# ЛАБОРАТОРНЫЕ РАБОТА №2

# МЕТОДЫ КОДИРОВАНИЯ ИНФОРМАЦИИ В КАНАЛАХ СВЯЗИ

# ПОСТРОЕНИЕ И РЕАЛИЗАЦИЯ ЭФФЕКТИВНЫХ КОДОВ

Целью работы является усвоение принципов построения и технической реализации кодирующих и декодирующих устройств эффективных кодов.

## 1.1. Указания к построению кодов

Учитывая статистические свойства источника сообщений, можно минимизировать среднее число двоичных символов, требующихся для выражения одной буквы сообщений, что при отсутствии шума позволяет уменьшить время передачи или емкость запоминающего устройства. Такое эффективное кодирование базируется на основной теореме Шеннона для каналов без шума.

К. 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 менее 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.

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

Код строится следующим образом: буквы алфавита сообщений выписываются в таблицу в порядке убывания вероятностей их встречаемости. Затем их разделяют на две группы так, чтобы суммы вероятностей встречаемости букв в каждой из групп были бы по возможности одинаковыми. Всем буквам верхней половины в качестве первого символа записывается – 1, а всем нижним – 0. Каждая из полученных групп, в свою очередь, разбивается на две подгруппы с одинаковыми суммарными вероятностями и т.д. Процесс повторяется до тех пор, пока в каждой подгруппе не останется по одной букве.

Все буквы будут закодированы различными последовательностями символов из “0” и ”1” так, что ни одна более длинная кодовая комбинация не будет начинаться с более короткой, соответствующей другой букве. Код, обладающий этим свойством, называется индексным. Это позволяет вести запись текста без разделительных символов и обеспечивает однозначность декодирования.

Рассмотрим алфавит из 8 букв. Ясно, что при обычном (не учитывающем вероятностей встречаемости их в сообщениях) кодировании для пред-

ставления каждой буквы требуется 3 символа ( $\log_2 M = \log_2 8 = 3$ ), где $M$ – количество букв в алфавите.

Наибольший эффект "сжатия" получается в случае, когда вероятности встречаемости букв представляют собой целочисленные отрицательные степени двойки. Среднее число символов на букву в этом случае точно равно энтропии. В более общем случае для алфавита из 8 букв среднее число символов на букву будет меньше трех, но больше энтропии алфавита *H(M)*.

$$H(M) \le l_{cp} \le \log_2 M$$

Для алфавита, приведенного в табл.1.1, энтропия *H(M)* равна 2.76, а среднее число символов на букву:

$$l_{cp} = \sum_{i=1}^{8} l_i p_i = 2.84 \ ,$$

где $l_i$ – количество символов для обозначения *i*-ой буквы.

**Таблица 1.1**

| Буква | Вероятности | Кодовая комбинация | № деления |
|---|---|---|---|
| $A_1$ | 0.22 | 11 | II |
| $A_2$ | 0.20 | 101 | III |
| $A_3$ | 0.16 | 100 | I |
| $A_4$ | 0.16 | 01 | IV |
| $A_5$ | 0.10 | 001 | V |
| $A_6$ | 0.10 | 0001 | VI |
| $A_7$ | 0.04 | 00001 | VII |
| $A_8$ | 0.02 | 00000 | |

Следовательно, некоторая избыточность в кодировании букв осталась. Из теоремы Шеннона следует, что эту избыточность можно устранить, если перейти к кодированию блоками.

Рассмотрим сообщения, образованные с помощью алфавита, состоящего всего из двух букв $A_1$ и $A_2$ с вероятностями появления соответственно

$P_1$ ($A_1$)=0,9 и $P_2$($A_2$)=0,1.

Поскольку вероятности не равны, то последовательность из таких букв будет обладать избыточностью. Однако, при побуквенном кодировании мы никакого эффекта не получим.

Действительно, на передачу каждой буквы требуется символ либо 1, либо 0, в то время как энтропия равна 0,47. При кодировании блоками, включающими по две буквы, получим табл. 1.2. Так как буквы статистически не связаны, вероятности встречаемости блоков определяют как произведение вероятностей составляющих их букв.

**Таблица 1.2**

| Буква | Вероятности | Кодовая комбинация | № деления |
|---|---|---|---|
| $A_1A_1$ | 0.81 | 1 | I |
| $A_1A_2$ | 0.09 | 01 | II |
| $A_2A_1$ | 0.09 | 001 | III |
| $A_2A_2$ | 0.01 | 000 | |

Среднее число символов на блок получается равным 1.29, а на букву – 0.645.

