МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Кафедра вычислительных систем и сетей

МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ ДЛЯ ПОДГОТОВКИ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

по курсу «**Теория кодирования**»

на весенний семестр для студентов заочного отделения

специальности 40 01 01

«Программное обеспечение информационных технологий»

Старший преподаватель кафедры ВСиС Калинцев С.В.

ПОЛОЦК 2013

ВВЕДЕНИЕ

Предложенные методические рекомендации предназначены для студентов заочного факультета специальностей 1-40 01 01 – «Программное обеспечение информационных технологий» для облегчения усвоения курса – «Теория кодирования» и помощи студентам-заочникам в решении типовых задач по рассматриваемому курсу.

Вопросы, предложенные для самоподготовки студентов, помогут укрепить и углубить знания студентов по основным темам дисциплины, а решение практической части позволит сформировать навыки расчета задач кодирования.

СОДЕРЖАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К РЕШЕНИЮ ЗАДАЧ

В следующих методических указаниях приводятся основные определения, а также алгоритмы кодирования их оценки. Кроме того, рассмотрены примеры решения типовых задач, соответствующих задачам, предлагаемым студентам для самостоятельного решения.

1. Рассмотрим некоторые коды, позволяющие сделать передачу информации более эффективной, что достигается путем сжатия сообщения. Начнем с *кода Шеннона–Фано*. Алгоритм состоит в следующем: буквы исходного алфавита сообщения выписываются в столбец в порядке убывания их вероятностей; производится разбиение на две группы с равной по возможности суммарной вероятностью, всем буквам верхней группы в качестве первого символа кодовой комбинации приписывается «1», а нижней – «0»; затем производятся следующие разбиения до тех пор, пока в каждой подгруппе не останется одна буква (при каждом разбиении появляется новый символ кодовой комбинации по правилам, изложенным выше).

Эффективность кода рассчитывается следующим образом

$$æ=\frac{H(Z)}{l_{cp}\log_2 h},$$

причем в случае двоичного алфавита формула имеет вид

$$æ=\frac{H(Z)}{l_{cp}}, \qquad (1)$$

где средняя длина кодовой комбинации равна

$$l_{cp}=\sum_i n(z_i)\,p(z_i), \qquad (2)$$

$n(z_i)$ – число символов в кодовой комбинации. Эффективность является безразмерной величиной и всегда меньше либо равна 1, т.е. æ ≤ 1. Чем ближе этот показатель к единице, тем эффективнее код.

<u>Пример 1.</u> Проведем кодирование методом Шеннона–Фэно и рассчитаем характеристики кода. Пусть исходный алфавит состоит из восьми букв и заданы их вероятности. Проведем разбиения по алгоритму Шеннона–Фэно и составим кодовые комбинации.

i	$p(z_i)$	кодовые комбинации			
z_1	0,25	1	<u>1</u>		
z_2	0,20	<u>1</u>	0		
z_3	0,15	0	1	<u>1</u>	
z_4	0,10	0	<u>1</u>	0	
z_5	0,10	0	0	1	<u>1</u>
z_6	0,10	0	0	<u>1</u>	0
z_7	0,06	0	0	0	<u>1</u>
z_8	0,04	0	0	0	0

Рассчитаем среднюю длину кодовой комбинации по формуле (2)

$$l_{cp} = 0{,}25 \cdot 2 + 0{,}20 \cdot 2 + 0{,}15 \cdot 3 + 0{,}10 \cdot 3 + 0{,}10 \cdot 4 +$$
$$+ 0{,}10 \cdot 4 + 0{,}06 \cdot 4 + 0{,}04 \cdot 4 \approx 2{,}85.$$

Эффективность кода, согласно (1), равна

$$æ \approx \frac{2{,}79}{2{,}85} \approx 0{,}98.$$

Эффективность кодирования можно увеличить, если проводить кодирование блоков, состоящих из нескольких букв алфавита.

<u>Пример 2</u>. Проведем блоковое кодирование по методу Шеннона–Фано. Пусть алфавит состоит из двух независимых букв с заданными вероятностями $p(z_1)$=0,8 и $p(z_2)$=0,2.

