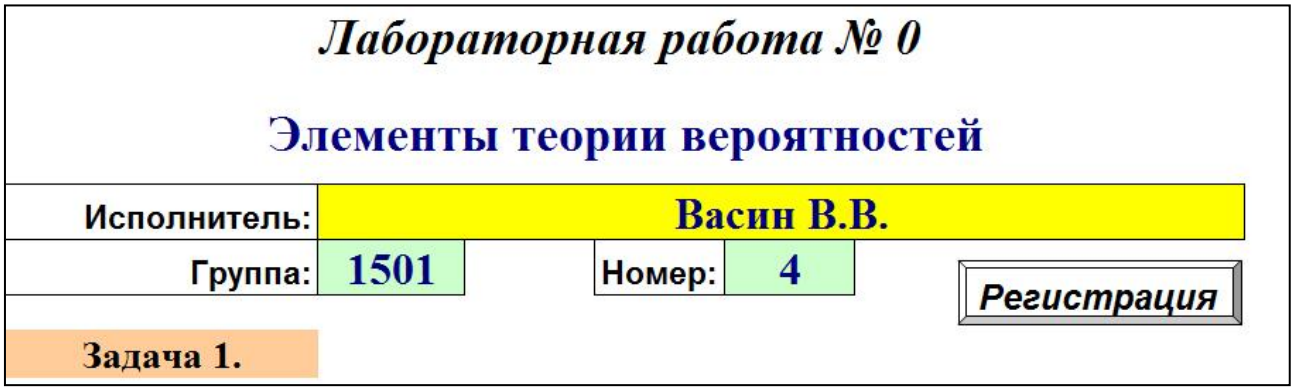
Автор будет весьма признателен, если отзывы по материалам практикума и предложения по его совершенствованию Вы направите по адресу: 620075, РОССИЯ, г. Екатеринбург, ул. К. Либкнехта, 9. Уральский государственный педагогический университет, Институт МФИиТ, кафедра ИКТО. Mail: b.e.starichenko@uspu.su

Б.Е. Стариченко

[1] Стариченко Б.Е. Теоретические основы информатики: учебник для вузов. – М.: Изд-во Горячая линия-Телеком, – 2014. – 367 с.

Общий порядок выполнения работ практикума

1. Для выполнения работ практикума студенты используют экранные формы, которые содержат задания и одновременно служат заготовками отчетов. Студент загружает форму, выполняет в ней работу, производит необходимые расчеты, оформляет и сдает файл на проверку преподавателю.
2. Настоящий практикум применяется при изучении курса «Теоретические основы информатики» в течении несколько лет. При этом его работы постоянно совершенствуются и изменяются. По указанной причине в любом текущем календарном году должны использоваться только те версии файлов лабораторных работ, в названии которых присутствует именно этот год, например, *ТОИ_ЛР-2-2018.xls.*
3. Для выполнения работы нужные файлы копируются в папку на локальном компьютере. Место расположения исходных файлов указывает преподаватель (сетевой диск, облако дисциплины). Работать с файлами можно в любых версиях MS Office, начиная с v. 2003. Использование сетевых версий электронных таблиц (например, Google-Таблицы), а также таблиц Open Office невозможно, поскольку ими не поддерживаются макросы и VBA-фрагменты, которые используются в работах практикума.
4. При открытии рабочего файла в приложении (MS Excel или MS Word) НЕ СЛЕДУЕТ отключать макросы; если они отключены настройками приложения – нужно их подключить.
5. Сразу после открытия файла необходимо зарегистрироваться, введя свою фамилию, код группы и свой порядковый номер в списке группы в соответствующие поля (см. рис. 1). После этого следует нажать экранную клавишу [**Регистрация**].



После нажатия [**Регистрация**] произойдет следующее:

- файл с отчетом по работе будет сохранен с названием, включающем регистрационные данные (фамилию), в ту папку, из которой он был открыт; в дальнейшем именно с ним нужно будет производить операции и затем сдавать его на проверку;
- клавиша [**Регистрация**] исчезнет с экрана – повторная регистрация из данного файла уже невозможна; при необходимости следует открыть исходный файл и повторить начальные операции;

- будет сформирован и выведен на экран индивидуальный вариант заданий лабораторной работы и персональные данные к ней; вариант формируется случайным образом, поэтому при повторной регистрации данные окажутся другими;
- в конце листа с заготовками отчета на желтом поле выводится номер контрольного теоретического вопроса и сам вопрос, на который должен ответить студент; по номеру он соответствует вопросу из описания лабораторной работы в настоящем пособии;
- в определенные (и, естественно, недоступные для студента) места записываются индивидуальные метки, по которым преподаватель может контролировать авторство представленного отчета; изменение регистрационных данных после процедуры регистрации не изменяет меток.

6. Если работа состоит из нескольких частей, размещенных на разных листах электронной таблицы, *регистрация производится только на первом листе*, а на остальные регистрационные данные просто копируются.
7. Помимо правильного выполнения заданий работы, обязательными являются следующие элементы отчета: содержательные выводы и ответ на контрольный вопрос; их отсутствие приводит к снижению оценки за работу.
8. Взаимодействие с преподавателем – представление отчета, получение оценки, возможность задать вопрос или направить сообщение и пр. – осуществляется в рамках электронной информационной образовательной среды дисциплины, в которую открывается доступ студенту в начале изучения курса. Порядок взаимодействия, а также все организационные моменты: график выполнения работ, сроки представления отчетов, критерии оценивания, система штрафных баллов – доводятся до сведения студента.
9. Как правило, на подготовку и сдачу отчета по лабораторной работе, выполняемой в аудитории, для студентов очной формы отводится 2 недели с даты ее выполнения; опоздание со сдачей влечет снижение общей оценки за работу. Для работ, которые вынесены на самостоятельное выполнение, ограничений по времени сдачи не устанавливается.
10. Для получения зачета по практикуму студент должен обеспечить долю его выполнения не ниже некоторой границы, установленной преподавателем. Как правило, это требует выполнения всех включенных в практикум работ.

Лабораторная работа № 0.
Элементы теории вероятностей

***Учебная задача*:**

Освоить основные соотношения теории вероятностей, необходимые для использования при дальнейшем изучении курса «Теоретические основы информатики».

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Приложение А.

Рабочий файл: **ТОИ_Лр-0.xls**

Необходимые теоретические сведения

Расчетные формулы

6. Вероятность одного (любого) из *m* благоприятных исходов при общем числе *равновероятных исходов n*:

$$p = \frac{m}{n} \qquad (0.1)$$

7. Условие нормировки вероятностей при *n* исходах:

$$\sum_{i=1}^{n} p_i = 1 \quad , \qquad (0.2)$$

где p_i – вероятность *i*-го исхода.

8. *Дополнительные вероятности*: если возможны только два исхода (*A* и *B* = $\bar{A}$), то

$$p(B) = p(\bar{A}) = 1 - p(A) \qquad (0.3)$$

9. Среднее значение набора дискретных величин x_1, $x_2 \ldots x_n$, вероятности появления которых, соответственно, p_1, $p_2 \ldots p_n$, равно:

$$< x >= \sum_{i=1}^{n} p_i \cdot x_i \qquad (0.4)$$

10. Вероятность (*p*) *совместного* наступления *k независимых* событий с вероятностями p_1, $p_2 \ldots p_k$ (*умножение* вероятностей):

$$p = p_1 \cdot p_2 \cdot \ldots \cdot p_k = \prod_{i=1}^{k} p_i \qquad (0.5)$$

11. Вероятность какого-либо одного (любого) из нескольких благоприятных *независимых и несовместных* событий *A, B, , K* равна (*сложение* вероятностей):

$$p(A \vee B \vee ... \vee K) = p(A) + p(B) + ... + p(K) \quad (0.6)$$

12. Вероятность события *B* при условии, что имело место влияющее на него событие *A*, называется *условной вероятностью*. Обозначается $p_A(B)$.
13. Правило *умножения* вероятностей, если события *A* и *B* *не являются независимыми*:

$$p(A \wedge B) = p(A) \cdot p(B) - p_A(B) \quad (0.7)$$

14. Общее правило *сложения* вероятностей:

$$p(A \vee B) = p(A) + p(B) - p(A) \cdot p_A(B) \quad (0.8)$$

15. Формула *полной вероятности*:

$$p(A) = \sum_{i=1}^{n} p(B_i) \cdot p_{B_i}(A) \quad (0.9)$$

Порядок выполнения работы

1) Скопируйте в свою рабочую папку файл **ТОИ_Лр-0.xls** и откройте его. Если требуется, установите **средний** уровень безопасности макросов (меню ***Сервис→****Макрос→Безопасность*). Зарегистрируйтесь на листе <**Теория вероятностей**>.
2) Для выполнения *Заданий 1-3* необходимо воспользоваться формулами (0.1), (0.3), (0.5).
3) Решение *Задания 4* подразумевает использование формулы (0.4), а для проверки промежуточных расчетов (0.2).
4) *Задание 5*, в котором требуется произвести доказательства, выполняется в рабочих тетрадях (конспектах); вывод может быть помещен в отчет с помощью редактора формул или вставкой картинки (фото) с решением.
5) *Задания 6-8* предполагают использование формул (0.1) и (0.5)-(0.9).
6) В *Заданиях 9* и *10* соотношения теории вероятностей применяются для анализа текстов.
7) В отчете после заданий необходимо дать ответ на теоретический вопрос. Список вопросов приведен ниже.
8) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Почему в определении вероятности количество попыток $N \to \infty$? Зависит ли вероятность случайного события от числа проведенных однотипных опытов, в которых оно проявляется? Почему?

2. Какие исходы случайного события названы *равновероятными*? Как обосновывается *равновероятность* при решении практических задач (например, что вероятности выпадения «орел» и «решка» одинаковы и равны 0,5)?
3. Дайте определения *дополнительным случайным событиям*; приведите примеры. Получите формулу, связывающую их вероятности.
4. Какие случайные события называются *независимыми*? *Совместными*? *Несовместными*? К какой из этих категорий следует отнести дополнительные события?
5. Выполняется ли условие нормировки вероятностей (0.2), если исходы случайного события не равновероятны? Ответ обоснуйте.
6. В каком случае среднее значение можно находить по формуле (0.4)?
7. Верно ли, что чем больше требований мы выдвигаем при выборе чего-либо (вещи, работы, партнера по жизни и т.п.), тем меньше вероятность найти подходящий вариант? Ответ обоснуйте.
8. В чем отличие *условной* и *безусловной* вероятностей? Можно ли считать, что для независимых событий они совпадают?
9. Верно ли, что условная вероятность может быть как больше, так и меньше безусловной? Приведите примеры.
10. Мы наугад открываем страницу книги и наугад выбираем строку и номер буквы в ней. Можно ли считать случайным событием выбор конкретной буквы? Одинаковы ли вероятности «наткнуться» таким образом на различные буквы русского алфавита? Ответ обоснуйте.

Лабораторная работа № 1.
Энтропия. Информация.

***Учебная задача*:**

Научиться оценивать энтропию и количество информации в опытах со случайными исходами.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Глава 2.

Рабочий файл: **ТОИ_Лр-1.xls**

Необходимые теоретические сведения

Расчетные формулы

1. Энтропия опыта с *n равновероятными исходами*

$$H = \log_2 n \ , \qquad (1.1)$$

где n – количество равновероятных исходов.

2. Энтропия опыта с *n не равновероятными исходами*

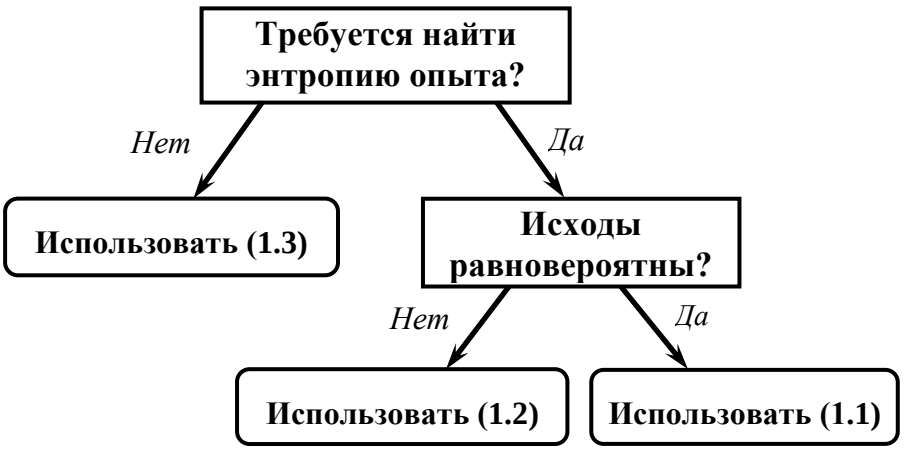
$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \ , \qquad (1.2)$$

где p_i – вероятность i-го исхода.

3. Энтропия *отдельного исхода* i:

$$H_i = -\log_2 p_i \ , \qquad (1.3)$$

Общий алгоритм анализа хода решения задач:



Формулы (1.1)-(1.3), а также алгоритм могут быть использованы и при решении задач, связанных *с нахождением количества информации*.

Выборочный каскад

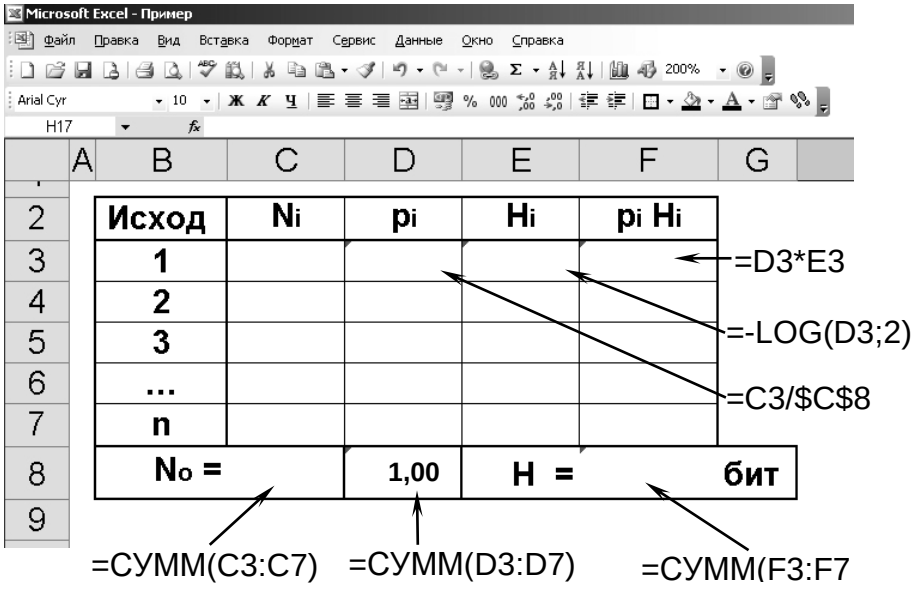
Число вопросов (k) с бинарными равновероятными ответами («да» – «нет») («выборочный каскад»), которые необходимо задать, для того чтобы полностью снять неопределенность опыта (ситуации) с n равновероятными исходами:

$$k \geq \log_2 n \qquad (1.4)$$

Поскольку количество вопросов выражается целым числом, при использовании формулы (1.4) k следует считать *равным ближайшему целому числу, большему логарифма*.

