

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Институт информационных технологий

Кафедра физико-математических дисциплин

**А. И. Митюхин**

**ТЕОРИЯ ИНФОРМАЦИИ  
ТЕОРИЯ КОДИРОВАНИЯ  
(ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ)**

*Рекомендовано УМО по образованию в области информатики  
и радиоэлектроники в качестве учебно-методического пособия  
для специальности 1-40 01 01*

*«Программное обеспечение информационных технологий»,  
1-39 03 02 «Программируемые мобильные системы»*

Минск БГУИР 2017



УДК [621.321 +004.932](076)  
ББК 32.811 я73+32.973.26-018.2я73  
М67

Рецензенты:  
Кафедра полиграфического оборудования и систем  
обработки информации учреждения образования  
«Белорусский государственный технологический университет»  
(протокол № от . .2016);

Заведующий лабораторией идентификации систем  
ОИПИ НАН Беларуси,  
доктор технических наук, профессор А.А. Дудкин

**Митюхин, А. И.**  
М67 Теория информации : учеб.-метод. пособие  
/ А. И. Митюхин. – Минск : БГУИР, 2017. – 161 с. : ил.  
ISBN 978-985-543-190-0.

Рассмотрены понятия теории информации – эффективное и помехоустойчивое кодирование, защита информации. Дано описание алгоритмов позволяющих уменьшать объем передаваемых, хранимых или распределяемых данных. Представлены основные методы кодирования и декодирования линейных кодов, корректирующих ошибки. Методы и алгоритмы теории информации изложены с учетом их практической направленности. Учебный материал содержит примеры решения задач на основе практического использования доступного математического аппарата. Изложение тем сопровождается иллюстрациями и упражнениями.

УДК [621.321 +004.932](076)  
ББК 32.811 я73+32.973.26-018.2я73

ISBN 978-985-543-190-0

@ Митюхин А. И.  
@ Белорусский государственный  
университет информатики  
и радиоэлектроники

## Содержание

ВВЕДЕНИЕ.....	5
ЧАСТЬ 1. ИНФОРМАЦИЯ. ЭФФЕКТИВНОЕ КОДИРОВАНИЕ.....	8
1. МОДЕЛЬ КАНАЛА ПЕРЕДАЧИ, ХРАНЕНИЯ, ОБРАБОТКИ И РАСПРЕДЕЛЕНИЯ ИНФОРМАЦИИ.....	8
1.1. Обобщенная модель канала передачи, хранения, обработки и распределения информации.....	8
1.2. Эталонная модель взаимосвязи открытых систем.....	10
1.3. Первичное кодирование информации.....	10
2. КАЧЕСТВЕННАЯ И КОЛИЧЕСТВЕННАЯ ОЦЕНКА ИНФОРМАЦИИ.....	12
2.1. Дискретный источник информации без памяти.....	12
2.2. Канал передачи информации.....	13
2.3. Дискретный канал без памяти.....	14
2.4. Характеристики дискретного канала без памяти.....	15
2.5. Модель связанных источников.....	19
2.6. Количественная оценка информации.....	23
2.7. Энтропия.....	25
2.8. Относительная избыточность источника.....	29
3. ЭФФЕКТИВНОЕ КОДИРОВАНИЕ ДЛЯ ДИСКРЕТНОГО ИСТОЧНИКА БЕЗ ПАМЯТИ.....	31
3.1. Условия взаимной однозначности алфавитного кодирования.....	31
3.2. Эффективное кодирование.....	33
3.3. Неравенство Крафта.....	36
3.4. Средняя длина кодового слова.....	38
4. ТЕОРЕМА ШЕННОНА О КОДИРОВАНИИ ДЛЯ КАНАЛА БЕЗ ШУМА (первая теорема Шеннона).....	40
4.1. Энтропия блокового источника.....	40
4.2. Первая теорема Шеннона.....	40
4.3. Сжатие данных.....	44
4.4. Энтропийное кодирование методом Хаффмена.....	47
4.5. Универсальный алгоритм сжатия.....	59
5. КАНАЛЫ БЕЗ ПАМЯТИ И ПЕРЕДАЧА ИНФОРМАЦИИ.....	67
5.1. Двоичный симметричный канал без памяти.....	67
5.2. Комбинирование источников.....	70
5.3. Совместная энтропия.....	70
5.4. Условная энтропия.....	71
5.5. Соотношение между совместной и условной энтропией.....	73
5.6. Пропускная способность канала.....	75
5.7. Дифференциальная энтропия.....	79
5.8. Пропускная способность непрерывного канала.....	80
5.9. Статистические характеристики каналов.....	83
6. ВВЕДЕНИЕ В ЗАЩИТУ ИНФОРМАЦИИ.....	86
6.1. Составляющие информационной безопасности.....	86
6.2. Информационные угрозы и атаки.....	87

6.3. Модели разграничения доступа к информации.....	88
6.4. Обеспечение целостности данных в инфокоммуникационных системах и сетях.....	91
6.5. Общие сведения по классической криптографии.....	92
6.6. Алгоритмы блочного шифрования.....	95
6.7. Ассиметричные алгоритмы шифрования.....	96
ЧАСТЬ 2. ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ.....	103
1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ.....	103
1.1. Основная теорема Шеннона кодирования для канала с шумом (вторая теорема Шеннона).....	103
1.2. Возможность исправления ошибок помехоустойчивым кодом.....	104
2. АЛГЕБРАИЧЕСКИЕ СТРУКТУРЫ.....	113
2.1. Группы.....	113
2.2. Разложение группы на смежные классы.....	121
2.3. Определение смежного класса кода.....	123
2.4. Кольцо.....	125
2.5. Конечные поля.....	127
2.6. Представление элементов конечного поля Галуа $GF(p^m)$ .....	129
2.7. Векторные пространства и подпространства.....	133
3. ЛИНЕЙНЫЕ КОДЫ.....	134
3.1. Линейные коды, исправляющие ошибки: построение и основные свойства.....	134
3.2. Вектор ошибок.....	136
3.3. Порождающая и проверочная матрица систематического линейного кода.....	136
3.4. Кодирование линейным кодом.....	139
3.5. Линейный код Рида-Маллера.....	141
3.6. Линейный код Хэмминга.....	144
3.7. Совершенные и квазисовершенные коды.....	146
3.8. Вычисление минимального веса по проверочной матрице кода.....	146
4. МЕТОДЫ ДЕКОДИРОВАНИЯ ЛИНЕЙНЫХ КОДОВ.....	147
4.1. Декодирование кода на основе принципа максимального правдоподобия.....	149
4.2. Декодирование по синдрому.....	152
4.3. Декодирование кода Хэмминга.....	154
4.4. Вычисление вероятности ошибки декодирования.....	153
Литература.....	158

## ВВЕДЕНИЕ

Теория информации это раздел науки, возникший в середине прошлого века. Умение применять на практике результаты теории информации стало важным для специалиста, создающего современные инфокоммуникационные системы. Теория информации возникла из статистической теории связи. Лишь частично отвечая на вопросы о путях и способах технической реализации аппаратуры, эта теория позволяет вычислить эффективность системы передачи, хранения, обработки и распределения информации, определить максимально возможную эффективность системы. Для широкого класса информационных задач при определенных знаниях или допущениях относительно статистики шумов стало возможным построение аппаратуры, работающей на основе оптимального приема кодированных сигналов. Такие сигналы строятся на основе трех составляющих теории информации – теории эффективного кодирования, теории помехоустойчивого кодирования и теории криптографического кодирования.

После того, как выбрана и обоснована конкретная схема оптимального приемника, фильтра, обнаружителя и т. п. возникают вопросы выбора метода кодирования, класса кода, способа защиты информации от несанкционированного доступа к ней.

Большая часть передаваемых, хранимых, распределяемых, преобразуемых данных соответствует звуковой, графической или видеоинформации. Увеличиваются технические затраты на хранение данных, предъявляются более высокие требования по экономии канального частотного ресурса. Алгоритмы теории информации позволяют уменьшить объем данных, используемых для представления информации.

Задача теории информации – при известной статистике шумов выбрать такое множество передаваемых сигналов, чтобы правдоподобие правильного декодирования принимаемых сообщений было максимальным. При этом важно найти не только хороший код, но и эффективный алгоритм декодирования.

Теория помехоустойчивых кодов (кодов, контролирующих ошибки) является одной из ветвей теории цифровой обработки сигналов (ЦОС). Существует тесная связь теории информации – кодирования и теории ЦОС. Но данные дисциплины развивались различными путями: одна разрабатывалась в основном алгебраистами, а другая – в основном инженерами. Первые результаты по теории информации появились в конце 40-х годов в работах К. Шеннона (Shannon Claude, амер. ученый), Голея (M. J. E. Golay, амер. ученый) и Р. Хэмминга (R. Hamming, амер. ученый). Можно определить следующие основные исторические этапы развития теории информации:

– 1948 г., К. Шеннон сформулировал и доказал теоремы кодирования для дискретного канала. К. Шеннон показал, что с каждым каналом передачи информации связано число  $C$ . Это число определяет пропускную способность канала и измеряется в битах в секунду. Если требуемая от информационной системы скорость передачи информации  $R_i$  (измеряемая в битах в секунду) меньше

С, то используя коды, контролирующие ошибки, для данного канала можно построить такую информационную систему, что вероятность ошибки на выходе декодера будет сколь угодно мала;

- 1950 г., Р. Хэмминг описал класс кодов, исправляющих независимые одиночные ошибки;

- 1952 г., Д. А. Хаффмен (D. A. Huffman, амер. ученый) показал, что, разработанный им алгоритм эффективного кодирования позволяет строить класс оптимальных префиксных кодов;

- 1960 г., Р. К. Боуз (R. C. Bose, инд.-амер. ученый), Д. К. Рой-Чоудхури (D. K. R-Chaudhari, инд.-амер. ученый) и независимо Р. К. Хоквингем, 1959 (R. C. Hocquenghem, фран. ученый) открыли двоичные коды, исправляющие кратные независимые ошибки (коды Боуза- Чоудхури- Хоквингема (БЧХ-коды));

- 1963 г., И. С. Рид (I. C. Reed, амер. ученый) и Г. Соломон (G. Solomon, амер. ученый) предложили модификацию БЧХ-кодов для не двоичных каналов (коды Рида-Соломона (РС-коды)). Эти коды нашли применение для исправления пакетов и модулей ошибок;

- (1960 – 1970) г., с появлением микросхем средней степени интеграции началось практическое воплощение методов теории информации в каналах с большим уровнем помех. Применялись низкоскоростные коды максимальной длины (М-последовательности), коды Рида-Маллера (РМ-коды) и др. Кроме того, были разработаны новые эффективные алгоритмы декодирования (Питерсон, Берлекэмп, Мэсси и др.).

- 1977 г., разработан метод сжатия на основе словаря А. Лемпелем (Abraham Lempel, изр. ученый) Я. Зивом (Jacob Ziv, изр. ученый). Метод является основой алгоритмов сжатия ZIP, ARJ, gzip и др.

Методы теории информации используются:

- для защиты данных в памяти вычислительных устройств, для передачи данных в вычислительных системах (такие системы очень чувствительны к очень малой доле ошибок, т. к. даже одиночная ошибка может нарушить всю программу вычислений);

- в цифровых оптических дисках (компакт-дисках);

- в системах со сжатием данных;

- в системах связи с ограничением на передаваемую мощность, например, в системах ретрансляции через спутник, где увеличение мощности обходится очень дорого;

- в системах цифрового телевидения; обработки изображения;

- в системах передачи информации разного назначения, например, в системах с пакетной коммутацией и разделением во времени, где длинные двоичные сообщения разделяются на пакеты, и пакет передается в отведенное временное окно. Из-за нарушения синхронизации пакеты могут быть утеряны. Кодирование позволяет обеспечить надежную синхронизацию в такой системе.

Кодирование применяется для защиты специальных радиоэлектронных

систем гражданского и военного назначения, например, радиолокационных и радионавигационных систем, систем видеогарантии от воздействия:

- непреднамеренных помех типа белого шума;
- преднамеренных помех специального типа, (например, сосредоточенных в спектре сигнала – узкополосных, или широкополосных с кодовыми видами модуляции).

Алгоритмы теории информации используются для защиты информационных комплексов от случайного и несанкционированного доступа к информации; повышения надежности радиоэлектронных и вычислительных систем, делая их нечувствительными к отказам и сбоям.

Рассмотрение вопросов теории информации начнём с представления общей модели информационного канала.

## **ЧАСТЬ 1. ИНФОРМАЦИЯ. ЭФФЕКТИВНОЕ КОДИРОВАНИЕ**





Корректирующий кодер вводит информационную избыточность в передаваемое сообщение с целью обнаружения и (или) исправления ошибок сравнительно небольшой кратности (число ошибок  $t = 1, 2, 3, 4$ ).

Кодер канала (помехоустойчивый кодер) также предназначен для минимизации влияния помех на передаваемую информацию и в большинстве своем используется в специальных радиоэлектронных системах:

- системах дальнего космоса;
- спутниковых навигационных системах (например, GPS "NAVSTAR" (Global Positioning System "NAVSTAR"), "ГЛОНАСС" (глобальная навигационная спутниковая система));
- системах скрытной связи;
- радиолокационных системах дальнего обнаружения целей, системах наведения на цель с повышенной точностью (точное оружие);
- системах мобильной и фиксированной связи третьего поколения CDMA (Code Division Multiple Access – кодовое разделение каналов, множественный доступ).

Модулятор преобразует множество дискретных сигналов канального кодера в непрерывные сигналы, которые передаются по каналам.

Физической средой передачи информации – каналом может служить:

- радиоканал;
- проводной канал;
- оптический канал;
- магнитная лента;
- компакт-диск;
- запоминающее устройство (ЗУ) и т. п.

*Замечание.* Если в качестве канала передачи использовать ЗУ, то это канал передачи информации во времени в отличие, например, от радиоканала передачи информации в пространстве.

В канале формируется смесь сигнала и помехи вида

$$y(t) = x(t)\mu(t) + n(t),$$

где –  $x(t)$  передаваемый непрерывный сигнал,  $\mu(t)$  мультипликативная помеха,  $n(t)$  аддитивная помеха (как правило, шум с гауссовским распределением).

Декодер источника восстанавливает ту избыточность, которая была ранее устранена на передающей стороне.

#### *Замечания*

1. Техническая реализация составляющих рис. 1.1 с номерами 3, 4, 5, 6, 10, 11, 12, 13 осуществляется на цифровой элементной базе.

2. В элементе 2 производится дискретизация по времени и квантование по уровню входной аналоговой реализации (сообщения) с формированием сим-

волов в двоичном или  $q$ -ичном алфавите.

## **1.2. Эталонная модель взаимосвязи открытых систем**

Для построения эффективных информационных систем для каналов с различной средой необходимо использовать и другие модели. Наиболее известна так называемая эталонная модель взаимосвязи открытых систем, где в обобщенном виде рассмотрены функции, выполняемые на различных уровнях. Модель представляет семиуровневую архитектуру.

1. На физическом уровне реализуется канал.
2. На канальном уровне реализуется процедуры кодирования по сжатию, шифрованию, помехоустойчивому кодированию информации.
3. На сетевом уровне реализуется передача информации от источника к адресату.
4. Транспортный уровень управляет сквозной передачей пакетов, с коррекцией ошибок.
5. Сеансовый уровень контролирует соединения между оконечными системами.
6. На уровне представления выполняются операции сжатия данных, защиты информации, преобразования форматов для обеспечения эффективного и безопасного взаимодействия.
7. Прикладной уровень предоставляет различные сетевые службы.

## **1.3. Первичное кодирование информации**

Дискретный поток двоичных символов, сформированный аналого-цифровым преобразователем (АЦП), как правило, преобразуется в форму наиболее подходящую для конкретных применений. Например, в цифровой электронике в таких приборах, как вольтметры и частотомеры, с точки зрения восприятия информации имеет преимущество десятичная система счисления. В таких приборах используются 7-сегментные индикаторы при вводе и выводе буквенно-цифровых данных. Требуется промежуточное преобразование простого двоичного кода в некоторое другое двоичное отображение – первичное кодирование информации. Кроме того, в реальных каналах передачи информации, из-за недопустимых частотных, амплитудных и др. искажений, простой двоичный код часто оказывается непригодным для передачи, хранения и обработки информации.

Первичное кодирование может осуществляться посредством весовых кодов. Веса – это величины, равные степени по основанию два, на которые умножаются двоичные цифры. Примером весового кода является двоично-десятичный код, где веса битов это значения: 8, 4, 2, 1. В двоично-десятичном коде каждая десятичная цифра задается словом из 4 двоичных цифр (битов). Например, число 127 записывается как (000100100111). Вес наименьшей значащей двоичной цифры равен 1. К весовым относятся также коды, построенные

с использованием двоичной, восьмеричной, шестнадцатеричной системы счисления. Названные коды применяются для кодирования команд, операндов и других данных, предназначенных для применения в цифровых устройствах, микропроцессорах, компьютерах и пр.

На этапе первичного кодирования используются и буквенно-цифровые коды. Кроме двоичного кодирования десятичных цифр, буквенно-цифровые коды позволяют получить двоичное изображение текстовых символов, чисел, знаков препинания и управляющих символов. Одним из таких кодов является ASCII-код (American Standard Code for Information Interchange – Американский стандартный код для обмена информацией) – это сравнительно давно принятый стандарт для представления данных в цифровой форме. Каждый символ обозначается числом от 32 до 127. Используется семь битов для идентификации символа и один бит добавляется таким образом, чтобы количество единиц в кодовом слове было четным. Получается блоковый равномерный код с контролем четности длиной 8 бит (байт). Например, символу DEL соответствует кодовое слово (1 1 1 1 1 1 1 1). Символ \$ – кодируется словом ASCII-кода как (0 0 1 0 0 1 0 0).

### 1.2.1. Рефлексные коды

В двоичном коде при переходе от кодирования одного десятичного числа к другому может происходить одновременное изменение двоичных цифр в нескольких разрядах. Например, при переходе от изображения числа  $15 \rightarrow 01111$  к числу  $16 \rightarrow 10000$  одновременно изменяются цифры в пяти разрядах. Это может являться источником ошибок при кодировании информации. Эффективным средством борьбы с ошибкой неоднозначности считывания является использование рефлексных кодов (отраженных кодов). Типичным представителем таких кодов является код Грея. Основным свойством кода Грея является то, что любые два соседние кодовые слова различаются только в одном разряде. На рис. 1.2. показаны соответствия между десятичным числом, простым двоичным кодом и двоичным кодом Грея.

$$\begin{array}{c} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array} .$$

Рис. 1.2. Код Грея

Пусть последовательно считываются числа 4 и 5. Им соответствуют слова

кода Грея (0 1 1 0) и (0 1 1 1). Прием же после числа 4  $\rightarrow$  (01 1 1 0) слова с отличием, например, в двух двоичных символах, свидетельствует о возникшей ошибке.

## **2. КАЧЕСТВЕННАЯ И КОЛИЧЕСТВЕННАЯ ОЦЕНКА**

## ИНФОРМАЦИИ

Под термином «информация» понимаются сведения, известия, которые описывают некоторое событие, руководство к действию, или свойство какого-либо объекта. Эти сведения могут быть представлены определенными символами, буквами алфавита, или в каком-то другом виде, например, изображением объекта «интереса», словами, и пр. Формой представления информации является сообщение. В конкретных информационных системах сообщение может использоваться, передаваться, становиться объектом хранения, распределения, преобразования.

Н. Виннер (N. Wiener, амер. ученый) определил информацию как объект нематериальной природы: «Информация есть информация, а не материал или энергия». В основе теории информации является положение о том, что любой источник информации можно описать вероятностными категориями, которые могут быть измерены. Каждое сообщение содержит в себе определенную информацию. Однако одни сообщения переносят больше информации, чем другие. Если в прогнозе погоды сообщается, что 1 января температура воздуха в Минске достигнет  $+20^{\circ}\text{C}$ , то это сообщение характеризуется очень большим количеством информации. Такое событие является неожиданным, редким, вероятность  $p$  его появления стремится к нулю,  $p \rightarrow 0$ . В сообщении, что 1 января температура воздуха в Минске ожидается  $-5^{\circ}\text{C}$  не является неожиданным. Вероятность  $p$  его появления стремится к единице,  $p \rightarrow 1$ . В сообщении о высоковероятностном событии содержится мало информации. Вероятность события является мерой его неожиданности (неопределенности) и связана с количественной мерой информации. Можно предположить, что количество информации о событии, обратно величине вероятности его появления, т. е.

$$I \sim \log \frac{1}{p} \sim -\log p, \quad (2.1)$$

где  $I$  – количество информации, полученное с появлением сообщения с вероятностью  $p$ . Чтобы определить понятие количества информации, как измеряемой величины, необходимо вначале рассмотреть канал передачи информации и свойства источника информации.

### 2.1. Дискретный источник информации без памяти

Дискретный источник информации без памяти  $X$  в дискретный момент времени  $i$  формирует символ  $x_i$  случайной последовательности символов. Выход источника есть случайная величина. Множество исходных символов  $X = \{x_1, x_2, \dots, x_m\}$  называется алфавитом источника  $X$ , а элементы  $x_i$  – буквами или символами. Символами источника могут быть буквы, цифры или некие абстрактные знаки. Каждый  $n$ -ый символ из конечного алфавита  $X = \{x_1, x_2, \dots, x_m\}$  источника появляется на выходе с вероятностью  $p_i$ . Вероятности появления символов источника задаются в виде множества  $\{p_1, p_2, \dots, p_m\}$ . Все вероятности символов в сумме должны давать значение 1, т. е.

$$\sum_{i=0}^m p_i = 1,$$

где  $m$  – число различных символов множества определяет размерность используемого алфавита.

Например, если  $X = \{x_1 = a, x_2 = b, x_3 = c\}, X = \{a, b, c\}, m = 3, \{p_1 = \frac{1}{2}, p_2 = \frac{1}{3}, p_3 = \frac{1}{6}\}$ .

Алфавиту  $X = \{0, 1\}, m = 2$ , соответствует двоичный источник без памяти. Выходными символами источника являются символы  $x_1 = 0$  и  $x_2 = 1$ . Обозначим  $p_1$  вероятность появления символа  $x_1$ . Тогда выражение  $p_2 = (1 - p_1)$  представляет вероятность появления символа  $x_2 = 1$ .

*Определение 2.1.* Два источника  $X = \{x_1, x_2, \dots, x_m\}$  и  $T = \{t_1, t_2, \dots, t_u\}$  являются независимыми, если совместная вероятность  $p_{x,t}$  каждой пары  $(x, t)$  событий  $x \in X, t \in T$  равна произведению

$$p_{x,t} = p_x p_t,$$

где  $p_x$  и  $p_t$  вероятности появления символов источников  $X$  и  $T$ .

### 2.1.1. Блоковый источник информации

*Определение 2.2.* Если выходом источника являются последовательности (блоки) из  $n$  одиночных статистически независимых символов алфавита  $X = \{x_1, x_2, \dots, x_m\}$ , то такой источник называется источником с  $n$ -кратным расширением  $X^n$  источника  $X$ .

*Замечание.* Источник  $X^n$  называют также блоковым источником.

На выходе блокового источника можно сформировать  $m^n$  символов. Например, для  $X = \{0, 1\}, n = 2$  возможное множество символов источника с 2-кратным расширением источника  $X$  есть

$$X^2 = \{c_1, c_2, c_3, c_4\},$$

где  $c_1 = (00), c_2 = (01), c_3 = (10), c_4 = (11)$

образуют множество, состоящее из  $m^n = 2^2 = 4$ -х символов блокового источника.

### 2.2. Канал передачи информации

Передача информации, формируемой источником, осуществляется посредством использования канала передачи информации. Канал – это некоторая физическая среда, соединяющая источник информации с получателем. Примеры каналов: проводная телефонная линия, среда распространения электромагнитных волн (радиоканал) используемая, например, при соединении компьютера, имеющем Wi-Fi-адаптер, с сетью Internet. CD (компакт-диск) это тоже канал в виде ЗУ и пр. На рис. 2.1. изображена обобщенная математическая модель системы передачи информации, включающая канал.

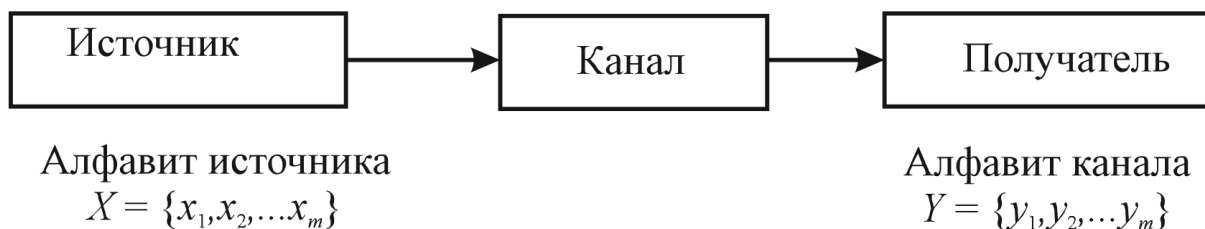


Рис. 2.1. Модель системы передачи информации

*Замечание.* Эффективность каналов передачи (хранения) информации возрастает с переходом на недвоичные символы.

### 2.3. Дискретный канал без памяти

Алфавиты передаваемых и принимаемых символов сообщения должны совпадать. Однако, из-за воздействия помех (шума) полученный символ может отличаться от переданного. В этом случае принимаемые символы называют алфавитом канала. На рис. 2.1 они обозначены как  $Y = \{y_1, y_2, \dots, y_m\}$  и приемник можно так же считать источником информации.

Каналы с шумами характеризуется условными вероятностями  $p(y|x)$  для всех  $x \in X$  и  $y \in Y$ .

*Определение 2.3.* Условная вероятность  $p(y|x)$  понимается как вероятность того, что на выходе канала (входе приемника) появился символ  $y$ , при условии, что на выходе источника  $X$  был сформирован символ  $x$ .

*Замечание.* Условные вероятности  $p(y|x)$  называются вероятностями перехода канала.

*Определение 2.4.* Если имеется конечное число входов и выходов канала, и принимается, что вероятность  $p(y|x)$  не зависит от вероятностей появления предыдущих символов входа, то канал называется дискретным каналом без памяти.

В ряде приложений, например, связанных с задачами обнаружения сигналов на фоне помех, условную вероятность  $p(x|y)$  называют апостериорной (послеопытной) вероятностью.

