

## 4 ЭФФЕКТИВНОЕ КОДИРОВАНИЕ ПО МЕТОДУ ШЕННОНА-ФЭНО И ХАФФМЕНА

### Лабораторная работа №4

#### 4.1 Теоретические сведения

Учитывая статистические свойства источника сообщения, можно минимизировать среднее число двоичных символов, требующихся для выражения одной буквы сообщения, что при отсутствии шума позволяет уменьшить время передачи или объем запоминающего устройства.

Такое эффективное кодирование базируется на основной теореме Шеннона для каналов без шума. Шеннон доказал, что сообщения, составленные из букв некоторого алфавита, можно закодировать так, что среднее число двоичных символов на букву будет сколь угодно близко к энтропии источника этих сообщений, но не меньше этой величины.

Теорема не указывает конкретного способа кодирования, но из нее следует, что при выборе каждого символа кодовой комбинации необходимо стараться, чтобы он нес максимальную информацию. Следовательно, каждый символ должен принимать значения 0 или 1 по возможности с равными вероятностями, и каждый выбор должен быть независим от значений предыдущих символов.

*Эффективное* или экономичное кодирование используется для уменьшения объемов информации на носителе-сигнале таким образом, чтобы устранить избыточность.

Для кодирования символов исходного алфавита используются двоичные коды переменной длины: чем больше частота символа, тем короче его код. Эффективность кода определяется средним числом двоичных разрядов для кодирования одного символа.

При эффективном кодировании существует предел сжатия, ниже которого не «спускается» ни один метод эффективного кодирования – иначе будет потеряна информация. Этот параметр определяется предельным значением двоичных разрядов возможного эффективного кода

$$l_{np} = - \sum_{i=1}^n f_i \log_2 f_i,$$

где  $n$  - мощность кодируемого алфавита,

$f_i$  частота  $i$ -го символа кодируемого алфавита.

Для случая отсутствия статистической взаимосвязи между буквами конструктивные методы построения эффективных кодов были даны впервые Шенноном и Фэно. Их методики существенно не отличаются, и поэтому соответствующий код получил название кода Шеннона - Фэно.

##### 4.1.1 Метод Шеннона-Фэно

Упорядоченный в порядке невозрастания вероятностей список букв делится на две последовательные части так, чтобы суммы вероятностей входящих в них букв как можно меньше отличались друг от друга. Буквам из первой части приписываем символ 0, а буквам из второй части – символ 1. Далее точно так же поступаем с каждой из полученных частей, если она содержит хотя бы две буквы. Построенный код является префиксным, и ему соответствует насыщенное кодовое дерево.

В алгоритме Фэно кодовое дерево строится от корня, а в алгоритме Хаффмана – начиная с листьев. Это отличие позволяет в алгоритме Хаффмана полнее использовать специфику данного распределения вероятностей и строить оптимальный код. Алгоритм Фэно строит код, близкий к оптимальному.

Метод Шеннона-Фэно соответствует требованию оптимального кодирования. Алгоритм построения кода Шеннона-Фэно состоит в том, что кодируемые символы (буквы) разделяются на две равновероятные подгруппы: для символов 1-й подгруппы на втором месте ставится 0, а для 2-й подгруппы – 1 и т.д.

#### **Пример выполнения**

Построить таблицу кодов алфавита методом Шеннона-Фэно. Записать двоичным кодом фразу «теория информации».

Решение.

Составим таблицу кодов алфавита. Берутся первые шесть букв (от – до т). Сумма их вероятностей равна 0,498, на все остальные (от н до ф) приходится 0,502. Первые шесть букв будут иметь на первом месте 0, остальные 1.

Далее снова первая группа делится на две приблизительные равновероятные подгруппы: (от – до щ) и (от е до т) и т.д. Для всех букв первой подгруппы на втором месте ставится 0, а второй подгруппы – 1.