Обобщенная расчетная схема

Перед началом работы рекомендуется создать на странице MS Excel лабораторной работы следующую обобщенную схему расчетов:



В общем случае рассматривается опыт с n не равновероятными исходами. В приведенной схеме в колонке N_i указывается (вводится) количество состояний, связанных с i исходом. Суммирование N_i в ячейке **C8** позволяет определить общее число исходов N_0. p_i – вероятность исхода i (обратите внимание, в ячейке **D8** найдена сумма вероятностей всех исходов; в случае правильных вычислений она равна 1 по условию нормировки суммы вероятностей). В колонке H_i – рассчитывается энтропия каждого из исходов по формуле (1.3). В колонке $p_i \cdot H_i$ рассчитываются «весовые» энтропии исходов. В ячейке **F8** по формуле (1.2) находится энтропия опыта.

При решении задач, связанных с определением количества информации, можно воспользоваться тем же алгоритмом и той же расчетной схемой, заменив в формулах (1.1) – (1.3) и в схеме H на I.

Примеры решения задач

Задача 1. a) Чему равна энтропия, связанная с выпадением цифры «6» при броске кубика? b) Чему равна энтропия броска, если выпала цифра «6»?

Решение:

a) Согласно алгоритму:

Вопрос 1: «Требуется найти энтропию опыта?»

Ответ 1: «Нет» (поскольку речь идет об энтропии отдельного исхода)

Следовательно, выбирается формула (1.3). Для того чтобы ею воспользоваться, необходимо найти вероятность исхода; поскольку исходы равновероятны и $n = 6$,

$$p = \frac{1}{n} = \frac{1}{6}$$

Следовательно, энтропия исхода

$H_6 = -\log_2 p = \log_2 6 = 2{,}585$ *бит*

b) Согласно алгоритму:

Вопрос 1: «Требуется найти энтропию опыта»?

Ответ 1: «Да»

Вопрос 2: «Исходы равновероятны?»

Ответ 2: «Нет»

Следовательно, выбирается формула (1.2). Для того чтобы ею воспользоваться, необходимо найти вероятности всех исходов и применить обобщенную схемуй решения:

Исход	$\mathbf{N_i}$	$\mathbf{p_i}$	$\mathbf{H_i}$	$\mathbf{p_i H_i}$
«1»-«5»	5	0,833	0,263	0,219
«6»	1	0,167	2,585	0,431
N_o=	6	1,000	**H_0=**	**0,650**

Таким образом, энтропия броска (опыта) оказывается равной $H_0 = 0{,}650$ бит.

Задача 2. В корзине находится 25 шаров: 3 – красных, 8 – белых, 9 – желтых, 5 – синих. Опыт состоит в извлечении случайным образом одного шара. Какое количество информации связано с исходом опыта?

Решение:

Поскольку речь идет об информации, связанной *с опытом*, исходы которого *не равновероятны*, воспользуемся расчетной схемой в Excel.

Исход	$\mathbf{N_i}$	$\mathbf{p_i}$	$\mathbf{I_i}$	$\mathbf{p_i I_i}$
красный	3	0,12	3,059	0,367
белый	8	0,32	1,644	0,526
желтый	9	0,36	1,474	0,531
синий	5	0,20	2,322	0,464
N_o=	25	1,00	**I_0=**	**1,888**

Таким образом, с извлечением шара связано $I_0 \approx 1{,}89$ бит информации.

Задача 3. Сколько вопросов с ответами «Да»-«Нет» необходимо задать человеку, чтобы узнать месяц его рождения? Какова оптимальная последовательность вопросов?

Решение:

Количество возможных исходов $n = 12$; при условии, что рождение человека может произойти в любом месяце, их следует считать равновероятными – это позволяет воспользоваться формулой (1.4). $\log_2 12 = 3{,}58$; поскольку $k \geq \log_2 12 = 4$.

Следовательно, для получения гарантированного ответа требуется задать 4 вопроса. Однако, если многократно производить такое «отгадывание», то примерно в половине случаев ответ будет получен за 3 вопроса, поскольку значение логарифма близко к 3,5.

Оптимальной следует считать организацию процедуры отгадывания по схеме выборочного каскада (половинного деления).

Порядок выполнения работы

1) Скопируйте в свою рабочую папку файл **ТОИ_Лр-1.xls** и откройте его. Если требуется, установите **средний** уровень безопасности макросов (меню ***Сервис***→*Макрос*→*Безопасность*). Зарегистрируйтесь на листе <**Энтропия**>. На листе <**Общ_сх>** реализуйте приведенную выше расчетную схему для 4-6 элементов (исходов) (например, для рассмотренной ранее Задачи 2).
2) Решение задач, в которых требуется определить энтропию или количество информации, следует начинать с анализа условия по представленному выше алгоритму.
3) Решите задачи, приведенные на страницах <**Энтропия>** и <**Информация**>. При необходимости таблицу со схемой решения копируйте с листа <**Общ_сх>.**
4) Задания, в которых требуется произвести доказательства, выполняются в рабочих тетрадях (конспектах); они могут быть помещены в отчет с помощью редактора формул или вставкой картинки с решением.
5) При выполнении задания Э5 исходная функция должна быть записана аналитически, представлена в табличном виде (протабулирована) с 20-25 промежуточными значениями, по которым и должен быть построен график.
6) Не забудьте привести ответ на теоретический вопрос, который выведен после задачи Э5 на листе <**Энтропия**>. Список вопросов приведен ниже.
7) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Почему в определении энтропии как меры неопределенности выбрана логарифмическая зависимость межу H и n? Почему выбран логарифм по основанию 2?
2. Поясните, можно ли формулу (1.2) интерпретировать следующим образом: *энтропия опыта с не равновероятными исходами равна средней энтропии всех его исходов*. Почему?
3. Обоснуйте предложенную в работе схему решения задач.
4. Почему в формуле (1.4) используется знак «≥»? При каких условиях его можно заменить знаком «=»?
5. Прокомментируйте результат решения задачи Э2: почему более меткий стрелок оказывается менее надежным?
6. Объясните, почему максимум энтропии опыта в задаче Э5 приходится на $p = 0,5$?
7. В задаче И2 получилось, что с отгадыванием двузначного числа по цифрам и того же числа в целом связано одинаковое количество информации. Однако, число вопросов с ответами «да»-«нет» в этих двух методах отгадывания нужно задать разное. Почему?
8. Прокомментируйте результат решения задачи И5.
9. Почему в задаче И8 различаются средние информации на знак русского и английского алфавита и для русского она выше?
10. Прокомментируйте результат решения задачи И10: что означает различие избыточностей сравниваемых языков?

Теоретические Основы Информатики

Лабораторный практикум

Лабораторная работа № 2.

Исследование статистических характеристик текста

***Учебная задача*:**

Определить некоторые статистические характеристики сгенерированного текста. Сделать заключение о его характере.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Глава 2. п. 2.3.

Рабочие файлы: **ТОИ_Лр-2.xls, Generator.doc**

Необходимые теоретические сведения

Определения:

Сообщения, в которых вероятность появления любого от дельного знака алфавита в любом месте сообщения остается неизменной и не зависит от того, какие знаки предшествовали данному, называются **шенноновскими**.

Сообщения, в которых существуют статистические связи (корреляции) между знаками или их сочетаниями, называются **сообщениями с памятью**.

Другими словами, размещение знаков в шенноновских сообщениях являются событиями число случайными, независимыми друг от друга. В сообщениях с памятью из-за корреляций размещение знаков не является чисто случайным.

Математические основания

Для того чтобы проверить, является ли некоторое сообщение **шенноновским** или **с памятью**, необходимо произвести частотное исследование текста и определить вероятности (относительные частоты) каждого из N знаков алфавита p_i $(i = 1...N)$, а также вероятности всех возможных сочетаний двух букв p_{ik} $(i,k = 1...N)$ (при учете только двухбуквенных корреляций). Если для всех пар ik появление в тексте знака a_i не оказывает влияния на другое появление после него знака a_k (т.е. эти случайные события являются *независимыми*), то, согласно формулам теории вероятностей, вероятность появления такой пары должна быть равна:

$$p_{ik}^{(нез)} = p_i p_k \qquad (2.1)$$

Критерием того, что сообщение является шенноновским, очевидно будет выполнение соотношения $p_{ik} \approx p_{ik}^{(нез)}$ *для всех пар* ik. Равенства могут не быть строгими из-за конечности длины текста (допустима разница в несколько тысячных).

Заметные отличия этих двух величин (0,01 и более) хотя бы для некоторых пар ik будут свидетельствовать о несправедливости модели

независимых знаков в сообщении. Это, в свою очередь, означает, что между знаками имеются связи (корреляции) и появление (или непоявление) в тексте какого-то знака оказывает влияние на вероятность появления последующего знака. Т.е. сообщение следует считать *сообщением с памятью*.

Другим критерием является отношение средних информаций на знак однобуквенного (исходного) алфавита и *двухбуквенного*, составленного из всех пар $a_i a_k$. Очевидно, если в исходном алфавите n знаков, то различных пар из них может быть образовано $N = n^2$, следовательно, именно столько знаков будет содержать двухбуквенный алфавит.

Для шенноновских сообщений среднее количество информации на знак двухбуквенного алфавита будет ровно в 2 раза больше, чем у однобуквенного, поскольку знаки независимы. Для сообщений с памятью из-за существования корреляций, т.е. элементов упорядочения, это отношение будет меньше 2.

Таким образом, для выявления характера текста требуется определить *вероятности* всех входящих в него знаков, а также вероятности всех возможных парных сочетаний знаков. Для этого, в свою очередь, требуется найти *количество вхождений* каждого знака в текст и количество вхождений всех парных комбинаций знаков – в этом состоит статистическое исследование текста.

Поскольку закономерности имеют статистический характер, для большей надежности результата исследование требуется производить с текстом значительной длины – в нашем случае – несколько тысяч (или десятков тысяч) знаков.

Порядок выполнения работы

1) Скопируйте в свою рабочую папку файлы **ТОИ_Лр-2.xls** и **Generator.doc**, откройте их. Если требуется, установите **средний** уровень безопасности макросов (меню ***Сервис→****Макрос→Безопасность*) и в MS Word, и MS Excel. Зарегистрируйтесь на листе <**Энтропия**>.
2) В документе **Generator** в правое окно введите любые 3 символа (первичный алфавит), в левое – объем текста (15000-20000 знаков); нажмите экранную клавишу [**Генерация**]. Сохраните сгенерировавнный текст в рабочую папку, добавив в имя файла свою фамилию.
3) Используя функции поиска и замены текстового редактора MS Word, найдите количество вхождения каждого из трех символов в тексте. Результаты занесите в табл. 1 отчета.
4) Дополните табл. 1, определите среднюю информацию на знак исходного алфавита I_{1u}.
5) В *Задании 2* введите все возможные двухбуквенные сочетания в табл. 2. Производя анализ текста, определите вероятности появления всех двухбуквенных сочетаний (P_{ij}), результаты занесите в табл. 2 (*следует принять, что* ***соседом последнего знака является первый знак****, т.е. общее число двухбуквенных сочетаний должно совпадать с количеством знаков в исходном тексте*).

6) По данным табл. 1 рассчитайте $(P_{ij})_{нез}$, найдите разности с P_{ij}. Сделайте заключение о характере текста (шенноновский, с памятью) по этому критерию.
7) Считая двухбуквенные сочетания знаками нового алфавита, дополните табл. 2 и определите среднюю информацию на знак этого алфавита ($I_{1д}$). Найдите отношение $I_{1д}$ и $I_{1и}$. Сделайте заключение о характере текста (шенноновский, с памятью) по этому критерию.
8) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
9) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Постройте математическое доказательство того, что для шенноновских сообщений (текстов) отношение средних информаций на знак одно- и двухбуквенных алфавитов всегда равно 2.
2. Почему в сообщениях с памятью отношение средних информаций на знак одно- и двухбуквенных алфавитов меньше 2?
3. В лабораторной работе использовался текст длиной до 20000 знаков при количестве знаков первичного алфавита 3. Поясните, имеется ли связь между числом знаков в алфавите и длиной анализируемого текста? Почему?
4. В лабораторной работе использовался текст длиной до 20000 знаков при количестве знаков первичного алфавита 3. Имеет ли смысл увеличивать эту длину, например, до сотен тысяч знаков? Почему?
5. Что изменится в выполнении работы, если исходный алфавит текста будет содержать иное количество знаков (например, 4).
6. Что изменится в расчетах, если потребуется учесть трехбуквенные сочетания?
7. Почему при выполнении *Задания 2* возникает проблема при подсчете числа парных сочетаний из одинаковых знаков?
8. При выполнении *Задания 2* возникает проблема при подсчете числа парных сочетаний из одинаковых знаков. Каким образом она решается и почему?
9. Почему при совпадении значений P_{ij} и $(P_{ij})_{нез}$ для всех пар *ik* сообщение считается шенноновским?
10. Можно ли предложенный в работе алгоритм проверки характера сообщения (текста) применить к текстам на естественном языке (например, русском)? Если «да», то при каких условиях? Если «нет», то почему?

Лабораторная работа № 3.
Сравнение методов кодирования информации

***Учебная задача*:**

Освоить различные методы кодирования текстовой информации, сравнить их эффективность.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Глава 3.

Необходимые теоретические сведения

Пусть имеется первичный алфавит *{A}*, содержащий *N* знаков с вероятностями их появления в тексте p_i.

Избыточность первичного двоичного кода определяется по формуле:

$$Q(A,2) = \frac{K(A,2)}{I_1^{(A)}} - 1, \qquad (3.1)$$

где *K(A,2)* – средняя длина кодовой комбинации;

$I_1^{(A)}$ – средняя информация на знак первичного алфавита.

При известных p_i найти $I_1^{(A)}$ можно по формуле Шеннона:

$$I_1^{(A)} = -\sum_{i=1}^{N} p_i \log_2 p_i \qquad (3.2)$$

Если кодируемые тексты (сообщения) являются *шенноновскими* (а только такие мы рассматриваем), то p_i имеют фиксированные значения для каждой буквы, никоим образом не связанные с методами кодирования. По указанной причине при сопоставлении разных методов средняя информация на знак первичного алфавита вычисляется один раз.

Для нахождения *K(A,2)* необходимо знать вероятности появления знаков первичного алфавита p_i, а также длины кодовых комбинаций каждого знака k_i; тогда:

$$K(A,2) = \sum_{i=1}^{N} k_i p_i \qquad (3.3)$$

Длины кодов k_i зависят от выбранного метода кодирования. Очевидно, из-за этого и избыточности для различных методов оказываются разными.

Суть первой части работы состоит в определении *K(A,2)* по (3.3) для разных методов кодирования одного и того же алфавита, вычисления избыточности метода по формуле (3.1) и сопоставлении на основании нее эффективности различных методов.

Во второй части работы производится сопоставление эффективности алфавитного и блочного кодирования.