Обратная условная вероятность канала –  $p(x|y)$  определяет вероятность приема символа  $x$ , если на выходе канала имеется символ  $y$ . Условная вероятность  $p(x|y)$  позволяет оценить качество приема.

Дискретный канал с алфавитом символов источника (входом канала)  $X = \{x_1, x_2, \dots, x_m\}$  и символов источника (выходом канала)  $Y = \{y_1, y_2, \dots, y_m\}$  описывается графом переходных вероятностей  $p(y|x)$ . На рис. 2.2 показано графическое описание дискретного канала с двумя символами на входе и выходе канала.



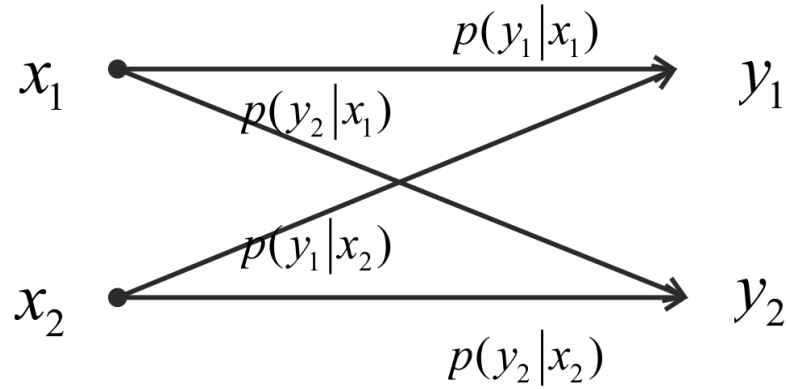


Рис. 2.2. Модель дискретного канала передачи информации с набором вероятностей перехода

## 2.4. Характеристики дискретного канала без памяти

### 2.4.1. Множества. Вероятность

Рассмотрение дискретного канала без памяти следует начать с напоминания некоторых определений теории множеств и теории вероятностей.

Предположим, что элементарное (одиночное) событие  $x$  происходит на множестве  $X = \{x\} = \{x_1, x_2, \dots, x_m\}$  всех возможных элементарных событий (выборочном пространстве). Каждое элементарное событие принадлежит одному и только одному  $x_i$  из множества  $X = \{x_1, x_2, \dots, x_m\}$ . Событие  $x \in X$  является подмножеством выборочного пространства. Заметим, что если  $X$  – множество, содержащее  $m$  элементов, то количество различных подмножеств равно  $2^m$ . Например, для  $\{x\} = \{0, 1\}$  подмножества являются:  $\{0\}, \{1\}, \{0, 1\}, \{\emptyset\}$ .

Множество чисел  $P = \{p(x_1), p(x_2), \dots, p(x_m)\}$  задает распределение вероятностей  $p(x)$  на множестве  $\{x\}$ , если выполняется условие нормировки

$$\sum_{i=1}^m p(x_i) = 1.$$

Предположим также, что элементарное событие  $y$  происходит на выборочном пространстве  $Y = \{y\} = \{y_1, y_2, \dots, y_m\}$ . Событие  $y \in \{y\}$  является также подмножеством выборочного пространства.

Множество  $P = \{p(y_1), p(y_2), \dots, p(y_m)\}$  задает распределение вероятностей  $p(y)$  на множестве  $\{y\}$  с условием

$$\sum_{i=1}^m p(y_i) = 1.$$

На множествах  $X$  и  $Y$  определены операция пересечения  $X \cap Y$

(произведение событий) и операция объединения  $X \cup Y$  (сумма событий).

С точки зрения теории вероятностей событие  $X \cap Y$  можно характеризовать как одновременное осуществление событий  $X$  и  $Y$ .

Множества  $X, Y, Z \dots$  – являются попарно непересекающимися (попарно несовместны), если никакие два из них не имеют общих элементов, т. е. если

$$X \cap Y = \emptyset, X \cap Z = \emptyset, \dots, Y \cap Z = \emptyset, \dots \quad (2.2)$$

На множествах  $X$  и  $Y$  определена операция разности  $(X - Y)$  множеств. Пусть заданы два множества элементов декартова произведения

$$X = \{(0, 1), (0, 3), (4, 5)\} \text{ и } Y = \{(0, 3), (0, 6)\}.$$

Тогда разность  $(X - Y) = \{(0, 1), (4, 5)\}$ . Рис. 2.3. иллюстрирует операцию разности двух множеств.

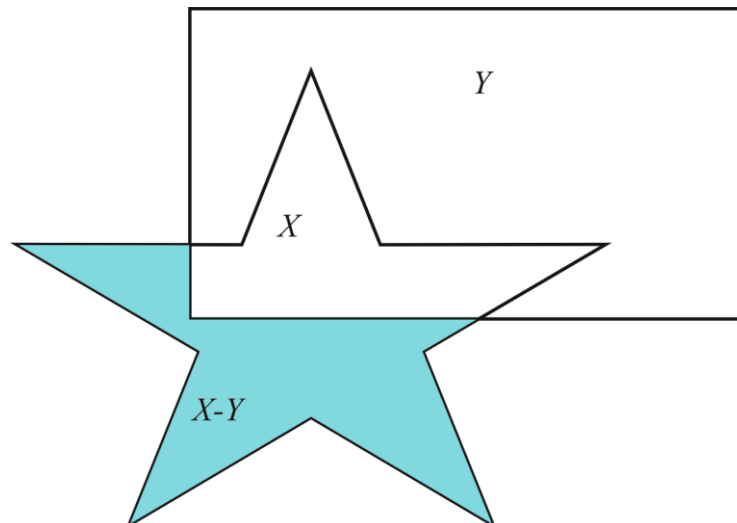


Рис. 2.3. Разность множеств  $(X - Y)$

Далее покажем, что на пересекающихся множествах  $X$  и  $Y$  справедлива формула

$$P(X \cup Y) = P(X) + P(Y) - P(X \cap Y).$$

На рис. 2.4. показаны пересекающиеся множества  $X$  и  $Y$ , а также непересекающиеся множества  $(X - Y)$ ,  $(Y - X)$ ,  $(X \cap Y)$ .

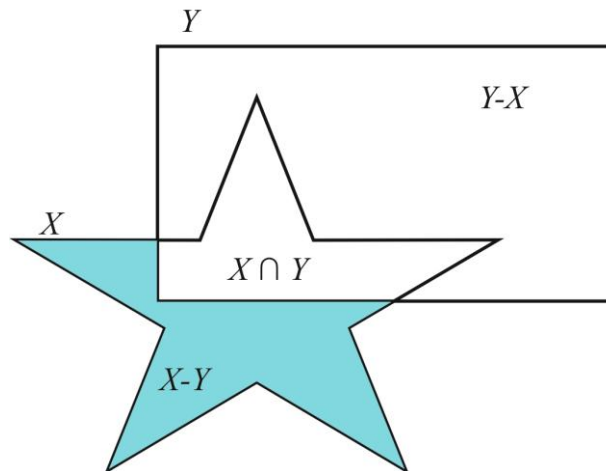


Рис. 2.4. Пересечение и разности множеств

Рассматривая рис. 2.4, можно записать следующие выражения для пересекающихся множеств:

$$X \cup Y = (X - Y) \cup (Y - X) \cup (X \cap Y);$$

$$P(X \cup Y) = P((X - Y) \cup (Y - X) \cup (X \cap Y)).$$

Так как  $(X - Y)$ ,  $(Y - X)$ ,  $(X \cap Y)$  – непересекающиеся множества,

$$P(X \cup Y) = P(X - Y) + P(Y - X) + P(X \cap Y). \quad (2.3)$$

Как видно, рис. 2.4, множество  $X$  состоит из непересекающихся множеств:

$$X = (X - Y) \cup (X \cap Y).$$

Тогда справедливо

$$P(X) = P((X - Y) \cup (X \cap Y)) = P(X - Y) + P(X \cap Y).$$

По аналогии получаем выражения:

$$Y = (Y - X) \cup (X \cap Y);$$

$$P(Y) = P((Y - X) \cup (X \cap Y)) = P(Y - X) + P(X \cap Y).$$

Сумма вероятностей  $P(X) + P(Y)$  равна

$$P(X) + P(Y) = P(X - Y) + P(X \cap Y) + P(Y - X) + P(X \cap Y).$$

Далее сгруппируем слагаемые следующим образом:

$$P(X) + P(Y) = [P(X - Y) + P(Y - X) + P(X \cap Y)] + P(X \cap Y). \quad (2.4)$$

Подставляя (2.3) в (2.4), получаем:

$$P(X) + P(Y) = P(X \cup Y) + P(X \cap Y).$$

Таким образом, на пересекающихся множествах  $X$  и  $Y$  справедлива формула вероятности

$$P(X \cup Y) = P(X) + P(Y) - P(X \cap Y). \quad (2.5)$$

Если события несовместны, т. е.  $X \cap Y = \emptyset$ , то формула (2.4) записывается в виде

$$P(X \cup Y) = P(X) + P(Y). \quad (2.6)$$

#### 2.4.2. Условная вероятность. Теорема Байеса

Необходимо определить вероятность события  $X$  при условии, что событие  $Y$  уже произошло. В этом случае выборочным пространством события  $X$  служит пространство  $Y$ , а событие  $X$  ограничивается вероятностью  $P(X \cap Y)$ , как показано на рис. 2.5.

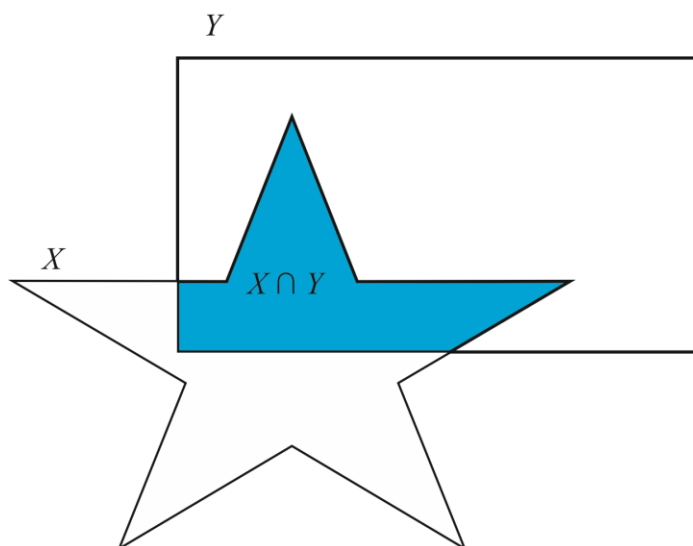


Рис. 2.5. Пересечение множеств

Тогда вероятность того, что произойдет как событие  $X$ , так и событие  $Y$  определяется формулой

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)}. \quad (2.7)$$

Аналогично, вероятность наступления  $Y$  при условии, если событие  $X$  произошло, определяется как

$$P(Y|X) = \frac{P(Y \cap X)}{P(X)} \quad (2.8)$$

Вероятности  $P(X \cap Y)$  и  $P(Y \cap X)$  тождественны совместной вероятности

$P(X, Y)$  появления двух событий  $X$  и  $Y$ . Так как операция пересечения удовлетворяет аксиоме коммутативности  $X \cap Y = Y \cap X$ , тогда из (2.6) следуют равенства:

$$P(X \cap Y) = P(X|Y)P(Y);$$

$$P(X, Y) = P(X|Y)P(Y). \quad (2.9)$$

Аналогично, из (2.8) следуют равенства:

$$P(Y \cap X) = P(Y|X)P(X);$$

$$P(Y, X) = P(Y|X)P(X). \quad (2.10)$$

Так как  $P(X, Y) = P(Y, X)$ , то

$$P(X|Y)P(Y) = P(Y|X)P(X) \quad (2.11)$$

Решая уравнение (2.11) относительно  $P(X|Y)$  получаем

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \quad (2.12)$$

Формула (2.12) известна как теорема Байеса (Bayes' theorem), или формула апостериорной вероятности. Эта формула имеет важное прикладное значение. Экспериментально вычисляя на выходе канала вероятность  $P(Y)$  формирования выходных символов, имея априорные значения вероятностей  $P(X)$  символов входа канала и зная свойства канала (переходные вероятности  $P(Y|X)$ ), можно найти вероятность  $P(X|Y)$  получения символов источника  $X$  на приемной стороне и, следовательно, иметь оценку качественных характеристик системы передачи информации.

*Замечание.* Оптимальная обработка сигналов и изображений реализуется на основе алгоритма, использующего принципы теоремы Байеса.

## 2.5. Модель связанных источников

Теоретическое описание дискретного канала без памяти может использовать представление передачи информации как связь двух источников: источника  $X$  и источника  $Y$ . В этом случае выход связанных источников может описываться парой событий  $(x_i, y_j)$ . Формирование символа  $x_i$  источника  $X$  связано с формированием символа  $y_j$  источника  $Y$  и наоборот. На рис. 2.6. показана модель связанных источников.

Пусть источник  $X$  формирует множество несовместных событий  $X = \{x_1, x_2, \dots, x_m\}$  таких, что одно из них непременно произойдет.

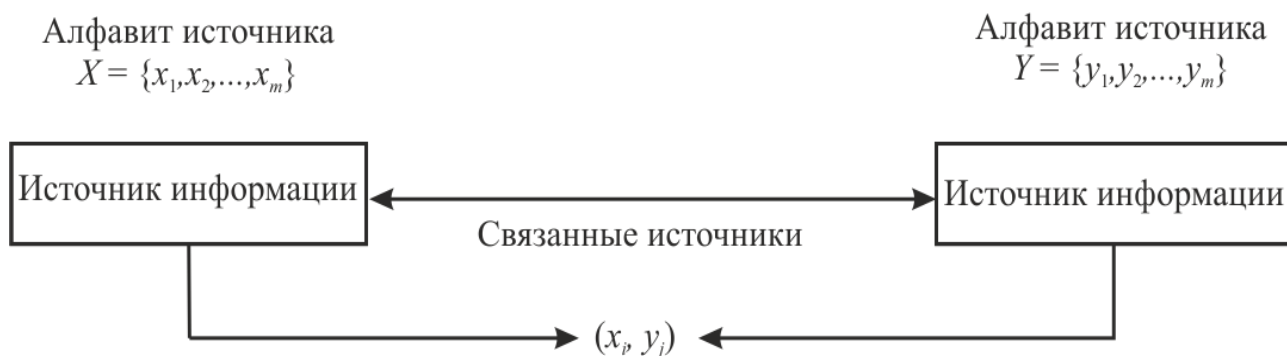


Рис. 2.6. Модель связанных источников

Напомним, два события  $x_i$  и  $x_j$  несовместны, если они взаимно исключают друг друга, т. е.  $x_i \cap x_j = \emptyset, i \neq j$ . Все элементарные события являются взаимоисключающими. Каждое элементарное событие принадлежит одному и только одному  $x_i$  из множества  $X = \{x_1, x_2, \dots, x_m\}$ .

Связанный, с источником  $X$ , источник  $Y$  формируются несовместные события  $\{y_1, y_2, \dots, y_m\}$ . Очевидно, объединение  $(x_1 \cup x_2 \cup \dots \cup x_m)$  всех  $x_i \in X$  дает пространство элементарных событий  $X$ . Объединение  $(y_1 \cup y_2 \cup \dots \cup y_m)$  всех  $y_i \in Y$  дает пространство  $Y$ . Для связанных источников любое событие  $y_i$  осуществляется только одновременно с некоторым событием  $x_j$  (и наоборот), рис. 2.6. Символически это утверждение можно записать как

$$y_i = (y_i x_1 \cup y_i x_2 \cup \dots \cup y_i x_m) \quad (2.13)$$

События  $yx_i$  и  $yx_j$  попарно несовместны (не пересекаются, т. е.  $yx_i \cap yx_j = \emptyset, i \neq j$ ), (2.3). Вероятность некоторого события  $y_i$  равна

$$p(y_i) = p(y_i x_1 \cup y_i x_2 \cup \dots \cup y_i x_m)$$

Для таких событий справедлива формула (2.6) когда их вероятности событий складываются. Тогда получаем следующее выражение вероятности некоторого события  $y_i$ :

$$p(y_i) = p(y_i x_1) + p(y_i x_2) + \dots + p(y_i x_m). \quad (2.14)$$

Рассмотрим формулу (2.14) на примере.

*Пример 2.1.* Имеется два связанных источника без памяти

$X = \{x_1, x_2\}, x_1 = 0, x_2 = 1$  и  $Y = \{y_1, y_2\}, y_1 = 0, y_2 = 1$ .

Из выражения (2.14) получают значения вероятностей  $y_1$  и  $y_2$ :

$$p(y_1) = p(y_1 x_1) + p(y_1 x_2); \quad (2.15)$$

$$p(y_2) = p(y_2 x_1) + p(y_2 x_2). \quad (2.16)$$

Подставляя выражение (2.10 ( $P(Y, X) = P(Y|X)P(X)$ )) совместной вероятности  $P$  в (2.15) и (2.16), получаем формулы:

$$p(y_1) = p(y_1|x_1)p(x_1) + p(y_1|x_2)p(x_2), \quad (2.17)$$

$$p(y_2) = p(y_2|x_1)p(x_1) + p(y_2|x_2)p(x_2). \quad (2.18)$$

Таким образом, распределение вероятностей  $p(y_i)$  символов источника  $Y$  и распределение вероятностей  $p(x_i)$  источника  $X$  связано следующим выражением:

$$p(y_i) = \sum_{j=1}^2 p(y_i|x_j)p(x_j), \quad (2.19)$$

где  $p(y_i|x_j)$  – условные вероятности появления символов связанных источников  $X$  и  $Y$ .

Возвращаясь к модели канала передачи информации, показанной на рис. 2.1, выражение (2.19) характеризует распределение вероятностей  $p(y_i)$  символов выхода канала (входа приемника), исходя из распределения вероятностей  $p(x_i)$  входа канала (выхода источника  $X$ ) и данных о переходных характеристиках  $p(y_i|x_j)$  дискретного канала без памяти.

Формула (2.19) имеет важное практическое значение, поскольку априорное знание статистических свойств  $p(y_i|x_j)$  канала передачи, (хранения) информации и статистических свойств  $p(x_i)$  исходного источника позволяет вычислить вероятности символов на выходе канала.

Используя формулу Байеса, можно найти вероятность  $p(x_i|y_i)$  получения символов источника  $X$  на приемной стороне. Полученные значения характеризуют такой важнейший показатель информационной системы как достоверность (точность) приема (обработки) информации.

В общем случае, для источников  $X = \{x_1, x_2, \dots, x_m\}$  и  $Y = \{y_1, y_2, \dots, y_m\}$  (для входа и выхода канала) формула (2.19) примет вид

$$p(y_i) = \sum_{j=1}^m p(y_i|x_j)p(x_j). \quad (2.20)$$

*Пример 2.2.* Имеется дискретный канал с входным источником  $X = \{x_1, x_2\}$ ,  $x_1 = 0, x_2 = 1$  и выходным источником  $Y = \{y_1, y_2\}$ ,  $y_1 = 0, y_2 = 1$ . Символы источника  $X$  появляются с вероятностью  $p(x_1) = 0,9$  и  $p(x_2) = 0,1$ .

В канале имеются шумы. Переходные вероятности канала соответственно равны:

$$p(y_1|x_1) = 0,99; p(y_2|x_1) = 0,01; p(y_2|x_2) = 0,95; p(y_1|x_2) = 0,05.$$

1. Определить вероятности появления символов на выходе канала с шумами.

Решение. По формуле (2.19) получаем:

$$p(y_1) = p(y_1|x_1)p(x_1) + p(y_1|x_2)p(x_2) =$$

$$= 0,99 \cdot 0,9 + 0,05 \cdot 0,1 = 0,896;$$

$$\begin{aligned} p(y_2) &= p(y_2|x_1)p(x_1) + p(y_2|x_2)p(x_2) = \\ &= 0,01 \cdot 0,9 + 0,95 \cdot 0,1 = 0,104. \end{aligned}$$

2. Найти вероятность  $p(x_i|y_i)$  получения символов источника  $X$  на приемной стороне в условиях присутствия шумов в канале.

Решение. По теореме Байеса (ф.  $P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}$ ) используя выражение для одиночных символов

$$p(x_i|y_j) = \frac{p(y_j|x_i)p(x_i)}{p(y_j)},$$

получаем:

$$p(x_1|y_1) = \frac{p(y_1|x_1)p(x_1)}{p(y_1)} = \frac{0,99 \cdot 0,9}{0,896} = 0,9944.$$

$$p(x_2|y_2) = \frac{p(y_2|x_2)p(x_2)}{p(y_2)} = \frac{0,95 \cdot 0,1}{0,104} = 0,9134.$$

Как видно, в канале с шумом значение вероятности правильного приема символа зависит от степени его повторяемости. Чем чаще он повторяется, тем достовернее прием.

Значения вероятностей ошибок в реальных каналах зависят от многих факторов:

- свойств физических каналов;
- свойств сигналов, которые являются физическими переносчиками сообщений;
- метода обработки сигналов на приемной стороне и пр;
- отношения средней мощности сигнала к средней мощности шума  $\frac{P_s}{P_N}$  на выходе канала передачи информации.

Величину отношения мощности сигнала к мощности шума часто выражают в логарифмическом масштабе

$$\frac{P_s}{P_N} = (10 \log_{10} \frac{P_s}{P_N}) \text{ dB}.$$

Например, для цифрового телевизионного вещания с хорошим качеством стандартные значения отношения  $\frac{P_s}{P_N}$  составляют (60 – 70) dB. В этом случае отношение  $\frac{P_s}{P_N}$  превышает величину  $10^6$ .

## 2.6. Количественная оценка информации

Понятие количества информации, предложенное К. Шенноном в 1948 году, определяется при выполнении трех аксиом.



1. Информация события (символа)  $x_i \in X$ , появляющегося с вероятностью  $p_i$ , имеет положительное значение

$$I(p_i) \geq 0.$$

2. Аксиома суммируемости информации.

Если независимые события  $(x_i, x_j)$  появляются с вероятностью  $p_i$  и  $p_j$ , то вероятность совместного события  $x_i$  и  $x_j$  равна  $P(x_i, x_j) = p_i \cdot p_j$ .

Напомним, если исход одного события не влияет на исход другого, то такие события называются независимыми.

*Пример 2.3.* Пусть двоичный источник без памяти  $X = \{0,1\}$ , формирует символ  $x_1$  с вероятностью  $p = 0,2$  и символ  $x_2$  с вероятностью  $(1 - p) = 0,8$ . Вероятность появления сообщения вида  $(x_2 x_1 x_1 x_2 x_1) = (10010)$  равна

$$P(x_2 x_1 x_1 x_2 x_1) = P(10010) = p^3 (1 - p)^2 = 0,2^3 (1 - 0,2)^2 = 0,00512.$$

Совместная информация двух независимых событий  $(x_i, x_j)$  с вероятностью совместного события  $p(x_i, x_j) = p_i \cdot p_j$  равна сумме их информаций

$$I(p_{i,j}) = I(p_i) + I(p_j).$$

Если Вы получили сообщение о том, что 1 июня температура воздуха в Минске достигнет  $+20^\circ\text{C}$  и, что экзамен состоится в аудитории 505-3 – это независимые события. Содержание этого сложного сообщения равняется сумме информации о погоде и экзамене.

3. Информация является непрерывной функцией вероятности события.

*Определение 2.5.* Количество информации, передаваемое источником при появлении одного символа  $x_i$  с вероятностью  $p$ , равно

$$I = \log \frac{1}{p} = -\log p. \quad (2.16)$$

Для логарифма может быть использовано основание 10, основание 2, основание  $e$  натуральных логарифмов. Разные основания только изменяют единицы меры информации. Измерение объема информации по формуле (2.16) впервые было предложено Р. В. Л. Хартли (R. V. L. Hartley (1888 – 1970), амер. ученый) в 1928 году. При использовании логарифмов с основанием 10 количество информации измеряется в единицах Хартли.

*Примеры*

2.4.1. Пусть  $p = 10^{-5}$ . Получаем  $I = \log_{10} \frac{1}{p} = -\log_{10} 10^{-5} = 5$  единиц информации Хартли.

2. 4.2. Пусть  $p = 10^{-1}$ . Получаем одну единицу информации Хартли

$$I = \log_{10} \frac{1}{p} = -\log_{10} 10^{-1} = 1.$$

2.4.3. Пусть  $p = 1$  (событие непременно состоится). Получаем ноль единиц информации Хартли

$$I = \log_{10} \frac{1}{p} = -\log_{10} 1 = 0.$$

При использовании логарифмов с основанием 2 количество информации измеряется в битах.

*Примеры*

2.5.1. Пусть  $p = \frac{1}{2}$ . Получаем

$$I = \log_2 \frac{1}{1/2} = -\log_2 \frac{1}{2} = 1 \text{ бит.}$$

2.5.2. Пусть  $p = \frac{1}{32}$ , тогда

$$I = \log_2 \frac{1}{1/32} = -\log_2 \frac{1}{32} = 5 \text{ бит.}$$

*Пример 2.6.* Пусть передается сообщение  $c = (x_1 x_2 x_3 x_4 x_5) = (10010)$  составленное из независимых символов  $x_i \in \{0,1\}$ . События  $x_i$  появляются с вероятностью  $p_i = \frac{1}{2}$ . Количество информации в этом сообщении равно

$$I = \log_2 \frac{1}{P(x_1 x_2 x_3 x_4 x_5)} = \log_2 \frac{1}{(\frac{1}{2})^5} = -\log_2 2^{-5} = 5 \text{ бит.}$$

Полученное значение соответствует сумме информаций 5 независимых событий

$$I(c) = I(p_1) + \dots + I(p_5) = 5 \text{ бит.}$$

На рис. 2.7 показан график, характеризующий количественное изменение  $I$  в зависимости от вероятности события.