Кодирование блоков, включающих по три буквы, дает еще больший эффект. Среднее число символов на блок в этом случае равно 1,59, а на букву – 0,53, что всего на 12% больше энтропии. Теоретический минимум $H(M)$=0,47 может быть достигнут при кодировании блоков, включающих бесконечное число букв:

$$\lim_{n\to\infty} l_{CP} = H(A).$$

Следует подчеркнуть, что уменьшение $l_{ср}$ при увеличении числа букв в блоке не связано с учетом статистических связей между соседними буквами, так как нами рассматривались алфавиты с некоррелированными буквами. Повышение эффективности определяется лишь тем, что набор вероятностей, получающийся при укрупнении блоков, можно делить на более близкие по суммарным вероятностям подгруппы.

Для учета взаимосвязи между буквами текста, кодирование очередной буквы необходимо вести с учетом предыдущей последовательности букв в зависимости от глубины этой связи. При таком кодировании энтропия на одну букву уменьшается, но существенно усложняется система кодирования, поскольку приходится учитывать не один столбец вероятностей, а $M^m$ столбцов, где $m$ – глубина взаимосвязи между соседними буквами.

Рассмотренная нами методика Шеннона - Фено не всегда приводит к однозначному построению кода, так как, разбивая на подгруппы, большей по суммарной вероятности можно сделать как верхнюю, так и нижнюю подгруппы. Вероятности, приведенные в табл.1.1, можно было бы разбить иначе (табл. 1.3):

**Таблица 1.3**

| Буква | Вероятности | Кодовая комбинация | № разбиения |
|---|---|---|---|
| $A_1$ | 0.22 | 11 | II |
| $A_2$ | 0.20 | 10 | I |
| $A_3$ | 0.16 | 011 | IV |
| $A_4$ | 0.16 | 010 | III |
| $A_5$ | 0.10 | 001 | V |
| $A_6$ | 0.10 | 0001 | VI |
| $A_7$ | 0.04 | 00001 | VII |
| $A_8$ | 0.02 | 00000 | |

При этом среднее число символов на букву оказывается равным 2.80. Таким образом, построенный код может оказаться не самым лучшим. При построении эффективных кодов с основанием $m>2$ неопределенность становится еще больше. 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 первый столбец в порядке убывания вероятностей. Две последние вероятности объединяются в одну вспомогательную, которой приписывается суммарная вероятность. Вероятности букв, не участвующих в объединении, и полученная суммарная вероятность снова располагаются в порядке убывания вероятностей в дополнительном столбце, а две последние вероятности снова объединяются. Процесс продолжается до тех пор, пока не получим единственную вспомогательную вероятность равную единице. Поясним методику на примере (табл. 1.4). Значения вероятностей примем те же, что и в табл. 1.1.

Для получения кодовой комбинации, соответствующей данной букве, необходимо проследить путь перехода ее вероятности по строкам и столбцам табл. 1.4. Для наглядности построим кодовое дерево. Из точки, соответствующей вероятности 1, направим две ветви, причем ветви с большей вероятностью присвоим символ 1, а с меньшей – 0. Такое последовательное ветвление продолжим до тех пор, пока не дойдем до вероятности каждой буквы. Кодовое дерево для алфавита букв, рассматриваемого в нашем примере, приведено на рис. 1.1.

**Таблица 1.4**

| Буква | Вероятности | Вспомогательные столбцы | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $A_1$ | 0.22 | 0.22 | 0.22 | 0.26 | 0.32 | 0.42 | 0.58 | 1 |
| $A_2$ | 0.20 | 0.20 | 0.20 | 0.22 | 0.26 | 0.32 | 0.42 | |
| $A_3$ | 0.16 | 0.16 | 0.16 | 0.20 | 0.22 | 0.26 | | |
| $A_4$ | 0.16 | 0.16 | 0.16 | 0.16 | 0.20 | | | |
| $A_5$ | 0.10 | 0.10 | 0.16 | 0.16 | | | | |
| $A_6$ | 0.10 | 0.10 | 0.10 | | | | | |
| $A_7$ | 0.04 | 0.06 | | | | | | |
| $A_8$ | 0.02 | | | | | | | |

Теперь, двигаясь по кодовому дереву от единицы через промежуточные вероятности к вероятностям каждой буквы, можно записать соответствующую ей кодовую комбинацию: $A_1$ - 01, $A_2$ - 00, $A_3$ - 111, $A_4$ - 110, $A_5$ - 100, $A_6$- 1011, $A_7$ - 10101, $A_8$ - 10100. При этом получим $l_{ср} = 2{,}80$ символа на букву.