Часть 1. Методы алфавитного кодирования

Рабочий файл: **ТОИ_Лр-3(1).xls**

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файлы **ТОИ_Лр-3(1).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов (меню ***Сервис→****Макрос→Безопасность*). Зарегистрируйтесь на листе <**Отчет**>.
2) В *Задании 1* по заданным вероятностям знаков определите среднюю информацию на знак.
3) В *Задании 2* по количеству знаков первичного алфавита определите длину кодовой комбинации для равномерного кодирования, произведите кодирование, найдите избыточность. Результат занесите в табл. в *Задании 6*.
   ВАЖНО: в Заданиях 3-5 перед применением методов кодирования знаки алфавита должны быть расположены в порядке убывания их вероятностей
4) Постройте коды в *Заданиях 3-5*, определить их избыточности. Результаты занесите в табл. в *Задании 6*.
5) По табл. в *Задании 6* по избыточностям полученных кодов сопоставьте эффективности и сделайте выводы относительно использованных методов кодирования.
6) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
7) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Почему средняя информация на знак первичного алфавита оказывается одинаковой для всех методов алфавитного кодирования? При каком условии это справедливо?
2. Чем определяется длина равномерного кода для заданного алфавита? Как ее найти?
3. Почему 1 байт = 8 бит?
4. Зачем необходимы разделители кодов при неравномерном кодировании? В каких случаях разделители не требуются?
5. В чем состоит идея использования неравномерных кодов? Из каких соображений выбирается длина кода для того или иного знака алфавита?
6. Что такое префиксные коды? В чем смысл условия Фано?
7. Почему в методе Хаффмана количество промежуточных алфавитов при свертывании на 2 меньше числа знаков первичного алфавита?
8. Может ли длина какого-либо кода в методе Хаффмана превышать длину равномерного кода для того же алфавита?
9. Всегда ли эффективность равномерного кодирования оказывается ниже по сравнению с эффективностью метода Хаффмана?
10. В чем состоит значение метода Хаффмана для теории и практики кодирования?

Часть 2. Сопоставление алфавитного и блочного кодирования

Рабочий файл: **ТОИ_Лр-3(2).xls**

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файл **ТОИ_Лр-3(2).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов (меню ***Сервис→****Макрос→Безопасность*). Зарегистрируйтесь на листе <**Отчет**>.
2) Выполните *Задание 1*:
   - для заданного алфавита найдите среднюю информацию на знак;
   - произведите кодирование алфавита по методу Хаффмана, найдите среднюю длину кода (*перед применением метода не забудьте расположить знаки в порядке убывания их вероятностей*);
   - определите избыточность алфавитного кодирования.
3) Выполните *Задание 2*:
   - выпишите в таблицу все парные комбинации знаков (их количество – n^2, где n – число знаков первичного алфавита);
   - полагая, что мы имеем дело с шенноновскими сообщениями, рассчитайте вероятности появления всех пар P_{ij}; проверьте выполнение условия нормировки вероятностей;
   - вычислите среднюю информацию на знак двухбуквенного алфавита; убедитесь, что она строго в 2 раза больше, чем для однобуквенного;
   - упорядочите по убываю вероятностей знаки в таблице; произведите кодирование двухбуквенного алфавита по методу Хаффмана; определите избыточность кода.
4) Произведите сопоставление избыточностей кодов одно- и двухбуквенного алфавитов, сделайте выводы.
5) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
6) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. В чем значение первой теоремы Шеннона для кодирования?
2. Чем определяется минимальная средняя длина алфавитного кода? При каком методе кодирования она достигается?
3. Как соотносятся средние информации на знак при алфавитном и блочном кодировании? Почему?
4. Почему избыточность блочного кода оказывается меньше избыточности алфавитного кода, если в обоих случаях использовался метод Хаффмана?
5. Существует ли предел снижения избыточности кода путем построения блочных кодов? Если «да», то чем он определяется?

Лабораторная работа № 4.

Кодирование числовой информации

***Учебная задача*:**

Освоить формы представление числовой информации в недесятичных системах счисления. Изучить порядок обработки кодов чисел в компьютере.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Глава 4.

Рабочие файлы: **ТОИ_Лр-4(1,2).xls, ТОИ_Лр-4(3).xls**

Часть 1. Системы счисления

Рабочий файл: **ТОИ_Лр-4(1,2).xls**, лист <**Отчет 1**>

Необходимые теоретические сведения

Система счисления – это правило записи чисел с помощью заданного набора специальных знаков – цифр.

Любое целое число Z может быть представлено в позиционной системе счисления с основанием p в виде многочлена:

$$Z_p = a_{k-1} \cdot p^{k-1} + a_{k-2} \cdot p^{k-2} + \ldots + a_1 \cdot p^1 + a_0 \cdot p^0 = \sum_{j=0}^{k-1} a_j \cdot p^j, \qquad (4.1)$$

где коэффициенты разложения a_j удовлетворяют условию: $0 \le a_j \le p - 1$

Аналогично, любая правильная дробь $0,Y$ также может быть представлена в виде многочлена с отрицательными показателями степеней:

$$0,Y_p = a_1 \cdot p^{-1} + a_2 \cdot p^{-2} + \ldots + a_m \cdot p^{-m} + \ldots = \sum_{j=1}^{k} a_j \cdot p^{-j} \qquad (4.2)$$

Одно и то же число может быть представлено в различных системах счисления. В связи с этим возникает вопрос о порядке перевода числа, представленного в одной системе счисления (СС), в СС с другим основанием.

Перевод целых десятичных чисел Z в СС с основанием p может быть реализован посредством рекуррентных соотношений:

$$Z_0 = Z_{(10)};$$
$$Z_{i+1} = Z_i \text{ div } p; \qquad a_i = Z_i \bmod p, \qquad (4.3)$$

где посредством функции div находится результат целочисленного деления, а функция mod дает остаток от целочисленного деления. В пакете MS Excel эти результаты могут быть получены посредством функций ЦЕЛОЕ(Z_i/p) и ОСТАТ(Z_i/p).

Процедура прекращается на том шаге, когда оба значения – Z_{i+1} и a_i – окажутся равными *0*. Полученное число *считывается в обратном порядке*.

Перевод простых десятичных дробей 0,Y в СС с основанием p может быть реализован посредством следующих рекуррентных соотношений:

$$Y_0 = 0,Y_{(10)};$$
$$a_i = \text{int}(Y_i \cdot p);\ Y_{i+1} = Y_i \cdot p - a_i \quad , \qquad (4.4)$$

где функция int означает нахождение целой части числа путем отбрасывания дробной (без округления). В MS Excel результат может быть получен использованием ЦЕЛОЕ($Y_i \cdot p$).

Следует сознавать, что при переводе дробей может возникнуть ситуация, когда конечная дробь в 10-ной СС превратиться в бесконечную дробь в СС с основанием *p* или наоборот. Поэтому процедура прекращается на том шаге, когда оба значения – Y_i и a_i – окажутся равными *0* либо по достижении желательной точности вычислений.

Перевод целых чисел Z_p в 10-ю СС может быть реализован на основании представления исходного числа в форме (4.1) и последующего проведения вычислений по правилам 10-ной арифметики.

Аналогично, *перевод простых дробей $0,Y_p$ в 10-ю СС* может быть реализован на основании представления исходного числа в форме (4.2) и последующего проведения вычислений по правилам 10-ной арифметики.

При переводе *чисел в естественной форме представления* (смешанных чисел) отдельно и независимо переводятся целая и дробная части и затем объединяются в единое число.

Перевод чисел из одной недесятичной СС (*p*) в другую недесятичную (*q*) необходимо начинать с анализа соотношения между *p* и *q*. Для «хороших» соотношений действуют 2 правила:

Правило 1. *Для преобразования целого числа $Z_p \rightarrow Z_q$ в том случае, если системы счисления связаны соотношением $q = pr$, где r – целое число большее 1, достаточно Z_p разбить* ***справа налево*** *на группы по r цифр и каждую из них независимо перевести в систему q.*

Правило 2. *Для преобразования целого числа $Z_p \rightarrow Z_q$ в том случае, если системы счисления связаны соотношением $p = qr$, где r – целое число большее 1, достаточно каждую цифру Z_p заменить соответствующим r-разрядным числом в системе счисления q, дополняя его при необходимости незначащими нулями слева до группы в r цифр.*

Переходы $Z_8 \rightarrow Z_{16}$ *и* $Z_{16} \rightarrow Z_8$, очевидно, удобнее осуществлять через промежуточный переход к двоичной системе. Например,

$$172_8 = 001111010_2 = 7A_{16}.$$

Подобным же образом происходит перевод дробных числе с той лишь разницей, что он осуществляется от десятичного разделителя вправо.

Если *p* и *q* не связаны указанными выше соотношениями, перевод следует осуществлять через промежуточных переход к 10-й СС:

$$Z_p \rightarrow Z_{10} \rightarrow Z_q$$

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файлы **ТОИ_Лр-4(1,2).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе <**Отчет 1**>.
2) Ознакомьтесь с реализацией алгоритма перевода (4.3) на листе <**Перевод 10–р**>. Выполните упражнения (*1.a*) *Задания 1*. При необходимости модернизируйте расчетную схему для увеличения разрядности преобразуемых целых чисел. Предусмотрен автоматизированный контроль правильности вычислений.
3) Ознакомьтесь с реализацией алгоритма перевода (4.1) на листе <**Перевод р–10**>. Выполните упражнения (*1.b*) *Задания 1*. При необходимости модернизируйте расчетную схему для увеличения разрядности преобразуемых и получаемых дробных чисел. Предусмотрен автоматизированный контроль правильности вычислений.
4) Ознакомьтесь с реализацией алгоритма перевода (4.4) на листе <**Перевод 10–р**>. Выполните упражнения (*2.a*) *Задания 2*. При необходимости модернизируйте расчетную схему для увеличения разрядности преобразуемых целых чисел. Предусмотрен автоматизированный контроль правильности вычислений.
5) Ознакомьтесь с реализацией алгоритма перевода (4.2) на листе <**Перевод р–10**>. Выполните упражнения (*2.b*) *Задания 2*. При необходимости модернизируйте расчетную схему для увеличения разрядности преобразуемых и получаемых дробных чисел. Предусмотрен автоматизированный контроль правильности вычислений.
6) С помощью построенных расчетных схем выполните упражнения по переводу смешанных чисел в *Задании 3*.
7) В *Задании 4a* требуется осуществить перевод чисел между СС, основания которых связаны между собой в соответствии с Правилами 1 и 2, приведенными в теоретическом введении. В *Задании 4b* по имеющимся данным необходимо заполнить пустые ячейки таблицы для СС 2-8-16 также с использованием Правил 1 и 2.
8) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
9) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. В чем отличие аддитивных и аддитивно-мультипликативных СС? Приведите примеры.
2. Можно ли унарную систему счисления считать позиционной СС с основанием 1? Почему?
3. Почему при переводе дробного числа из одной системы счисления в другую конечная дробь может стать бесконечной или наоборот? Приведите примеры.

4. Каким образом перевести число в естественной форме представления (имеются целая и дробная части) из одной недесятичной СС в другую недесятичную?
5. Сформулируйте правило перевода целых чисел между СС, основание одной из которых является целочисленной степенью другого основания. Приведите пример.
6. Сформулируйте правило перевода дробных чисел между СС, основание одной из которых является целочисленной степенью другого основания. Приведите пример.
7. Как удобнее осуществлять перевод между СС с основаниями $8 \leftrightarrow 4$? Почему?
8. Усовершенствуйте реализацию алгоритма перевода *целых чисел* между СС $10 \rightarrow p$, представленного на листе <**Перевод 10 – р**>, таким образом, чтобы в некоторой ячейке сразу выдавался результат перевода (как единое число).
9. Усовершенствуйте реализацию алгоритма перевода *дробных чисел* между СС $10 \rightarrow p$, представленного на листе <**Перевод 10 – р**>, таким образом, чтобы в некоторой ячейке сразу выдавался результат перевода (как единое число).
10. Почему при обработке чисел в компьютере используется двоичная СС?

Часть 2. Нормализация чисел

Рабочий файл: **ТОИ_Лр-4(1,2).xls**, лист <**Отчет 2**>

Необходимые теоретические сведения

Число X_p называется *нормализованным в системе счисления p*, если оно представлено в виде:

$$X_p = \pm M_p \cdot p^{\pm k_p} \quad , \qquad (4.5)$$

где M_p – мантисса нормализованного числа, удовлетворяющая условию:

$$\frac{1}{p} \le M_p < 1 \quad , \qquad (4.6)$$

k_p – порядок нормализованного числа, представленный в СС *p*.

Например,

$(3213202{,}12)_4 = 0{,}321320212_4 \cdot 4^{13_4}$

$(0{,}0000004321)_5 = 0{,}4321 \cdot 5^{-11_5}$

Для перевода десятичного числа в естественной форме представления в нормализованное число в СС *p* можно сначала перевести число в естественной форме из *10* в *p*, пользуясь описанными выше алгоритмами, а затем нормализовать его в СС *p*.

Например, необходимо представить число $16{,}5_{10}$ в нормализованной форме в 8-ричной СС. Пользуясь алгоритмами (4.3) и (4.4), получаем: $17{,}375_{10} = 21{,}3_8 = 0{,}213_8 \cdot 8^2$; таким образом, $M_8 = 0{,}213$; $k_8 = 2$.

Для перевода числа из естественной формы представления в СС *p* в нормализованное число в СС *q* можно сначала перевести число в естественной форме из *p* в *10*-ю СС по алгоритмам (4.1) и (4.2), затем в естественную форму в СС *q*, а затем нормализовать его в СС *q*.

Преобразование числа из нормализованной формы в СС *p* в нормализованное число в СС *q* удобнее осуществлять через промежуточный переход к 10-ой СС.

При переводе нормализованных чисел основания которых связаны целочисленными степенями, нужно исходное число представить в естественной форме, перевести в новую СС согласно Правилам 1 и 2, затем нормализовать в новой СС. Например:

p = 9: q = 3

$$X_9 = 0{,}283765 \cdot 9^{4_9} = 2837{,}65_9 = 02221021{,}2012_3 = 0{,}22210212012_3 \cdot 3^{21_3} = X_3$$

Порядок выполнения работы:

1) Откройте из своей рабочей папки файл с отчетом по 1-й части ЛР **ТОИ_Лр-4(1,2).xls**. Если требуется, установите **средний** уровень безопасности макросов. Повторная регистрация не требуется. Перейдите на лист <**Отчет 2**>.

2) Используя реализованные в предыдущей части данной ЛР алгоритмы перевода целых и дробных чисел в различные СС, выполните *Задания 1-2*(a-c), *3a*. При выполнении заданий *2c* и *3a* преобразования необходимо осуществлять через десятичную СС в качестве промежуточной. Результаты занесите в соответствующие таблицы.
3) В задании *3b* выполните преобразования между СС 2↔8↔16) без перехода к 10-ой СС в качестве промежуточной.
4) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
5) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. В чем преимущества и недостатки представления чисел в нормализованной форме?
2. Почему мантисса нормализованного числа должна удовлетворять условию (4.6)?
3. Сформулируйте правило построения нормализованного числа из числа в естественной форме представления для произвольной СС.
4. Сформулируйте правило построения нормализованного десятичного числа из числа в естественной форме представления в произвольной СС.
5. Сформулируйте обобщенное правило преобразований нормализованных чисел между СС, основание одной из которых является целочисленной степенью другого основания.

Часть 3. Операции с кодами чисел

Рабочий файл: **ТОИ_Лр-4(3).xls**, лист <**Отчет 3**>

Необходимые теоретические сведения

Размещение кодов чисел в регистрах

Для представления двоичных чисел в памяти и регистрах компьютера отводится конечное число разрядов. По этой причине можно говорить, что в компьютере производится обработка ***кодов чисел***, которые представляют собой запись чисел в ограниченной разрядной сетке.

При выполнении данной лабораторной работы будет использоваться 16-ти-разрядная сетка. Нумерация разрядов сетки осуществляется *справа налево*, младший разряд имеет номер «0» (соответственно, старший разряд – номер «15»).