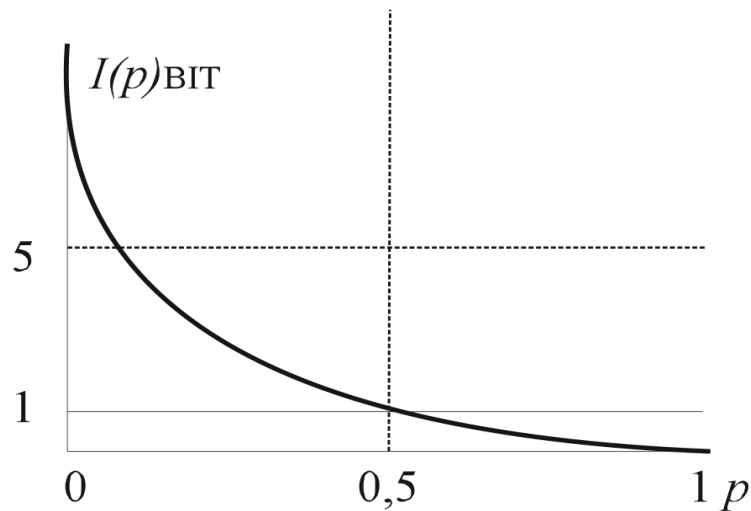


Рис. 2.7. Изменение количества информации  $I$  в зависимости от вероятности возникновения события

Как видно, с уменьшением вероятности появления события или увеличением его неопределенности, количество информации возрастает. Определение информации можно трактовать как некоторое отражение возникновения событий.

### Упражнения

2.1. Вычислить количество информации выдаваемой источником, если размерность алфавита  $X = \{x_1, x_2, \dots, x_m\}$  равна  $m = 3$ . Вероятность появления символов источника: события  $p_1 = 0,15$ ;  $p_2 = 0,5$ ;  $p_3 = 0,35$ .

2.2. Пусть передается сообщение  $(x_1 x_2 x_2 x_1 x_3 x_4 x_5 x_6)$ . Вычислить количество информации в этом сообщении. Размерность алфавита  $X = \{x_1, x_2, \dots, x_6\}$  равна  $m = 6$ . Вероятность появления символов источника:  $p_1 = 0,05$ ;  $p_2 = 0,15$ ;  $p_3 = 0,05$ ;  $p_4 = 0,4$ ;  $p_5 = 0,2$ ;  $p_6 = 0,15$ .

## 2.7. Энтропия

Пусть двоичный дискретный источник без памяти  $X = \{x_1, x_2\}$ ,  $m = 2$  формирует символ  $x_1 = 0$  с вероятностью  $p$  и символ  $x_2 = 1$  с вероятностью  $(1 - p)$ . Если получен символ  $x_1$ , то это сообщение оценивается количеством информации, равным

$$I(x_1) = -\log p.$$

Аналогично, при приеме символа  $x_2$  количество полученной информации определяется как

$$I(x_2) = -\log(1 - p).$$

Одной из характеристик двоичного дискретного источника без памяти является среднее количество (ожидаемое количество) информации выдаваемой источником. Так как источник формирует случайные события, то математическое ожидание определяется по формуле

$$E(I) = \sum_{i=1}^m p_i I(x_i) = \sum_{i=1}^2 p_i I(x_i) = p_1 I(x_1) + p_2 I(x_2) = \\ = -p \log p - (1-p) \log(1-p).$$

*Пример 2.7.* Пусть двоичный источник формирует символ  $x_1 = 0$  с вероятностью  $p = 0,2$  и символ  $x_2 = 1$  с вероятностью  $(1-p) = 0,8$ . Сообщение оценивается количеством информации, равным

$$I(x_1) = -\log 0,2 = 2,3219 \text{ бит}, I(x_2) = -\log 0,8 = 0,3219 \text{ бит}.$$

Полученные значения  $I(x_i)$  соответствуют точкам на графике, показанном на рис. 2.3.

Среднее значение количества информации источника

$$E(I) = 0,2 \cdot 2,3219 + 0,8 \cdot 0,3219 = 0,7219 \text{ бита}.$$

*Определение 2.6.* Энтропия  $H$  источника информации – это средняя информация, полученная для всех возможных событий.

Энтропия источника (*пример 2.7*) равна  $H = 0,7219$  бит/символ. В этом примере энтропия источника информации определяется как математическое ожидание количества информации

$$E(I) = H.$$

Для дискретного источника двух независимых событий  $X = \{0,1\}$  с вероятностями  $p$  и  $(1-p)$  энтропия определяется как

$$H = -p \log_2 p - (1-p) \log_2(1-p). \quad (2.17)$$

*Замечание.* Энтропия двоичного источника, вычисляемая по формуле (2.17) называется функцией (формулой) Шеннона.

На рис. 2.8 показан график энтропии двух событий как функция вероятности. Максимальное значение энтропии равно 1 бит/символ, когда

$$p_1 = p_2 = \frac{1}{2}.$$

$$H = -p \log_2 p - (1-p) \log_2(1-p) = \\ = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \text{ бит/символ}.$$

Это соответствует наибольшей неопределенности для двух событий. Для значения  $p$  равного нулю или единице события имеют полную определенность, и никакая информация не передается.

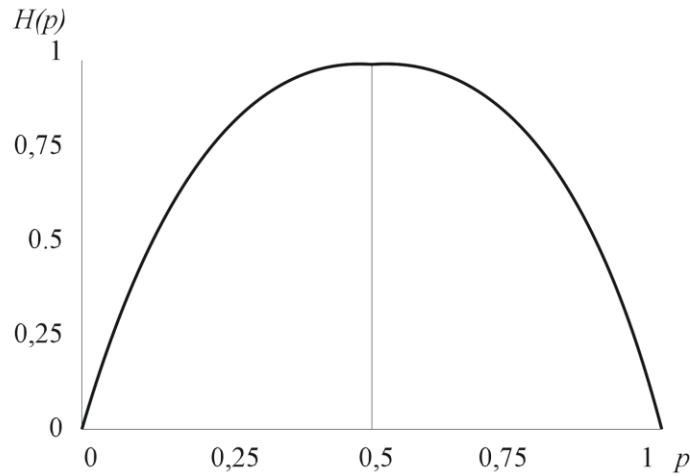


Рис. 2.8. Энтропия двоичного источника

*Определение 2.7.* Энтропия дискретного источника без памяти с символами алфавита  $X = \{x_1, x_2, \dots, x_m\}$  и соответствующими вероятностями  $p_1, p_2, \dots, p_m$  равна

$$H = H(p_1, p_2, \dots, p_m) = \sum_{i=1}^m -p_i \log p_i. \quad (2.18)$$

*Замечания.*

1. Величину (2.18) называют также неопределенностью источника.

2. Из *Определения 2.7.* следует понятие энтропии как среднее количество информации, приходящейся на один символ источника.

*Пример 2.8.* Вычислим энтропию источника с алфавитом из четырех символов  $X = \{x_1, x_2, x_3, x_4\} = \{a, b, c, d\}$  с вероятностями  $p_1 = \frac{1}{2}, p_2 = \frac{1}{4}, p_3 = \frac{1}{8}, p_4 = \frac{1}{8}$ .

*Решение*

$$\begin{aligned} H &= \sum_{i=1}^4 -p_i \log_2 p_i = -\left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{4} \log_2 \frac{1}{4} + \frac{1}{8} \log_2 \frac{1}{8} + \frac{1}{8} \log_2 \frac{1}{8}\right) = \\ &= \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1,75 \text{ бит/символ.} \end{aligned}$$

### 2.7.1. Свойства энтропии

1. Энтропия  $H(p_1, p_2, \dots, p_m) = \sum_{i=1}^m -p_i \log p_i$  является неотрицательной непрерывной функцией вероятностей событий  $p_1, p_2, \dots, p_m$ .

Доказательство неотрицательности  $H(p_1, p_2, \dots, p_m) \geq 0$  очевидно. Так как  $\log p_i \leq 0$ , то  $-\log p_i \geq 0$  для всех значений  $i = 1, 2, \dots, m$ .

2. Для дискретного источника без памяти с равной вероятностью  $p_i = \frac{1}{m}$  энтропия увеличивается с увеличением размерности алфавита  $m$ .

**Пример 2.9.** Вычислим энтропию источника с алфавитом из четырех символов  $X = \{x_1, x_2, x_3, x_4\} = \{a, b, c, d\}$  с равными вероятностями  $p_1 = \frac{1}{4}, p_2 = \frac{1}{4}, p_3 = \frac{1}{4}, p_4 = \frac{1}{4}$ .

Решение

$$H = \sum_{i=1}^4 -p_i \log_2 p_i = -\left(\frac{1}{4} \log_2 \frac{1}{4} + \frac{1}{4} \log_2 \frac{1}{4} + \frac{1}{4} \log_2 \frac{1}{4} + \frac{1}{4} \log_2 \frac{1}{4}\right) = 2 \text{ бит/символ.}$$

В *примере 2.8*, где разные значения вероятностей  $p_i$ ,  $H = 1,75$  бит/символ.

**Теорема 2.1.** Если все события имеют одинаковую вероятность  $p_1 = \dots = p_i \dots = p_m$ , энтропия дискретного источника без памяти максимальна и равна

$$H_0 = \log_2 m \text{ бит/символ.}$$

В этом случае неопределенность источника максимальна и источник передает максимально возможное среднее количество информации, приходящее на один символ (см. рис. 2.3 и *пример 2.9*).

**Определение 2.8.** Величина  $H_0$  определяет емкость дискретного источника как системы хранения информации.

3. Источник без памяти с разными значениями вероятности появления символов алфавита обладает энтропией, меньшей  $\log_2 m$ .

Сравнивая источник *примера 2.8*, где  $m = 4$ ,  $H = 1,75$  бит/символ с источником такого же размера, но с одинаковыми значениями вероятностями  $p_1 = p_2 = p_3 = p_4 = \frac{1}{4}$ , имеем

$$H = 1,75 < H_0 = \log_2 m = 2.$$

4. Энтропия блокового источника равна

$$H' = nH,$$

где  $H$  – энтропия источника одиночных символов.

**Пример 2.10.**

1) Источник формирует символы  $X = \{x_1, x_2\} = \{0, 1\}$  с вероятностями  $\{p_1 = \frac{1}{3}, p_2 = \frac{2}{3}\}$ . Размерность алфавита  $m = 2$ . Энтропия источника одиночных символов равна

$$H = \sum_{i=1}^2 -p_i \log_2 p_i = \frac{1}{3} \cdot 1,585 + \frac{2}{3} \cdot 0,585 = 0,918 \text{ бит/символ.}$$

2) Имеется блочный источник  $X^2 = \{c_1, c_2, c_3, c_4\}$ ,  $n = 2$ . Символы  $c_1 = (00), c_2 = (01), c_3 = (10), c_4 = (11)$  получены расширением источника одиночных символов  $X = \{x_1, x_2\} = \{0, 1\}$  с вероятностями  $\{p_1 = \frac{1}{3}, p_2 = \frac{2}{3}\}$ .

Вычислить энтропию источника  $X^2$ .

Решение

1. Вычисляем вероятности появления символов источника  $X^2$ . Вероятность  $P(x_i, x_j)$  совместного события  $x_i$  и  $x_j$  равна

$$P(x_i, x_j) = p_i \cdot p_j.$$

Получаем следующие значения вероятностей:

$$P(c_1) = p_1 \cdot p_1 = \frac{1}{9},$$

$$P(c_2) = p_1 \cdot p_2 = \frac{2}{9},$$

$$P(c_3) = p_2 \cdot p_1 = \frac{2}{9},$$

$$P(c_4) = p_2 \cdot p_2 = \frac{4}{9}.$$

2. Энтропия источника равна

$$\begin{aligned} H' = \sum_{i=1}^4 -P(c_i) \log_2 P(c_i) &= \frac{1}{9} \cdot 0,585 + \frac{2}{9} \cdot 2,1699 + \frac{2}{9} \cdot 2,1699 + \\ &+ \frac{4}{9} \cdot 1,1699 = 1,83 \text{ бит/символ.} \end{aligned}$$

Как видно, энтропия блочного источника определяется свойством 4,

$$H' = nH = 2 \cdot 0,918 = 1,83 \frac{\text{бит}}{\text{символ}}.$$

## 2.8. Относительная избыточность источника

*Определение 2.9.* Избыточность дискретного источника без памяти  $X = \{x_1, x_2, \dots, x_m\}$  – это разность между емкостью  $H_0$  источника и энтропией источника

$$R = H_0 - H. \quad (2.19)$$

*Определение 2.10.* Относительной избыточностью источника называется величина

$$r = \frac{R}{H_0} = 1 - \frac{H}{H_0}. \quad (2.20)$$

*Пример. 2.11.* Используя данные примера 2.8 (источник  $X = \{x_1, x_2, x_3, x_4\}$  с разными вероятностями,  $H = 1,75$ ) и значение  $H_0 = 2$  для такого же источника, но с одинаковыми вероятностями, получаем величину избыточности

$$R = 2 - 1,75 = 0,25.$$

Относительная избыточность источника равна

$$r = 1 - \frac{1,75}{2} = 0,125 \cong 12,5\%.$$

### *Упражнения*

2.3. Вычислить энтропию дискретного источника без памяти с символами алфавита  $X = \{a, b\}$  с вероятностью  $p_a = \frac{6}{8}, p_b = \frac{1}{4}$ .

2.4. Вычислить энтропию дискретного источника без памяти с символами алфавита  $X = \{a, b, c\}$  с вероятностью  $p_a = \frac{1}{2}, p_b = \frac{1}{3}, p_c = \frac{1}{6}$ .

2.5. Источник формирует символы  $X = \{x_1, x_2, \dots, x_6\} = \{A, K, N, D, E, !\}$ . Вероятности символов задаются множеством  $\{p_1 = 0,05, p_2 = 0,15, p_3 = 0,05, p_4 = 0,4, p_5 = 0,2, p_6 = 0,15\}$ .

2.5.1. Вычислить энтропию дискретного источника.

2.5.2. Вычислить емкость дискретного источника.

2.5.3. Вычислить избыточность дискретного источника.

2.5.4. Вычислить относительную избыточность дискретного источника.



### 3. ЭФФЕКТИВНОЕ КОДИРОВАНИЕ ДЛЯ ДИСКРЕТНОГО ИСТОЧНИКА БЕЗ ПАМЯТИ

#### 3.1. Условия взаимной однозначности алфавитного кодирования

Кодирование связано с преобразованием выходных символов дискретного источника (событий источника) в последовательность символов заданного кодового алфавита. На рис. 3.1 изображена математическая модель системы передачи информации с кодированием.

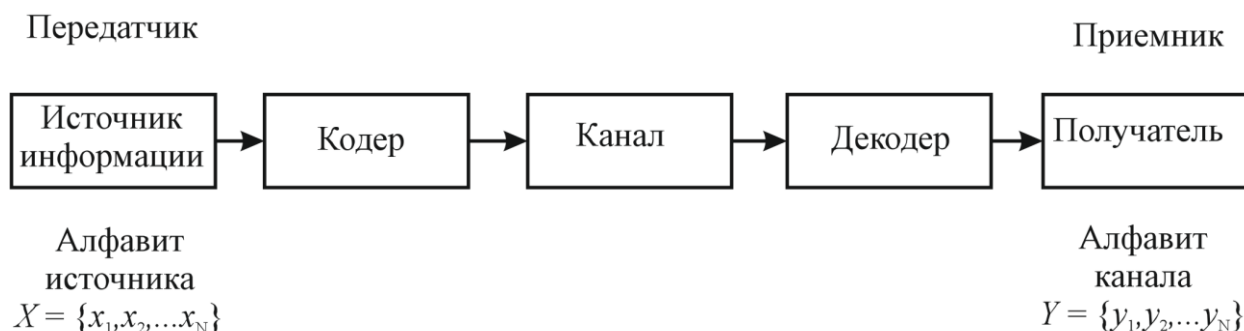


Рис. 3.1. Система передачи информации с кодированием

*Определение 3.1.* Код источника – это множество дискретных последовательностей всех событий, представленных символами кодового алфавита.

В качестве кодового алфавита часто используются символы двоичного  $\{0,1\}$  или бинарного алфавита  $\{1, -1\}$ . Например, слово  $x = (x_1 x_2 \dots x_7) = (-1 - 1 - 111 - 11)$  представлено символами бинарного алфавита источника,  $m = 2$ . В этом случае размерность кодового алфавита равна двум. На практике размерность (обозначается  $q$ ) кодового алфавита может быть большей.

*Определение 3.2.* Последовательности символов называются кодовыми словами или кодовыми векторами.

В результате кодирования осуществляется однозначное присвоение кодовых слов символам источника. Недопустимо присваивать разным символам источника одинаковые кодовые слова. Код должен удовлетворять условию сингулярности, когда каждое кодовое слово соответствует уникальному символу дискретного источника. Как правило, кодовые слова получают  $n$ -кратным расширением источника одиночных символов.

*Определение 3.3.* Длина кода  $n$  (значность) – число символов кодового слова.

Говорят, что двоичное слово  $x = (x_1 x_2 \dots x_n) = (1110010)$  имеет длину  $n = 7$ , слово  $l = (abbcab)$  длиной  $n = 6$ . Параметр  $n$  определяет следующие особенности кодов. Коды бывают:

- равномерные (блоковые),  $n = \text{const}$ ;
- неравномерные,  $n = \text{var}$ .

Код можно представить в виде списка, в котором каждому кодовому слову однозначно соответствует символ источника.

*Пример 3.1.* Пусть для передачи сообщения "DANKE" используются следующие кодовые слова равномерного кода:

$A \rightarrow (000);$   
 $K \rightarrow (010);$   
 $N \rightarrow (001);$   
 $D \rightarrow (111);$   
 $E \rightarrow (100).$

Для построения этого кода использовались символы двоичного источника  $X = \{0,1\}$ . Кодированному сообщению "DANKE" соответствует последовательность независимых символов  $x_i \in \{0,1\}$ ,

$DANKE \rightarrow (111000001010100).$

Пусть символы источника  $x_i$  появляются с вероятностью  $p_i = \frac{1}{2}$ . Количество информации в этом сообщении равно

$$I = \log_2 \frac{1}{P(x_1 \dots x_{15})} = \log_2 \frac{1}{(\frac{1}{2})^{15}} = -\log_2 2^{-15} = 15 \text{ бит.}$$

*Пример 3.2.* В случае неравномерного кодирования используем код со словами:

$A \rightarrow (00);$   
 $K \rightarrow (10);$   
 $N \rightarrow (010);$   
 $D \rightarrow (110);$   
 $E \rightarrow (111).$

Сообщению "DANKE" соответствует последовательность двоичных символов

$DANKE \rightarrow (1100001010111).$

Количество информации в этом сообщении равно

$$I = \log_2 \frac{1}{P(x_1 \dots x_{13})} = \log_2 \frac{1}{(\frac{1}{2})^{13}} = -\log_2 2^{-13} = 13 \text{ бит.}$$

Из примеров 3.1 и 3.2 следует очевидный вывод: неравномерный код более эффективный. Для передачи сообщения "DANKE" с использованием неравномерного кода потребовалось 13 бит. При равномерном кодировании того же сообщения затрачено 15 бит.

Правила кодирования должны отвечать следующим требованиям.

1. Необходимо добиваться высокой вероятности однозначного (правильного) декодирования исходной информации дискретного источника по закодированной последовательности.

2. Число символов кода, требуемого на один символ источника должно быть минимальным.

### 3.2. Эффективное кодирование

Основная идея эффективного кодирования базируется на использовании коротких кодовых слов для событий, характеризующихся высокой вероятностью. В этом случае уменьшается средняя длина закодированных сообщений. При этом должно обеспечиваться однозначное декодирование (желательно без введения дополнительных символов – меток синхронизации между кодовыми словами).

*Определение 3.4.* Код является эффективным, если он имеет наименьшую возможную среднюю длину кодового слова.

Многие алгоритмы эффективного кодирования в качестве однозначно декодируемого кода используют префиксные моментальные коды.

*Определение 3.5.* Префиксный код – это множество кодовых слов, в котором каждое кодовое слово не совпадает с началом более длинного слова.

#### 3.2.1. Моментальные коды

*Определение 3.6.* Если процесс однозначного декодирования каждого кодового слова осуществляется сразу же после приема всех символов кодового слова и принимается решение о соответствующем символе источника, то код с таким свойством называется моментальным.

Коды, рассмотренные в *примерах 3.1* и *3.2*, относятся к моментальным. Критерием моментальности кода является то, что ни одно слово, не совпадает с началом более длинного кодового слова.

##### 3.2.2.1. Код с запятой

Примером моментального кода служит код с запятой. Нуль в конце слова означает запятую, разделяющую кодовые слова. При получении нуля моментально принимается решение о декодировании соответствующего символа. На рис. 3.2 показан пример множества слов кода с запятой:

$$\begin{aligned}c_1 &\rightarrow (0); \\c_2 &\rightarrow (10);\end{aligned}$$

$$\begin{aligned}c_3 &\rightarrow (110); \\c_4 &\rightarrow (1110); \\c_5 &\rightarrow (11110).\end{aligned}$$

Рис. 3.2. Код с запятой

Например, последовательность  $c = (1111001101101011100)$  декодируется как  $c_5c_1c_3c_3c_2c_4c_1$ . Очевидно, введение разделительного символа уменьшает эффективность кодирования. На практике желательно использовать коды без запятой, когда при установленной синхронизации возможна передача кодированной информации без специального разделения кодовых слов. Кодовую конструкцию моментального кода удобно иллюстрировать с помощью кодового дерева.

### 3.2.2. Кодовое дерево

Кодовое дерево имеет начальную точку отсчета (корень). Из этой точки изображаются одна или две ветви. Ветвям присваиваются значения символов 0 и 1. Слева располагаются ребра, соответствующие символу 0, справа – ребра, соответствующие символу 1. Ветви заканчиваются промежуточными узлами. Затем из этих узлов строятся еще одна или две ветви и т. д. пока не будет изображен конечный узел, соответствующий каждому символу источника – кодовому слову. Кодовое слово получается в результате движения по ветвям от корня и записи символов 0 и 1 ветвей. Для однозначно декодируемого кода не должно быть конечных узлов на более длинном пути, заканчивающемся узлом, т. е. кодовым словом. Рассмотрим пример изображения кодового дерева для кодового алфавита  $\{0, 1\}$ .

*Пример 3.3.* Пусть символы источника задаются множеством  $X = \{A, K, N, D, E\}$ . Для кодирования источника используем префиксный код со словами:

$$\begin{aligned}A &\rightarrow (00); \\K &\rightarrow (10); \\N &\rightarrow (010); \\D &\rightarrow (110); \\E &\rightarrow (111).\end{aligned}$$

На рис. 3.3 показано кодовое дерево неравномерного кода. Черные точки (конечные узлы дерева) соответствуют словам кода.

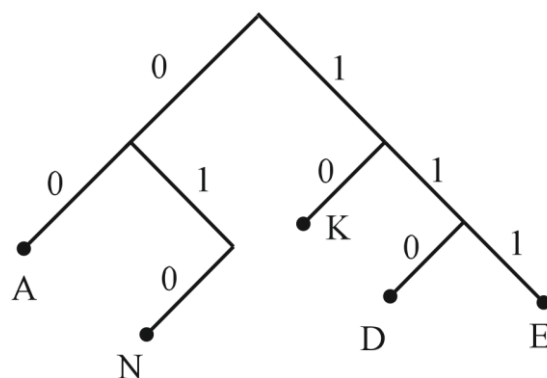


Рис. 3.3. Кодовое дерево неравномерного префиксного кода

Изображение кодового дерева равномерного кода из *примера 3.1*, ( $A \rightarrow (000)$ ,  $K \rightarrow (010)$ ,  $N \rightarrow (001)$ ,  $D \rightarrow (111)$ ,  $E \rightarrow (100)$ ) показано на рис. 3.4.

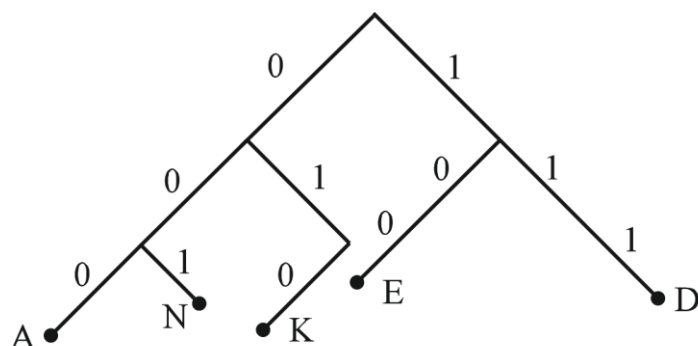


Рис. 3.4. Кодовое дерево равномерного кода

Изображение множества слов кода с запятой  $\{c_1 \rightarrow (0), c_2 \rightarrow (10), c_3 \rightarrow (110), c_4 \rightarrow (1110), c_5 \rightarrow (11110)\}$  показано на рис. 2.5.

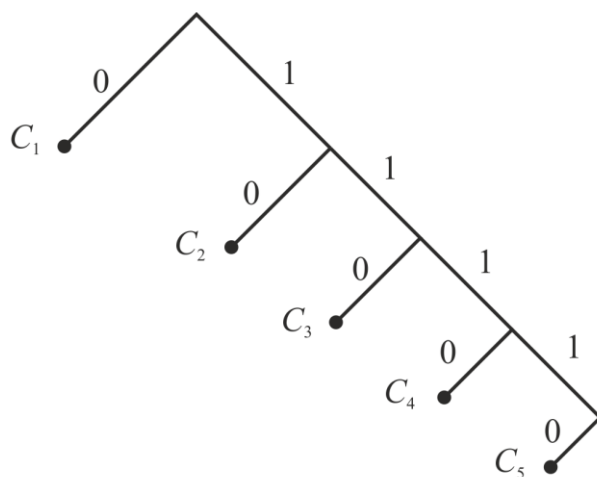


Рис. 3.5. Кодовое дерево кода с запятой

### Упражнения

3.1. Построить кодовое дерево кода  $X = \{x_1, x_2, \dots, x_8\}$ ,

где  $x_1 = (01), x_2 = (00), x_3 = (111), x_4 = (110), x_5 = (100), x_6 = (1011),$   
 $x_7 = (10101), x_8 = (10100).$

3.2. Построить кодовое дерево кода  $X = \{x_1, x_2, \dots, x_{11}\},$   
 где  $x_1 = (0001), x_2 = (001), x_3 = (01), x_4 = (010), x_5 = (0111), x_6 = (0110),$   
 $x_7 = (1000), x_8 = (1001), x_9 = (101), x_{10} = (110), x_{11} = (111).$

3.3.1. Является ли следующий код:

$A \rightarrow (01),$   
 $K \rightarrow (10),$   
 $N \rightarrow (011),$   
 $D \rightarrow (101)$

однозначно декодируемым?

3.3.2. Построить кодовое дерево этого кода.

### 3.3. Неравенство Крафта

Для ответа на вопрос, будет ли предлагаемое множество кодовых слов кода при декодировании точно соответствовать исходной информации источника, применяется неравенство Крафта.