Очевидно, что при кодировании по одной букве l_{cp}=1 и $æ_1$=0,72.

Проведем кодирование блоков, состоящих из двух букв. Ниже приведена таблица с разбиениями и соответствующими кодовыми комбинациями.

$z_i\, z_j$	$p(z_i\, z_j)$	код. комб.
$z_1\, z_1$	0,64	$\underline{1}$
$z_1\, z_2$	0,16	0 $\underline{1}$
$z_2\, z_1$	0,16	0 0 $\underline{1}$
$z_2\, z_2$	0,04	0 0 0

При расчете вероятностей блоков использовали теорему умножения вероятностей для независимых событий

$$p(z_i\, z_j) = p(z_i) \cdot p(z_j).$$

Рассчитаем основные характеристики

$$l_{cp} = 0{,}64 \cdot 1 + 0{,}16 \cdot 2 + 0{,}16 \cdot 3 + 0{,}04 \cdot 3 \approx 1{,}56;$$

$$æ_2 \approx \frac{2 \cdot 0{,}72}{1{,}56} \approx 0{,}92\,.$$

При кодировании блоков из трех букв эффективность возрастает еще больше.

$z_1\, z_1\, z_1$	0,512	$\underline{1}$
$z_1\, z_1\, z_2$	0,128	0 1 $\underline{1}$
$z_2\, z_1\, z_1$	0,128	0 $\underline{1}$ 0
$z_1\, z_2\, z_1$	0,128	0 0 $\underline{1}$
$z_1\, z_2\, z_2$	0,032	0 0 0 1 $\underline{1}$
$z_2\, z_2\, z_1$	0,032	0 0 0 $\underline{1}$ 0
$z_2\, z_1\, z_2$	0,032	0 0 0 0 $\underline{1}$
$z_2\, z_2\, z_2$	0,008	0 0 0 0 0

$$l_{cp} = 0{,}512 \cdot 1 + 0{,}128 \cdot 9 + 0{,}032 \cdot 15 + 0{,}008 \cdot 5 \approx 2{,}184;$$

$$æ_3 \approx \frac{3 \cdot 0{,}72}{2{,}184} \approx 0{,}98\,.$$

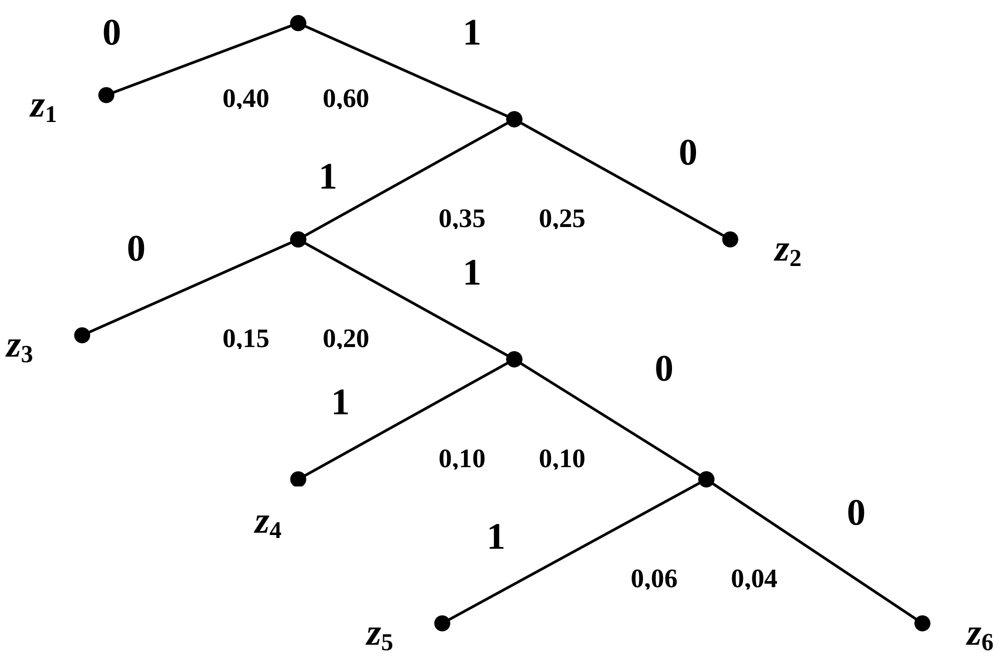
Таким образом, имеем $\text{æ}_1 < \text{æ}_2 < \text{æ}_3$.