При записи ***целых чисел без знака*** (положительных) заполнение сетки производится, *начиная с младшего разряда*. Очевидно, максимальное число, которое может быть записано в 16-ти-разрядную сетку, будет равно $2^{16} - 1 = 65535$ (во всех ячейках стоят *1*). При попытке записать большее по величине число возникает ошибочная ситуация, называемая *переполнением* (число выходит за установленную сетку).

Запись ***целых чисел со знаком*** в отведенную разрядную сетку осуществляется в *дополнительном коде*. Для двоичных целых чисел он строится по следующим правилам:

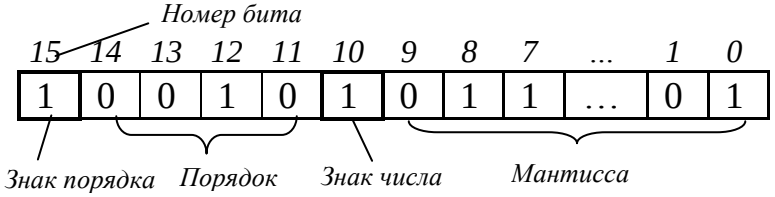
- для $Z_2 \geq 0$ дополнительный код совпадает с самим числом ($DK = Z_2$);
- для $Z_2 < 0$ дополнительный код совпадает с дополнением модуля числа, т.е. $DK = D(|Z_2|,k)$.

В свою очередь, *дополнение модуля двоичного числа* строится следующим образом:

- инвертировать представление исходного числа в отведенной разрядной сетке;
- к последнему (младшему) разряду инвертированного представления прибавить *1* по правилам двоичной арифметики.

Таким образом, 15-й (старший) бит целого положительного числа всегда содержит *0*, а отрицательного числа *1*. Очевидно, в такой форме представления максимальным по модулю будет число $2^{15} - 1 = 32767$ – при его превышении возникает переполнение.

При размещении ***двоичного нормализованного числа*** запись всех составляющих (знак числа, мантисса, знак порядка и порядок) осуществляется в строго определенные ячейки разрядной сетки. Поскольку в данной лабораторной работе используется 16-ти-разрядная сетка, примем следующий порядок размещений в ней вещественных чисел:



Т.к. значение мантиссы лежит в интервале $0{,}1_2 \le M_2 < 1$, ноль в разряде целых и разделитель десятичных разрядов в представление не включается, т.е. мантисса содержит *только цифры дробной части*.

Таким образом, наибольший порядок числа в данной форме представления равен $2^4 - 1 = 15$; следовательно, максимальным оказывается число $2^{15} = 32767_{10}$.

Поскольку соотношение между числом разрядов в двоичном и десятичном представлении числа: $k_2 = 3{,}322 \cdot k_{10}$, 10-битная двоичная мантисса обеспечивает точность представления и вычисления до 3-х знаков в десятичной мантиссе – все разряды, следующие за тысячными, будут пропадать, что приводит к погрешности записи и обработки вещественных чисел.

Операции с двоичными числами выполняются согласно правилам двоичной арифметики:

Сложение производится согласно таблице сложения, которая для двоичных чисел имеет вид:

$0 + 0 = 0$ $\qquad$ $0 + 1 = 1$

$1 + 0 = 1$ $\qquad$ $1 + 1 = \overset{\frown}{10}$

В последнем случае в том разряде, где находились слагаемые, оказывается 0, а 1 переносится в старший разряд.

Умножение производится согласно таблице умножения:

$0 \cdot 0 = 0$ $\qquad$ $0 \cdot 1 = 0$

$1 \cdot 0 = 0$ $\qquad$ $1 \cdot 1 = 1$

Операции сложения и умножения являются *двуместными*, поскольку выполняются над двумя числами и, следовательно, требуют два регистра для записи обрабатываемых чисел и один для записи результата.

Над целыми числами со знаком определены операции сложения и умножения. *Операция вычитания как самостоятельная отсутствует*, поскольку она эквивалентна *сложению с дополнительным кодом вычитаемого*. Операция деления над целыми числами не определена.

Сложение и вычитание нормализованных чисел возможно только *при совпадении их порядков*. Поэтому перед выполнением операции должен быть произведен сдвиг мантиссы меньшего слагаемого в разрядной сетке таким образом, чтобы порядки совпали. Сложение разрядов мантиссы осуществляется побитно; при необходимости результат сложения нормализуется.

Умножение нормализованных чисел производится следующим порядком:

- сложить порядки сомножителей;
- перемножить мантиссы;
- при необходимости нормализовать результат умножения.

Деление нормализованных чисел эквивалентно умножению на число, у которого изменен на противоположный знак порядка.

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файлы **ТОИ_Лр-4(3).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе <**Отчет 3**>.
2) Используя реализованные ранее алгоритмы перевода целых и дробных чисел в различные СС, выполните *Задание 1(a, b)*. Результаты занесите в соответствующие таблицы. При возникновении переполнения сделайте отметку («1») в соответствующей ячейке таблицы. Предусмотрена автоматизированная проверка.

   Рекомендация*: перед записью десятичных чисел в разрядную сетку удобнее сначала перевести их в 8-ричную СС.*
3) При выполнении *Задания 1c* следует разместить слагаемые в двух первых строках таблицы (регистрах r_1 и r2), а результат – в третьей (регистр r_3). Предусмотрена автоматизированная проверка.

   Важно: ***1) записью чисел в разрядную сетку должна производиться в дополнительном коде;***

   2) Появление «1» в Дополнительном разряде в операции вычитания (сложения с дополнительным кодом) не является переполнением и не свидетельствует об ошибке выполнения операции.
4) При выполнении *Задания 2* нужно использовать установленное выше распределение разрядов для записи нормализованного двоичного числа. Для удобства десятичные числа рекомендуется сначала перевести в 8-ричную СС.

   Важно: ***сначала необходимо осуществить перевод числа в 2-ю СС, затем его нормализовать и только после этого записывать в сетку.***
5) Ознакомьтесь с примером выполнения *Задания 2c*. В нем, показано, что умножение вещественных нормализованных двоичных чисел сводится к операциям сдвига и сложения. По аналогии проведите предложенные вычисления. Предусмотрена автоматизированная проверка.
6) В *Задании 2d* требуется показать на примере, что при выполнении операций с вещественными числами в конечной разрядной сетке возможно возникновение погрешности вычислений.
7) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
8) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Что такое «код числа»? Какие коды применяются в компьютерах?
2. Чем различается представление целых чисел со знаком и без знака в конечной разрядной сетке?

3. Сформулируйте правило построения дополнительного кода числа. Приведите пример.
4. Как представляется вещественное число в конечной разрядной сетке? Чем определяется точность представления и обработки чисел?
5. В чем преимущества и недостатки кодирования вещественных чисел с помощью нормализованной двоичной формы?
6. Опишите порядок действий при сложении двух нормализованных двоичных чисел в конечной разрядной сетке.
7. Опишите порядок действий при вычитании двух нормализованных двоичных чисел в конечной разрядной сетке.
8. Опишите порядок действий при умножении двух нормализованных двоичных чисел в конечной разрядной сетке.
9. Опишите порядок действий при делении двух нормализованных двоичных чисел в конечной разрядной сетке.
10. В каких случаях при обработке кодов вещественных чисел возникает погрешность вычислений? Почему? Может ли погрешность не возникать?

Лабораторная работа № 5.

Исследование дискретных двоичных каналов передачи информации

***Учебная задача*:**

На основе экспериментирования с компьютерными моделями двоичных каналов выявить их характеристики.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Глава 5.

Рабочий файл: **ТОИ_Лр-5(1,2).xls**

Необходимые теоретические сведения

Определения:

Дискретным называется канал связи, используемый для передачи сообщений в дискретной форме представления (с помощью знаков).

Дискретный канал называется **двоичным**, если он используется для передачи знаков двоичного алфавита.

Дискретный канал называется **однородным**, если характеризующие его апостериорные вероятности не изменяются с течением времени.

Дискретный двоичный канал называется **симметричным**, если для апостериорных вероятностей выполняются условия:

$$p_0(1) = p_1(0) = p; \qquad p_0(0) = p_1(1) = 1 - p,$$

где p – вероятность возникновения ошибки при передаче знака («*0*» или «*1*») по каналу.

Пропускная способность дискретного канала *без помех* численно равна максимальной (или несущей) частоте полосы пропускания:

$$C_0 = \nu_m \qquad (5.1)$$

Пропускная способность *однородного двоичного симметричного канала с помехами* может быть вычислена:

$$C = \nu_m \cdot [1 + (1 - p) \cdot \log_2(1-p) + p \cdot \log_2 p], \qquad (5.2)$$

где p – вероятность ошибки передачи, ν_m – частота передачи.

Дискретный двоичный канал **называется каналом со стиранием**, если на приемном конце канала возможно появление знака, который не может быть интерпретирован ни как «0», ни как «1» (это эквивалентно стиранию исходного знака и появлению нового (обозначим «s»), которого не было в начальном двоичном алфавите на входном конце канала).

Пропускная способность однородного двоичного симметричного канала со стиранием может быть вычислена:

$$C = \nu_m \cdot \{(1 - q)[1 - \log_2(1-q)] + (1 - p - q)\cdot\log_2(1 - p - q) + p\cdot\log_2 p], \qquad (5.3)$$

где p – вероятность возникновения ошибки при передаче знака (замена «*0*» на «*1*» или «*1*» на «*0*»); q – вероятность стирания (замена «*0*» или «*1*» знаком стирания); ν_m – частота передачи.

Исследование:

В лабораторной работе представлена экранная модель дискретного канала связи. Поскольку канал двоичный, он имеет 2 входа для отправки сигналов «*0*» и «*1*». У канала без стирания выходов также 2; канал со стиранием имеет 3 выхода (добавляется знак стирание «*s*»).

Задача исследования состоит в определении пропускной способности каналов, для чего требуется знать вероятности искажения p (и стирания q) сигналов. Экспериментирование состоит в том, что подаются сигналы на вход (N сигналов) и фиксируются сигналы на выходном конце канала связи. Из-за воздействия помех в канале часть исходных сигналов будет инвертироваться или стираться. Например, при прохождении сигнала «1», на выходном конце обязательно появится некоторое количество «0» (и «s»). Очевидно, что:

$$N = N(0) + N(1) + N(s)$$

Если отслеживалось прохождение «1», то, определив $N(0)$, можно найти вероятность искажения «1»:

$$p_0(1) = N(0)/N.$$

Аналогичным путем можно определить $p_1(0)$.

Поскольку появления помех являются случайными событиями, для надежного отслеживания их воздействия требуется производить опыты с большим числом посылаемых сигналов (несколько десятков тысяч). В работе предусмотрен также 3-х кратный повтор опытов с разными N с последующим усреднением полученных вероятностей.

Если оказывается, что $p_1(0) = p_0(1)$ (на самом деле, значения могут незначительно различаться из-за ограниченности статистической выборки), то делается заключение, что *канал симметричный*, что позволяет для расчета его пропускной способности использовать выражение (5.2) (или (5.3) для канала со стиранием).

Значение ν_m генерируется в индивидуальном варианте работы при регистрации – по нему можно найти C_0 – пропускную способность канала без помех.

В работе предусмотрена автоматизированная проверка правильности проведения исследования и вычислений.

Часть 1. Исследование двоичного канала без стирания

Рабочий файл: **ТОИ_Лр-5(1,2).xls**, лист <**Двоичный канал**>

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файлы **ТОИ_Лр-5(1,2).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе <**Двоичный канал**>.
2) При выполнении *Задания 1* в окне (ячейке) «***Число сигналов***» указывается количество *парных* сигналов (*0* и *1*), которые будут поданы на вход канала (например, при указании числа сигналов 1000 в канал буду посланы 1000 «*0*» и 1000 «*1*»). В процессе исследования нужно занести в таблицы результаты подсчета на выходном конце числа инвертированных (искаженных) сигналов и по ним найти $p_1(0)$ в первой таблице и $p_0(1)$ во второй. Из их сравнения сделать вывод *о возможности считать канал симметричным*. Если это приближение оказывается принятым, за вероятность искажения сигнала p принять среднее из $p_1(0)$ и $p_0(1)$.
3) В *Задании 2* по заданной несущей частоте ν и вероятности p вычисляются пропускные способности канала без помех (C_0) и симметричного с помехами (C). Значение C_0 следует округлить до целых. Предусмотрен автоматизированный контроль правильности вычислений.
4) В *Задании 3* следует построить граф канала связи (используя встроенный графический редактор MS Excel).
5) При выполнении *Задания 4* необходимо протабулировать функцию

$$\frac{C}{C_0}(p)$$

для симметричного двоичного канала в интервале $p \in [0,1]$ (20-25 точек) и построить ее график.
6) Сформулируйте выводы по работе.
7) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
8) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Тождественны ли понятия «канал связи» и «линия связи»? Ответ обоснуйте.
2. Верно ли, что характер канала (дискретный или аналоговый) определяется свойствами среды, в которой происходит передача сигналов?
3. Что общего и в чем отличие характеристик «пропускная способность канала связи» и «скорость передачи информации»?
4. Каким образом должны соотноситься скорость передачи информации и пропускная способность канала связи, чтобы при передаче не происходило потери информации? Почему?

5. В чем состоит влияние помех на процесс передачи информации по дискретному каналу связи? Как влияют помехи на пропускную способность канала?
6. Что такое «граф канала»? Что он отражает? Каковы правила его построения?
7. Как на графе канала отражается то обстоятельство, что канал является *двоичным симметричным*?
8. Каким образом в выполненной лабораторной работе доказывается, что канал является симметричным? Зачем это делается?
9. В работе получен результат $C<C_0$. Какова причина? Всегда ли при искажении входных сигналов справедливо это соотношение?
10. Почему при вероятности искажения сигнала $p = 1$ пропускная способность симметричного канала оказывается такой же, как у канала без помех?

Часть 2. Исследование двоичного канала со стиранием

Рабочий файл: **ТОИ_Лр-5(1,2).xls**, лист <**Канал со стиранием**>

Порядок выполнения работы:

1) Откройте из своей рабочей папки файл с отчетом по 1-й части ЛР **ТОИ_Лр-5.xls**. Если требуется, установите **средний** уровень безопасности макросов. Повторная регистрация не требуется. Перейдите на лист <**Канал со стиранием**>.
2) В *Задании 1* по аналогии с 1-й частью работы проведите исследование с целью определения вероятностей искажения (p) и стирания (q). Знак стирания обозначен «*s*». Убедитесь, что канал является симметричным.
3) В *Задании 2* проведите вычисление пропускной способности канала. Проверьте правильность расчетов. Проведите сопоставление полученного значения с результатами из Части 1 данной ЛР, а также с пропускной способностью канала без помех.
4) В *Задании 3* постройте граф канала.
5) В *Задании 4* для полученного значения q постройте график зависимости
$$\frac{C}{C_0}(p)$$
Сопоставьте полученный характер зависимости с аналогичной зависимостью из Части 1 данной ЛР.
6) Сформулируйте выводы по работе.
7) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
8) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Что такое «стирание» при передаче сигналов? Почему оно происходит?
2. Всегда ли при передачи двоичных сигналов возникает стирание? Чем может определяться необходимость его учета?
3. Как соотносятся вероятности искажения (инверсии) (p) и стирания (q) для дискретного канала связи? Почему?
4. Каким образом наличие стирания влияет на пропускную способность канала связи? Почему?
5. Как влияет вероятность стирания на характер зависимости пропускной способности от вероятности искажения сигнала? Предложите объяснение.