*Определение 3.7.* Для построения однозначно декодируемого  $q$ -ичного кода, содержащего  $m$  кодовых слов с длинами  $n_1, n_2, \dots, n_m$ , необходимо и достаточно, чтобы выполнялось неравенство Крафта

$$\sum_{i=1}^m q^{-n_i} \leq 1, \quad (3.1)$$

где  $q$  обозначает число разных символов (размерность) кодового алфавита.

Для двоичного алфавита ( $q = 2$ ) неравенство Крафта записывается как

$$\sum_{i=1}^m q^{-n_i} = \sum_{i=1}^m 2^{-n_i} = 2^{-n_1} + 2^{-n_2} + \dots + 2^{-n_m} \leq 1. \quad (3.2)$$

Из (3.2) следует, что неравенство Крафта для двоичного равномерного кода примет следующий вид:

$$\sum_{i=1}^m 2^{-n_i} \equiv m 2^{-n} \leq 1. \quad (3.3)$$

Например, для  $m = 3, n = 2$  имеем:

$$\sum_{i=1}^3 2^{-n_i} = 2^{-n_1} + 2^{-n_2} + 2^{-n_3} = 2^{-2} + 2^{-2} + 2^{-2} = 3 \cdot 2^{-2} = \frac{3}{4} < 1.$$

Рассмотрим примеры применения неравенства Крафта.

*Пример 3.4.* Используются следующие кодовые слова длиной  $n = 3$  равномерного кода:

$$A \rightarrow (000);$$

$$K \rightarrow (010);$$

$$N \rightarrow (001);$$

$$D \rightarrow (111).$$

$$E \rightarrow (100).$$

Удовлетворяет ли код неравенству Крафта?

Решение. Так как  $n_1 = n_2 = \dots = n_5 = n = 3$ ,

$$m2^{-n} = 5 \cdot 2^{-3} = \frac{5}{8} < 1.$$

Данный код однозначно декодируемый.

*Замечание.* Для двоичного равномерного кода максимальное число кодовых слов равно

$$m = 2^n.$$

Это значение позволяет закодировать  $2^n$  символов блочного источника.

*Пример 3.5.* Пусть для кодирования используется префиксный код со словами:

$$A \rightarrow (00);$$

$$K \rightarrow (10);$$

$$N \rightarrow (010);$$

$$D \rightarrow (110);$$

$$E \rightarrow (111).$$

Удовлетворяет ли код неравенству Крафта?

Решение. Так как  $n_1 = n_2 = 2, n_3 = n_4 = n_5 = 3, m = 5$ ,

$$\sum_{i=1}^5 2^{-n_i} = \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} = \frac{7}{8} < 1.$$

Данный код также однозначно декодируемый.

### Упражнения

3.4. Является ли код  $X = \{x_1, x_2, \dots, x_8\} =$   
 $= \{x_1 = (01), x_2 = (00), x_3 = (111), x_4 = (110), x_5 = (100), x_6 = (1011),$   
 $x_7 = (10101), x_8 = (10100)\}$

однозначно декодируемым?

3.5. Является ли код  $X = \{x_1, x_2, \dots, x_{11}\} =$   
 $\{x_1 = (0001), x_2 = (001), x_3 = (01), x_4 = (010), x_5 = (0111), x_6 = (0110),$   
 $x_7 = (1000), x_8 = (1001), x_9 = (101), x_{10} = (110), x_{11} = (111)\}.$

однозначно декодируемым?

3.6. Из каких следующих значений длин кодовых слов можно построить однозначно декодируемый код?

3.6.1.  $n_1 = 2, n_2 = 2, n_3 = 2, n_4 = 3, n_5 = 3.$

3.6.2.  $n_1 = 1, n_2 = 2, n_3 = 3, n_4 = 3, n_5 = 8.$

3.6.3.  $n_1 = 1, n_2 = 2, n_3 = 3, n_4 = 4, n_5 = 4$ .

### 3.4. Средняя длина кодового слова

*Определение 3.8.* Мерой эффективности кода является его средняя длина кодовых слов

$$L_n = \sum_{i=1}^m P_i l_i, \quad (3.4)$$

где  $m$  – число символов источника с  $n$ -кратным расширением источника одичных символов,  $P_1, \dots, P_m$  – вероятности символов источника с  $n$ -кратным расширением,  $l_1, \dots, l_m$  – длина соответствующих кодовых слов.

*Пример 3.6.* Пусть для передачи сообщения "DANKE" используется следующие кодовые слова равномерного кода:

$A \rightarrow (000);$   
 $K \rightarrow (010);$   
 $N \rightarrow (001);$   
 $D \rightarrow (111);$   
 $E \rightarrow (100).$

Для построения этого кода использовались символы двоичного источника  $X = \{0,1\}$ . Пусть код характеризуется вероятностями  $P_1 = P_2 = P_3 = P_4 = P_5 = \frac{1}{5}$ . Код имеет среднюю длину

$$L_n = \sum_{i=1}^5 \frac{1}{5} l_i = \frac{1}{5} 3 + \dots + \frac{1}{5} 3 = \frac{15}{5} = 3.$$

*Пример 3.7.* Пусть для передачи сообщения "DANKE" используется следующие кодовые слова неравномерного кода:

$A \rightarrow (00);$   
 $K \rightarrow (10);$   
 $N \rightarrow (010);$   
 $D \rightarrow (110);$   
 $E \rightarrow (111).$

Для построения этого кода использовались символы двоичного источника  $X = \{0,1\}$ . Пусть код характеризуется вероятностями  $P_1 = P_2 = P_3 = P_4 = P_5 = \frac{1}{5}$ . Код имеет среднюю длину

$$L_n = \sum_{i=1}^5 \frac{1}{5} l_i = \frac{1}{5} 2 + \frac{1}{5} 2 + \frac{1}{5} 3 + \frac{1}{5} 3 + \frac{1}{5} 3 = 2,6.$$

Очевидно, более эффективной является та информационная система, которая использует коды с минимально возможными длинами кодовых слов.



### 3.4.1. Средняя длина кодового слова и энтропия

Понятие энтропии источника как среднее количество информации, передаваемое одним символом источника, отражает связь величины энтропии с величиной средней длины слов эффективного кода. Это отражение выражается двумя соотношениями.

*Соотношение 1.* Неравенство длины кода.

Средняя длина  $L$  двоичного однозначно декодируемого кода удовлетворяет неравенству

$$L \geq H. \quad (3.5)$$

Выражение (3.5) определяет минимально достижимую среднюю длину эффективного кода.

*Пример 3.8.* Пусть используется префиксный код со словами:

$$\begin{aligned} A &\rightarrow (00); \\ K &\rightarrow (10); \\ N &\rightarrow (010); \\ D &\rightarrow (110); \\ E &\rightarrow (111). \end{aligned}$$

Вероятности символов источника характеризуются множеством  $\{P(A), \dots, P(E)\} \rightarrow \{p_1 = \frac{1}{2}, p_2 = \frac{1}{4}, p_3 = \frac{1}{8}, p_4 = \frac{1}{16}, p_5 = \frac{1}{16}\}$ . Энтропия источника равна:

$$H = \sum_{i=1}^5 -p_i \log_2 p_i = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{16} \cdot 4 + \frac{1}{16} \cdot 4 = \frac{7}{4} = 1,875 \text{ бит/символ.}$$

Средняя длина кодового слова равна:

$$L = \sum_{i=1}^m p_i l_i = \frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{16} \cdot 3 + \frac{1}{16} \cdot 3 = \frac{9}{4} = 2,25.$$

В рассмотренном примере для средней длины кодового слова выполняется *Соотношение 1*:

$$L = 2,25 > H = 1,875.$$

*Соотношение 2.* Пределы средней длины кодового слова.

Можно получить кодирование источника двоичным кодом, у которого средняя длина кодового слова удовлетворяет выражению

$$H \leq L \leq H + 1. \quad (3.6)$$

При этом выражение

$$L \leq H + 1 \quad (3.7)$$

определяет максимально возможное значение средней длины двоичного эффективного кода. Если вернуться к *примеру 3.8*, для кода имеем

$$H = 1,875 < L = 2,25 < H + 1 = 1,875 + 1.$$

## 4. ТЕОРЕМА ШЕННОНА О КОДИРОВАНИИ ДЛЯ КАНАЛА БЕЗ ШУМА (первая теорема Шеннона)

Если в канале передачи информации вероятность  $p \rightarrow 0$  (ошибки маловероятны), основное требование к информационной системе – это представление символов источника в максимально компактной форме. Первая теорема Шеннона определяет минимально достижимую длину кодового слова на символ источника. Учитывая статистические свойства источника, можно более эффективно передавать (обрабатывать) информацию. Если высоковероятным символам поставить в соответствие более короткие длины кодовых слов, а маловероятным символам – слова большей длины, в этом случае достигается увеличение скорости передачи информации. Количество передаваемой информации за единицу времени также увеличится.

### 4.1. Энтропия блокового источника

Пусть имеется источник блоковых символов  $X^n$ . Эти блоки имеют длину  $n$ . Первая теорема Шеннона утверждает, что если  $n \rightarrow \infty$ , можно произвести кодирование блоковых символов источника  $X^n$  кодом, у которого средняя длина кодового слова будет приближаться к энтропии  $H$  источника  $X$ .

Если рассматривать блоки символов длиной  $n$  как новые независимые события с энтропией  $H' = nH$  (см. свойство 4 энтропии блокового источника) в соответствии с выражением (3.5) можно записать

$$H' \leq L \leq H' + 1, \quad (4.1)$$

где  $L$  – средняя длина слова источника  $X^n$ . Это же значение  $L$  определяет среднюю длину кодового слова последовательности  $n$  символов источника  $X$ .

Перепишем (4.1) как

$$nH \leq L \leq nH + 1.$$

где  $H$  – энтропия источника одиночных символов. Полученное выражение преобразуем к виду

$$H \leq \frac{L}{n} \leq H + \frac{1}{n}, \quad (4.2)$$

где  $\frac{L}{n}$  – средняя длина слова на один символ источника  $X$  (на одно событие).

Таким образом, расширяя источник одиночных символов, т. е. увеличивая значение  $n$  – длину блока, при  $n \rightarrow \infty$  можно закодировать символы блокового источника кодовыми словами со средней длиной на символ источника  $X$ , приближающейся к значению энтропии источника  $X$ .

### 4.2. Первая теорема Шеннона

Кодируя блоковые последовательности источника без памяти  $X = \{x_1, x_2, \dots, x_m\}$  с энтропией  $H$ , можно построить  $q$ -ичный префиксный код, в

котором средняя длина кодового слова удовлетворяет выражению

$$\lim_{n \rightarrow \infty} \frac{L_n}{n} = H,$$

где  $L_n$  – средняя длина кодовых слов префиксного кода.

*Замечание.* Составляющая  $\frac{L_n}{n} = H + \frac{1}{n}$  выражения (4.2) определяет максимальное значение средней длины кодового слова префиксного кода.

*Вывод.* Эффективное кодирование информации требует использования источника с  $n$ -кратным расширением источника  $X$ .

*Пример 4.1.* Вычислим энтропию двоичного источника с символами алфавита  $X = \{a, b\}$  с вероятностью  $p_1 = \frac{7}{8}, p_2 = \frac{1}{8}$ .

Решение. Применяя формулу Шеннона (2.17), получаем

$$H = -p_1 \log_2 p_1 - p_2 \log_2 p_2 = -\frac{7}{8} \log_2 \frac{7}{8} - \frac{1}{8} \log_2 \frac{1}{8} =$$

$$= \frac{7}{8} \cdot 0,1926 + \frac{1}{8} \cdot 3 = 0,54 \text{ бит/символ.}$$

Источник дает менее 1 бита информации на символ. Символы источника кодируются словами  $a \rightarrow 0, b \rightarrow 1$ . Длина кодовых слов равна  $l_1 = l_2 = 1$ .

Средняя длина слова кода равна

$$L = \sum_{i=1}^2 p_i l_i = \frac{7}{8} \cdot 1 + \frac{1}{8} \cdot 1 = 1.$$

*Пример 4.2.* Для увеличения количества передаваемой информации, используем расширение источника одиночных символов *примера 4.1*. Построим блоковый источник  $X^2 = \{(aa), (ab), (ba), (bb)\} = \{c_1, c_2, c_3, c_4\}$ .

Решение. Вероятности появления символов с 2-кратным расширением источника одиночных символов равны:

$$\begin{aligned} P_1 &= P(c_1) = p_1 p_1 = \frac{49}{64}; \\ P_2 &= P(c_2) = p_1 p_2 = \frac{7}{64}; \\ P_3 &= P(c_3) = p_2 p_1 = \frac{7}{64}; \\ P_4 &= P(c_4) = p_2 p_2 = \frac{1}{64}. \end{aligned}$$

Для кодирования блокового источника применим префиксный код:

$$\begin{aligned} (aa) &\rightarrow 0; \\ (ab) &\rightarrow 10; \end{aligned}$$

$$\begin{aligned}(ba) &\rightarrow 110; \\ (bb) &\rightarrow 111.\end{aligned}$$

Средняя длина  $L_n$  этого кода равна:

$$L_n = \sum_{i=1}^4 P_i l_i = \frac{49}{64} \cdot 1 + \frac{7}{64} \cdot 2 + \frac{7}{64} \cdot 3 + \frac{1}{64} \cdot 3 = \frac{87}{64}.$$

Средняя длина слова на один символ источника равна:

$$\frac{L_n}{n} = \frac{\frac{87}{64}}{2} = \frac{87}{128} = 0,68.$$

Как видно, средняя длина кода источника  $X = \{a, b\}$  уменьшилась с

$$L = 1 \text{ до } \frac{L_n}{n} = 0,68 > H = 0,54.$$

Энтропия блокового источника равна

$$\begin{aligned}H' = \sum_{i=1}^4 -P(c_i) \log_2 P(c_i) &= \left(-\frac{49}{64} \log_2 \frac{49}{64}\right) + \left(-\frac{7}{64} \log_2 \frac{7}{64}\right) + \left(-\frac{7}{64} \log_2 \frac{7}{64}\right) + \\ &+ \left(-\frac{1}{64} \log_2 \frac{1}{64}\right) = 1,08 \frac{\text{бит}}{\text{символ}(H')}.\end{aligned}$$

Заметим, энтропия блокового источника в  $n$  раз больше энтропии соответствующего источника одиночных символов

$$H' = n H = 2 \cdot 0,54 = 1,08 \frac{\text{бит}}{\text{символ}(H')}.$$

*Пример 4.3.* Источник формирует символы  $X = \{x_1, x_2\} = \{a, b\}$  с вероятностями  $\{p_1 = \frac{1}{3}, p_2 = \frac{2}{3}\}$ . Размерность алфавита  $m = 2$ . Энтропия источника равна

$$H = \sum_{i=1}^2 -p_i \log_2 p_i = \frac{1}{3} \cdot 1,585 + \frac{2}{3} \cdot 0,585 = 0,918 \text{ бит/символ}.$$

Символы источника кодируются словами  $a \rightarrow 0, b \rightarrow 1$ . Длина кодовых слов равна  $l_1 = l_2 = 1$ .

*Пример 4.4.* Для увеличения количества передаваемой информации, используем источник одиночных символов примера 4.3. Построим блоковый источник  $X^2 = \{(aa), (ab), (ba), (bb)\} = \{c_1, c_2, c_3, c_4\}$ . Выход этого источника принимает одно из состояний множества  $C = \{c_1, c_2, c_3, c_4\}$ .

1. Вычислим вероятности появления символов множества  $C = \{c_1, c_2, c_3, c_4\}$  как независимых событий источника  $X = \{x_1, x_2\}$  с вероятностями символов  $x_1 = a \rightarrow p_1 = \frac{1}{3}$  и  $x_2 = b \rightarrow p_2 = \frac{2}{3}$ .

$$P(c_1) = p_1 \cdot p_1 = \frac{1}{9}, P(c_2) = p_1 \cdot p_2 = \frac{2}{9},$$

$$P(c_3) = p_2 \cdot p_1 = \frac{2}{9}, P(c_4) = p_2 \cdot p_2 = \frac{4}{9}.$$

Для кодирования блокового источника применим следующий равномерный блоковый код:

$$(aa) \rightarrow 00;$$

$$(ab) \rightarrow 10;$$

$$(ba) \rightarrow 01;$$

$$(bb) \rightarrow 11.$$

2. Средняя длина  $L_n$  этого кода равна:

$$L_n = \sum_{i=1}^m P(c_i) l_i = \frac{1}{9} \cdot 2 + \frac{2}{9} \cdot 2 + \frac{2}{9} \cdot 2 + \frac{4}{9} \cdot 2 = 2.$$

3. Средняя длина на один символ источника равна:

$$\frac{L_n}{n} = \frac{2}{2} = 1.$$

Как видно, средняя длина кода источника  $X = \{a, b\}$  не уменьшилась

$$\frac{L_n}{n} = 1 > H = 0,918,$$

т. к. в этом примере использовалось равномерное кодирование источника.

4. Энтропия блокового источника равна

$$H' = \sum_{i=1}^4 -P(c_i) \log_2 P(c_i) = \frac{1}{9} \cdot 0,585 + \frac{2}{9} \cdot 2,1699 + \frac{2}{9} \cdot 2,1699 +$$

$$+ \frac{4}{9} \cdot 1,1699 = 1,83 \frac{\text{бит}}{\text{символ}(H')}.$$

Энтропия блокового источника в 2 раза больше энтропии соответствующего источника одиночных символов.

$$H' = n H = 2 \cdot 0,918 = 1,83.$$

Следует отметить, что применение рассмотренного метода эффективного представления информации приводит к увеличению сложности технической реализации кодирования, увеличения сложности декодирования, увеличения времени на процесс декодирования.

### Упражнения

4.1. Источник формирует символы  $X = \{x_1, x_2\} = \{a, b\}$  с вероятностями  $\{p_1 = \frac{9}{10}, p_2 = \frac{1}{10}\}$ . Имеется блочный источник с двукратным расширением  $X^2 = \{(aa), (ab), (ba), (bb)\} = \{c_1, c_2, c_3, c_4\}$ . Для кодирования блочного источника применяется префиксный код:

$$\begin{aligned}c_1 &\rightarrow (0); \\c_2 &\rightarrow (10); \\c_3 &\rightarrow (110); \\c_4 &\rightarrow (111).\end{aligned}$$

4.1.1. Вычислить энтропию источника.

4.1.2. Вычислить энтропию блочного источника.

4.1.3. Вычислить среднюю длину слова декодируемого кода.

4.1.4. Вычислить среднюю длину слова на один символ источника  $X$ .

4.2. Источник формирует символы  $X = \{x_1, x_2\}$  с вероятностями  $\{p_1 = \frac{9}{10}, p_2 = \frac{1}{10}\}$ . Имеется блочный источник с трехкратным расширением  $X^3 = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ . Для кодирования блочного источника применяется префиксный код:

$$\begin{aligned}c_1 &\rightarrow (1); \\c_2 &\rightarrow (011); \\c_3 &\rightarrow (010); \\c_4 &\rightarrow (001); \\c_5 &\rightarrow (00011); \\c_6 &\rightarrow (00010); \\c_7 &\rightarrow (00001); \\c_8 &\rightarrow (00000).\end{aligned}$$

4.2.1. Вычислить энтропию источника.

4.2.2. Вычислить энтропию блочного источника.

4.2.3. Вычислить среднюю длину слова декодируемого кода.

4.2.4. Вычислить среднюю длину слова на один символ источника  $X$ .

### 4.3. Сжатие данных

В настоящее время большая часть передаваемых, хранимых, распределяемых, преобразуемых данных соответствует звуковой, графической или видеоинформации. В этом случае реализация современных инфокоммуникационных систем неизбежно усложняется. Увеличиваются технические затраты на хранение данных, предъявляются более высокие требования по экономии канального частотного ресурса. Сжатие данных позволяет уменьшить объем данных, используемых для представления информации. Даже при постоянном росте емкости хранения данных и пропускной способности каналов сжатие остается необходимым и существенным компонентом информационных технологий. Различают два основных метода кодирования данных используемых для

сжатия: кодирование с потерями и кодирование без потерь.

Идея сжатия основывается на возможности устранения или уменьшения избыточности передаваемых (хранимых) данных. Сигнал, несущий информацию, можно сжать путем удаления из него имеющейся избыточности.

Различают два вида избыточности.

1. Видео-аудио (субъективная) избыточность, которую можно устранить с некоторой потерей информации, сравнительно мало влияющей на качество воспроизводимого изображения или звука.

Например, в обычных телевизионных каналах, передача видеoinформации осуществляется в полосе 6 МГц. Хотя видеоспектр имеет значительно большее значение, часть спектральных составляющих отбрасывается. В этом случае исходная видеoinформация не может быть полностью восстановлена.

Примером сжатия с потерями, когда отбрасывается несущественная информация, может служить система цифрового покомпонентного телевидения (Digital Video Broadcasting, DVB). В покомпонентном телевидении разделённые сигналы яркости  $Y$  и цветоразностные сигналы  $R - Y$  и  $B - Y$  квантуются на 256 уровней. Длина кодового слова, соответствующего каждому уровню яркости равна  $n = 8$ . При дискретизации телевизионного сигнала с частотой Найквиста-Котельникова ширина полосы частот цифрового полного телевизионного сигнала составляет величину

$$W = f_{d_Y}n + f_{d_R}n + f_{d_B}n = 13,5 \cdot 8 + 6,75 \cdot 8 + 6,75 \cdot 8 = 216 \text{ МГц},$$

где частота дискретизации яркостного канала  $f_{d_Y} = 13,5$  МГц, частота дискретизации цветоразностных сигналов  $f_{d_R} = f_{d_B} = 6,75$  МГц.

Скорость передачи информации достигает значения  $R = 216 \frac{\text{Мбит}}{\text{с}}$ . В этом случае за одну секунду передаются данные объемом 216 Мбит.

Телевидение высокой четности для формата HDTV (High Definition Television) имеет примерно удвоенную разрешающую способность по горизонтали и как минимум, удвоенную разрешающую способность по вертикали и соответствующее увеличение объема цифрового информационного потока до значения  $\approx 576$  Мбит. Стандартный телевизионный каналный частотный ресурс находится в диапазоне 48 – 862 МГц. Очевидно, без эффективного сжатия невозможно реализовать технологию многоканальной передачи информации в формате HDTV. В этой технологии сжатие (кодирование) видеосигнала реализуется на основе стандарта MPEG (Motion Pictures Experts Group). Стандарт разработан Экспертной группой по вопросам движущихся изображений. Кодирование основано на удалении, невоспринимаемой органами зрения, отдельных спектральных составляющих видеосигнала.

2. Статистическая избыточность, связанная с корреляцией и предсказуемостью обрабатываемых данных. Такая избыточность может быть полностью устранена без потери информации и исходные данные могут быть полностью восстановлены. Технология сжатия осуществляется за счет применения эффективного кодирования данных источников. В этом случае под избыточностью



понимают частое повторение символов информационного сообщения, повторяемость слов и предложений сообщений. Примером такого алгоритма является кодирование источника с помощью кода Хаффмена.

Эффективность сжатия оценивается коэффициентом

$$K = \frac{N-M}{N}, \quad (4.3)$$

где  $N$  обозначает затраты на передачу (хранение) данных без сжатия,  $M$  – затраты на передачу (хранение) данных со сжатием.

Например, передача (хранение) яркостей всех 64 пикселей фрагмента изображения размером  $8 \times 8$  реализуется посредством использования  $N = 2^6 \cdot 2^3 = 2^9 = 512$  бит. Здесь значение яркости каждого пикселя кодируется двоичным кодом длиной 8 бит. Пусть с использованием метода сжатия по стандарту *MPEG-2* получено  $M = 128$  бит. Тогда эффективность сжатия

$$K = \frac{512-128}{512} = 0,75,$$

что соответствует 75%.

Преобразуем формулу (4.3) следующим образом:

$$\begin{aligned} KN &= N - M, \\ N &= KN + M, \end{aligned}$$

$$\frac{N}{M} = \frac{KN + M}{M}.$$

В этом случае отношение вида  $\frac{N}{M}$  характеризует выигрыш в записи данных. Для рассматриваемого примера выигрыш равен

$$\frac{N}{M} = \frac{KN + M}{M} = \frac{0,75 \cdot 512 + 128}{128} = 4.$$

*Пример 4.5.* Оценить эффективность сжатия стереофонического аудио усовершенствованной системы кодирования звука *MPEG-2 AAC* (Advanced Audio Coding; стандарт разработан в 1998 году в Институте интегральных схем Фраунгофера – *IIS-A*, Германия) со скоростью 112 Кбит/с. В качестве сравнения рассмотрим кодирование по стандарту стереофонического аудио компакт дисков (*CD-Audio*), где сжатия практически нет. Стандартная частота дискретизации  $f_d$  непрерывного аудио сигнала для *CD*  $f_d = 44,1$  КГц. Длина кодовых слов  $n = 16$ .

Решение. Скорость передачи стереоданных аудио- *CD* определяется как

$$R = 2 \cdot f_d \cdot n = 2 \cdot 44,1 \cdot 16 = 1411,2 \text{ Кбит/с.}$$

Эффективность сжатия

$$K = \frac{1411,2 - 112}{1411,2} \cong 0,92.$$

Выигрыш в записи данных

$$\frac{N}{M} = \frac{1411,2}{112} = 12,6.$$

*Замечание.* Сравнительно высокая степень сжатия методом кодирования ААС отражается на качественных характеристиках воспроизводимого аудио сигнала.

Одним из первых методов сжатия без потерь является метод Шеннона-Фано.

#### 4.3.1. Коды Шеннона-Фано

Код по своему построению удовлетворяет свойству префикса. Средняя длина моментального кода Шеннона-Фано приближается к границе, определяемой энтропией. Однако метод не всегда дает оптимальное пространство кодовых слов. Алгоритм необязательно минимизирует среднюю длину слова.

#### 4.4. Энтропийное кодирование методом Хаффмена

Ранее было показано, чем более неравномерно распределены вероятности появления символов источника, тем меньше энтропия. Так, в *примере 4.1* для  $X = \{a, b\}$ ,  $p_1 = \frac{7}{8}$ ,  $p_2 = \frac{1}{8}$ , было получено  $H = 0,54$  бит/символ. Для другого источника с большей неопределенностью (*пример 4.3*,  $X = \{x_1, x_2\} = \{a, b\}$ ,  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{2}{3}$ .) энтропия источника  $H = 0,918$  бит/символ. С увеличением неопределенности появления случайного события энтропия увеличивается.