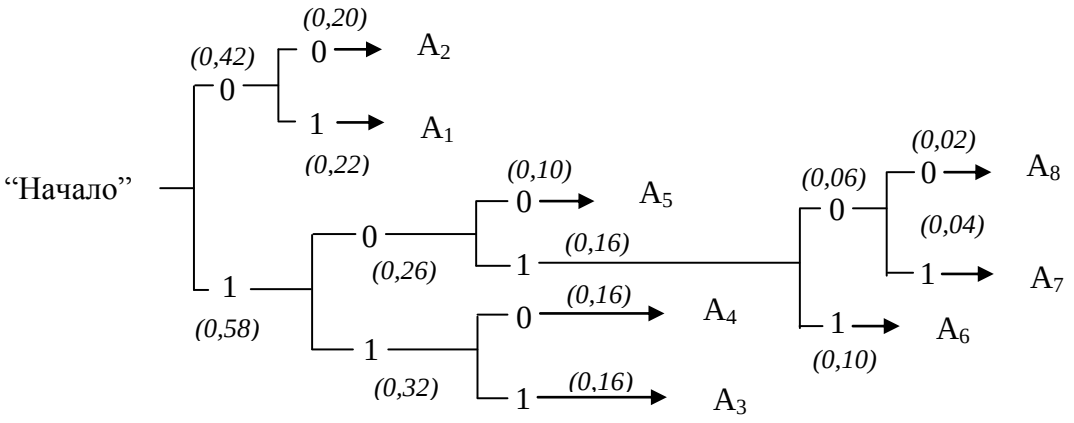


**Рис. 1.1. Кодовое дерево**

Отметим в заключение особенности систем эффективного кодирования.

Одна из особенностей обусловлена различием в длине кодовых комбинаций для разных букв. Если буквы выдаются через равные промежутки времени, то кодирующее устройство через равные промежутки времени выдает комбинации различной длины. Поскольку линия связи используется эффективно только в том случае, когда символы поступают в нее с постоянной скоростью, то на выходе кодирующего устройства должно быть предусмотрено буферное устройство ("упругая" задержка). Оно запасает символы по мере их поступления и выдает их в линию связи с постоянной скоростью. Аналогичное устройство необходимо и на приемной стороне.

Вторая особенность связана с возникновением задержки в передаче информации. Наибольший эффект достигается при кодировании длинными блоками, а это приводит к необходимости накапливать буквы, прежде чем сопос-

тавить им определенную последовательность символов. При декодировании задержка возникает снова. Общее время задержки может быть велико, особенно при появлении блока, вероятность которого мала. Это следует учитывать при выборе длины кодируемого блока.

Еще одна особенность заключается в специфическом влиянии помех на достоверность приема. Одиночная ошибка может перевести передаваемую кодовую комбинацию в другую, не равную ей по длительности. Это повлечет за собой неправильное декодирование целого ряда последующих комбинаций, который называют треком ошибки. Специальными методами построения эффективного кода трек ошибки стараются свести к минимуму.

## ЗАДАНИЕ

1. Изучить описание, изучить методы построения и технической реализации эффективных кодов.

2. По конкретным значениям вероятностей встречаемости букв, заданных студенту преподавателем или выбранных самостоятельно (отличающихся от рассмотренного в описании, не более 12 букв и нетривиальный случай), построить эффективный код, используя методики Шеннона-Фено и Хаффмена.

3. Вычислить энтропию источника и среднюю длину комбинации полученного кода.

4. Подготовить небольшой текст на 15–20 букв для построения дерева кодирования.

5. Зарисовать таблицу и дерево Хаффмена.

6. Подсчитать выигрыш от записи текста эффективным кодом.

## Требования к отчету

Отчет должен включать:

1. Таблицу построения эффективного кода по методике Шеннона-Фено;

2. Таблицу и кодовое дерево, иллюстрирующие построение эффективного кода по методике Хаффмена.

3. Результаты расчетов энтропии источника и среднюю длину кода для буквы, отдельно для заданного Вами алфавита из 12 букв и текста.