Алгоритм *кода Хаффмана* состоит в следующем. Буквы исходного алфавита выписываются в столбец в порядке убывания их вероятностей. Последние две буквы столбца объединяются в одну – вспомогательную букву, которой приписывается суммарная вероятность. Затем формируется следующий столбец с учетом новой буквы по принципу убывания вероятностей. Процесс повторяется и продолжается до тех пор, пока останется одна буква с вероятностью, равной 1. Кодовые комбинации легко получить, построив кодовое дерево. Вершиной дерева является последняя буква, процесс ветвления проводится с учетом полученной таблицы, двигаясь в обратном направлении. Каждому из двух ребер, участвующих в объединении, приписывается кодовый символ: ребру с большей вероятностью – «1», с меньшей – «0». Двигаясь от вершины дерева до одной из букв алфавита по соответствующим ребрам, получаем ее кодовую комбинацию.

Пример 3. Проведем кодирование по методу Хаффмена. Исходный алфавит состоит из шести букв с заданными вероятностями. Составим таблицу.

z_i	$p(z_i)$	Вспомогательные столбцы				
z_1	0,40	0,40	0,40	0,40	0,60	1,00
z_2	0,25	0,25	0,25	0,35	0,40	
z_3	0,15	0,15	0,20	0,25		
z_4	0,10	0,10	0,15			
z_5	0,06	0,10				
z_6	0,04					

Строим кодовое дерево и выписываем кодовые комбинации букв.



z_1	z_2	z_3	z_4	z_5	z_6
0	10	110	1111	11101	11100

Характеристики кода рассчитываются по тем же формулам, что и для кода Шеннона–Фано

$$l_{ср} = 0{,}40 \cdot 1 + 0{,}25 \cdot 2 + 0{,}15 \cdot 3 + 0{,}10 \cdot 4 + 0{,}10 \cdot 4 + 0{,}06 \cdot 5 + 0{,}04 \cdot 5 \approx 2{,}25;$$

$$æ \approx \frac{2{,}20}{2{,}25} \approx 0{,}98.$$

Код Хаффмана можно использовать и для кодирования блоков из букв так, как это было рассмотрено выше для кода Шеннона–Фано, что увеличит эффективность передачи информации.

2. Важнейшей задачей кодирования является обеспечение достоверной передачи информации. Поэтому обнаружение и исправление ошибок приобретает актуальное значение. Рассмотрим некоторые коды, позволяющие решать эти задачи.

Код с проверкой четности позволяет обнаружить однократную ошибку, т. е. ошибку в одном разряде двоичного слова. Алгоритм кодирования состоит в следующем: к передаваемому двоичному слову добавляем в конце один символ («0» или «1») с тем, чтобы сумма всех символов была равна 0. Суммирование проводится по модулю 2 (mod 2)

$$0 \oplus 0 = 0,$$
$$0 \oplus 1 = 1,$$

$$1 \oplus 0 = 1,$$
$$1 \oplus 1 = 0.$$

При декодировании проверяем сумму символов принятого слова. Если она не равна 0, значит, имеем однократную ошибку.

Пример 4. Проведем кодирование сообщения $a = 010110$ кодом с проверкой четности.