Из первой теоремы Шеннона следует, что средняя длина кодового слова источника может находиться в диапазоне

$$H \leq L \leq H + 1$$

и определяется величиной энтропии источника.

Таким образом, кодирование будет тем эффективнее, чем ближе средняя длина кодового слова приближается к энтропии и чем меньше значение энтропии. Метод эффективного кодирования источника информации, основанный на понятии энтропии, называется энтропийным. Один из основных методов такого кодирования известен как кодирование Хаффмена (D.A. Huffman, амер. ученый). В 1952 году Хаффмен показал, что разработанный им алгоритм эффективного кодирования позволяет строить класс оптимальных префиксных кодов без запятой, т. е. неравномерных кодов. Эти коды позволяют кодировать символы источника с минимальной избыточностью. Кодирование Хаффмена формирует оптимальный код для дискретных источников без памяти. Средняя длина  $L$  кодового слова кода Хаффмена приближается к энтропии источника  $H$ . В настоящее время коды Хаффмена используются при цифровой обработке изображений и звуков. Они составляют основную часть стандартов сжатия изо-

бражений и звука MPEG, H.264, JPEG (Joint Photographic Experts Group, JPEG разработан Объединенной группой экспертов по обработке фотографических изображений).

#### 4.4.1. Алгоритм Хаффмена

Алгоритм включает в себя выполнение ряда итерационных действий.

1. Упорядочение. Необходимо расставить символы источника в порядке уменьшения их вероятностей.

2. Редукция. Необходимо объединить два символа с наименьшими вероятностями в один символ.

3. Переупорядочение. Необходимо расставить символы в порядке уменьшения их вероятностей.

4. Продолжить процессы 2 и 3 до тех пор, пока все символы не будут объединены. В случае, когда несколько символов имеют одинаковые вероятности, объединяются те из них, которые имели до этого меньшее число объединений.

5. Кодирование. Необходимо начать процесс кодирования с последнего объединения. Приписать первой компоненте составного символа значение 0, а второй компоненте значение 1. Продолжать этот процесс до тех пор, пока все символы не будут закодированы.

*Пример 4.6.* Источник формирует следующие символы  $X = \{x_1, x_2, \dots, x_6\} = \{A, K, N, D, E, !\}$ . Вероятности символов задаются множеством:  $\{p_1 = 0,05, p_2 = 0,15, p_3 = 0,05, p_4 = 0,4, p_5 = 0,2, p_6 = 0,15\}$ . Построить код Хаффмена.

Решение. Алгоритм реализуется по схеме, приведенной на рис. 4.1. Упорядочение

<i>D</i>	<i>E</i>	<i>K</i>	!	<i>A</i>	<i>N</i>
0,4	0,2	0,15	0,15	0,05	0,05

Редукция

<i>D</i>	<i>E</i>	<i>K</i>	!	<i>A N</i>
0,4	0,2	0,15	0,15	0,1

Редукция

<i>D</i>	<i>E</i>	<i>K</i>	! <i>A N</i>
0,4	0,2	0,15	0,25

Упорядочение

<i>D</i>	! <i>A N</i>	<i>E</i>	<i>K</i>
0,4	0,25	0,2	0,15

Редукция

<i>D</i>	! <i>A N</i>	<i>E K</i>
0,4	0,25	0,35

Упорядочение

<i>D</i>	<i>E K</i>	! <i>A N</i>
0,4	0,35	0,25

Редукция

$D$	$EK!AN$
0,4	0,6

Упорядочение

$EK!AN$	$D$
0,6	0,4

Редукция

$DEK!AND$
1

Рис. 4.1. Последовательность шагов алгоритма Хаффмена

Далее изобразим кодовое дерево, как показано на рис. 4.2.

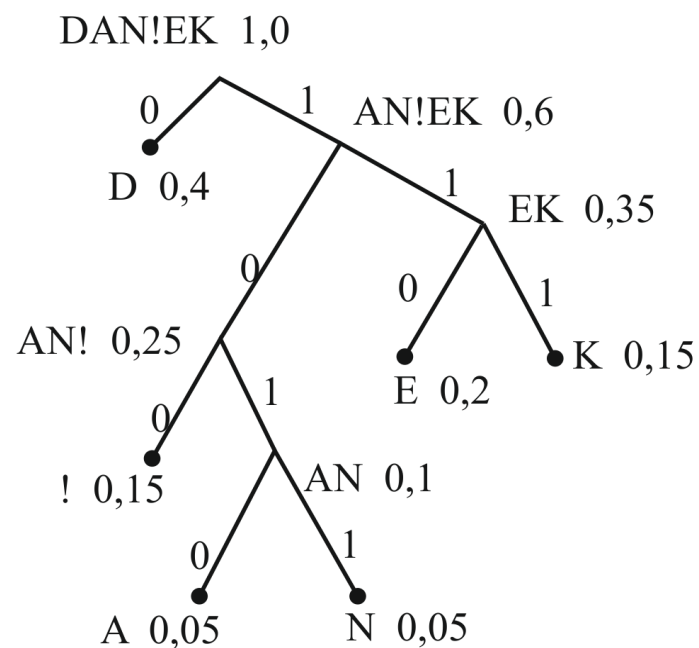


Рис. 4.2. Кодовое дерево источника  $X = \{A, K, N, D, E, !\}$

Соответственно, кодовые слова кода Хаффмена имеют вид:

$A \rightarrow 1010;$   
 $N \rightarrow 1011;$   
 $! \rightarrow 100;$   
 $E \rightarrow 110;$   
 $K \rightarrow 111;$   
 $D \rightarrow 0.$

Средняя длина полученного однозначно декодируемого кода равна

$$L_n = \sum_{i=1}^6 p_i l_i = 0,05 \cdot 4 + 0,15 \cdot 3 + 0,05 \cdot 4 + 0,4 \cdot 1 + 0,2 \cdot 3 + 0,15 \cdot 3 = 2,3.$$

Энтропия источника равна

$$H = \sum_{i=1}^6 -p_i \log_2 p_i = (-0,05 \log_2 0,05) + (-0,15 \log_2 0,15) + \\ + (-0,05 \log_2 0,05) + (0,4 \log_2 0,4) + \\ + (0,2 \log_2 0,2) + (-0,15 \log_2 0,15) = 2,25 \frac{\text{бит}}{\text{символ}}.$$

Полученные значения средней длины  $L_n$  кода и энтропии  $H$  удовлетворяют выражению (3.6)

$$H \leq L \leq H + 1, \\ 2,25 < 2,3 < 3,25.$$

#### 4.4.2. Эффективность кода

*Определение 4.1.* Эффективностью кода или фактором сжатия называется отношение энтропии к средней длине кода

$$\eta = \frac{H}{L_n}.$$

В примере 4.6 эффективность кода равна

$$\eta = \frac{2,25}{2,3} = 0,976.$$

#### 4.4.3. Коды Хаффмена блоковых источников

Пусть источник формирует одиночные символы  $X = \{a, b\} = \{0, 1\}$  с вероятностями  $\{p_1 = \frac{1}{3}, p_2 = \frac{2}{3}\}$ . Энтропия источника равна  $H = 0,918$  бит/символ (см. пример 4.3). Кодирование этого источника кодом Хаффмена приводит к двум словам с минимальной длиной  $L = 1$ .

1. Имеется блоковый источник с двукратным расширением  $X^2 = \{(00), (01), (10), (11)\} = \{c_1, c_2, c_3, c_4\}$ . Символы  $c_1, c_2, c_3, c_4$  получены расширением источника одиночных символов  $X = \{a, b\}$  с вероятностями  $\{p_1 = \frac{1}{3}, p_2 = \frac{2}{3}\}$ . Вероятности  $P(c_i)$  появления символов источника  $X^2$  определяются множеством

$$\{P(c_1) = \frac{1}{9}, P(c_2) = P(c_3) = \frac{2}{9}, P(c_4) = \frac{4}{9}\}.$$

Энтропия источника  $X^2$

$$H^2 = \sum_{i=1}^4 -P(c_i) \log_2 P(c_i) = 1,83 \text{ бит/символ (см. пример 4.4).}$$

Кодирование блокового источника  $X^2$  реализуем на основе алгоритма Хаффмена. Алгоритму соответствует последовательность шагов, рис. 4.3.

Упорядочение

$c_4$	$c_2$	$c_3$	$c_1$
4	2	2	1
$\frac{4}{9}$	$\frac{2}{9}$	$\frac{2}{9}$	$\frac{1}{9}$

Редукция

$c_4$	$c_2$	$c_1 c_3$
4	2	3
$\frac{4}{9}$	$\frac{2}{9}$	$\frac{3}{9}$

Упорядочение

$c_4$	$c_1 c_3$	$c_2$
4	3	2
$\frac{4}{9}$	$\frac{3}{9}$	$\frac{2}{9}$

Редукция

$c_4$	$c_1 c_3 c_2$
4	5
$\frac{4}{9}$	$\frac{5}{9}$

Упорядочение

$c_1 c_3 c_2$	$c_4$
5	4
$\frac{5}{9}$	$\frac{4}{9}$

Редукция

$c_1 c_3 c_2 c_4$
1

Рис. 4.3. Последовательность шагов алгоритма Хаффмена

Далее изобразим кодовое дерево кода Хаффмена, рис. 4.4.

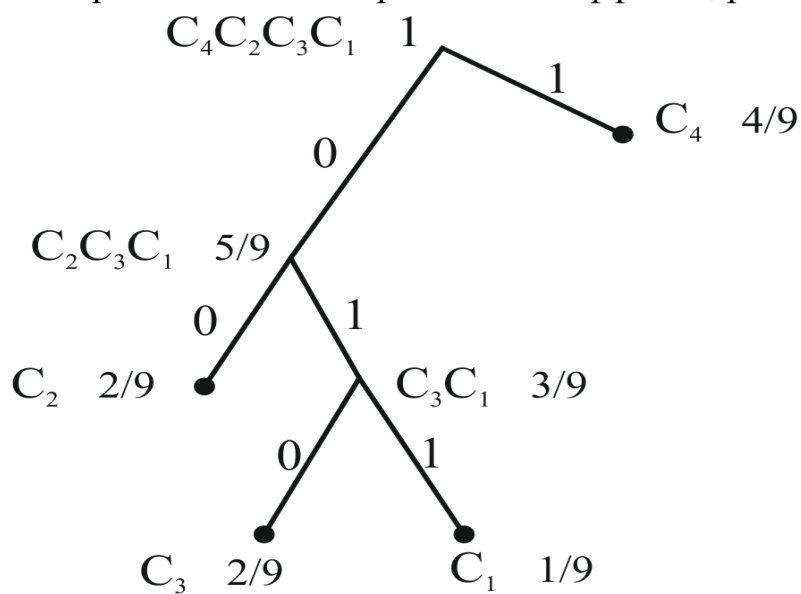


Рис. 4.4. Кодовое дерево

Кодовые слова имеют вид:

$$\begin{aligned}c_1 &\rightarrow (011); \\c_2 &\rightarrow (00); \\c_3 &\rightarrow (010); \\c_4 &\rightarrow (1).\end{aligned}$$

Средняя длина полученного двоичного кода Хаффмена равна

$$L_n = \sum_{i=1}^4 P(c_i) l_i = \frac{1}{9} \cdot 3 + \frac{2}{9} \cdot 2 + \frac{2}{9} \cdot 3 + \frac{4}{9} \cdot 1 = \frac{17}{9} \cong 1,88.$$

Средняя длина слова на один символ источника равна:

$$\frac{L_n}{n} = \frac{17}{9 \cdot 2} = \frac{17}{18} \cong 0,944.$$

Как видно, средняя длина кода источника  $X = \{a, b\}$  уменьшилась с

$$L = 1 \text{ до } \frac{L_n}{n} \cong 0,94 > H = 0,918.$$

2. Имеется блочный источник с трехкратным расширением

$$\begin{aligned}X^3 &= \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\} = \\&= \{(000), (001), (010), (011), (100), (101), (110), (111)\}.\end{aligned}$$

Символы блочного источника  $X^3$  получены расширением источника одиночных символов  $X = \{0,1\}$  с вероятностями  $\{p_1 = \frac{1}{3}, p_2 = \frac{2}{3}\}$ . Вероятности  $P(c_i)$  появления символов источника  $X^3$  определяются как

$$\begin{aligned}P(c_1) &= p_1 \cdot p_1 \cdot p_1 = \frac{1}{27}, \\P(c_2) &= p_1 \cdot p_1 \cdot p_2 = \frac{2}{27}, \\P(c_3) &= p_1 \cdot p_2 \cdot p_1 = \frac{2}{27}, \\P(c_4) &= p_1 \cdot p_2 \cdot p_2 = \frac{4}{27}, \\P(c_5) &= p_2 \cdot p_1 \cdot p_1 = \frac{2}{27}, \\P(c_6) &= p_2 \cdot p_1 \cdot p_2 = \frac{4}{27}, \\P(c_7) &= p_2 \cdot p_2 \cdot p_1 = \frac{4}{27},\end{aligned}$$

$$P(c_8) = p_2 \cdot p_2 \cdot p_2 = \frac{8}{27}.$$

Энтропия источника  $X^3$

$$H^3 = nH = 3 \cdot 0,918 = 2,754.$$

Кодирование блокового источника  $X^3$  реализуем на основе алгоритма Хаффмена. Структуре алгоритма соответствует последовательность шагов, показанная на рис. 4.6.

Исходные данные

$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$
1	2	2	4	2	4	4	8
$\frac{1}{27}$	$\frac{2}{27}$	$\frac{2}{27}$	$\frac{4}{27}$	$\frac{2}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{8}{27}$

Упорядочение

$c_8$	$c_7$	$c_6$	$c_4$	$c_5$	$c_3$	$c_2$	$c_1$
8	4	4	4	2	2	2	1
$\frac{8}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{2}{27}$	$\frac{2}{27}$	$\frac{2}{27}$	$\frac{1}{27}$

Редукция

$c_8$	$c_7$	$c_6$	$c_4$	$c_5$	$c_3$	$c_2 c_1$
8	4	4	4	2	2	3
$\frac{8}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{2}{27}$	$\frac{2}{27}$	$\frac{3}{27}$

Упорядочение

$c_8$	$c_7$	$c_6$	$c_4$	$c_2 c_1$	$c_5$	$c_3$
8	4	4	4	3	2	2
$\frac{8}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{3}{27}$	$\frac{2}{27}$	$\frac{2}{27}$

Редукция

$c_8$	$c_7$	$c_6$	$c_4$	$c_2 c_1$	$c_5 c_3$
8	4	4	4	3	4
$\frac{8}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{3}{27}$	$\frac{4}{27}$

Упорядочение

$c_8$	$c_7$	$c_6$	$c_4$	$c_5 c_3$	$c_2 c_1$
8	4	4	4	4	3
$\frac{8}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{3}{27}$

Редукция

$c_8$	$c_7$	$c_6$	$c_4$	$c_5 c_3 c_2 c_1$
8	4	4	4	7
$\frac{8}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{4}{27}$	$\frac{7}{27}$



Упорядочение

$c_8$	$c_5 c_3 c_2 c_1$	$c_7$	$c_6$	$c_4$
8	7	4	4	4
$\overline{27}$	$\overline{27}$	$\overline{27}$	$\overline{27}$	$\overline{27}$

Редукция

$c_8$	$c_6 c_4$	$c_5 c_3 c_2 c_1$	$c_7$
8	8	7	4
$\overline{27}$	$\overline{27}$	$\overline{27}$	$\overline{27}$

Редукция

$c_8$	$c_6 c_4$	$c_7 c_5 c_3 c_2 c_1$
8	8	11
$\overline{27}$	$\overline{27}$	$\overline{27}$

Упорядочение

$c_7 c_5 c_3 c_2 c_1$	$c_8$	$c_6 c_4$
11	8	8
$\overline{27}$	$\overline{27}$	$\overline{27}$

Редукция

$c_7 c_5 c_3 c_2 c_1$	$c_8 c_6 c_4$
11	16
$\overline{27}$	$\overline{27}$

Упорядочение

$c_8 c_6 c_4$	$c_7 c_5 c_3 c_2 c_1$
16	11
$\overline{27}$	$\overline{27}$

Редукция

$c_8 c_6 c_4 c_7 c_5 c_3 c_2 c_1$
1

Рис. 4.6. Последовательность шагов алгоритма Хаффмена  
Далее представим кодовое дерево, рис. 4.7 и код Хаффмена:

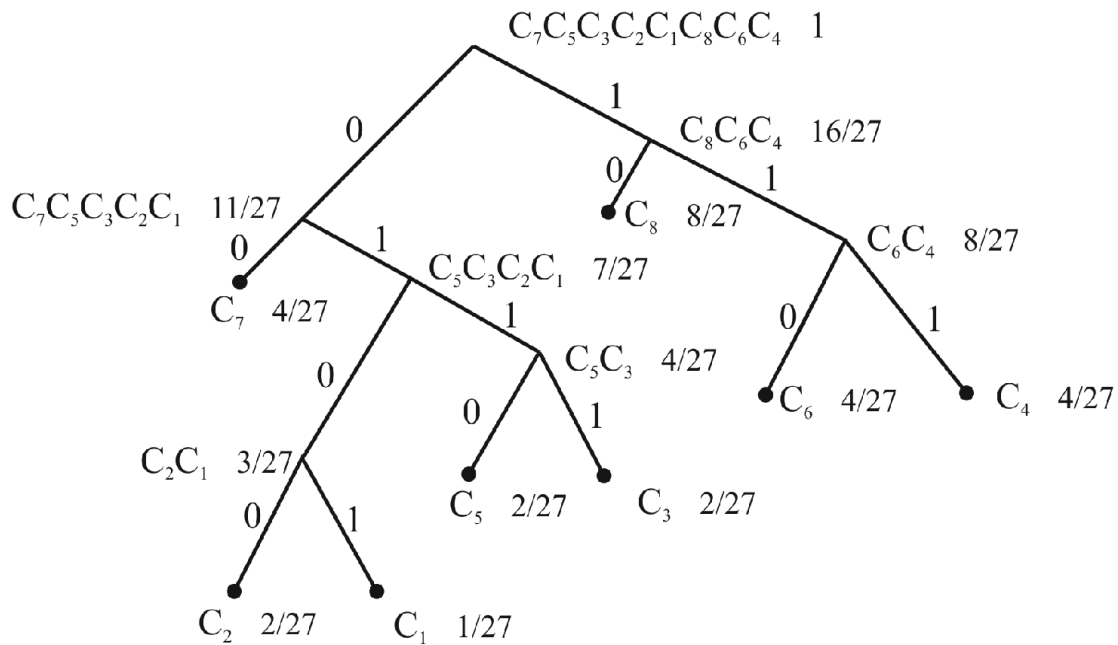


Рис. 4.7. Кодовое дерево

$c_1 \rightarrow (0101);$   
 $c_2 \rightarrow (0100);$   
 $c_3 \rightarrow (0111);$   
 $c_4 \rightarrow (111);$   
 $c_5 \rightarrow (0110);$   
 $c_6 \rightarrow (110);$   
 $c_7 \rightarrow (00);$   
 $c_8 \rightarrow (10).$

Средняя длина полученного двоичного кода Хаффмена равна

$$\begin{aligned}
 L_n = \sum_{i=1}^8 P(c_i) l_i &= \frac{1}{27} \cdot 4 + \frac{2}{27} \cdot 4 + \frac{2}{27} \cdot 4 + \frac{4}{27} \cdot 3 + \frac{2}{27} \cdot 4 + \frac{4}{27} \cdot 3 + \\
 &+ \frac{4}{27} \cdot 2 + \frac{8}{27} \cdot 2 = \frac{76}{27} \cong 2,81.
 \end{aligned}$$

Средняя длина слова на один символ источника равна:

$$\frac{L_n}{n} \cong \frac{2,81}{3} \cong 0,938.$$

Как видно, средняя длина кода источника  $X = \{0,1\}$  уменьшилась с  $L = 1$  до  $\frac{L_n}{n} \cong 0,938 > H = 0,918$ .

Результаты расчетов для блочных источников сведены в табл. 4.1. Анализ данных таблицы позволяет сформулировать следующие выводы.

1. С увеличением степени блокового источника значение средней длины (на один символ блокового источника) слова кода Хаффмена уменьшается и стремится к энтропии одиночного источника.

2. С увеличением степени блокового источника увеличивается эффективность кода или сжатие источника.

3. Недостатком алгоритма Хаффмена является требование априорного знания значений вероятностей появления символов, или оценок этих вероятностей.

Таблица. 4.1

Эффективность кодирования источников

Источник одиночных символов $X$	Блоковый источник $X^2$	Блоковый источник $X^3$
Символы $c_1 \rightarrow (0)$ $c_2 \rightarrow (1)$  Энтропия источника $H = 0,918$ бит/символ	Код Хаффмена $c_1 \rightarrow (011)$ $c_2 \rightarrow (00)$ $c_3 \rightarrow (010)$ $c_4 \rightarrow (1)$	Код Хаффмена $c_1 \rightarrow (0100)$ $c_2 \rightarrow (0101)$ $c_3 \rightarrow (0111)$ $c_4 \rightarrow (111)$ $c_5 \rightarrow (0110)$ $c_6 \rightarrow (110)$ $c_7 \rightarrow (00)$ $c_8 \rightarrow (10)$
Средняя длина кода на один символ источника $X$  1	Средняя длина кода на один символ источника $X^2$  $\cong 0,944$	Средняя длина кода на один символ источника $X^3$  $\cong 0,938$
Эффективность кода $(\eta = \frac{H}{L_n})$ $\eta = \frac{0,918}{1} = 0,918$	Эффективность кода $\eta = \frac{1,836}{1,88} \cong 0,944$	Эффективность кода $\eta = \frac{2,754}{2,81} \cong 0,980$

#### 4.4.4. Декодирование кода Хаффмена

При приеме символа кодового слова декодирование начинается с начальной точки отсчета дерева (с корня дерева). Если входному символу соответствует значение 1, следует двигаться по ветви с присвоенным значением 1. Если принимается 0, следует идти по ветви, соответствующей значению 0. При попадании на конечный узел дерева принимается решение о принятом символе. При попадании в узел, из которого выходят две ветви, следующий принятый символ (0 или 1) указывает, по какой ветви следует двигаться. Движение по дереву продолжается до достижения конечного узла. Для наглядности процесса декодирования кодового слова (1011) изобразим кодовое дерево декодера источника  $X = \{A, K, N, D, E, !\}$ , (см. пример 4.6). Очевидно, декодирование ко-

довой последовательности (1011) по дереву приводит к символу  $N$ .

*Замечание.* Передача информации посредством эффективного кодирования требует использования каналов, характеризующихся повышенной надежностью. Вероятность ошибки в таком канале должна быть сравнительно малой. Даже одиночная ошибка в потоке кодированной информации приводит к неправильному декодированию сообщения источника. Рассмотрим это замечание на примере.

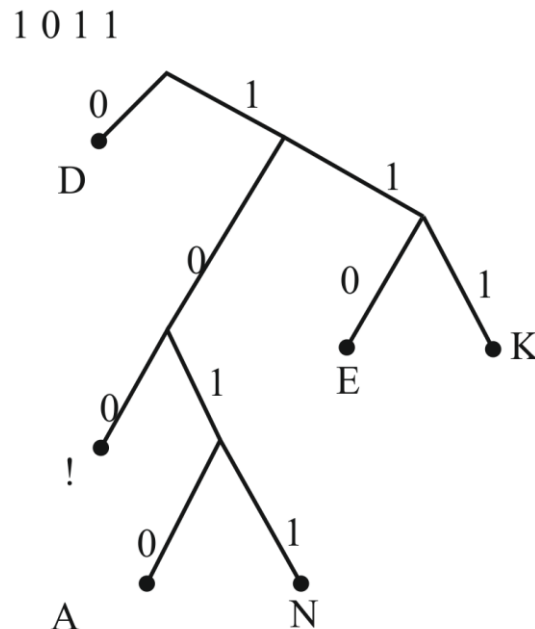


Рис. 4.8. Кодовое дерево декодера источника  $X = \{A, K, N, D, E, !\}$

*Пример 4.7.* Пусть для передачи сообщения "DANKЕ!" используется код Хаффмена, полученный в *примере 4.6*. Код состоит из следующих слов:

$A \rightarrow 1010;$   
 $N \rightarrow 1011;$   
 $! \rightarrow 100;$   
 $E \rightarrow 110;$   
 $K \rightarrow 111;$   
 $D \rightarrow 0.$

Сообщению "DANKЕ" соответствует поток двоичных символов

$$u = 010101011111110100.$$

Пусть в процессе передачи этого потока из-за воздействия помехи в канале возникла одиночная ошибка в 4-м двоичном символе. На вход декодера поступила последовательность

$$y = 010001011111110100.$$

Декодирование входной последовательности  $y = 010001011111110100$

с использованием кодового дерева, рис. 4.9 приводит к формированию ошибочного сообщения «D!DNKE!».

Легко убедиться, что структура кодового дерева отвечает следующему последовательному соответствию символов:  $0 \rightarrow D$ ,  $100 \rightarrow !$ ,  $0 \rightarrow D$ ,  $1011 \rightarrow N$ ,  $111 \rightarrow K$ ,  $110 \rightarrow E$ ,  $100 \rightarrow !$ .

Как видно, один бит, принятый с ошибкой, приводит к тому, что часть символов или все последующие символы будут декодированы неправильно.

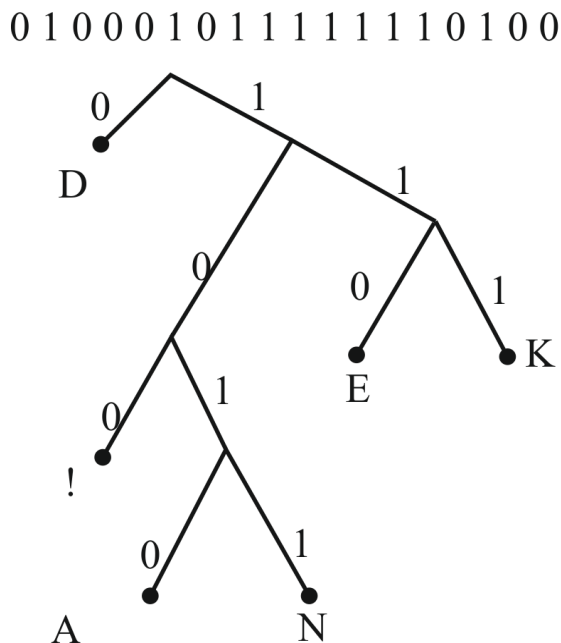


Рис. 4.9. Декодирование двоичного потока символов

Очевидными недостатками энтропийного метода кодирования являются:

- необходимо априорное знание вероятностных характеристик (статистик) символов источника;
- сжатие данных снижает избыточность и поэтому понижает надежность передачи информации. Вероятность ошибочного приема информации увеличивается.