Теоретические Основы Информатики

Лабораторный практикум

Лабораторная работа № 6.

Построение помехоустойчивых кодов

***Учебная задача*:**

Освоить построение канонического систематического кода и кода Хемминга.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Глава 6.

Рабочий файл: **ТОИ_Лр-6(1,2).xls**

Часть 1. Построение канонического систематического кода

Рабочий файл: **ТОИ_Лр-6(1,2).xls**, лист <**Отчет 1**>

Пример выполнения заданий работы

Условие задания:

Имеется производящая матрица канонического систематического кода:

$$G_{7,4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

По данной исходной матрице необходимо выполнить следующие построения:

(1) Определить характеристики кода (n, k, S, S_p, $F(n,k)$).
(2) Построить все разрешенные кодовые комбинации.
(3) Определить минимальное кодовое расстояние и по нему установить возможности кода по обнаружению и исправлению ошибок.
(4) Составить уравнения проверок.
(5) Построить схему кодера.
(6) Построить проверочную матрицу.
(7) Составить таблицу исправлений.
(8) Построить схему декодера.

Выполнение заданий

По информационной части заданной производящей матрицы видно, что она представлена диагональной подматрицей, следовательно, код является *каноническим* и для его построения применимы матричные операции.

(1) Из анализа структуры матрицы $G_{7,4}$ видно, что она содержит 7 столбцов и 4 строки. Следовательно, длина кодовой комбинации $n = 7$; число информационных бит $k = 4$; их разность дает число проверочных бит $r = n - k = 3$. Таким образом, требуется построить канонический систематический код (7,4). Избыточность кода $F(n, k) = r/k = 0{,}75$.

Общее количество кодовых комбинаций $S = 2^n = 128$.

Число разрешенных кодовых комбинаций $S_p = 2^k = 16$

(2) Среди разрешенных кодовых комбинаций всегда имеется тривиальная $U^{(0)}=\{0000000\}$. Еще 4 кода могут быть записаны непосредственно из строк матрицы $G_{7,4}$: $U^{(1)}=\{1000110\}$, $U^{(2)}=\{0100101\}$, $U^{(3)}=\{0010011\}$, $U^{(4)}=\{0001111\}$.

Остальные 11 кодов получаются ***построением линейных комбинацией*** базисных кодовых векторов.

$U^{(5)}=U^{(1)} \oplus U^{(2)} = \{1100011\};$

$U^{(6)}=U^{(1)} \oplus U^{(3)} = \{1010101\};$

$U^{(7)}=U^{(1)} \oplus U^{(4)} = \{1001001\};$

$U^{(8)}=U^{(2)} \oplus U^{(3)} = \{0110110\};$

$U^{(9)}=U^{(2)} \oplus U^{(4)} = \{0101010\};$

$U^{(10)}=U^{(3)} \oplus U^{(4)} = \{0011100\};$

$U^{(11)}=U^{(1)} \oplus U^{(2)} \oplus U^{(3)} = U^{(5)} \oplus U^{(3)} = \{1110000\};$

$U^{(12)}=U^{(1)} \oplus U^{(2)} \oplus U^{(4)} = U^{(5)} \oplus U^{(4)} = \{1101100\};$

$U^{(13)}=U^{(1)} \oplus U^{(3)} \oplus U^{(4)} = U^{(6)} \oplus U^{(4)} = \{1011010\};$

$U^{(14)}=U^{(2)} \oplus U^{(3)} \oplus U^{(4)} = U^{(8)} \oplus U^{(4)} = \{011101\};$

$U^{(15)}=U^{(1)} \oplus U^{(2)} \oplus U^{(3)} \oplus U^{(4)} = U^{(11)} \oplus U^{(4)} = \{1111111\}.$

(3) По минимальному количеству единиц в построенных кодовых комбинациях находится ***минимальный вес кодовой комбинации***: $d_{min} = 3$. В свою очередь, по нему можно установить кратности обнаруживаемой и исправляемой ошибок из условий:

$$\theta_{об} = d_{\min} - 1 = 2; \;\; \theta_u = \mathrm{int}\left(\frac{d_{\min} - 1}{2}\right) = 1 \;\;,$$

где функцией **int** в формуле обозначена операция нахождения целой части числа путем отбрасывания дробной части без округления.

(4) Для получения уравнений проверок нужно каждую строку (i) производящей матрицы домножить на u_i и произвести суммирование в столбцах:

$$\begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} p_1 \\ p_2 \\ p_3 \end{array} \left(\begin{array}{cccccc} u_1 & 0 & 0 & 0 & u_1 & u_1 & 0 \\ 0 & u_2 & 0 & 0 & u_2 & 0 & u_2 \\ 0 & 0 & u_3 & 0 & 0 & u_3 & u_3 \\ 0 & 0 & 0 & u_4 & u_4 & u_4 & u_4 \end{array} \right)$$

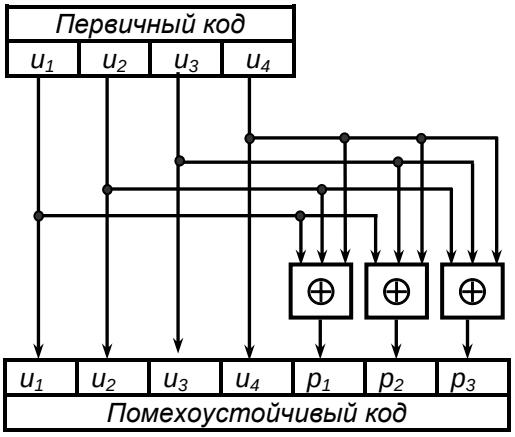
Три последних столбца дают нужные уравнения проверок; суммирование: во всех уравнениях производится по mod 2.

$$p_1 = u_1 \oplus u_2 \oplus u_4$$

$$p_2 = u_1 \oplus u_3 \oplus u_4$$

$$p_3 = u_2 \oplus u_3 \oplus u_4$$

(5) Имея уравнения проверок можно построить схему кодирующего устройства (кодера).



(6) Проверочную матрицу можно построить из уравнений проверок либо из производящей матрицы. Для этого нужно ***транспонировать*** проверочную подматрицу производящей матрицы:

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}^T = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Справа к ней пристраивается единичная диагональная размером (r,r); в рассматриваемом примере (3,3). Таким образом, полностью проверочная матрица будет выглядеть следующим образом:

$$H_{7,3} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

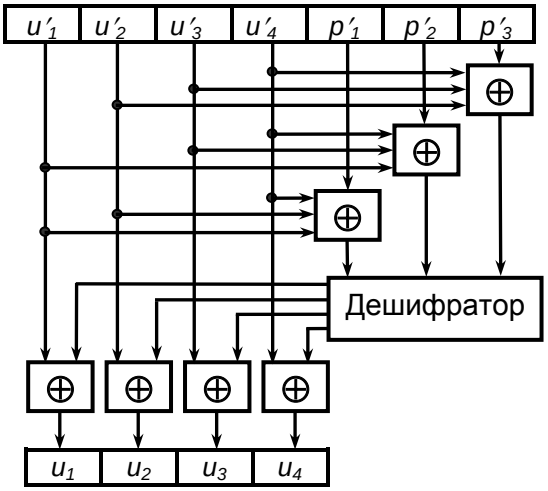
Ясно, что при необходимости из проверочной матрицы $H_{n,r}$ можно, произведя обратные действия, получить производящую $G_{n,k.}$

(7) Таблица исправлений может быть построена из следующих соображений.

Предположим, что ошибочно передан бит u_1, а все остальные знаки кода переданы верно. Это скажется на значениях тех проверочных бит, в которые входит u_1 – а нашем случае – p_1 и p_2. Таким образом, искажению u_1 будет соответствовать синдром ошибки $Q = (110)$. Здесь *1* – соответствует номеру проверочного уравнения, в котором фиксируется ошибка, а *0* – тому, где ошибки нет. Для исправления необходимо инвертировать ошибочный бит, т.е. к его значению прибавить *1 по mod 2*, к остальным же битам не прибавлять ничего (или, что то же самое, прибавить *0*). Исправлять ошибки в контрольных битах p_r, даже если они будут там обнаружены, при декодировании не требуется. Таким образом, получаем полную ***таблицу исправлений:***

Синдром	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
Вектор ошибки	001	010	011	100	101	110	111
Ошибочная позиция кода	p_1	p_2	u_3	p_3	u_2	u_1	u_4
Выход дешифратора	0000	0000	0010	0000	0100	1000	0001

(8) Непосредственно исправление ошибочного бита производит блок декодера, называемый ***дешифратором***: если синдром указывает на ошибку в одном из проверочных бит, исправлений в информационных битах не требуется и дешифратор выдает *0* на все выходы; если синдром указывает на ошибку в каком-либо информационном бите, дешифратор посылает *1* на вход сумматора по mod *2*. Схема декодера систематического кода для построенного кода будет выглядеть следующим образом:



Задания выполнены.

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файл **ТОИ_Лр-6(1,2).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе <**Отчет 1**>. В процессе регистрации будет сгенерирован индивидуальный вариант и представлена исходная матрица G или H. Дальнейшая работа производится с ней.
2) Если исходной является ***производящая матрица*** *G*, выполните *Задания 1-8* в последовательности, описанной выше.
3) Если исходной является ***проверочная матрица*** *H*, задания выполняются в следующем порядке:
   (1) Определить характеристики кода (n, k, S, S_p, $F(n, k)$).
   (2) Составить уравнения проверок.
   (3) Построить производящую матрицу.
   (4) Определить минимальное кодовое расстояние и по нему установить возможности кода по обнаружению и исправлению ошибок.
   (5) Построить все разрешенные кодовые комбинации.
   (6) Построить схему кодера.
   (7) Составить таблицу исправлений.
   (8) Построить схему декодера.

4) Схемы кодера и декодера строятся с использованием экранных заготовок элементов схем и при необходимости – встроенного графического редактора MS Excel.
5) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
6) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Дайте определение помехоустойчивого кода. За счет чего код приобретает качество помехоустойчивости?
2. Укажите типы помехоустойчивых кодов. Какой из них требует больше проверочных бит при равной кратности ошибки и почему?
3. Какой помехоустойчивый код называется *систематическим* (*линейным*)? Почему?
4. Чем определяется *избыточность* помехоустойчивого кода? Каковы варианты ее уменьшения?
5. Что такое «*вес кодовой комбинации*»? Какое значение имеет эта величина для характеристик помехоустойчивости кода?
6. Почему матрица кода *G* называется *производящей*? Каковы правила ее построения?
7. Почему матрица *H* называется «*проверочной*»?
8. Что такое «*синдром ошибки*»? Каким образом он устанавливается? Могут ли разные биты помехоустойчивого кода иметь одинаковые синдромы ошибок? Почему?
9. Каковы функции дешифратора в декодере? Какова идея его работы?
10. Можно ли для описания характеристик систематического кода использовать схему кодера (или декодера)?

Часть 2. Построение помехоустойчивого кода Хемминга

Рабочий файл: **ТОИ_Лр-6(1,2).xls**, лист <**Отчет 2**>

Порядок выполнения работы:

1) Откройте из своей рабочей папки файл с отчетом по 1-й части ЛР **ТОИ_Лр-6(1,2).xls**. Если требуется, установите **средний** уровень безопасности макросов. Повторная регистрация не требуется. Перейдите на лист <**Отчет 2**>.
2) В *Задании 1* по сгенерированному первичному коду требуется построить помехоустойчивый код Хемминга. Предусмотрена автоматизированная проверка правильности построения.
3) В *Задании 2* необходимо определить характеристики кода (n, k, S, S_p, $F(n, k)$), а также кратность исправляемой и обнаруживаемой ошибок.
4) В *Задании 3* требуется самостоятельно записать 5 исходных (первичных) кодовых комбинаций с тем же n, что было предложено в *Задании 1*, и для них построить коды Хемминга.
5) В *Задании 4* требуется записать уравнения проверок для первых 4-х проверочных бит кода Хемминга.
6) В *Задании 5* требуется локализовать и исправить ошибку передачи в предложенном коде. Предусмотрена проверка решения.
7) *Задание 6* подразумевает построение из предложенных элементов схемы кодера для кода Хемминга.
8) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
9) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Можно ли отнести код Хемминга к линейным? Почему?
2. В чем отличия кода Хемминга от других систематических кодов?
3. Передаются знаки в байтовой кодировке. Сколько проверочных бит будет содержать код Хемминга и какова будет его избыточность?
4. Можно ли утверждать, что схема кодера для построения кода Хемминга является универсальной и позволяет строить их при любой длине первичного кода n?
5. Каким образом происходит локализация ошибки передачи в коде Хемминга? Как она в дальнейшем исправляется?

Лабораторная работа № 7.

Изучение дискретных устройств обработки информации

***Учебная задача*:**

Освоить методы проектирования и описания дискретных устройств обработки информации.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Глава 11.

Рабочий файл: **ТОИ_Лр-7(1,2).xls**

Введение

Дискретные устройства обработки информации – абстрактные устройства, осуществляющие преобразование информации в дискретной форме представления по заданному алгоритму.

Дискретное устройство называется ***двоичным***, если оно преобразует информацию, представленную двоичным алфавитом.

Примем следующую классификацию дискретных устройств обработки информации:

- устройства без памяти – *комбинационные схемы*;
- устройства с конечным числом ячеек памяти – *конечные автоматы*;
- устройства с бесконечной памятью (пример – машина Тьюринга).

Первые два типа устройств изучаются в данной лабораторной работе; исследование *машины Тьюринга* предусматривается в ЛР № 8.

Часть 1. Комбинационные схемы

Рабочий файл: **ТОИ_Лр-7(1,2).xls**, лист <**Комбинационные схемы**>

Необходимые теоретические сведения

Базовые понятия

Комбинационная схема может обеспечить обработку двоичных сигналов, подаваемых на ее вход, в соответствии с заданной логической функцией (системой функций), связывающей выходные сигналы с входными. Обработка производится комбинацией (схемой) *логических вентилей* (*логических элементов*), реализующих три основные логические функции: И, ИЛИ и НЕ – они образуют *базис* элементов. При конструировании схем выходы вентилей могут присоединяться ко входам других вентилей или быть вершинами (выходами) схемы; *не может существовать соединений выхода данного вентиля с его входом* (обратная цепь).

В общем случае комбинационная схема описывается совокупностью *трех* компонентов: алфавитом входных сигналов X, алфавитом выходных сигналов Y и *функцией* (системой функций) *выходов* θ, связывающей выходные сигналы с входными. Поскольку рассматриваются только двоичные устройства, входной и выходной алфавиты совпадают – это двоичный алфавит, содержащий знаки $\{0, 1\}$. Таким образом, двоичная комбинационная схема задается логической функцией (системой функций) выходов $\theta_j(x_1, x_2, \ldots x_N)$. Количество функций выхода равно *числу выходов* схемы ($y_j = \theta_j$).

Имеются три взаимосвязанные формы представления дискретных устройств без памяти: *графическая* (комбинационная схема), *функциональная* (система логических функций, реализуемый схемой), *табличная* (таблица значений на входах и выходах схемы – *таблица истинности*). По любой из указанных форм могут быть построены две другие.

В отношении комбинационных схем выделяется две группы задач: *анализа* и *синтеза*.

Задача анализа: по имеющейся схеме определить функции обработки (функции выходов) и построить таблицу значений.

Задачи синтеза: по заданным функциям обработки или таблице значений построить комбинационную схему.

Пример

Пример решения задачи **синтеза** комбинационной схемы.