Суммируем символы заданного сообщения, в результате получим
$$0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1.$$
Следовательно, добавочный символ равен 1 и закодированное сообщение имеет вид
$$b = 0101101.$$

Теперь рассмотрим код, позволяющий исправить однократную ошибку. Это – *код Хэмминга*. Разберем алгоритм кода. При кодировании сообщение разбивается на слова длиной m, к слову добавляется r контрольных символов. Таким образом, закодированное слово имеет длину $n = m + r$, причем выполнено
$$2^r \geq n + 1.$$
Например, если $m = 4$, то легко видеть, что
$$2^3 \geq (4 + 3) + 1 = 8$$
и таким образом, число контрольных символов $r = 3$ и длина кодового слова $n = 7$. В этом случае имеем (4, 7) код Хэмминга. Контрольные символы размещаются в разрядах с номерами, равными степеням 2, т. е. $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, и т. д. Информационные символы располагаются по порядку в оставшихся разрядах. Если исходное слово $a = a_1\, a_2\, a_3\, a_4$, а закодированное – $b = b_1b_2b_3b_4b_5b_6b_7$, то для определения значений контрольных символов b_1, b_2, b_4 используем следующую систему
$$\begin{cases} b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0, \\ b_2 \oplus b_3 \oplus b_6 \oplus b_7 = 0, \\ b_1 \oplus b_3 \oplus b_5 \oplus b_7 = 0. \end{cases} \tag{13}$$

Пример 5. Закодируем двоичное слово $a = 0101$ кодом (4, 7) Хэмминга. Учитывая, что $b_3 = 0$, $b_5 = 1$, $b_6 = 0$, $b_7 = 1$. Для определения контрольных символов используем систему (13)
$$\begin{cases} b_4 \oplus 1 \oplus 0 \oplus 1 = 0, \\ b_2 \oplus 0 \oplus 0 \oplus 1 = 0, \\ b_1 \oplus 0 \oplus 1 \oplus 1 = 0. \end{cases}$$
Следовательно, $b_4 = 0$, $b_2 = 1$, $b_1 = 0$. Таким образом, закодированное слово имеет вид $b = 0100101$.

Рассмотрим процесс декодирования. Если обозначить искаженное слово через $c = c_1c_2c_3c_4c_5c_6c_7$. Тогда, используя следующую систему, определяем разряд, в котором произошла ошибка.

$$\begin{cases} c_4 \oplus c_5 \oplus c_6 \oplus c_7 = e_1, \\ c_2 \oplus c_3 \oplus c_6 \oplus c_7 = e_2, \\ c_1 \oplus c_3 \oplus c_5 \oplus c_7 = e_3. \end{cases} \quad (14)$$

Двоичное число $(e_1e_2e_3)_2$ определяет номер разряда, в котором произошла ошибка.

Пример 6. Декодируем сообщение c = 0101111, если при кодировании использовался (4, 7) код Хэмминга.

Используя систему (12), определяем разряд, в котором произошла ошибка

$$\begin{cases} 1 \oplus 1 \oplus 1 \oplus 1 = 0, \\ 1 \oplus 0 \oplus 1 \oplus 1 = 1, \\ 0 \oplus 0 \oplus 1 \oplus 1 = 0. \end{cases}$$

Следовательно, ошибка в разряде с номером 2: $010_2 = 2_{10}$. Исправляя ошибку и исключая контрольные символы, получаем информационное сообщение a = 0111.

ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ ДЛЯ САМОПОДГОТОВКИ

1. Коды Хаффмена и области их применения.
2. Арифметическое сжатие и области его применения.
3. Сжатие информации с помощью спектральных преобразований.
4. Словарные методы сжатия. Метод Зива-Лемпела.
5. Основные шаги и принцип работы алгоритма сжатия JPEG.
6. Основные шаги и принцип работы алгоритма сжатия JPEG2000.
7. Основные параметры помехоустойчивых кодов.
8. Вероятность правильного и ошибочного декодирования кодового слова.
9. Задание линейных кодов и кодирование информации.
10. Декодирование помехоустойчивых кодов по максимуму правдоподобия.
11. Декодирование помехоустойчивых кодов по синдрому.
12. Мажоритарное декодирование помехоустойчивых кодов.
13. Кодирование и декодирование информации с помощью кодов Хэмминга.
14. Коды Рида-Маллера: основные параметры, формирование, кодирование и декодирование информации, применение.
15. Коды Рида-Соломона: основные параметры, формирование, кодирование и декодирование информации, применение.
16. Параметры и построение сверточных кодов.
17. Описание сверточных кодов с помощью многочленов и матриц.
18. Поле Галуа. Области применения.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ ПОДГОТОВКИ

Задача 1.