#### 4.4.5. Адаптивный алгоритм Хаффмена

Адаптивный алгоритм эффективного кодирования реализуется с помощью двух операций.

1. Вначале выполняется кодирование источника в предположении, что все символы имеют равные вероятности появления.
2. По мере накопления знаний о статистических характеристиках источника выполняется кодирование по алгоритму Хаффмена.

#### Упражнения

4.3.1. Размерность алфавита источника  $X = \{x_1, x_2, \dots, x_6\}$  равна  $m = 6$ . Построить дерево Хаффмена для следующих значений вероятностей появления символов источника:

$$p_1 = 0,05; p_2 = 0,15; p_3 = 0,05; p_4 = 0,4; p_5 = 0,2; p_6 = 0,15.$$

4.3.2. Записать слова кода Хаффмена.

4.3.3. Декодировать последовательность  $X = 1110111011110101101$ , используя полученный код Хаффмена.

4.4.1. Источник символов  $\{A, K, N, D, E\}$  характеризуется следующими вероятностями:

$$p_A = \frac{1}{5}; p_K = \frac{1}{5}; p_N = \frac{1}{5}; p_D = \frac{1}{5}; p_E = \frac{1}{5}.$$

4.4.2. Записать слова кода Хаффмена.

4.5.1. Источник формирует символы  $X = \{x_1, x_2\}$  с вероятностями  $\{p_1 = \frac{9}{10}, p_2 = \frac{1}{10}\}$ . Имеется блочный источник с трехкратным расширением  $X^3 = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ . Построить дерево Хаффмена блочного источника.

4.5.2. Записать слова кода Хаффмена.

4.6.1. Источник формирует символы алфавита  $X = \{x_1, x_2, \dots, x_8\} = \{A, K, N, D, E, !, *, J\}$  с вероятностями их появления

$$\{p_1 = 0,5; p_2 = 0,1; p_3 = 0,1; p_4 = 0,1; p_5 = 0,06; p_6 = 0,04, \\ p_7 = 0,05; p_8 = 0,05\}.$$

Построить кодовое дерево Хаффмена.

4.6.2. Запишите код Хаффмена.

4.7.1. Для передачи звука используется 8 уровней квантования. Распределению уровней отвечает гистограмма со следующими значениями:

$$\{p_1 = 0,3; p_2 = 0,23; p_3 = 0,15; p_4 = 0,08; p_5 = p_6 = p_7 = p_8 = 0,06\}.$$

Построить кодовое дерево Хаффмена.

4.7.2. Записать слова кода Хаффмена.

4.7.3. Вычислить среднюю длину кода, кодирующего звук.

4.7.4. Вычислить энтропию источника звука.

4.8.1. Для передачи изображения используется 6 уровней квантования. Распределению уровней отвечает гистограмма со следующими значениями:

$$\{p_1 = 0,27; p_2 = 0,21; p_3 = 0,23; p_4 = 0,15; \\ p_5 = p_6 = 0,07\}.$$

Построить кодовое дерево Хаффмена.

4.8.2. Записать слова кода Хаффмена.

4.8.3. Вычислить среднюю длину кода, кодирующего изображение.

4.8.4. Вычислить энтропию источника изображения.

#### **4.5. Универсальный алгоритм сжатия**

Реализация этого алгоритма не требует предварительных знаний статистики символов источника и, по сути, является адаптивным. Универсальный алгоритм сжатия информации основан на идее использования словаря последо-

вательностей символов, слов, фраз и пр., или по-другому – образцов, встречающихся в несжатых данных. Образцам последовательных символов, встречающихся в текстах, изображениях, данных ставятся в соответствие кодовые слова. При этом каждое кодовое слово представляется определенным индексом. В процессе сжатия последовательности в несжатых данных заменяются кодовыми словами, которые имеются в словаре. Чем больше объем словаря из этих последовательных символов (образцов), и чем чаще они встречаются в несжатых данных, тем больше выигрыш в сжатии. Универсальный метод сжатия эффективен при архивации текстовой информации при обработке изображений, звуков и др.

#### **4.5.1. Алгоритм эффективного кодирования Лемпеля – Зива**

Метод сжатия на основе словаря был разработан А. Лемпелем (Abraham Lempel) Я. Зивом (Jacob Ziv) в 1977 году и известен как LZ77. Метод LZ77 является основой алгоритмов сжатия ZIP, ARJ, gzip и др. применяемых в компьютерах, используется в растровом файловом формате сжатия изображений PNG (Portable Network Graphic).

*Замечание.* В формате PNG используется разновидность кодирования LZ77, включающая кодирование Хаффмена.

Многие реальные источники информационных символов характеризуются тем, что не всегда символы удовлетворяют свойству независимости. Например, в любом языке вероятность появления той или иной буквы зависит от предыдущей буквы. В этом случае говорят о межсимвольной зависимости, коррелированности. Кроме того, практически в любом тексте слова, фразы повторяются. Можно сказать, что текст состоит из образцов, которые образуют некоторый словарь. Кодирование текста можно свести к некоторому алгоритму выбора образцов из словаря.

Процесс кодирования методом Лемпеля – Зива начинается сразу после поступления выходных символов источника на вход кодера. Кодирование информации производится по следующему алгоритму.

1. Передающая сторона записывает в специальный буфер поиска то, что было уже отправлено (символ, последовательность символов, слово, фразу и пр.). Принимающая сторона также записывает то, что было уже получено для осуществления декодирования.

2. При подготовке следующего фрагмента текста передающая сторона находит в ранее переданном фрагменте образцы.

3. Далее идет процесс передачи не самих образцов, а только информации об этих образцах в виде ссылок.

4. Ссылка записывается в форме трех указателей  $(x, y, z)$ :

–  $x$  указывает относительный адрес образца в буфере поиска. Адрес определяется местом (позицией) записи образца в буфере поиска. Значение позиции определяется числом позиций, которые надо пройти в обратном направлении от текущего символа до образца, если в обратном направлении в буфере, где начинается образец;

–  $y$  обозначает длину образца – число совпадающих символов образца и символов несжатых данных;

–  $z$  обозначает следующую букву в буфере, которая отличается от продолжения фразы в словаре образцов. Например, ссылке  $(7, 4, A)$  соответствует новый текст, состоящий из 4 букв образца, который начинается с 7-й буквы в обратном направлении буфера, и что в новом тексте следующая за образцом идет буква A.

Таким образом, кодирование осуществляется посредством использования ссылок  $(x, y, z)$ . Закодированная информация представляется последовательностью этих ссылок.

*Пример 4.8.* Необходимо передать сообщение: ДЕКОДИРОВАНИЕ\_КОДА с помощью алгоритма кодирования LZ77.

1. Процесс передачи сообщения начинается с кодирования первой буквы сообщения.

Сообщение

Д.

Кодовое слово выглядит как  $Д \rightarrow (0, 0, Д)$ . Символ Д еще не содержится в буфере поиска (словаре образцов), поэтому:

адрес  $x \rightarrow 0$ ;

длина образца  $y \rightarrow 0$ ;

следующая буква  $z \rightarrow Д$ .

*Замечание.* Если символ не содержится в словаре образцов, он кодируется словом вида

$(0, 0, \text{символ})$ .

Такое слово называется «нулевой фразой». При декодировании оно распознается по двум нулям.

2. Буфер поиска (словарь образцов) еще пуст.

Буфер поиска имеет вид

–.

3. Далее идет кодирование и передача следующей буквы Е. Формируется вновь слово «нулевая фраза»

$(0, 0, E)$ .

4. В буфере поиска уже записана буква Д.

Буфер поиска имеет вид

Д.

5. Далее идет кодирование словами «нулевая фраза» других символов:



К, О.

6. Далее передается ссылка (4, 1, И), которая указывает, что уже передавалась буква Д, а также показывает следующую за ней букву И. Символ И (указатель z в слове (4, 1, И)) – это следующая буква кодируемого сообщения. Фактически эта ссылка (кодовое слово) соответствует передаче двух букв. Число 4 кодового слова (4, 1, И) соответствует позиции в обратном направлении от текущего передаваемого символа «Д» до записанного уже ранее в буфер поиска этого символа фразы ДЕКО. Последующие шаги алгоритма иллюстрируются в табл. 4.2.

Таблица. 4.2.

Алгоритм кодирования LZ7

N	Буфер поиска	Буфер для предварительной записи сообщения	Кодовое слово (x y z)
1	–	ДЕКОДИРОВАНИЕ_КОДА	(0, 0, Д)
2	Д	ЕКОДИРОВАНИЕ_КОДА	(0, 0, Е)
3	ДЕ	КОДИРОВАНИЕ_КОДА	(0, 0, К)
4	ДЕК	ОДИРОВАНИЕ_КОДА	(0, 0, О)
5	ДЕКО	ДИРОВАНИЕ_КОДА	(4, 1, И)

Последующий процесс кодирования показан в продолжении таблицы.

Продолжение таблицы 4.2.

6	ДЕКОДИ	РОВАНИЕ_КОДА	(0, 0, Р)
7	ДЕКОДИР	ОВАНИЕ_КОДА	(4, 1, В)
8	ДЕКОДИРОВ	АНИЕ_КОДА	(0, 0, А)
9	ДЕКОДИРОВА	НИЕ_КОДА	(0, 0, Н)
10	ДЕКОДИРОВАН	ИЕ_КОДА	(6, 1, Е)
11	ДЕКОДИРОВАНИЕ	–	(0, 0, _)
12	ДЕКОДИРОВАНИЕ_	КОДА	(12, 3, А)
	ДЕКОДИРОВАНИЕ_КОДА		

Таким образом, передаваемому сообщению соответствует последовательность ссылок (слов):

{(0, 0, Д) (0, 0, Е) (0, 0, К) (0, 0, О) (4, 1, И) (0, 0, Р) (4, 1, В) (0, 0, А) (0, 0, Н) (6, 1, Е) (0, 0, \_) (12, 3, А)}.

Метод кодирования LZ приводит к сжатию тогда, когда затраты на кодирование оказываются в среднем меньше по сравнению с кодированием кодом ASCII.

Сравним затраты на кодирование слова «КОДА». Фраза, состоящая из четырех букв, закодированная ASCII-кодом записывается  $N = 32$  битами. Кодовое слово (12, 3, А) → «КОДА» требует только  $M = 24$  бит. Эффективность сжатия

$$K = \frac{32 - 24}{32} = 0,25,$$

что соответствует 25%. Для длинных текстов кодирование LZ методом позволяет получить эффективность сжатия в пределах  $K = 50 - 60\%$ .

Для длинных текстов LZ77 кодирование практически полностью устраняет избыточность. В этом случае средняя длина кодового слова на один символ источника (текста) стремится к энтропии текста.

#### 4.5.2. Декодирование LZ-кода

LZ –декодер осуществляет декодирование каждого кодового слова по идентичному словарю буфера поиска, который создается на приемной стороне. При поступлении на вход декодера первых 5-и кодовых слов (см. *пример 4.8*) получаем символы передаваемого сообщения, табл. 4.3

Таблица 4.3

Декодирование LZ-кода			
N	Кодовое слово (x y z)	Буфер поиска	Сообщение
1	(0, 0, Д)	–	Д
2	(0, 0, Е)	Д	Е
3	(0, 0, К)	ДЕ	К
4	(0, 0, О)	ДЕК	О
5	(4, 1, И)	ДЕКО	ДИ

В слове (4, 1, И) число 4 – это указатель  $x$  позиции в обратном направлении от текущего принимаемого символа до записанного уже ранее в буфер поиска образца. Символ И (указатель  $z$  в слове (4, 1, И)) – это следующая буква декодируемого сообщения. Последующий процесс декодирования показан в продолжении таблицы.

Продолжение таблицы 4.3

6	(0, 0, Р)	ДЕКОДИ	Р
7	(4, 1, В)	ДЕКОДИР	ОВ
8	(0, 0, А)	ДЕКОДИРОВ	А
9	(0, 0, Н)	ДЕКОДИРОВА	Н
10	(6, 1, Е)	ДЕКОДИРОВАН	ИЕ
11	(0, 0, _)	ДЕКОДИРОВАНИЕ	_
12	(12, 3, А)	ДЕКОДИРОВАНИЕ_	КОДА
		ДЕКОДИРОВАНИЕ_КОДА	

Недостатком алгоритма LZ77 является конечный размер буферов памяти. Типовой размер памяти буфера поиска фраз составляет величину  $W_p = 2^{12} = 4096$ . Размер памяти буфера для записи кодируемой информации  $W_i = 2^4$ . Если образец повторяется, но предыдущий пример его является более удаленным в прошлом, чем длина буфера поиска, сделать ссылку становится невоз-

можным. Кроме того, большие размеры адреса  $x$  и длины  $y$  образца могут потребовать относительно большого числа бит для их записи при формировании кодового слова (ссылки).

#### 4.5.3. Алгоритм кодирования Лемпеля – Зива – Уэлча

Модификацией алгоритма эффективного кодирования LZ77 является алгоритм LZ78, также разработанный Зивом и Лемпелем в 1978 году. В алгоритме LZ78 фактически буферы памяти не имеют конечный размер и изменена структура кодового слова. Модификацией алгоритма LZ78 является метод LZW предложенный Т. Уэлчем (Terry Welsh) в 1984 году. Метод LZW используется в таких растровых файловых форматах сжатия изображений как:

- GIF (Graphic Interchange Format – Формат обмена графическими данными), находит применение в сети Интернет;

- TIFF (Tagged Image File Format – Формат представления графической информации), находит применение при подготовке печатных документов.

Кодирование по алгоритму LZW применяется в методах сжатия JPEG-LS, PNG, в векторном файловом формате PDF (Portable Document Format).

Алгоритм кодирования LZW, как и LZ77, основывается на свойстве межсимвольной зависимости символов данных, повторяемости образцов. В процессе кодирования – декодирования создается словарь образцов. Как и в LZ77, образцы соответствуют символам, словам, фразам, предложениям и пр. Процесс сжатия осуществляется посредством использования ссылок, представляемых в виде индексов. На приемной стороне также создается словарь, который используется для декодирования кодированной информации синхронно с кодером.

Отличие метода кодирования LZW от LZ77 состоит в следующем.

1. Процесс кодирования начинается с загрузки в словарь образцов (буфер поиска) некоторого множества базовых символов алфавита, слов, фраз.

2. Образцы индексируются, например, десятичным номером.

3. При передаче символов сообщения, если в словаре находится нужный образец, отправляется только его индекс.

4. Процесс формирования образцов носит динамический характер.

*Пример 4.9.* Необходимо передать сообщение ДЕКОДЕР\_КОДА с помощью алгоритма кодирования LZW.

1. Множество базовых символов алфавита передаваемого сообщения состоящее из 7-и символов

$$\{\text{Д, Е, К, О, Р, \_, А}\}$$

формирует начальный словарь.

2. Символам начального словаря соответствуют десятичные индексы:

$$\text{Д} \rightarrow 1, \text{Е} \rightarrow 2, \text{К} \rightarrow 3, \text{О} \rightarrow 4, \text{Р} \rightarrow 5, \_ \rightarrow 6, \text{А} \rightarrow 7.$$

3. Первая буква сообщения передается в виде ссылки (кодového слова)  
(1).

4. Так как последовательности символов ДЕ в базовом словаре нет, в словарь записывается новый образец в виде числа (8), т. е. ДЕ → (8).

5. Далее передается символ Е в виде кодového слова  
(2),

и последовательность символов ЕК записывается в словарь под очередным индексом (9).

6. Далее продолжаются однобуквенные передачи символов К, О, и запись в словарь последовательностей КО, ОД.

7. Затем передается индексом (8) сразу два символа ДЕ, так как в словаре уже имеется этот образец.

По мере расширения словаря передаются все более длительные последовательности символов сообщения. Длина кодového слова также увеличивается. Процесс кодирования иллюстрируется данными табл. 4.4 и табл. 4.5

Таблица 4.4  
Словарь

Индекс	Словарь образцов
1	Д
2	Е
3	К
4	О
5	Р
6	—
7	А
8	ДЕ
9	ЕК
10	КО
11	ОД
12	ДЕР
13	Р_
14	_К
15	КОД
16	ДА
17	А...

Таблица 4.5

Кодирование методом LZW

Сообщение	Д	Е	К	О	ДЕ	Р	—	КО	Д	А
Кодовые слова	1	2	3	4	8	5	6	10	1	7

Метод кодирования LZW приводит к сжатию тогда, когда затраты на кодирование оказываются в среднем меньше по сравнению с кодированием кодом ASCII. Затраты на кодирование сообщения ДЕКОДЕР\_КОДА ASCII-кодом составляют

$$N = 12 \times 8 = 96 \text{ бит.}$$

Кодирование LZW-кодом требует использования

$$M = 10 \times 8 = 80 \text{ бит.}$$

Эффективность сжатия

$$K = \frac{96-80}{96} \cong 0,16,$$

что соответствует 16%.

Как видно, даже на малой длине сообщения достигается сжатие. В типовом случае LZW кодирование обеспечивает сжатие текстового файла исполняемого кода примерно наполовину исходного размера.

#### 4.5.4. Декодирование LZW-кода

LZW декодер осуществляет декодирование каждого кодового слова по идентичному словарю, который создается на приемной стороне. При поступлении на вход декодера последовательности значений индексов (см. *пример 4.9*)

1, 2, 3, 4, 8, 5, 6, 10, 1, 7,

для каждого индекса выбираются соответствующие последовательности символов из словаря образцов, табл. 4.6. В результате получаем принятое сообщение.

Таблица 4.6

Декодирование LZW-кода

Кодовые слова	1	2	3	4	8	5	6	10	1	7
Сообщение	Д	Е	К	О	ДЕ	Р	_	КО	Д	А

#### Упражнения

4.9.1. Используйте алгоритм кодирования LZ77 для сжатия сообщения ТЕОРИЯ ИНФОРМАЦИИ – ТЕОРИЯ КОДИРОВАНИЯ.

4.9.2. Оцените эффективность сжатия.

4.10.1. Используйте алгоритм кодирования LZW для сжатия сообщения ТЕОРИЯ ИНФОРМАЦИИ – ТЕОРИЯ КОДИРОВАНИЯ.

4.10.2. Оцените эффективность сжатия.

## **5. КАНАЛЫ БЕЗ ПАМЯТИ И ПЕРЕДАЧА ИНФОРМАЦИИ**

### **5.1. Двоичный симметричный канал без памяти**

Двоичный симметричный канал (ДСК) является математической двоичной моделью взаимодействия двух дискретных источников без памяти. Напомним, приемник также можно считать источником информации. Двоичный сим-

метричный канал является также моделью передачи информации по каналу с аддитивным белым гауссовским шумом. Графическое представление такого канала показано на рис. 5.1.

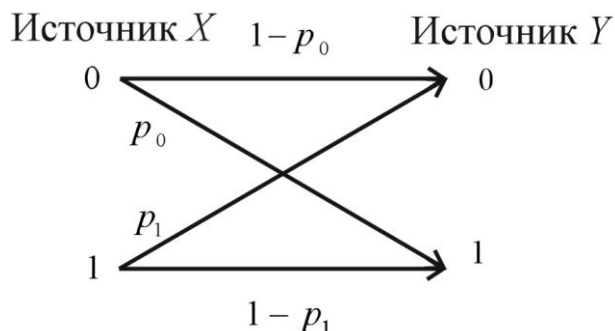


Рис. 5.1. Модель двоичного симметричного канала передачи информации

Входными символами канала являются символы  $x_1 = 0$  и  $x_2 = 1$ . Выходными символами источника являются символы  $y_1 = 0$  и  $y_2 = 1$ . Если при передаче информации выходные символы канала совпадают с входными ( $x_1 = y_1$  и  $x_2 = y_2$ ), пространство событий отражается подмножеством

$$\{00, 11\}.$$

Из-за воздействия шума возможен неправильный прием символов, когда выход канала не совпадает с входом ( $y_1 \neq x_1$  и  $y_2 \neq x_2$ ). В этом случае пространство таких событий отражается подмножеством

$$\{01, 10\}.$$

Передача информации по модели, показанной на рис. 5.1, может отображаться в виде пространства событий определяемым множеством

$$\{00, 01, 10, 11\}.$$

Это пространство событий описывается вероятностями перехода символов входа  $X$  в символы выхода  $Y$ .

Достоверной передаче символов  $\{00, 11\}$  соответствуют вероятности  $p(0|0)$  и  $p(1|1)$ , т. е.

$$\{00 \rightarrow p(0|0), \{11 \rightarrow p(1|1)\},$$

Передача с ошибками характеризуется условными вероятностями  $p(1|0)$  и  $p(0|1)$  искаженной передачи символов, т. е.

$$\{10 \rightarrow p(1|0), \{01 \rightarrow p(0|1)\},$$

Если вероятности искажений  $p(0|1) \cong p(1|0) = p$ , то канал называется двоичным симметричным.

Пусть  $p$  вероятность перехода (вероятность ошибки). Тогда достоверная передача информации – это событие, происходящее с вероятностью  $(1 - p)$ .

Для описания ДСК воспользуемся выражением (2.14). Оно определяет вероятность выхода  $p(y_i)$  ДСК через распределение вероятностей источника  $X$  (априорные вероятности  $p(x_j)$ ) и вероятностей перехода  $p(y_i|x_j)$  по формуле

$$p(y_i) = \sum_{j=1}^2 p(y_i|x_j)p(x_j).$$

Тогда

$$p(y_1) = p(y_1|x_1)p(x_1) + p(y_1|x_2)p(x_2),$$

$$p(y_2) = p(y_2|x_1)p(x_1) + p(y_2|x_2)p(x_2).$$

Легко увидеть, что этим выражениям соответствует матричная форма

$$\begin{bmatrix} p(y_1) \\ p(y_2) \end{bmatrix} = \begin{bmatrix} p(y_1|x_1) & p(y_1|x_2) \\ p(y_2|x_1) & p(y_2|x_2) \end{bmatrix} \begin{bmatrix} p(x_1) \\ p(x_2) \end{bmatrix}. \quad (5.1)$$

Двоичный симметричный канал определяется четырьмя вероятностями перехода для значений  $0 \leq p \leq 0,5$ .

*Замечание.* Матрица

$$\mathbf{P} = \begin{bmatrix} p(y_1|x_1) & p(y_1|x_2) \\ p(y_2|x_1) & p(y_2|x_2) \end{bmatrix}$$

называется матрицей переходных вероятностей канала (матрицей канала).

Так как для ДСК справедливо  $p(0|1) \cong p(1|0) = p$ , то

$$p(y_1|x_2) = p(y_2|x_1) = p, \quad p(y_1|x_1) = p(y_2|x_2) = (1 - p). \quad (5.2)$$

Подставляя выражения (5.2) в  $\mathbf{P}$ , распределение вероятностей выходных символов канала записывается в виде

$$\begin{bmatrix} p(y_1) \\ p(y_2) \end{bmatrix} = \begin{bmatrix} 1 - p & p \\ p & 1 - p \end{bmatrix} \begin{bmatrix} p(x_1) \\ p(x_2) \end{bmatrix}. \quad (5.3)$$

Вероятность того, что на выходе канала будет символ  $y_1$  равна

$$p(y_1) = (1 - p)p(x_1) + pp(x_2).$$

Вероятность того, что на выходе канала будет символ  $y_2$  равна

$$p(y_2) = pp(x_1) + (1 - p)p(x_2).$$

*Пример 5.1.* Входными символами ДСК являются символы  $x_1 = 0$  и  $x_2 = 1$ . Пусть канал характеризуется вероятностью ошибки  $p = 0,05$ . Источник символов описывается вероятностями  $p(x_1) = 0,9$  и  $p(x_2) = 0,1$ . Матрица ДСК имеет вид



$$P = \begin{bmatrix} 0,95 & 0,05 \\ 0,05 & 0,95 \end{bmatrix}.$$

Вероятность того, что на выходе канала будет символ  $y_1$  равна

$$p(y_1) = (1 - p)p(x_1) + pp(x_2) = 0,95 \cdot 0,9 + 0,05 \cdot 0,1 = 0,86.$$

Вероятность того, что на выходе канала будет символ  $y_2$ , равна

$$p(y_2) = pp(x_1) + (1 - p)p(x_2) = 0,05 \cdot 0,9 + 0,95 \cdot 0,1 = 0,14.$$

2. Найти вероятность  $p(x|y)$  достоверного получения символов источника  $X$  приемной стороной.

Решение. Воспользуемся формулой (2.7) из теоремы Байеса

$$p(x_i|y_j) = \frac{p(y_j|x_i)p(x_i)}{p(y_j)}.$$

Вероятность получения символов источника  $X$  по ДСК вычисляется из выражения

$$p(x_i|y_i) = \frac{p(y_i|x_i)p(x_i)}{p(y_i)}.$$

С учетом выполнения условий

$$p(y_1|x_1) = p(y_2|x_2) = (1 - p)$$

для ДСК, получаем:

$$p(x_1|y_1) = \frac{(1 - p)p(x_1)}{p(y_1)} = \frac{0,95 \cdot 0,9}{0,86} = 0,9941;$$

$$p(x_2|y_2) = \frac{(1 - p)p(x_2)}{p(y_2)} = \frac{0,95 \cdot 0,1}{0,14} = 0,678.$$

Как видно из примера, в условиях воздействия шума в канале большее значение вероятности правильного приема символа получается, когда он чаще передается. На основе этого фундаментального принципа введения избыточности (повторяемости) строятся все оптимальные алгоритмы обработки сигналов в шумах, помехоустойчивое кодирование и пр.

## 5.2. Комбинирование источников

Не теряя общего представления о комбинировании  $l$  источников, ограничимся рассмотрением двух источников,  $l = 2$ .

*Определение 5.1.* Под комбинированием источников  $X = \{x_1, x_2, \dots, x_m\}$  и  $Y = \{y_1, y_2, \dots, y_n\}$  понимают источник  $(X, Y)$ , который характеризуется множе-

ством  $\{x, y\}$ . Источник  $(X, Y)$  включает в себя пары совместных событий  $(x, y)$  из  $X$  и  $Y$ .