Пусть для некоторой схемы таблица истинности имеет вид:

x_1	x_2	y_1	y_2
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

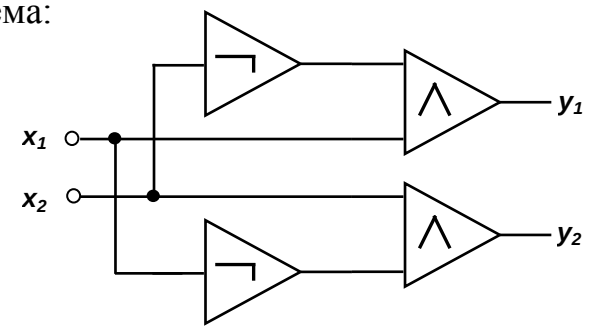
По данной таблице необходимо определить *функции обработки* (*функции выходов*), а также *построить комбинационную схему*.

Из представленной таблицы видно, что схема имеет два независимых входа (x_1 и x_2) и два выхода (вершины) – y_1 и y_2. Анализируя исходную таблицу, можно усмотреть следующие соотношения, связывающие входные и выходные сигналы:

$$y_1 = \overline{x}_1 \wedge x_2;$$

$$y_2 = x_1 \wedge \overline{x}_2$$

Схема:



Задача решена.

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файлы **ТОИ_Лр-7(1,2).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов. Зарегистрируйтесь на листе <**Комбинационные схемы**>. При выполнении работы предусмотрено использование экранного справочника формул математической логики.
2) В *Задании 1a* нужно по схеме, мысленно задавая различные комбинации входных сигналов, определить сигнал на выходе и занести в таблицу. Предусмотрена проверка правильности заполнения таблицы. По ней требуется построить и записать аналитический вид функции выходов.
3) В *Задании 1b* требуется построить комбинационную схему, эквивалентную заданной. Порядок выполнения: по исходной схеме определить логическую функцию обработки, упростить ее (пользуясь соотношениями математической логики – справочник можно вызвать экранной кнопкой), построить схему для упрощенной функции, составить таблицу значений, проверить, что для исходной схемы она также справедлива. Для построения следует использовать предоставленные графические заготовки и при необходимости – встроенный графический редактор MS Excel.
4) *Задание 1c* предполагает построение схемы, аналогичной стоящей перед входами двоичного триггера; имея два входа и два выхода, схема не позволяет появиться на выходе (и, соответственно, на входе триггера) определенной комбинации (например, 11). Задача имеет несколько вариантов решения – достаточно представить любой.
5) В *Задании 2a* необходимо подавать различные комбинации сигналов на входы «черного ящика» и отслеживать сигналы на выходе (комбинации «0» и «1» вводятся в ячейки, обозначенные x_1 и x_2). По этим наблюдениям заполняется таблица значений. По таблице следует установить функции выходов и по ним построить схему.
6) В *Заданиях 2b* и *2c* требуется по заданной таблице (функции) построить две недостающие формы представления комбинационной схемы.
7) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
8) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Что называется «дискретным устройством по обработке информации»? Почему оно называется «дискретным»?
2. Прокомментируйте классификацию дискретных устройств по обработке информации по объему их памяти.
3. Какие дискретные устройства по обработке информации называются «комбинационными схемами»?

4. Верно ли, что можно построить универсальную комбинационную схему, которая обеспечивала бы обработку информации по любому алгоритму? Почему?
5. Верно ли, что компьютер можно рассматривать как комбинационную схему? Почему?
6. В чем особенность двоичных комбинационных схем?
7. Что такое «базис комбинационных схем»? Какие элементы он включает? Почему?
8. Какие существуют способы описания комбинационных схем? В чем достоинства каждого?
9. Укажите общий порядок решения задач анализа комбинационных схем.
10. Укажите общий порядок решения задач синтеза комбинационных схем.

Часть 2. Конечные автоматы

Рабочий файл: **ТОИ_Лр-7(1,2).xls**, лист <**Конечные автоматы**>

Необходимые теоретические сведения

Базовые понятия

В базис *конечных автоматов* помимо логических вентилей входят также *элементы памяти*. К наиболее распространенным элементам памяти следует отнести *задержку*, *триггер* и *двоичный счетчик*. При выполнении заданий данной лабораторной работы будет использоваться *элемент задержки*, который может включаться между выходом и входом одного вентиля (образуя, тем самым, обратную цепь).

Конечный автомат описывается совокупностью *пяти* компонентов: алфавитом входных сигналов X, алфавитом выходных сигналов Y, алфавитом внутренних состояний Q, *функцией* (системой функций) *выходов*, связывающих выходные сигналы с входными сигналами и внутренними состояниями (памятью) $y_j = \theta_j(x_1, x_2, \ldots x_N, q_1, \ldots q_K)$, и функцией (системой функций) *переходов* Ψ, связывающих внутренние состояния на данном такте с выходные сигналы и внутренними состояниями на предыдущем такте $q_i = \Psi_i(x_1, x_2, \ldots x_N, q'_1, \ldots q'_K)$. Поскольку рассматриваются только двоичные устройства, входной и выходной алфавиты, а также алфавит внутренних состояний совпадают – это двоичный алфавит, содержащий знаки $\{0, 1\}$. Таким образом, двоичный конечный автомат задается логическими функциями (системами функций) переходов Ψ и выходов θ. Количество функций переходов равно числу внутренних состояний автомата (т.е. числу элементов памяти); количество функций выходов равно числу выходов схемы.

Способами описания конечных автоматов являются: схема автомата, функциональная таблица (таблица значений функций переходов и выходов), таблица истинности, диаграмма Мура. По любой из указанных форм могут быть построены остальные.

В отношении конечных автоматов выделяется две группы задач: *анализа* и *синтеза*.

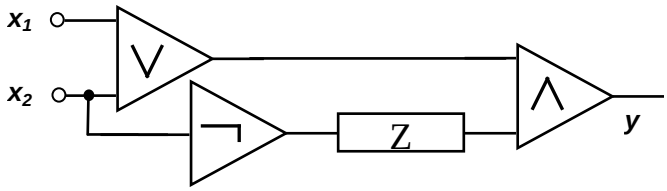
Задача анализа: по имеющейся схеме автомата определить функции обработки (функции переходов и выходов), построить таблицу значений и диаграмму автомата.

Задачи синтеза: по заданным функциям обработки, таблице значений или диаграмме построить схему автомата.

Примеры

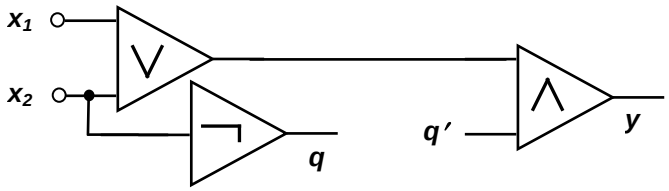
Пример решения задачи **анализа** конечного автомата.

Пусть имеется схема конечного автомата.



Требуется описать ее работу, т.е. построить таблицу допустимых значений (таблицу истинности), определить значения функции выходов и переходов, построить диаграмму Мура.

Для решения задачи анализа используется *метод устранения задержек* – задержка исключается из схемы, но добавляется новая вершина (q) на входе задержки и новый полюс (q') на ее выходе:



Для схемы без задержек можно записать автоматные функции:

$$y = (x_1 \vee x_2) \wedge q';$$

$$q = \overline{x}_2$$

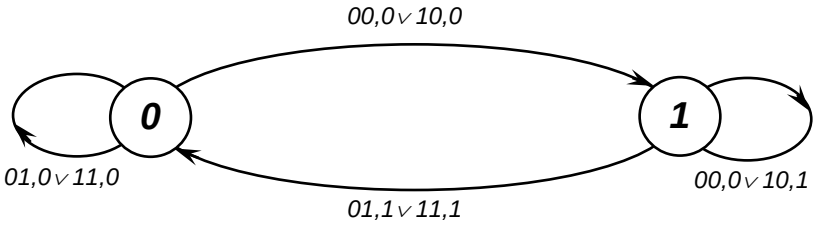
Теперь можно построить таблицу значений (таблицу истинности) для этой системы уравнений, считая все три полюса независимыми и задавая на них различные значения входных сигналов; очевидно, возможны 8 разных их сочетаний; по ним вычисляются q и y:

x_1	x_2	q'	q	y
0	0	0	1	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

На основании таблицы значений можно построить функциональную таблицу (таблицу функций). Ее строки и столбцы соответствуют входным сигналам x_i и состояниям на предыдущем такте q'. В клетках размещаются состояния текущего такта q и значение выходного сигнала y (разделяются запятой). Поскольку схема имеет два входа, в колонке x должны быть записаны все возможные сочетания входных сигналов (очевидно, их 4); при записи указывается сначала сигнал на входе x_1, затем на x_2.

x \ q'	0	1
00	1,0	1,0
01	0,0	0,1
10	1,0	1,1
11	0,0	0,1

На основании функциональной таблицы строиться диаграмма Мура. Вершинами графа являются возможные внутренние состояния (в рассматриваемом примере 0 и 1), ребра указывают возможность перехода из одного состояния в другое; ребру приписываются условия перехода: значения сигналов на входе и выходе.



Решение задачи **синтеза** автомата из логических элементов и элементов задержки производится в обратном порядке:

- заполняется таблица значений функций автомата (по принципу: «при условии подачи на вход таких-то комбинаций на выходе должно быть то-то»); другой вариант – представляются автоматные функции в виде таблицы или диаграммы;
- по таблице значений (или таблице функций) строится система булевских функций, описывающих работу автомата;
- по функциям определяется набор логических элементов и связей между ними;
- строится схема без задержек с промежуточными полюсами и вершинами;
- вводятся элементы задержки, устраняются промежуточные полюса и вершины.

Пример решения задачи **синтеза** конечного автомата.

Пусть имеется таблица значений для некоторого конечного автомата. По ней необходимо построить функциональную таблицу и диаграмму, установить явный вид автоматных функций, а также построить схему автомата. Требуется также оОпределить выходное слово, если входным словом (последовательность сигналов на входе) было *11001*, а начальным состоянием автомата было $q = 0$.

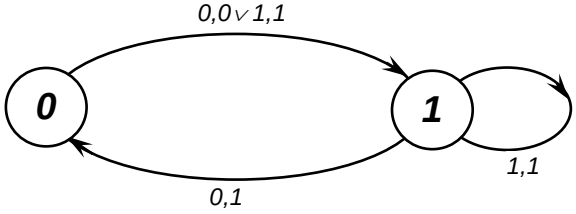
x	q'	q	y_2
0	0	1	0
0	1	0	1
1	0	1	1
1	1	1	1

Анализ условия: из таблицы видно, что схема имеет один вход (x), один выход (y) и один элемент задержки (q). Пока независимыми полюсами схемы следует считать x и q', а вершинами – y и q. На заключительном этапе между q' и q будет включена задержка.

По таблице значений строится таблица автоматных функций:

q / x	*0*	*1*
0	*1,0*	*0,1*
1	*1,1*	*1,1*

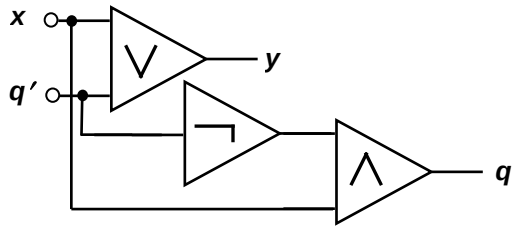
Соответствующая диаграмма Мура:



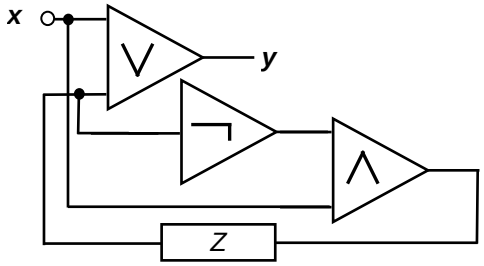
Система автоматных функций будет выглядеть следующим образом:

$$y = x \vee q' \; ; \; q = x \vee \overline{q}'$$

Схема без задержек с промежуточным полюсом $(q)'$ и вершиной q:



После подключения между q' и q элемента задержки окончательная схема автомата будет иметь вид:



Для определения выходного слова используется таблица значений, таблица функций или диаграмма. При этом работу автомата удобно представить по тактам:

	Такты					
	0	*1*	*2*	*3*	*4*	*5*
Вход (x)	–	1	1	0	0	1
Состояние (q)	0	1	1	0	1	1
Выход (y)	–	1	1	1	0	1

Таким образом, последовательность выходных сигналов: 11101

Порядок выполнения работы:

1) Откройте из своей рабочей папки файл с отчетом по 1-й части ЛР **ТОИ_Лр-7(1,2).xls**. Если требуется, установите **средний** уровень безопасности макросов. Повторная регистрация не требуется. Перейдите на лист <**Конечные автоматы**>.
2) В *Задании 1* нужно методом устранения задержек решить задачу анализа для представленной схемы. При построении схем и диаграмм можно использовать имеющийся в отчете конструктор схем, а также встроенный графический редактор MS Excel. На заключительном этапе работы следует определить последовательность сигналов на выходе (выходное слово) по последовательности на входе при том что начальным является состояние автомата 00. Предусмотрена автоматизированная проверка.
3) В *Задании 2а* необходимо «подавать» различные комбинации сигналов на входы «черного ящика» и отслеживать сигналы на выходе (комбинации «0» и «1» вводятся в ячейки, обозначенные x_1, x_2 и q'). Передача сигналов

осуществляется после нажатия экранной кнопки **[Такт]** Для очистки всех входов перед вводом значений необходимо нажать **[Сброс]**. На основании этих наблюдений заполняется таблица значений. По таблице следует установить автоматные функции, построить таблицу функций, найти вид функций переходов и выходов, построить диаграмму и схему.

4) В *Задании 2b* требуется по заданным автоматным функциям построить таблицу значений, таблицу функций, диаграмму, схему, а также определить выходное слово по заданному входному. Предусмотрена автоматизированная проверка.
5) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
6) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Почему конечные автоматы называются «конечными»? Чем они отличаются от комбинационных схем?
2. Что представляют собой элементы памяти конечных автоматов, каковы их функции? Укажите несколько типов элементов памяти.
3. Что такое «базис конечных автоматов»? Какие элементы он включает? Почему?
4. Укажите методы описания конечных автоматов.
5. Сформулируйте правила построения диаграмм Мура.
6. В чем суть метода устранения задержек при решении задач анализа конечного автомата?
7. Укажите общий порядок решения задач анализа конечных автоматов.
8. Укажите общий порядок решения задач синтеза конечных автоматов.
9. Каким образом в конечном автомате можно определить выходные сигналы по последовательности входных?
10. В чем суть «экспериментирования» в *Задании 2a*?

Лабораторная работа № 8.
Алгоритмическая машина Тьюринга

***Учебная задача*:**

Научиться программировать эмулятор машины Тьюринга.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», п.9.3.3.

Рабочий файл: **ALGO2000.exe**

Необходимые теоретические сведения

Описание машины Тьюринга

Машина Тьюринга – абстрактный исполнитель (абстрактная устройство по обработке дискретной информации). Машина Тьюринга (МТ) является расширением конечного автомата и, согласно тезису Чёрча-Тьюринга, способна имитировать все другие исполнители (с помощью задания правил перехода), каким-либо образом реализующие процесс пошаговой обработки, в котором каждый шаг обработки достаточно элементарен.