буквы	вероятности	символы кода									код									
		1-й	2-й	3-й	4-й	5-й	6-й	7-й	8-й	9-й										
-	0,145	0	0	0							000									
о	0,095			1								001								
е	0,074			1	1	0	0					0100								
а	0,064					1	0						0101							
и	0,064					1	0						0110							
т	0,056					1	1						0111							
н	0,056			1	0	0	0					1000								
с	0,047	1									1001									
р	0,041	1	1			0	0					10100								
в	0,039					1	0	1					10101							
л	0,036					1	0						10110							
к	0,028					1	1	1					10111							
м	0,026	1	0			0	0					11000								
д	0,026					1	1						11001							
п	0,024					1	0	0					11010							
у	0,021					1	1	1	0				110110							
я	0,019					1	1	1	1				110111							
ы	0,016					1	1	0	0	0				111000						
з	0,015								1	1	1	1				111001				
ь,ъ	0,015	1	1						1	0	0			111010						
б	0,015	1	1	1	1				1			111011								
г	0,014	1	1	0	0				0				111100							
ч	0,013				1				1	1	1				111101					
й	0,01				1				1	1	0	0			1111100					
х	0,009				1	1	1	0	1	0		11111010								
ж	0,008	1	1	1	1	1	0	1		11111011										
ю	0,007						1	1	1	1	0	0		11111100						
ш	0,006						1	1	1	1	1	1		11111101						
ц	0,003						1	1	1	1	1	1	0		111111100					
щ	0,003												1	1	1	1	1	0	1	111111101
э	0,003												1	1	1	1	1	1	0	111111110
ф	0,002												1	1	1	1	1	1	1	111111111

Фраза «теория информации» будет выглядеть следующим образом:

0111	0100	001	10100	0110	110111	000	0110	1000
т	е	о	р	и	я	пробел	и	н
111111111	001	10100	11000	0101	111111100	0110	0110	
ф	о	р	м	а	ц	и	и	

Для того, чтобы выяснить, является ли построенный код оптимальным, необходимо найти среднюю информацию, приходящуюся на один элементарный символ (0 или 1) и сравнить ее с максимально возможной информацией.

Определим среднюю информацию, содержащуюся в одной букве передаваемого текста, т.е. энтропия на одну букву

$$H(\mathfrak{b}) = - (0,145\log_2 0,145 + 0,95\log_2 0,95 + \dots + 0,002\log_2 0,002) = 4,42$$

Определим среднее число элементарных символов на букву (среднюю длину кодового слова) как произведение количества символов кода на вероятность появления данной буквы.

$$n_{\text{ср}} = 3*0,145 + 3*0,095 + 4*0,074 + \dots + 9*0,002 = 4,45$$

Информация на один элементарный символ

$$I_{\text{ср}} = \frac{H(\mathfrak{b})}{n_{\text{ср}}} = \frac{4,42}{4,45} = 0,994$$

Таким образом, информация на один символ близка к своему верхнему пределу 1. Следовательно, построенный код в целом отвечает принципу оптимальности.

#### 4.1.2 Методика Хаффмена

Методика Шеннона – Фэно не всегда приводит к однозначному построению кода. Ведь при разбиении на подгруппы можно сделать большей по вероятности как верхнюю, так и нижнюю подгруппу.

От указанного недостатка свободна методика Хаффмена. Она гарантирует однозначное построение кода с наименьшим для данного распределения вероятностей средним числом символов на букву.

Для двоичного кода методика сводится к следующему. Буквы алфавита сообщений выписываются в основной столбец в порядке убывания вероятностей. Две последние буквы объединяются в одну вспомогательную букву, которой приписывается суммарная вероятность. Вероятности букв, не участвовавших в объединении, и полученная суммарная вероятность снова располагаются в порядке убывания вероятностей в дополнительном столбце, а две последние объединяются. Процесс продолжается до тех пор, пока не получим единственную вспомогательную букву с вероятностью, равной единице.

Методика поясняется примером, представленным табл. 1.

Для составления кодовой комбинации, соответствующей данному сообщению, необходимо проследить путь перехода сообщения по строкам и столбцам таблицы.

Таблица 1

Буква	Вероятности	Вспомогательные столбцы						
		1	2	3	4	5	6	7
Z <sub>1</sub>	0.22	0.22	0.22	<u>0.26</u>	<u>0.32</u>	<u>0.42</u>	<u>0.58</u>	} → <u>1</u>
Z <sub>2</sub>	0.20	0.20	0.20	0.22	0.26	0.32	0.42	
Z <sub>3</sub>	0.16	0.16	0.16	0.10	0.22	0.26		
Z <sub>4</sub>	0.16	0.16	0.16	0.16	0.20			
Z <sub>5</sub>	0.10	0.10	<u>0.16</u>	0.16				
Z <sub>6</sub>	0.10	0.10	0.10					
Z <sub>7</sub>	0.04	} → <u>0.06</u>						
Z <sub>8</sub>	0.02							

Для наглядности строится кодовое дерево. Из точки, соответствующей вероятности 1, направляются две ветви, причем ветви с большей вероятностью присваивается символ 1, а с меньшей 0. Такое последовательное ветвление продолжаем до тех пор, пока не дойдем до каждой буквы. Кодовое дерево для алфавита букв, рассматриваемого в табл. 1, приведено на рисунке 1.