Вероятности  $p(x, y)$  пар совместных событий удовлетворяют выражению нормировки

$$\sum_{i=1}^m \sum_{j=1}^u p(x_i, y_j) = 1.$$

Например, если  $X = \{x_1, x_2\}$ ,  $Y = \{y_1, y_2\}$ , комбинированный источник характеризуется совместными событиями

$$(X, Y) = \{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)\}$$

и вероятностями  $p(x_i, y_j)$  совместных событий

$$\begin{aligned} \sum_{i=1}^2 \sum_{j=1}^2 p(x_i, y_j) &= \sum_{i=1}^2 [p(x_i, y_1) + p(x_i, y_2)] = \\ &= p(x_1, y_1) + p(x_1, y_2) + p(x_2, y_1) + p(x_2, y_2) = 1. \end{aligned}$$

### 5.3. Совместная энтропия

**Теорема 5.1.** Если источники  $X$  и  $Y$  независимы, то энтропия комбинированного источника  $(X, Y)$  равна сумме энтропий отдельных источников

$$H(X, Y) = H(X) + H(Y).$$

Напомним (см. *определение 2.1*), источники  $X = \{x_1, x_2, \dots, x_m\}$  и  $Y = \{y_1, y_2, \dots, y_u\}$  независимы, если совместная вероятность каждой пары  $(x, y)$  определяется как  $p_{x,y} = p_x p_y$ .

Доказательство теоремы. Пусть  $X = \{x_1, x_2\}$ ,  $Y = \{y_1, y_2\}$ . Тогда событие  $(x, y) \in \{X, Y\}$  имеет вероятность  $p_x p_y$ . Из определения энтропии можно записать для комбинированного источника  $(X, Y)$  выражение совместной энтропии:

$$\begin{aligned} H(X, Y) &= - \sum_{i=1}^2 \sum_{j=1}^2 p_{x_i} p_{y_j} \log_2 p_{x_i} p_{y_j} = \\ &= - \sum_{i=1}^2 \sum_{j=1}^2 p_{x_i} p_{y_j} (\log_2 p_{x_i} + \log_2 p_{y_j}) = \\ &= - \sum_{i=1}^2 p_{x_i} \sum_{j=1}^2 p_{y_j} (\log_2 p_{x_i} + \log_2 p_{y_j}) = \end{aligned}$$

$$= - \sum_{i=1}^2 p_{x_i} \log_2 p_{x_i} \sum_{j=1}^2 p_{y_j} - \sum_{i=1}^2 p_{x_i} \sum_{j=1}^2 p_{y_j} \log_2 p_{y_j}.$$

Все вероятности символов одиночных источников в сумме должны давать значение 1, поэтому  $\sum_{i=1}^2 p_{x_i} = \sum_{j=1}^2 p_{y_j} = 1$ . В результате получаем

$$H(X, Y) = - \sum_{i=1}^2 p_{x_i} \log_2 p_{x_i} - \sum_{j=1}^2 p_{y_j} \log_2 p_{y_j} = H(X) + H(Y). \quad (5.4)$$

*Утверждение 5.1.* Если источники  $X$  и  $Y$  не являются независимыми, всегда выполняется

$$H(X, Y) < H(X) + H(Y).$$

#### 5.4. Условная энтропия

Пусть имеются источники  $X$  (символ  $x_i \in X$ ) и  $Y$  (символ  $y_i \in Y$ ).

*Определение 5.2.* Условная энтропия  $H(Y|X)$  – это среднее количество информации на один символ источника  $Y$ , при условии наличия символа источника  $X$ .

Для конкретного значения  $y_i \in Y$  в соответствии с определением энтропии (см. (2.18),  $H = \sum_{i=1}^m -p_i \log p_i$ ) условная энтропия записывается в виде

$$H(Y|x_i) = - \sum_{y_j \in Y} p(y_j|x_i) \log_2 p(y_j|x_i). \quad (5.5)$$

*Замечание.* Выражение (5.5) называется частной условной энтропией источника  $Y$  для символа  $x_i$  источника  $X$ .

Средняя величина частных значений  $H(Y|x_i)$  по всем  $x_i \in X$  в соответствии с их вероятностями  $p(x_i)$  определяется по формуле

$$H(Y|X) = \sum_{x_i \in X} H(Y|x_i) p(x_i). \quad (5.6)$$

Подставив выражение (5.5) в (5.6), получаем

$$H(Y|X) = - \sum_{x_i \in X} p(x_i) (\sum_{y_j \in Y} p(y_j|x_i) \log_2 p(y_j|x_i)). \quad (5.7)$$

Формулу (5.7) можно записать в виде

$$H(Y|X) = - \sum_{x_i \in X} (\sum_{y_j \in Y} p(y_j|x_i) p(x_i) \log_2 p(y_j|x_i)). \quad (5.8)$$

Заменяя в (5.8) выражение  $p(y_j|x_i)p(x_i)$  на совместную вероятность  $p(x_i, y_j)$  появления двух событий  $y_j$  и  $x_i$ , (см. (2.10),  $p(x_i, y_j) = p(y_j|x_i)p(x_i)$ ), получим

$$H(Y|X) = - \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log_2 p(y_j|x_i). \quad (5.9)$$

По аналогии можно дать определение и записать выражение для условной энтропии  $H(X|Y)$ .

*Определение 5.3.* Условная энтропия  $H(X|Y)$  – это среднее количество информации на один символ источника  $X$ , при условии наличия символа источника  $Y$ .

$$H(X|Y) = - \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log_2 p(x_i|y_j). \quad (5.10)$$

*Пример 5.2.* Напомним, двоичный симметричный канал представляется в виде модели состоящей из двух источников. Вычислим условную энтропию ДСК. Входные символы канала (источник  $X$ ):  $x_1 = 0$  и  $x_2 = 1$ . Выходные символы канала (источник  $Y$ ):  $y_1 = 0$  и  $y_2 = 1$ . Вероятность ошибки в канале  $p = \frac{1}{8}$ .

Решение. Применяя формулу (5.9) условной энтропии, запишем выражение

$$\begin{aligned} H(Y|X) &= - \sum_{i=1}^2 \sum_{j=1}^2 p(x_i, y_j) \log_2 p(y_j|x_i) = \\ &= - \sum_{i=1}^2 [p(x_i, y_1) \log_2 p(y_1|x_i) + p(x_i, y_2) \log_2 p(y_2|x_i)] = \\ &\quad - [p(x_1, y_1) \log_2 p(y_1|x_1) + p(x_1, y_2) \log_2 p(y_2|x_1)] - \\ &\quad - [p(x_2, y_1) \log_2 p(y_1|x_2) + p(x_2, y_2) \log_2 p(y_2|x_2)] = \\ &= H(Y|x_1) + H(Y|x_2), \end{aligned}$$

$$\text{где } H(Y|x_1) = -[p(x_1, y_1) \log_2 p(y_1|x_1) + p(x_1, y_2) \log_2 p(y_2|x_1)] \text{ и} \quad (5.11)$$

$$H(Y|x_2) = -[p(x_2, y_1) \log_2 p(y_1|x_2) + p(x_2, y_2) \log_2 p(y_2|x_2)] \quad (5.12)$$

соответственно частная условная энтропия источника  $Y$  для символов  $x_1$  и  $x_2$  источника  $X$ .

Исходя из свойств переходных вероятностей ДСК:

$$p(x_i, y_i) = p(x_j|y_j) = 1 - p;$$

$$p(x_i|y_j) = p(x_j|y_i) = p,$$

выражение (5.11) можно записать в виде

$$\begin{aligned} H(Y|x_1) &= -[p(x_1|y_1) \log_2 p(y_1|x_1) + p(x_1|y_2) \log_2 p(y_2|x_1)] = \\ &= H(Y|0) = -[(1-p) \log_2(1-p) + p \log_2(p)] = H(P). \end{aligned} \quad (5.13)$$

Аналогично получаем выражение частной условной энтропии  $H(Y|x_2)$ :

$$\begin{aligned} H(Y|x_2) &= -[p(x_2|y_1) \log_2 p(y_1|x_2) + p(x_2|y_2) \log_2 p(y_2|x_2)] = \\ &= H(Y|1) = -[(1-p) \log_2 p(1-p) + p \log_2 p] = H(P). \end{aligned} \quad (5.14)$$

*Вывод.* Условная энтропия ДСК определяется функцией Шеннона  $H(P)$  (2.17).

Напомним, функция Шеннона характеризует дискретный источник двух независимых событий  $X = \{0, 1\}$  с вероятностями  $P$  и  $(1 - P)$ .

Применяя (5.13) и (5.14), получаем

$$\begin{aligned} H(Y|0) = H(Y|1) = H(P) &= -p \log_2 p - (1-p) \log_2(1-p) = \\ &= -\frac{1}{8} \log_2 \frac{1}{8} - \frac{7}{8} \log_2 \frac{4}{8} = 0,54 \text{ бит/символ.} \end{aligned}$$

*Выводы*

1. Из-за действия шумов в канале количество принимаемой информации уменьшилось почти в два раза.
2. Условная энтропия ДСК не зависит от вероятностей символов входа канала.

*Замечание.* В рассматриваемом примере источники  $X$  и  $Y$  связаны между собой каналом с шумами. В этом случае входные символы источника  $X$  позволяют делать некоторое предположение о символах источника  $Y$ . Эта заранее получаемая информация снижает степень неопределенности источника  $Y$ , и, следовательно, уменьшение среднего ожидаемого количества полученной информации – энтропии.

## 5.5. Соотношение между совместной и условной энтропией

Вновь обратимся к определению совместной энтропии комбинированного источника как математическое ожидание информации всех пар событий этих источников. Не теряя общего представления о комбинированном источнике, ограничимся рассмотрением двух источников  $X = \{x_1, x_2, \dots, x_m\}$  и  $Y = \{y_1, y_2, \dots, y_m\}$ . Совместное событие  $(x, y) \in (X, Y)$  характеризуется совместной вероятностью  $p_x, p_y$ . Из определения понятия энтропии можно записать для

комбинированного источника  $(X, Y)$ ,  $X = \{x_1, x_2\}$ ,  $Y = \{y_1, y_2\}$  следующее выражение:

$$H(X, Y) = - \sum_{i=1}^2 \sum_{j=1}^2 p(x_i, y_j) \log_2 p(x_i, y_j). \quad (5.15)$$

Заменяя в (5.15) совместную вероятность  $p(x_i, y_j)$  на выражение  $p(x_i)p(y_j|x_i) = p(y_j)p(x_i|y_j)$ , (см. (2.10)), получим

$$H(X, Y) = - \sum_{i=1}^2 \sum_{j=1}^2 p(x_i)p(y_j|x_i) \log_2(p(x_i)p(y_j|x_i)).$$

Раскрывая логарифм произведения, преобразуем последнее выражение к виду

$$\begin{aligned} H(X, Y) &= - \sum_{i=1}^2 \sum_{j=1}^2 p(x_i)p(y_j|x_i)(\log_2 p(x_i) + \log_2 p(y_j|x_i)) = \\ &= - \sum_{i=1}^2 \sum_{j=1}^2 p(y_j|x_i)(p(x_i) \log_2 p(x_i) + p(y_j|x_i)p(x_i) \log_2 p(y_j|x_i)) = \\ &= - \sum_{i=1}^2 \left[ \sum_{j=1}^2 p(y_j|x_i)(p(x_i) \log_2 p(x_i) + p(y_j|x_i)p(x_i) \log_2 p(y_j|x_i)) \right]. \end{aligned}$$

Далее вынесем за знак суммирования по индексу  $j$  вероятности с индексом суммирования по индексу  $i$ .

$$\begin{aligned} H(X, Y) &= - \sum_{i=1}^2 [(p(x_i) \log_2 p(x_i) \sum_{j=1}^2 p(y_j|x_i) + \\ &+ p(x_i) \sum_{j=1}^2 p(y_j|x_i) \log_2 p(y_j|x_i)]. \end{aligned}$$

Последнее выражение запишем в виде суммы двух слагаемых

$$H(X, Y) = - \sum_{i=1}^2 p(x_i) \log_2 p(x_i) \sum_{j=1}^2 p(y_j|x_i) - \sum_{i=1}^2 p(x_i) \sum_{j=1}^2 p(y_j|x_i) \log_2 p(y_j|x_i).$$

Так как

$$\sum_{j=1}^2 p(y_j|x_i) = 1 \text{ и } \sum_{i=1}^2 p(x_i) = 1, \text{ получаем}$$

$$H(X, Y) = - \sum_{i=1}^2 p(x_i) \log_2 p(x_i) - \sum_{j=1}^2 p(y_j) \log_2 p(y_j),$$

$$H(X, Y) = H(X) + H(Y|X). \quad (5.16)$$

Аналогично, можно записать формулу

$$H(X, Y) = H(Y) + H(X|Y). \quad (5.17)$$

Совместная энтропия комбинированного источника представляется суммой энтропии одного источника и частью энтропии другого источника.

*Вывод.* Если между источниками имеется статистическая зависимость, априорные знания свойств одного источника, приводят к уменьшению среднего количества информации на выходе этого источника

## 5.6. Пропускная способность канала

### 5.6.1. Средняя взаимная информация

Вновь рассмотрим систему связанных источников  $X$  и  $Y$  в виде модели дискретного канала, рис. 2.1. Из теоремы Байеса следует, что апостериорное знание вероятностей символов источника  $Y$  позволяет получить знание о источнике  $X$ . Апостериорная информация источника  $Y$  об источнике  $X$  уменьшает неопределенность источника  $X$  – энтропию  $H(X)$ . В терминах теории информации среднее количество информации, приходящееся на один символ источника  $X$ , но после получения апостериорных знаний о событиях источника  $Y$  (на выходе канала) – это условная энтропия  $H(X|Y)$  („новая“ энтропия источника  $X$ ).

Получение апостериорных знаний о событиях источника  $Y$  (на выходе канала) приводит к уменьшению среднего количества информации, приходящей на символ источника  $X$ .

*Определение 5.4.* Средняя взаимная информация событий источника  $X$  при условии наличия событий источника  $Y$  (на выходе канала) равна

$$I(X; Y) = H(X) - H(X|Y), \quad (5.18)$$

где  $H(X) = -\sum_{x_i \in X} p(x_i) \log_2 p(x_i)$ ,

$H(X|Y) = -\sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log_2 p(x_i|y_j)$ , (см. (5.10)).

Формула (5.18) определяет среднее количество информации о источнике  $X$  с учетом информации переданной источником  $Y$ .

Подставляя в (5.18) выражения  $H(X)$  и  $H(X|Y)$ , получаем формулу среднего количества информации источника  $Y$ :

$$I(X; Y) = \sum_{x_i \in X} p(x_i) \frac{1}{\log_2 p(x_i)} - \left( \sum_{y_j \in Y} \sum_{x_i \in X} p(x_i, y_j) \frac{1}{\log_2 p(x_i|y_j)} \right). \quad (5.19)$$

Формула позволяет осуществлять анализ системы передачи информации (источник – канал – источник).

Из формулы (5.19) следует, что среднее количество информации, получаемое при наблюдении источника  $Y$  (выхода канала), зависит:

– от статистических характеристик источника  $X$ , т. е. от распределения вероятностей символов  $p(x_i)$  источника  $X$ ;

– от статистических характеристик канала, т. е. от условных вероятностей  $p(x_i|y_j)$  канала (матрицы канала).

Рассмотрим характерные свойства канала.

1. В канале без шумов, когда вероятность ошибки  $p = 0$ , совместная вероятность  $p(y_j, x_i) = 0$  при  $i \neq j$ . В этом случае компонента

$$H(X|Y) = (\sum_{y_j \in Y} \sum_{x_i \in X} p(y_j, x_i) \frac{1}{\log_2 p(x_i|y_j)}) \quad (5.20)$$

выражения (5.19) равна нулю.

2. Если  $p = 0$ , при  $i = j$  вероятность  $p(x_i|y_j) = 1$ . Тогда значение выражения

$$\frac{1}{\log_2 p(x_i|y_j)} = -\log_2 p(x_i|y_j) = -\log_2 1 = 0$$

и компонента  $H(X|Y)$  также становится нулевой.

Отсюда следуют фундаментальные выводы теории информации.

1. Среднее количество информации, выдаваемое источником  $Y$ , в канале без шумов достигает максимума

$$I(X; Y)_{max} = H(X)$$

2. В канале без шумов вся информация источника  $X$  передается на выход канала достоверно (без потерь).

3. При увеличении вероятности ошибки  $p$  в канале среднее количество достоверно передаваемой информации  $I(X; Y)$  снижается.

Используя соотношения между совместной и условной энтропией (5.16) в форме

$$H(X) = H(X, Y) - H(Y|X)$$

и формулу (5.18), получаем другое выражение взаимной информации:

$$I(X; Y) = H(X) - H(X|Y) = H(X, Y) - H(X|Y) - H(Y|X). \quad (5.21)$$

### 5.6.2. Пропускная способность канала с матрицей переходных вероятностей



**Определение 5.5.** Пропускная способность  $C$  канала – это максимальная взаимная информация  $I(X; Y)_{\max}$ , которая может быть достигнута в канале с матрицей  $\mathbf{P}$  переходных вероятностей.

*Замечание.* Пропускная способность канала определяет максимальную скорость передачи информации, при которой она может передаваться без ошибок.

### 5.6.3. Пропускная способность двоичного симметричного канала

Рассмотрим ДСК с источником  $X = \{x_1 = 0, x_2 = 1\}$ . Вероятности символов источника  $X$  обозначим  $p(x_1)$  и  $p(x_2)$ . Канал характеризуется вероятностью ошибок  $p_e$  и матрицей канала

$$\mathbf{P} = \begin{bmatrix} 1 - p_e & p_e \\ p_e & 1 - p_e \end{bmatrix}.$$

На выходе канала формируются символы  $Y = \{y_1 = 0, y_2 = 1\}$ . Если известны входные (априорные) вероятности ДСК, выходные вероятности  $p(y_i)$  можно найти как (5.3)

$$\begin{bmatrix} p(y_1) \\ p(y_2) \end{bmatrix} = \mathbf{P} \begin{bmatrix} p(x_1) \\ p(x_2) \end{bmatrix} = \begin{bmatrix} 1 - p_e & p_e \\ p_e & 1 - p_e \end{bmatrix} \begin{bmatrix} p(x_1) \\ p(x_2) \end{bmatrix}.$$

Для определения пропускной способности ДСК необходимо вычислить среднюю взаимную информации  $I(X; Y)$ , т. е.

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

В примере 5.2 для ДСК была найдена условная энтропия

$$H(Y|X) = H(p_e) = -[(1 - p_e) \log_2(1 - p_e) + p_e \log_2 p_e].$$

Взаимная информация равна

$$I(X; Y) = H(Y) - H(p_e).$$

Пропускная способность ДСК определяется как

$$C = I(X; Y)_{\max} = \max \{H(Y) - H(p_e)\}.$$

Когда шумов в канале нет,  $x_1 = y_1, x_2 = y_2$ . Энтропия

$$H(Y) = H(X) = 1 \frac{\text{бит}}{\text{символ}}$$

достигает своего максимального значения в случае равенства вероятностей символов источника  $X$ . Поэтому пропускная способность ДСК характеризуется величиной

$$C = \{1 - H(p_e)\} \frac{\text{бит}}{\text{символ}}.$$

Рассмотрим ситуации возможные при передаче информации по ДСК.

1. Если ошибки не возникают,  $p_e = 0$ , условная энтропия (формула Шеннона)

$$H(Y|X) = -[p_e \log_2 p_e + (1 - p_e) \log_2 (1 - p_e)] = 0.$$

Обеспечивается максимальная пропускная способность  $C = 1$  ДСК. Осуществляется достоверная (надежная) передача всей информации.

2. Если  $p_e = 1$ , условная энтропия

$$H(Y|X) = -[p_e (\log_2 p_e + (1 - p_e) \log_2 (1 - p_e))] = 0.$$

В этом случае пропускная способность также равна  $C = 1$ . Среднее количество передаваемой информации также оказывается равным 1 бит/символ. Но в этом случае с вероятностью единица принятый символ не равен переданному символу. Процент ошибок достигает 100%. Невозможно определить, какой символ передавался 0 или 1.

3. Если  $p_e = 0,5$ ,  $H(p_e) = 1$  и

$$C = \{1 - H(p_e)\} = 0,$$

передача информации по ДСК невозможна.

На рис. 5.2 показана зависимость пропускной способности ДСК от вероятности ошибочного приема.

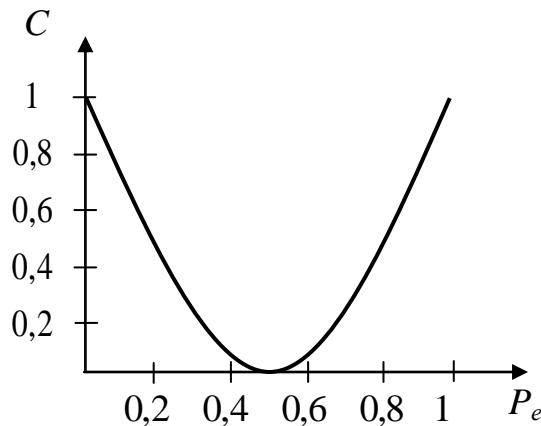


Рис. 5.1. Пропускная способность ДСК

При наличии шумов пропускная способность  $C$  в канале всегда меньше одного бита на символ, рис. 5.1. Из рисунка видно, что с ростом  $p_e$  от 0 до 0,5 пропускная способность  $C$  убывает от своего максимального значения, равного 1 до 0. Среднее количество передаваемой информации оказывается равным нулю.

*Пример 5.3.* Пусть в среднем один из каждых 100 символов принимается неправильно, т. е.  $p_e = 0,01$ . Определить пропускную способность ДСК.

Решение.  $C = [1 + 0,01 \log_2 0,01 + 0,99 \log_2 0,99] = 0,98 \frac{\text{бит}}{\text{символ}}.$

Ошибки в канале уменьшают среднее количество передаваемой информации. В примере, вместо одного бита достоверно получено 0,98 бита.

Известна другая форма записи пропускной способности ДСК.

*Определение 5.6.* Пропускная способность двоичного симметричного канала с вероятностью ошибки  $p_e$  равна

$$C = W[1 + p_e \log_2 p_e + (1 - p_e) \log_2 (1 - p_e)] \text{ бит/с},$$

где  $W = 1/\tau$  – тактовая частота следования информационных символов;  $\tau$  – длительность символа.

При отсутствии помех  $p_e = 0$ , пропускная способность достигает максимума

$$C_{\max} = W \text{ бит/с}.$$

## 5.7. Дифференциальная энтропия

Понятие дифференциальной энтропии вводится для непрерывных источников информации. Выходом такого источника является непрерывный (аналоговый) сигнал  $x(t)$  – случайная функция от времени  $t$ . Для непрерывных представлений энтропии, вместо вероятностей символов источников рассматриваются функции распределения плотностей переменных непрерывного источника. Введение функций распределения позволяет определять понятие энтропии и условной энтропии для непрерывных источников.

*Определение 5.7.* Дифференциальная энтропия – это среднее количество информации непрерывного источника определяется как

$$H(X) = - \int_{-\infty}^{\infty} f(x) \log_2 f(x) dx,$$

где  $f(x)$  обозначает функцию распределения вероятности случайного процесса непрерывного источника.

Основываясь на общих понятиях о дифференциальной энтропии можно утверждать, что энтропийное представление, описывающие дискретные источники без памяти и каналы, справедливы и для непрерывных источников и каналов.

## 5.8. Пропускная способность непрерывного канала

Качество системы передачи информации характеризуется вероятностью ошибки на бит (частотой ошибок на бит). При передаче сигналов по каналу с аддитивным гауссовским шумом вероятность ошибки на бит может быть уменьшена путем увеличения мощности передатчика, которая также является одной из характеристик качества системы. Лучшей из двух систем передачи

данных считается та, которая достигает желаемой частоты ошибок на бит при меньшей мощности передатчика.

Сообщение из  $k$  информационных бит имеет энергию

$$E_c = \sum_{n=0}^{k-1} |x(n)|^2.$$

Энергия сигнала, соответствующая одному информационному биту, определяется соотношением

$$E_b = \frac{E_c}{k} = \frac{1}{k} \sum_{n=0}^{k-1} |x(n)|^2. \quad (5.22)$$

*Замечание.* Проверочные символы, символы синхронизации (например, начала кодового слова, строчные, кадровые или символы канального обмена и др.) не несут информации и поэтому не могут участвовать в вычислении  $E_b$ .

Для сообщений, передаваемых со скоростью  $R_i$  информационных бит/с, величина  $E_b$  определяется из выражения

$$E_b = \frac{P_c}{R_i},$$

где  $P_c$  – средняя мощность сообщения.

На вход приемника поступает также и белый шум с односторонней спектральной плотностью  $N_0$  Вт/Гц. Очевидно, что на частоту ошибок на бит влияет только отношение  $\frac{E_b}{N_0}$ . Сравнительные качественные характеристики различных способов передачи сигналов можно получить, оценивая зависимости их вероятностей ошибок на бит от отношения  $\frac{E_b}{N_0}$ . Нижняя, теоретически достижимая в цифровой системе передачи информации, граница  $\frac{E_b}{N_0}$  (абсолютного значения) определяется из формулы пропускной способности непрерывного канала.

**Теорема 5.2.** Пропускная способность (Хартли – Шеннона) идеального канала равна

$$C = W \log_2(1 + P_c/P_N), \quad (5.23)$$

где  $W$  – ширина полосы частот канала;  $P_N = N_0 W$  – средняя мощность помех с нормальным законом распределения амплитуд и равномерным спектром в полосе частот канала.