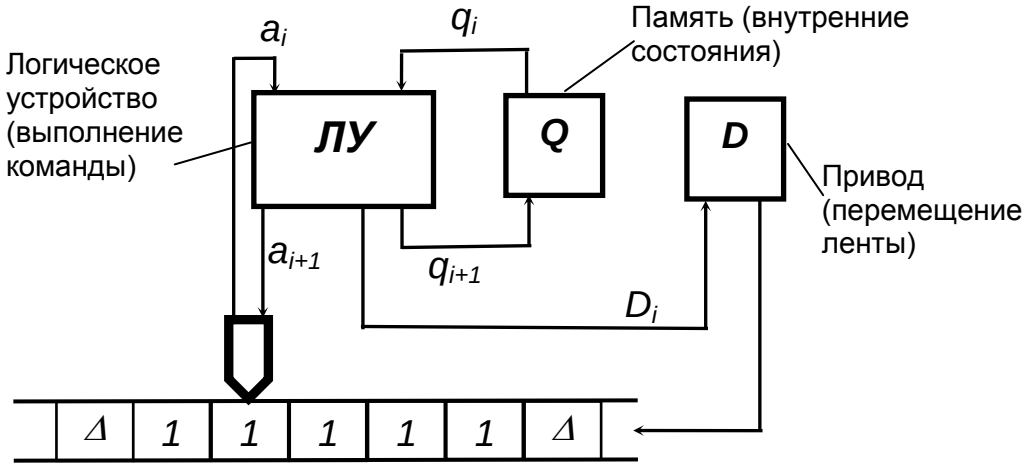


Рис. 1. Схема функционирования машины Тьюринга

В состав МТ входит *бесконечная в обе стороны лента*, разделенная на ячейки (внешняя память), *управляющее устройство* (УУ) с конечным числом состояний (внутренняя память) и считывающе-записывающее устройство (*обозреватель*), которое на каждом шаге обозревает какую-то ячейку ленты и по команде управляющего устройства может изменить ее содержание (см. рис. 1).

МТ для конкретной задачи задается перечислением элементов множества знаков внешнего алфавита *{A}*, множества внутренних состояний *{Q}* и набором правил, по которым работает машина. Они имеют вид: $a_j q_i \rightarrow a_j' D_k q_i'$, т.е. после обзора символа a_j обозревателем при нахождении УУ в состоянии q_i, в ячейку записывается символ a_j', УУ переходит в состояние q_i', а лента

совершает перемещение D_k. В D_k входят 3 команды: «сдвиг на 1 ячейку право», «сдвиг на 1 ячейку влево» и «остаться на месте». Хотя в «классической» машине Тьюринга производился сдвиг ленты относительно обозревателя, можно вести речь о перемещении обозревателя относительно ленты – именно так принято в используемом в данной лабораторной работе эмуляторе.

Для каждой комбинации $a_j q_i$ имеется *ровно одно* правило преобразования. Это означает, что УУ реализует функцию, сопоставляющую каждой паре входных сигналов $a_j q_i$ одну и только одну тройку выходных $a_j D_k q_i'$ – она называется *логической функцией машины* и обычно представляется в виде таблицы (*функциональной схемы машины*), столбцы которой обозначаются символами состояний, а строки – знаками внешнего алфавита.

Помимо знаков внешнего алфавита *{A}* на ленте может размещаться *пустой знак* (*Λ,* в нашем случае «_») – его отсылка в занятую ячейку эквивалентна стиранию ее содержания. Все ячейки, незанятые знаками *{A},* считаются занятыми пустыми знаками.

В алфавит внутренних состояний УУ *{Q}* помимо *рабочих состояний* (находясь в них МТ производит обработку информации) обязательно входит состояние *останова*, по достижении которого МТ прекращает работу.

Таким образом, если *{A}* содержит *N* знаков, а *{Q}* – *M* состояний, то общее число правил преобразования (число ячеек в функциональной таблице), очевидно, равно *(N+1)×(M+1)*. Вообще говоря, не все ячейки таблицы оказываются обязательными к заполнению. Программирование МТ состоит в разработке данной таблицы.

Перед началом работы МТ находится в *исходной конфигурации* – в ячейки ленты занесены знаки внешнего алфавита, обозреватель располагается над некоторой ячейкой.

Работа МТ происходит тактами. На каждом такте на вход УУ подается обозреваемый в данный момент знак ленты и текущее внутреннее состояние – на основании этих данных в соответствии с таблицей преобразований выбирается команда, в соответствии с которой в ячейку записывается новый символ, машина переходит в новое состояние, а обозреватель может быть сдвинут относительно ленты на 1 ячейку вправо или влево.

В зависимости от начальной конфигурации и таблицы преобразований возможны два варианта развития событий:

- после конечного числа тактов машина останавливается по команде останова; при этом на ленте оказывается конечная конфигурация, соответствующая выходной информации;
- остановки не происходит (*зацикливание*).

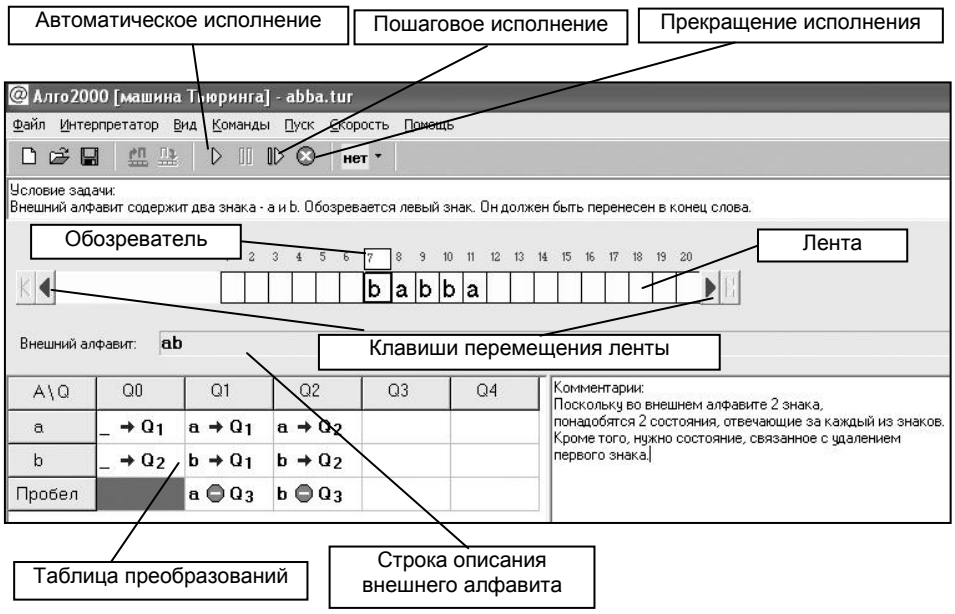
В первом случае говорят, что данная машина применима к начальной информации, во втором – нет. На практике это означает, что были допущены ошибки при составлении таблицы преобразований.

Вся совокупность входных конфигураций, при которых машина обеспечивает получение результата, образуют *класс решаемых задач*. Для каждого класса задач необходима своя МТ.

Работа с программой ALGO2000

Как уже отмечалось, МТ является *абстрактной* машиной, осуществляющей обработку дискретной информации. Это означает, что построить такую машину в принципе невозможно (хотя бы потому, что элементом МТ является бесконечная лента). Вместе с тем, программным путем можно реализовать экранный *эмулятор* МТ, который будет обладать всеми качествами МТ, за исключением неограниченной внешней памяти. Программирование таких эмуляторов весьма полезно для освоения алгоритмизации.

В настоящей лабораторной работе предполагается использование эмулятора МТ ALGO2000 (автор Р. Зартдинов). Ниже показано размещение элементов МТ на экране:



В программе принят следующий порядок ввода информации:

1) внешний алфавит вводится с клавиатуры в отведенную строку;
2) начальная конфигурация записывается на ленте с клавиатуры; при попытке ввода знака, не входящего во внешний алфавит, будет выдано предупреждение; начальное положение обозревателя устанавливается нажатием экранных клавиш перемещения ленты;
3) для ввода команды в ячейку таблицы нужно выделить ее щелчком мыши, после чего ввести команду с клавиатуры; приняты следующие обозначения:

- «пробел» или знак «_» (подчеркивание) – пустой знак;
- знаки внешнего алфавита – с клавиатуры;
- знаки «>» и «<» – сдвиг обозревателя относительно ленты, соответственно, вправо или влево;
- знак «!» – команда останова; после нее указывается номер любого состояния;
- состояние в команде задается только его номером.

Например, при наборе: *a>2* в ячейку будет записано: $\mathbf{a \rightarrow Q_2}$

- очистка элементов таблицы может осуществляться клавишей [Del] или с помощью меню, вызываемого правой клавишей мыши;
- запуск программы к исполнению производится при нажатии соответствующих пиктограмм в режиме *пошаговом* (трассировка) или *автоматическом*; правее находится пиктограмма принудительной остановки выполнения программы.

На рис. 1 показан пример решения задачи о переносе первого обозреваемого на ленте знака в конец конфигурации.

Порядок выполнения работы

1) Скопируйте в свою рабочую папку программу ALGO2000. Запустите ее к исполнению. При необходимости переведите интерпретатор в режим *Машина Тьюринга*. Для знакомства с интерфейсом программы повторите решение задачи, изображенной на рисунке, приведенном выше.
2) К решению предлагаются задачи нескольких уровне сложности: *1 уровень* – 2-3 внутренних состояния, *2 уровень* – для решения требуется 5-7 состояний, *3 уровень* – 10-15 состояний. Решите *Задачи тренинга* по 1-2 из каждой категории для лучшего освоения программирования машины Тьюринга.
3) Выполните задания индивидуального варианта. Номер варианта (N_{vi}) определяется по следующему правилу: $N_{vi} = 1 + N_i \bmod 10$, где i – Ваш порядковый номер в списке группы, а *mod* – результат целочисленного деления N_i на *10*, например, для $i = 12$ будет $N_{vi} = 3$. Вариант содержит 3 задания с оценками: 1-го уровня – 1 балл; 2-го уровня – 2 балла; 3-го уровня – 3 балла.
4) Преподавателю в качестве отчета по оговоренной схеме взаимодействия сдаются: 3 файла с решением задач, а также текстовый файл с ответом на теоретический вопрос. Номер вопроса соответствует номеру индивидуального варианта. Вопросы приведены ниже. Необходимо соблюдать двузначную нумерацию задач в индивидуальных заданиях, например, 5.3. (первая цифра – номер варианта, вторая – номер задачи).

Задачи тренинга

1-й уровень

1.1. **Инверсия** (файл ***1_inv.tur***)

Внешний алфавит: {**0, 1**}.

Начальная конфигурация: произвольная последовательность 0 и 1; обозревается произвольный знак последовательности.

МТ должна осуществить инверсию исходного слова, т.е. заменить все 0 на 1, а 1 на 0 и остановить обозреватель над первым знаком.

1.2. **Сложение в унарной СС** (файл ***1_un_plus.tur***)

Внешний алфавит: {1, +}

Начальная конфигурация: произвольные слагаемые в унарной системе счисления, разделенные знаком + (например, 11111+11); изначально обозревается произвольный знак.

МТ должна выполнить сложение и представить результат в унарной СС.

1.3. **Умножение на 2** (файл ***1_mult2.tur***)

Внешний алфавит: {**0, 1**}. Начальная конфигурация: двоичное целое число; обозревается произвольный знак.

Необходимо осуществить умножение числа на 2.

2-й уровень

2.1. **Добавление 1 к числу в 4-й СС** (файл ***2_4SS_plus1.tur***)

Начальная конфигурация представляет запись формулы сложения произвольного целого числа в 4-й СС и 1 (например, 1230+1=); обозревается произвольный знак конфигурации. На ленте должен быть представлен результат сложения.

2.2. **Дополнительный код числа** (файл ***2_DK.tur***)

Начальная конфигурация: на ленте ненулевое отрицательное двоичное число (знак «–» указан явно); обозревается произвольный знак. Требуется построить дополнительный код числа.

2.3. **Минус 1 в 3-й СС** (файл ***2_3SS-1.tur***)

A = {0,1,2,–}. На ленте запись выражения с вычитанием 1 из числа в троичной СС (например, *120–1=*); обозревается произвольный знак конфигурации. Выполнить вычитание чисел в 3-й СС.

3-й уровень

3.1. **Смена знаков** (файл ***3_Change.tur***)

A = {a,b}. Начальная конфигурация – произвольное слово из знаков алфавита; обозревается произвольный знак. Требуется в исходном слове поменять местами его первый и последний знаки.

3.2. Сложение в 5-й СС (файл ***3_5Sum.tur***)

Необходимо реализовать сложение в 5-й СС; начальная запись, например, *123+14=*; в конце на ленте результат сложения. Изначально обозревается произвольная позиция конфигурации.

3.3. ***Замена bb*** (файл ***3_bb.tur***)

A = {a,b}. На ленте слово из знаков алфавита; обозревается произвольный знак. Заменить все двукратные вхождения *bb* на *b*.

Варианты индивидуальных заданий

Вариант 1.

1.1. Необходимо реализовать умножение на 9 целого троичного числа. Изначально обозревается произвольная цифра числа.

1.2. *A = {0,1,+}.* На ленте запись выражения с суммированием двух чисел в двоичной системе счисления (например, *1001+11=*); обозревается произвольный знак конфигурации. Выполнить сложение чисел в 2-й СС.

1.3. *A = {a,b}.* Начальная конфигурация – произвольное слово из знаков алфавита; обозревается произвольный знак. Требуется удвоить каждый символ исходного слова (например, *bab* → *bbaabb*).

Вариант 2.

2.1. Необходимо проверить, является ли записанное число в 2-й СС четным Ответ: на ленте 1 (да), 0 (нет). Изначально обозревается произвольная цифра числа.

2.2. *A = {a,b}.* На ленте слово из знаков алфавита; обозревается произвольный знак. Необходимо определить, имеется ли вхождение двойных букв *a* или *b* в исходном слове на ленте. Ответ: на ленте 1 (да), 0 (нет).

2.3. Необходимо реализовать вычитание в 10-й СС; начальная запись, например, *103 – 12=*; в конце на ленте результат вычитания. Обозревается произвольная позиция конфигурации. У результата не должно оставаться незначащих нулей.

Вариант 3.

3.1. Необходимо у троичного целого числа удалить незначащие нули, если такие есть. Изначально обозревается произвольная цифра числа.

3.2. *A = {a,b}.* На ленте слово из знаков алфавита; обозревается произвольный знак. Заменить в исходном слове каждое вхождение *a* на *bb.*

3.3. Начальная конфигурация – запись произвольного четного числа в унарной СС; обозревается произвольная цифра. Требуется выполнить деление исходного числа на 2.

Вариант 4.

4.1. Необходимо проверить, является ли записанное число в 4-й СС четным Ответ: на ленте 1 (да), 0 (нет). Изначально обозревается произвольная цифра числа.

4.2. *A = {a,b}.* На ленте 2 слова из знаков алфавита, разделенные проделом (пустым знаком); обозревается произвольный знак правого слова. Требуется «склеить» слова, т.е. удалить разделяющий пробел.

4.3. *A = {I,0,1}.* Начальная конфигурация – запись произвольного числа в унарной СС; обозревается произвольная цифра. Необходимо преобразовать унарную запись числа в двоичную.

Вариант 5.