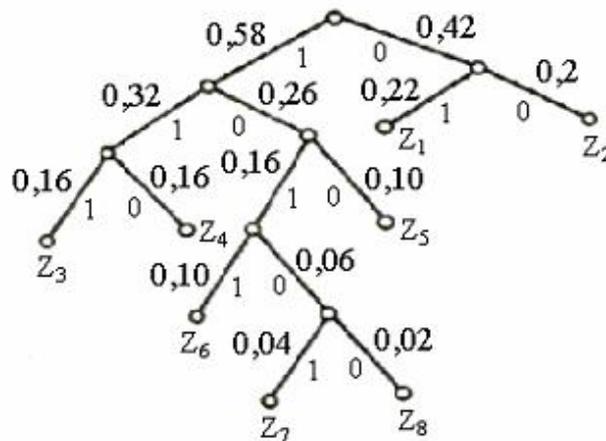


Рисунок 1 – Кодовое дерево

Теперь, двигаясь по кодовому дереву сверху вниз, можно записать для каждой буквы соответствующую ей кодовую комбинацию:

z1	z2	z3	z4	z5	z6	z7	z8
01	00	111	110	100	1011	10101	10100

Эффективность рассмотренных алгоритмов достигается благодаря присвоению более коротких кодовых комбинаций (кодовых символов) символам источника сообщений, вероятность которых более высока, и более длинных кодовых комбинаций - символам источника сообщений с малой вероятностью. Это ведет к различиям в длине кодовых символов и, как следствие, к трудностям при их декодировании. Для разделения отдельных кодовых символов можно использовать специальный разделительный

элемент, но при этом существенно снижается эффективность кода, т.к. средняя длина кодового символа фактически увеличивается на один элемент символа кода.

Целесообразнее обеспечить декодирование без введения дополнительных элементов символов. Этого можно добиться, если в эффективном коде ни одна кодовая комбинация не будет совпадать с началом более длинной кодовой комбинации. Коды, удовлетворяющие этому условию, называют префиксными кодами (префиксом или началом называют первый элемент в кодовом символе, а последний элемент – окончанием или постфиксом).

Коды, построенные по алгоритмам Шеннона–Фэно или Хаффмена, являются префиксными.

### 4.3 Задание на лабораторную работу

Задача1. Имеется алфавит символов и их вероятности, с которыми они встречаются в тексте. Построить таблицу кодов символов методом Шеннона-Фэно. Закодировать сообщение «вилка» и раскодировать заданную последовательность кодов.

а	в	л	и	е	с	к
0,3	0,2	0,15	0,1	0,1	0,08	0,07

Задача2. Построить таблицу кодов символов методами Шеннона-Фэно и Хаффмана.

Пусть  $A\{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$ ,  $P=(0,20;0,20;0,19;0,12;0,11;0,09;0,09)$ .

Задача3. Построить оптимальный неравномерный код методом Хаффмана. Данные:  $P_{a1}=0,22$ ,  $P_{a2}=0,58$ ,  $P_{a3}=0,01$ ,  $P_{a4}=0,03$ ,  $P_{a5}=0,16$ .

Задача4. Построить оптимальный код по методам Шеннона-Фэно и Хаффмана. Определить энтропию сообщения, сравнить среднюю длину кодового слова, построенного разными методами.

X1	X12	X3	X4	X5	X6	X7	X8	X9
0,35	0,15	0,13	0,09	0,09	0,08	0,05	0,04	0,02

### 4.4 Задание на СРОП

Решить примеры, выданные преподавателем.

### 4.5 Вопросы для защиты лабораторной работы

3.5.1 Что понимают под кодированием сообщения?

3.5.2 Какие коды называются равномерными?

3.5.3 Как строится код Шеннона-Фэно?

- 3.5.4. Как определяется число элементарных сигналов, приходящихся на одну букву сообщения?
- 3.5.5 Сформулировать основную теорему о кодировании.
- 3.5.6 Что называется декодированием сообщения?
- 3.5.7 Что называется блочным кодированием?
- 3.5.8 Объяснить принцип построения кода Хаффмана.
- 3.5.9 Назначение и цели эффективного кодирования.