Провести кодирование по одной и блоками по две буквы, используя метод Шеннона – Фано. Сравнить эффективности кодов. Известны вероятности появления символов $p(S_1)=p_1$, $p(S_2)=p_2$, $p(S_3)=p_3$.

$\boldsymbol{p_1}$	0,1
$\boldsymbol{p_2}$	0,15
$\boldsymbol{p_3}$	0,75

Задача 2.

Алфавит передаваемых сообщений состоит из независимых букв S_i. Вероятности появления каждой буквы в сообщении заданы. Определить и сравнить эффективность кодирования сообщений методом Хаффмана при побуквенном кодировании и при кодировании блоками по две буквы.

P(Si) = (0,6;0,2;0,08;0,12)

Задача 3.

Декодировать полученное сообщение *c*, если известно, что использовался (7, 4) – код Хэмминга. Провести кодирование кодом с проверкой четности.

C = 1100001

Задача 4.

Найти элементы расширенного поля Галуа. Составить таблицы сложения и умножения элементов поля с использованием заданного полинома. Объяснить результат.

GF ($\mathbf{2^n}$): $f(x) = 2x^4 + x + 1$

Задача 5.

Для заданных k и r построить проверочную и порождающую матрицы. Используя построенные матрицы закодировать произвольно выбранное информационное сообщение, внести одиночную ошибку, декодировать и объяснить полученный результат. Определить кодовое расстояние.

k=4, r=3

$$u_4 = u_0 + u_1 + u_2;$$
$$u_5 = u_0 + u_2;$$
$$u_6 = u_0 + u_2 + u_3;$$

СПИСОК ЛИТЕРАТУРЫ

Основная литература

1. Прикладная теория кодирования: Учеб.-метод. комплекс для студ. спец. 1-39 01 01 «Радиотехника»/Сост. Р.П. Богуш, А.В. Курилович; Под общей редакцией Р.П. Богуша. – Новополоцк: ПГУ,2005. – 256 с.
2. Теория прикладного кодирования. В 2 т. Т. 1: Учеб. пособие/ В.К. Конопелько, В.А. Липницкий, В.Д. Дворников и др.; Под ред. проф. В.К. Конопелько. – Мн.: БГУИР, 2004
3. Теория прикладного кодирования. В 2 т. Т. 2: Учеб. пособие/ В.К. Конопелько, В.А. Липницкий, В.Д. Дворников и др.; Под ред. проф. В.К. Конопелько. – Мн.: БГУИР, 2004
4. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки: Пер. с англ. / Под ред. Р.Д. Добрушина и С.И. Самойленко. – М.: Мир, 1976

Дополнительная литература

1. Вентцель Е.С. Теория вероятностей. М.: Высшая школа, 2002. М.: КНОРУС, 2010.
2. Душин В.К. Теоретические основы информационных процессов и систем. М.: Дашков и К., 2003.
3. Яблонский С.В. Введение в дискретную математику. М.: Высшая школа, 2002.
5. Пугачев В.С. Теория вероятностей и математическая статистика. М.: Физматлит, 2002.

КОНТАКТНАЯ ИНФОРМАЦИЯ

Необходимые консультации по всем возникающим вопросам, а также все необходимые учебные пособия можно получить:

1. на кафедре вычислительных систем и сетей УО «ПГУ», ауд. 156 УЛК №5 (корпус В) согласно графика консультаций или по раб.тел. (0214)**423031**;
2. по электронной почте **Marshall9@mail.ru**;
3. по телефону: (МТС) **+375298922998** (с 18-00 до 19-00);
4. на персональной странице http://marshall-k.narod.ru.