5.1. Знаки алфавита имеют следующие числовые коды: *a – 7; b – 9; c – 3; d – 6*. Начальная конфигурация: произвольная последовательность знаков алфавита; обозревается произвольный знак слова. Произвести кодирование.

5.2. *A = {0,1,–}.* На ленте запись выражения с вычитанием двух чисел в двоичной системе счисления (например, *1001–11=*); обозревается произвольный знак конфигурации. Выполнить вычитание чисел в 2-й СС.

5.3. *A = {a,b}.* Начальная конфигурация – произвольное слово из знаков алфавита; обозревается произвольный знак. удвоить исходное слово (например, *abb → abbabb*).

Вариант 6.

6.1. Необходимо проверить, делится ли записанное на ленте десятичное сило на 5; изначально обозревается произвольная цифра. Ответ: на ленте 1 (да), 0 (нет).

6.2. *A = {0,1,2,+}.* На ленте запись выражения с суммированием произвольного и одноразрядного чисел в 3-й СС (например, *1012+2=*); обозревается произвольный знак начальной конфигурации. Осуществить сложение.

6.3. *A = {a,b}.* Начальная конфигурация – произвольное слово из знаков алфавита; обозревается произвольный знак. Требуется перевернуть исходное слово (например, *abb → bba).*

Вариант 7.

7.1. Знаки алфавита имеют следующие числовые коды букв русского алфавита: *т – 4; е – 8; с – 2; в – 9, о – 3*. Начальная конфигурация: последовательность кодов; обозревается произвольный знак. Произвести декодирование.

7.2. *A = {0,1,2,3,–}.* На ленте запись выражения с вычитанием одноразрядного числа из произвольного в 4-й СС (например, *1032–3=*); обозревается произвольный знак начальной конфигурации. Осуществить вычитание.

7.3. *A = {a,b}.* Начальная конфигурация – произвольное слово из знаков алфавита; обозревается произвольный знак. Требуется разделить буквы слова пробелами (например, *abbab → a b b a b*).

Вариант 8.

8.1. На ленте запись двоичного числа. Необходимо определить, делится ли оно на 4. Ответ: на ленте 1 (да), 0 (нет). Изначально обозревается произвольная цифра числа.

8.2. *A = {0,1,–}.* Начальная конфигурация: на ленте произвольное ненулевое двоичное число (знаки «–» указан явно, «+» – не указывается); обозревается произвольный знак. Требуется построить дополнительный код числа.

8.3. *A = {a,b,0,1,2,3,4,5}.* Начальная конфигурация – произвольное слово из знаков *a* и *b*; обозревается произвольный знак. Необходимо подсчитать,

сколько раз (не более 5) в слово входит сочетание *aa*. Ответ должен быть записан на ленте справа от слова.

Вариант 9.

9.1. *A = {a,b}*. Необходимо определить, имеется ли вхождение двойных букв *a* в исходном слове на ленте. Ответ: на ленте 1 (да), 0 (нет). Изначально обозревается произвольный знак слова.

9.2. *A = {I,0,1}*. Начальная конфигурация – запись произвольного числа в унарной СС; обозревается произвольная цифра. Необходимо реализовать умножение на 2 в унарной системе счисления.

9.3. Начальная конфигурация – произвольное целое число в 3-й СС. Обозревается произвольная цифра числа. Требуется перевести его в 4-ю СС.

Вариант 10.

10.1. *A = {a,b,c}*. Начальная конфигурация: произвольная последовательность знаков внешнего алфавита; обозревается произвольный знак слова. Произвести замену знаков *a* на *b*, *b* на *c*.

10.2. Необходимо реализовать сложение в 10-й СС; начальная запись, например, *123+17=*; в конце на ленте результат сложения. Обозревается произвольная позиция конфигурации.

10.3. *A = {0,1}*. Необходимо изменить на обратный порядок следования цифр в записи целого двоичного числа (например, *10110* $\rightarrow$ *01101*). Изначально обозревается произвольная цифра числа.

Контрольные вопросы:

1. Почему машину Тьюринга можно считать предельным случаем конечного автомата? Почему МТ *не является* конечным автоматом?
2. Почему машина Тьюринга называется *абстрактной*?
3. Укажите назначение основных компонентов МТ?
4. Что такое «*конфигурация МТ*»?
5. Какова структура команды в МТ? В чем суть программирования МТ?
6. Опишите функционирование МТ в процессе решения задачи.
7. Если в процессе исполнения алгоритма на МТ происходит «зацикливание», о чем это может свидетельствовать?
8. Можно ли построить универсальную МТ, способную решить любую задачу по обработке информации в дискретной форме представления?
9. Что общего и чем различаются МТ и компьютер? Можно ли считать компьютер машиной Тьюринга?
10. Почему программа, которая была использована в данной лабораторной работе, называется «*эмулятор МТ*»?

Теоретические Основы Информатики

Лабораторный практикум

Лабораторная работа № 9.

Освоение методов криптографии

***Учебная задача*:**

Освоить одно- и двухключевые методы шифрования.

Перед выполнением работы рекомендуется прочитать учебник Б.Е. Стариченко «Теоретические основы информатики», Глава 7.

Часть 1. Методы симметричного шифрования (одноключевые)

Рабочий файл: **ТОИ_Лр-9(1).xls**

Необходимые теоретические сведения

Определения

Криптографическим преобразованием (шифрованием) называется приведение составляющих элементов исходной информации (слов, букв, цифр) с помощь специальных алгоритмов к виду, не позволяющему воспроизвести исходные данные без знания правила обратного преобразования (восстановления) или специального ключа.

Алфавит – конечное множество знаков, используемых для представления первичной информации.

Открытый (исходный) *текст* – упорядоченный набор знаков алфавита, обладающий семантической значимостью (смыслом).

Шифрованный (закрытый) *текст* (*криптограмма*) – данные, полученные после применения шифра к открытому тексту.

Шифр – алгоритм или однозначные отображения открытого текста в шифрованный.

Ключ – некоторый секретный параметр шифра (обычно – последовательность знаков алфавита), позволяющий выбрать для шифрования только одно конкретное преобразование из всего множества преобразований, составляющих шифр.

Шифрование – процесс нормального применения криптографического преобразования открытого текста на основе алгоритма и ключа, в результате которого возникает шифрованный текст.

Расшифровывание – процесс нормального применения криптографического преобразования шифрованного текста в открытый.

Дешифрование (дешифровка) – процесс извлечения открытого текста из шифрованного *без знания* шифра и/или криптографического ключа.

Методы симметричного шифрования, которые рассматриваются в данной работе***:***

- подстановки (замены): одноалфавитной замены (Цезаря), одноалфавитной многократной замены, Виженера (с ключевой фразой и автоключом);
- перестановки: с фиксированным периодом, блочной.

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файлы **ТОИ_Лр-9(1).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов (меню ***Сервис→****Макрос→Безопасность*). Зарегистрируйтесь на листе <**Криптография**>. В процессе регистрации будет сгенерировано индивидуальное задание: исходный текст, а также двузначная кодировка букв русского алфавита. Обратите внимание на комментарий (3) в области «Исходные данные»
2) В *Задании 1a* исходный алфавит нужно скопировать и вставить во вторую строку со сдвигом вправо или влево на указанный ключ – число знаков (в нижней строке показана шкала сдвига); скопировать часть, выходящую за исходный алфавит и поставить ее перед или после (в зависимости от направления сдвига); очевидно, начальные и конечные знаки алфавитов в обеих строках должны совпасть. Произвести шифрование текста с помощью «сдвинутого» алфавита. Проверить правильность шифрования.
3) В *Задании 1b* требуется произвести обратную операцию: дешифровать фразу, подобрав подходящий сдвиг алфавита. Критерием правильности выполнения задания будет «осмысленность» конечной фразы.
4) *Задание 1c* предполагает использование многократной (4-х кратной) замены: ключ содержит 4 цифры со знаками, по которым нужно построить 4 «сдвинутых» алфавита. Далее исходный текст записывается по колонкам в таблицу с 4-мя строками. Каждая строка шифруется независимо по соответствующему «сдвинутому алфавиту; результаты заносятся в другую таблицу. Зашифрованная фраза считывается по столбцам 2-й таблицы. Проверьте правильность шифрования.
5) В *Заданиях 1d* и *1e* для шифрования должен быть использован метод Виженера. Используются числовые коды букв алфавита, указанные в разделе «Исходные данные». Порядок определения кодов Виженера приведен в задании. Проверьте правильность шифрования.
6) В *Задания 2a* и *2b* шифрование осуществляется методом перестановок. Указания по применению методов содержатся в заданиях. Проверьте правильность шифрования.
7) *Задание 2c* предполагает дешифровку указанной фразы путем подбора ключа блочной перестановки.
8) *Задание 3* предполагает описание и пример Вашего метода шифрования.
9) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.

10) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. В чем схожесть и различие операций помехоустойчивого кодирования и шифрования сообщения?
2. Что такое криптостойкость? Какими факторами она определяется?
3. Почему методы симметричного шифрования называют еще методами с закрытым ключом и одноключевыми методами?
4. Почему считается, что метод шифрования Цезаря обладаем слабой криптостойкостью?
5. Как реализуются методы двухалфавитной замены? Можно ли к ним отнести *транлитерацию* – замену букв русского алфавита буквами латинского?
6. Предложите алгоритм шифрования-расшифрования с использованием книги (как шифровальной таблицы). К какой категории методов шифрования он относится? Как повысить крипкостойкость этого метода?
7. Поясните суть формулы расчета кода в методе Виженера.
8. Что является ключом в шифровании методом *гаммирования*? В чем достоинства и недостатки этого метода шифрования?
9. На основании каких сведений криптоаналитик может дешифровать зашифрованные тексты?
10. В чем отличия симметричных и асимметричных криптосистем? В чем их достоинства и недостатки?

Часть 2. Формирование ключей и шифрование в криптосистеме RSA

Рабочий файл: **ТОИ_Лр-9(2).xls**

Необходимые теоретические сведения

Определения:

- под *простым* числом понимается такое число, которое делится только на единицу и на само себя;
- *взаимно-простыми* называются такие числа, которые не имеют ни одного общего делителя, кроме единицы;
- под результатом операции *A mod B* понимается остаток от целочисленного деления *A* на *B* (*48 mod 9 = 3*).

Алгоритм RSA

Алгоритм основан на том, что нахождение больших простых чисел в вычислительном отношении осуществляется легко, но разложение на множители произведения двух таких чисел практически невыполнимо.

Алгоритм подразумевает 2 этапа работы: (1) генерация открытого и закрытого ключей; (2) использование ключей для шифрования-расшифрования текстов.

Этап 1. Генерация ключей.

1) Выбирается два больших ***простых*** числа q и p (например *100* десятичных разрядов; в данной работе – из интервала от 10 до 30), находится их произведение $n = p \cdot q$; число n будет ограничивать сверху допустимый объем передаваемого шифрованного фрагмента (например, при размере q и p по *100* десятичных разрядов, n будет содержать *200* десятичных разрядов, что соответствует длине шифруемого фрагмента около *660* бит; в нашем случае решается учебная задача, поэтому n содержит только *3* десятичных или *10* двоичных разрядов, чего, впрочем, достаточно для побайтного (алфавитного) шифрования).
2) Вычисляется значение функции Эйлера $\Phi(n) = (p - 1) \cdot (q - 1)$.
3) Выбирается произвольное достаточно большое число e, взаимно простое с $\Phi(n)$ – оно будет принято в качестве открытого ключа; в данной работе предлагается выбрать e, содержащее 2 десятичных разряда; e и n составляют открытый ключ, который сообщается всем пользователям, желающим отправить сообщение владельцу закрытого (секретного) ключа.
4) Выбирается произвольное d, удовлетворяющее уравнению $ed \bmod \Phi(n) = 1$ и условию и $d < \Phi(n)$.
5) За значение открытого ключа принимается пара (e, n), секретного ключа – пара (d, n).
6) Помимо ключей должно быть осуществлено числовое кодирование букв первичного алфавита – коды, естественно, должны быть известны как отправителям, так и получателю.

Этап 2. Шифрование-расшифрование текстов.

1) Шифрование текста производит его отправитель. Для этого коду каждой буквы текста (w_i) ставится в соответствие код шифра (u_i) по правилу:

$$u_i = (w_i)^e \ mod \ n \qquad (9.1)$$

Не зная закрытого ключа расшифровать последовательность кодов шифра невозможно, даже зная открытый ключ.

2) Расшифрование текста производит его получатель с помощью закрытого (секретного) ключа по правилу:.

$$w_i = (u_i)^d \ mod \ n \qquad (9.2)$$

Порядок выполнения работы:

1) Скопируйте в свою рабочую папку файл **ТОИ_Лр-9(2).xls**, откройте его. Если требуется, установите **средний** уровень безопасности макросов (меню ***Сервис→****Макрос→Безопасность*). Зарегистрируйтесь на листе <**Отчет**>.
2) *Задание 1*:
   - придумайте и введите в соответствующие ячейки значения p и q – ***простые*** числа из интервала от 10 до 30; вычислите n и $\Phi(n)$.
   - для подбора d нужно воспользоваться таблицей, в которой для каждого d найдено значение выражения $ed \ mod \ \Phi(n)$; задача – найти значение d, при котором выражение станет равно 1 при выполнении условия $d < \Phi(n)$; поскольку в данной работе d должно содержать не менее 2-х десятичных разрядов, перебор начинается с 10; выводятся одновременно по 25 значений; для перехода к следующим 25 значениям нужно в 1-й ячейке d указать величину на 25 единиц большую – в первом переходе 35 вместо 10, затем 60 вместо 35 и т.д., пока не будет найдено d со значением выражения 1 в нижней строке;
   - введите открытый и закрытый ключи в ячейке таблицы; проверьте правильность результата;
   - в *Задании 1b* требуется назначить 3-х-значные десятичные коды буквам русского алфавита; по-видимому, наиболее простым является вариант ввода кода буквы «А» и последующего автозаполнения для остальных букв алфавита.
3) В *Задании 2a* необходимо ввести текст побуквенно в ячейки первой строки таблицы; во вторую строку ввести числовые коды букв, согласно таблице кодов из задания *1b*. Далее с помощью алгоритма, реализующего расчетную схему (9.1) в нижней части отчета, для каждого кода w_i вычисляется код шифра u_i и заносится в 3-ю строку таблицы. Поскольку используются 3-х-значные коды, если в коде шифра оказывается меньше знаков, он должен быть дополнен до 3-х-значного нулями слева. Проверьте правильность шифрования.
4) В *Задании 2b* необходимо расшифровать текст с помощью закрытого ключа, используя реализацию алгоритма (9.2). Проверьте правильность расшифровки.

5) Дайте ответ на теоретический вопрос в конце отчета. Список вопросов приведен ниже.
6) Отчет сохраните в своей рабочей папке (облаке). По завершении работы представьте его преподавателю по оговоренной схеме взаимодействия.

Контрольные вопросы:

1. Каков принцип работы криптосистемы с открытым ключом?
2. В чем достоинства и недостатки алгоритма RSA?
3. Можно ли для числовых кодов первичного алфавита использовать ASCII-коды? Ответ обоснуйте.
4. Как можно было бы автоматизировать нахождение ключа *d*? Каков принцип работы такой программы?
5. Как можно было автоматизировать шифрование-дешифрование методом RSA? Какие модули должна содержать программа шифрования.

Учебное издание

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по курсу «Теоретические основы информатики»

Уральский государственный педагогический университет.
620017 Екатеринбург, пр-т Космонавтов, 26.
E-mail: uspu@uspu.me