### 5.8.1. Граница Шеннона

Определим границу абсолютного значения отношения  $\frac{E_b}{N_0}$ . Ширина полосы сигнала в формуле (5.21) не ограничена. Устремим  $W$  к бесконечности и найдем предельное значение пропускной способности канала  $C_\infty$ :

$$C_{\infty} = \lim_{W \rightarrow \infty} C = \lim_{W \rightarrow \infty} W \log_2 \left(1 + \frac{P_c}{N_0 W}\right).$$

Обозначим величину  $\frac{1}{W}$  символом  $\gamma$ , тогда можно записать

$$C_{\infty} = \lim_{\gamma \rightarrow 0} C = \lim_{\gamma \rightarrow 0} \frac{1}{\gamma} \log_2 \left(1 + \frac{P_c}{N_0} \gamma\right). \quad (5.24)$$

Функция (5.24) в точке  $\gamma = 0$ , принимая вид  $0/0$ , не определена. Для раскрытия неопределенности и вычисления предела воспользуемся правилом Лопиталя. Продифференцируем числитель и знаменатель (5.22) по  $\left(1 + \frac{P_c}{N_0} \gamma\right)$ .

$$C_{\infty} = \lim_{\gamma \rightarrow 0} \frac{\log_2 \left(1 + \frac{P_c}{N_0} \gamma\right)'}{\gamma'}. \quad (5.25)$$

Дифференцирование числителя (5.25) приводит к выражению

$$\log_2(x)' = \log_2 e.$$

Дифференцирование знаменателя приводит к

$$\frac{d\gamma}{d\left(1 + \frac{P_c}{N_0} \gamma\right)} = \frac{N_0}{P_c}.$$

Формулу (5.25) можно записать как

$$C_{\infty} = \lim_{W \rightarrow \infty} C = \frac{P_c}{N_0} \log_2 e = 1,443 \frac{P_c}{N_0}. \quad (5.26)$$

Выражения (5.26) и (5.23) определяют границу Шеннона. Подставив в последнее выражение значение мощности информационных бит

$$R_i E_b = P_c,$$

получим граничное значение  $\frac{E_b}{N_0}$  для максимальной скорости передачи информации  $R_i = C_{\infty}$ :

$$R_i \leq 1,443 \frac{P_c}{N_0} = 1,443 R_i \frac{E_b}{N_0};$$

$$1 \leq 1,443 \frac{E_b}{N_0};$$

$$0,69 \cong \frac{1}{1,443} \leq \frac{E_b}{N_0}.$$

*Замечание.* Нужно иметь в виду, что выражение пропускной способности

Шеннона справедливо только тогда, когда передаваемый сигнал образует аддитивную смесь с белым шумом. Кроме того, по своим статистическим свойствам сигнал подобен нормальному стационарному шуму с заданной средней мощностью и равномерной спектральной плотностью внутри полосы частот  $W$ .

**Теорема 5.3.** В системе, передающей информацию в условиях белого гауссовского шума с односторонней спектральной плотностью  $N_0$  необходимо, чтобы энергия на бит удовлетворяла неравенствам:

$$E_b \geq 0,69N_0;$$

$$\frac{E_b}{N_0} \geq 0,69 \cong -1,6 \text{ dB}.$$

*Вывод.* Для передачи одного информационного бита необходимо, чтобы отношение энергии на бит  $E_b$  к спектральной плотности мощности шума  $N_0$  было, как минимум 0,69.

Формула (5.26) приводит к очень важному практическому выводу:

– для случая малого отношения

$$\frac{P_c}{P_N} = \frac{\text{сигнал}}{\text{шум}} \ll 1$$

на входе приемника пропускная способность канала

$$C_\infty = 1,443 \frac{P_c}{N_0}$$

не зависит от ширины его пропускания, а определяется средней мощностью передаваемого сигнала и спектральной плотностью мощности шума (мощности, приходящейся на единицу полосы).

**Теорема 5.3** определяет нижний предел допустимого абсолютного отношения  $\frac{E_b}{N_0}$ . Верхний предел установлен экспериментально. При отношении сигнал/шум  $\frac{E_b}{N_0} \geq 12 \text{ dB}$  обеспечивается практически безошибочная передача информации. Следовательно, практическая необходимость применения помехоустойчивого кодирования и выбора соответствующего кода возникает лишь в том случае, когда соотношение  $\frac{E_b}{N_0}$  лежит в диапазоне минус 1,6 dB плюс 12 dB. Для специальных систем (космических, военных), где требуется повышенная надежность передачи, например, телеметрии, помехоустойчивое кодирование позволяет получить значительно большее значение отношения сигнал/шум на выходе декодера приемника, и тем самым практически не допустить возможность возникновения ошибки.

Формулу (5.23) можно записать в следующем виде:

$$I = WT \log_2(1 + P_c/P_N), \quad (5.27)$$

где  $I$  характеризует максимальное количество информации, передаваемое по каналу за время  $T$ . Из (5.27) следует, что при уменьшении отношения  $(P_c/P_N)$  или уменьшения  $\frac{E_b}{N_0}$  можно сохранить количество передаваемой информации, расширяя полосу сигнала или увеличивая время передачи. Выражение (5.23) имеет фундаментальное значение для теории информации.

### 5.9. Статистические характеристики каналов

Проектирование, разработка инфокоммуникационных систем различного назначения требует априорного знания значений вероятностей ошибок при передаче информации в том или ином канале. Статистика ошибок в различных каналах к настоящему времени исследована достаточно полно. Опубликованы результаты экспериментов, касающиеся измерения частоты ошибок на информационный бит и характера их группирования в каналах. Вероятность появления ошибок на выходе соответствующего канального приемника по данным отечественной и зарубежной литературы находится в пределах:

радиорелейного  $p_f = 10^{-4} - 10^{-5}$ ;

телефонного  $p_f = 10^{-3} - 10^{-5}$ ;

магнитной ленты  $p_f = 10^{-4} - 10^{-5}$ ;

телеметрического  $p_f = 10^{-6} - 10^{-10}$ ;

оптического диска  $p_f = 10^{-5} - 10^{-6}$ ;

космического телеметрического  $p_f = 10^{-12} - 10^{-23}$ .

В телеметрическом канале космического корабля многоразовых полетов (Space Shuttle) с помощью (127, 120) БЧХ-кода достигается вероятность ложного приема командной информации менее  $p_f = 6,62 \cdot 10^{-23}$ .

Для цифровых устройств различают следующие вероятности ошибок:

– вероятность ошибки из-за дефекта (отказа) элементов;

– вероятность ошибки из-за сбоев элементов.

Дефект – отказ какого-либо элемента, местоположение которого известно, а состояние не изменяется при входных воздействиях. Покажем это на примере матрицы оперативного ЗУ. Дефектной является третья ячейка памяти, рис. 5.2.

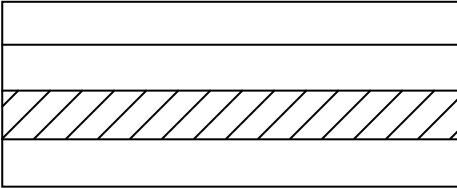
Записываемая информация	Дефект (состояние "1" 3-ей ячейки)	Считываемая информация
0		0
1		1
0		1
1		1

Рис. 5.2. Дефект ЗУ

Сбой это перемежающийся переход состояния элемента из правильного в неправильное и обратно (возможен и при входном воздействии). Для полупроводниковой памяти вероятность ошибки из-за отказа

$$p_{fa} = 10^{-5} - 10^{-6},$$

а вероятность ошибки из-за сбоев элементов

$$p_{fs} = 10^{-4} - 10^{-5}.$$

Для того чтобы гарантировать высокую надежность памяти необходимо иметь вероятность ошибки из-за отказа порядка

$$p_{fa} = 10^{-7} - 10^{-8}.$$

#### Упражнения

5.1. Определить пропускную способность ДСК с вероятностью  $p = 10^{-3}$ .

5.2. Определить пропускную способность непрерывного канала, если  $\frac{P_c}{N_0} = 0,1$ .

5.3. Показать, что отношение  $P_c/P_N$  определяет параметр  $(2^{\frac{R_i}{W}} - 1)$  системы передачи информации.

5.4.1. Вычислить отношение сигнал/шум по мощности на выходе космического канала Марс – Земля.

Средняя мощность сигнала на входе приемника определяется соотношением

$$S = \frac{S_s G_s A}{4\pi D^2} \frac{1}{B},$$

где  $S_s$  – средняя мощность сигнала передатчика;

$G_s$  – коэффициент усиления антенны передатчика;

$D$  – расстояние до приемника;

$A$  – эффективная площадь антенны приемника;

$B$  – коэффициент потерь мощности сигнала на входе приемника, учитывающий потери мощности (влияние ионосферы, тропосферы, неравномерности диаграмм направленности антенны передатчика и приемника, и др.).

Экспериментальные исследования космического канала показали, что величина  $B$  находится в диапазоне 1,2 – 2.

Расстояние  $D \approx 400$  млн. км.



Пусть  $S_s = 60$  Вт. Если принять коэффициент полезного действия передатчика 10% (что реально для несущей частоты передатчика космического аппарата порядка  $f = 1000$  МГц), передатчик должен иметь блок энергоснабжения мощностью 600 Вт. Для обеспечения такого расхода электричества на Марсе потребуются солнечные батареи с площадью панелей  $\approx 15$  м<sup>2</sup>.

Коэффициент  $G_s$  зависит от размеров антенны с параболическим рефлектором передатчика космического аппарата. Пусть диаметр антенны равен 1,5 м. В этом случае можно иметь коэффициент усиления  $G_s \approx 200$ .

Из-за большого расстояния между передатчиком и приемником, и необходимостью иметь приемлемое значение средней мощности сигнала на входе приемника, на Земле используются приемные антенны (антенные поля) большой площади. Пусть  $A = 600$  м<sup>2</sup>.

Исследования космических каналов большой протяженности показали, что основной помехой в них является белый шум со спектральной плотностью мощности

$$N_0 = kT,$$

где  $k = 1,38 \cdot 10^{-23} \frac{\text{Дж}}{\text{град}}$  – постоянная Больцмана,

$T$  – шумовая температура всех источников помех (собственные шумы устройств космического аппарата, Галактика, яркие звезды, Солнце, Луна, Земля, атмосфера и др.). На частоте  $f = 1000$  МГц шумовая температура  $T \approx 50^\circ\text{К}$ .

Пусть ширина полосы частот  $W$  канала равна  $W = 20$  КГц.

5.4.2. Вычислить пропускную способность космического канала Марс – Земля.

## 6. ВВЕДЕНИЕ В ЗАЩИТУ ИНФОРМАЦИИ

Защита информации – комплекс мероприятий, направленных на обеспечение информационной безопасности.

Под информационной безопасностью понимают защищенность информации и поддерживающей инфраструктуры от случайных и преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб пользователям информации и поддерживающей инфра-

структуре. Информационная безопасность не сводится только к защите от несанкционированного доступа к информации, это более широкое понятие. Субъект информационных отношений может пострадать (понести убытки, моральный ущерб) не только от несанкционированного доступа к информации, но и от повреждения элементов информационной системы.

Информационная безопасность в значительной степени зависит от надежности поддерживающей инфраструктуры к которой можно отнести системы электро-, водо- и теплоснабжения, средства коммуникации, обслуживающий персонал и др.

В определении информационной безопасности употреблено понятие «неприемлемый ущерб». Например, неприемлемым, недопустимым ущербом являются: нанесения вреда здоровью, окружающей среде, урон, нанесенный стране и пр. Часто порог неприемлемости имеет материальное (денежное) выражение. Застраховаться от всех видов ущерба невозможно. Тогда целью защиты информации становится уменьшение размеров ущерба до допустимых значений.

### **6.1. Составляющие информационной безопасности**

Спектр интересов субъектов, связанных с использованием инфокоммуникационных систем разделяют на следующие категории:

- обеспечение доступности;
- обеспечение конфиденциальности информационных ресурсов и поддерживающей инфраструктуры;
- обеспечение целостности информации.

Информационные системы необходимы для получения информационных услуг. Если услуги становятся по разным причинам недоступными, то наносится ущерб субъектам информационных отношений.

*Определение 6.1.* Доступность – это возможность получить информационную услугу за приемлемое время.

*Определение 6.2.* Конфиденциальность – это статус, представленный данным и определяющий требуемую степень их защиты.

*Определение 6.3.* Секретность – это понятие, которое употребляется по отношению к отдельным лицам, которые имеют право объявлять информацию закрытой, т. е. подлежащей защите.

*Определение 6.4.* Под целостностью понимают защищенность информации от разрушения и несанкционированного изменения.

Целостность подразделяют на:

- статическую, понимаемую как неизменность информационных объектов;

– динамическую, относящуюся к правильному выполнению сложных действий (транзакций).

Рецептура лекарств, характеристики комплектующих изделий, описание хода технологического процесса, база данных землетрясений, данные аэро –, космического дистанционного зондирования участков Земли – все это примеры информации, нарушение целостности которой может привести к неприемлемому ущербу. Преднамеренное искажение информации – это также нарушение целостности.

Введенные категории информационной безопасности рассматривают относительно независимо. Считается, что если все три категории реализуются, то обеспечивается информационная безопасность. В этом случае субъектам информационных отношений не будет нанесен неприемлемый ущерб.

*Замечание.* К поддерживающей инфраструктуре применимы те же требования целостности и доступности, что и к информационным системам.

## **6.2. Информационные угрозы и атаки**

*Определение 6.5.* Угроза – это потенциальная возможность определенным образом нарушить информационную безопасность.

*Определение 6.6.* Попытка реализации угрозы называется атакой, а тот, кто предпринимает такую попытку, называется злоумышленником.

Угроза является следствием наличия уязвимых мест в защите информационных систем (таких, например, как возможность доступа посторонних лиц к информационным подсистемам).

*Определение 6.6.* Промежуток времени от момента, когда появляется возможность использовать злоумышленником слабое место в защите, и до момента, когда пробел в защите ликвидируется, называется окном опасности.

Некоторые угрозы существуют в силу самой природы современных информационных систем. Например, угрозы отключения электричества или выхода параметров источника напряжения за допустимые пределы существуют в силу зависимости аппаратного обеспечения информационных систем от надежности и качественных характеристик электропитания. Иметь представление о возможных угрозах, а также об уязвимых местах защиты необходимо для того, чтобы выбирать эффективные и наиболее экономичные средства обеспечения информационной безопасности. Угрозы классифицируют по следующим критериям:

- по составляющим информационной безопасности (доступность, целостность, конфиденциальность);
- по компонентам информационных систем (аппаратура, поддерживаю-

щая инфраструктура, данные, программы);

- по способу осуществления: случайные и преднамеренные.

Случайные угрозы могут быть обусловлены физическими воздействиями стихийных природных явлений, не зависящих от человека. К угрозам случайного характера также относятся аварийные ситуации на объекте размещения информационной системы. Аварийные ситуации это – отказы аппаратуры системы, пожары, наводнения, ураганы, разряды атмосферного электричества и др.

Преднамеренные угрозы направлены против элементов и подсистем, образующих информационную систему.

### **6.3. Модели разграничения доступа к информации**

Устранение или уменьшение преднамеренных угроз основывается на использовании определенных моделей разграничения доступа к информации. Модели строятся с учетом следующих возможных злоумышленных действий:

- несанкционированный доступ к информации и ознакомление с хранящейся и циркулирующей в информационной системе конфиденциальной информацией;
- доступ локальных пользователей к информации, на работу с которой они не имеют полномочий;
- несанкционированное копирование сведений: данных и программ;
- кража физических носителей информации и оборудования, приводящая к утрате информации;
- умышленное уничтожение информации;
- несанкционированная модификация документов и баз данных;
- фальсификация сообщений;
- дезинформация, т. е. навязывание ложного сообщения и пр.

Конкретные модели разграничения доступом к информации в инфокоммуникационных системах должны учитывать следующие угрозы доступности.

1. Самыми частыми и самыми опасными (с точки зрения размера ущерба) являются случайные (непреднамеренные) ошибки лиц, обслуживающих информационные системы. По некоторым источникам, до 65% потерь – это следствие непреднамеренных ошибок (ошибки в программе), вызвавшие крах системы.

2. Отказ информационной системы, повреждение аппаратуры, разрушение данных (например, мощный кратковременный импульс способен разрушить данные на магнитных носителях).

3. Программные атаки на доступность, когда используется агрессивное потребление ресурсов (полосы пропускания сетей, вычислительной способности процессора или оперативной памяти).

4. Внедрение в атакуемые системы вредоносного программного обеспечения.

5. Отказ поддерживающей инфраструктуры (нарушение работы (случай-

ное или преднамеренное)), системы связи, электропитания, террористический акт и пр.

6. Стихийные бедствия. По статистике на долю огня, воды, землетрясений, ураганов и пр. приходится 13% потерь, нанесенных информационным системам.

7. Отказ пользователей (невозможность работать с информационной системой в силу отсутствия подготовки, технической поддержки, справочной литературы и др.).

### **6.3.1. Методы разграничения доступа и способы их реализации**

Общий подход по методам разграничения доступа к информации и способам их реализации основывается на стандартах информационной безопасности. Стандарты описывают средства, с помощью которых обеспечивается информационная безопасность.

Исторически первым стандартом, получившим широкое распространение и оказавшим влияние на базу стандартизации информационной безопасности во многих странах мира, стал стандарт Министерства обороны США «Критерии оценки доверенных компьютерных систем». Этот стандарт, называемый по цвету обложки «Оранжевая книга», был опубликован в 1983 году. Особенностью стандарта является то, что рассмотрению подлежат так называемые доверенные системы, т. е. информационные системы, которым можно оказать определенную степень доверия, с точки зрения информационной безопасности. «Оранжевая книга» поясняет понятие безопасной системы, которая управляет с помощью соответствующих средств, доступом к информации так, что только авторизованные лица или процессы, действующие от их имени, получают право читать, записывать, создавать и удалять информацию.

*Определение 6.7.* Доверенная система – это информационная система, использующая достаточные аппаратные и программные средства, чтобы обеспечить одновременную обработку информации разной степени секретности группой пользователей без нарушения прав доступа.

*Замечание.* Безопасность и доверие оцениваются с точки зрения управления доступом к данным, что является одним из средств обеспечения конфиденциальности и целостности (статической). Например, руководство режимного предприятия в первую очередь заботится о защите от несанкционированного доступа к информации, т. е. конфиденциальности. В частности, правила определяют, в каких случаях пользователь может работать с конкретными данными.

Если понимать политику безопасности узко, то это правила разграничения доступа к информации («Можно читать только то, что положено» или «Субъект может читать информацию из объекта, если уровень секретности субъекта не ниже, чем у объекта»).

Стандартные средства осуществляют разграничение доступа к информации на следующих уровнях.

1. Законодательный. В области информационной безопасности законы реально работают через нормативные акты, подготовленные соответствующими ведомствами.

2. Административный. Практическое осуществление административных мер защиты информации связано с ограничением доступа людей к аппаратуре, компьютерам, программам, обрабатываемой информации, данным и пр. На этом уровне устанавливаются способы доступа к информации и условия ее распространения, регламентируются процедуры выдачи допусков к данным. Часть из этих правил определяются законами и нормативными актами. Но большинство правил определяет организация на основе приказов и инструкций. При введении административных мер допуска возникают определенные проблемы. Реализация этих мер создает неудобства для пользователей. Эффективность административных мер может свестись к нулевой: список паролей будет лежать под стеклом, дверь то запирается, то открыта и т. д.

3. Процедурный. На этом уровне выделяют такие меры доступа как:

- управление персоналом, физическая защита;
- технический.

Реализация меры по управлению персоналом строится на двух принципах:

- разделение обязанностей;
- минимизация привилегий.

Принцип разделение обязанностей предписывает распределение ответственности. Например, один инженер разрабатывает схему процессора, а другой разрабатывает схему слежения за задержкой и пр.

Принцип минимизации привилегий предписывает выделять пользователям информационной системы только те права, которые необходимы для выполнения своих обязанностей. Назначение этого принципа – уменьшить ущерб от случайных или умышленных действий.

Основной принцип физической защиты доступом формулируется как «непрерывность защиты в пространстве и времени». Для физической защиты окон опасности быть не должно. Под физической защитой здесь понимается отражение попыток несанкционированного доступа к данным. Средства физического управления доступом:

- охрана;
- двери с замками;
- телекамеры;
- датчики движения и др.

Выделяют четыре вида охранных мер:

- охрана границ территории (зоны, окружающей здание);
- охрана самого здания;
- охрана входов в здание;
- охрана критических зон.

Для защиты границ территории используют ограды, инфракрасные или СВЧ детекторы, а также замкнутые телевизионные системы.

Для защиты здания оно должно быть построено из прочных материалов и иметь толстые стены. Здание фирмы IBM, например, имеет стены из железобетона толщиной 33 см.

Для обнаружения проникновения злоумышленника в критическую зону используют системы сигнализации (системы наблюдения за входом в помещение). К наиболее распространенным системам сигнализации относятся следующие.

1. Фотометрические системы обнаруживают изменение уровня освещенности.

2. Звуковые, ультразвуковые, СВЧ системы обнаружения реагируют на изменение частоты сигнала, отраженного от движущегося объекта.

3. Акустосейсмические (вибрационные) системы обнаруживают шум и вибрации.

4. Системы, реагирующие на приближение к объекту, обнаруживают нарушение структуры электромагнитного или электростатического поля.

Кроме того, посредством физической защиты реализуется управление носителями.

#### **6.4. Обеспечение целостности данных в инфокоммуникационных системах и сетях**

С целью нарушения статической целостности злоумышленник может:

- ввести неправильные данные;
- изменить данные;
- разрушить информацию деструктивными программными воздействиями (компьютерными вирусами и пр.).

Угрозами динамической целостности являются:

- нарушение атомарности сложных действий;
- переупорядочение;
- кража;
- дублирование данных или внесение дополнительных сообщений (сетевых пакетов и пр.).

Кроме того, случайные ошибки пользователей системы, обслуживающего персонала, грозят повреждением аппаратуры, разрушением программ и пр.

Выделяют следующие направления деятельности, относящиеся к обеспечению целостности данных.

1. Необходима поддержка пользователей – это консультирование, связанное с информационной безопасностью. Целесообразно фиксировать вопросы пользователей, чтобы выявлять их типичные ошибки.

2. Для обеспечения целостности и доступности поддерживающей инфраструктуры нужно защищать оборудование от краж и повреждений, выбирать оборудование с максимальным временем наработки на отказ, дублировать узлы, иметь запасные части.

3. Поддержка программного обеспечения – необходимо следить за тем, какое программное обеспечение установлено на компьютерах; необходим контроль неавторизованного доступа к программам и их изменениям.

Конфигурационное управление контролирует и фиксирует изменения в программной конфигурации. Фиксация изменений позволяет восстановить конфигурацию после аварии. Необходима защита от случайных, непродуманных модификаций. Предусмотреть возможность возврата к прошлой программной работающей версии.

Резервное копирование необходимо для восстановления программ и данных после аварии. Должны быть созданы полные эталонные копии программ системы. Копии размещаются в месте, защищенном от несанкционированного доступа.

Управление носителями, посредством физической защиты, обеспечивает целостность хранящейся вне компьютерных систем. Под физической защитой здесь понимается не только отражение попыток несанкционированного доступа, но и предохранения от вредных влияний окружающей среды (влага, жара, холод, магнетизм).

Документирование. В виде документов оформляется журнала учета носителей информации, план восстановления данных после аварии.

Для построения надежной защиты информации современная информационная система должна включать в себя и такие сервисы безопасности как:

- помехоустойчивое кодирование;
- криптографическое кодирование (шифрование).

## **6.5. Общие сведения по классической криптографии**

### **6.5.1. Криптографическое кодирование**

*Определение 6.8.* Криптография – это раздел прикладной математики, изучающий модели, методы, алгоритмы, программные и аппаратные средства преобразования информации.

Криптографическое кодирование (шифрование) относится к традиционным сервисам безопасности. Это наиболее мощное средство обеспечения информационной безопасности. Криптографическое кодирование необходимо для реализации трех сервисов безопасности:

- шифрования;
- контроля целостности;
- аутентификации.

*Определение 6.9.* Шифрование – это кодирование (преобразование) исходного текста, который носит название открытого текста, в шифрованный текст (криптограмму) с помощью секретного ключа.

Процесс создания криптограммы записывается как

$$C = E_k(m),$$



где  $m$  (message) – открытый текст,  $E$  (encryption) – шифрующая функция (кодирование, преобразование),  $k$  – секретный ключ,  $C$  – шифротекст.

Ключ  $k$  определяет также процесс, обратный шифрованию, который называют расшифрованием (дешифрованием)

$$m = D_k(C),$$

где  $D$  (decryption) – дешифрирующая функция (декодирование, обратное преобразование).

При этом должно выполняться тождество

$$m = D_k(C) = D_k(E_k(m)).$$

*Замечание.* Алгоритмы шифрования  $E$  и дешифрования  $D$  открыты, и секретность исходного текста  $m$  в шифротексте  $C$  зависит от ключа  $k$ .

Существуют два основных алгоритма шифрования.

В первом, процессы шифрования – дешифрования используют один и тот же секретный ключ отправителем и получателем информации. Этот метод используется в, так называемых, симметричных криптосистемах. В них ключ поставляется абонентам специальным конфиденциальным способом.

Второй метод шифрования основан на двух ключах. Первый, – открытый ключ, доступный всем пользователям информационной системы, применяется при шифровании. Второй ключ, математически связанный с первым – секретный. Он нужен при расшифровании текста. Такие криптосистемы называются асимметричными, или криптосистемами с открытым ключом.

*Определение 6.10.* Криптосистема – это система, реализованная программно, аппаратно или программно -аппаратно и осуществляющая криптографическое преобразование информации.

Аппаратная реализация имеет существенную стоимость, однако ей присущи и преимущества:

- высокая производительность;
- сравнительная простота;
- защищенность.

Программная реализация криптосистемы более практична, допускает гибкость в использовании.

### **6.5.2. Требования к криптосистемам защиты информации**

Основными требованиями являются:

- функциональное преобразование сообщения  $E_k(m)$  и обратное преоб-

разование зашифрованного сообщения  $D_k(C)$  должны быть сравнительно легко вычислимы;

- не зная ключ  $k$ , невозможно за заданное (реальное) время вычислить сообщение  $m$  по шифротексту  $C = E_k(m)$ . Не должно быть простых зависимостей между используемыми ключами;

- изменение длины ключа не должно вести к качественному ухудшению алгоритма шифрования;

- число операций, необходимых для определения ключа по фрагменту шифрованного текста должно быть не меньше общего числа ключей;

- число операций, необходимых для дешифрования информации путем перебора всевозможных ключей должно иметь строгую нижнюю оценку. При этом следует учитывать вычислительные возможности по числу операций в единицу времени современных суперкомпьютеров, возможности использования сетевых вычислений;

- знание алгоритма шифрования не должно влиять на надежность защиты.

### 6.5.3. Криптоанализ

Криптоанализ занимается задачами, обратными задачам криптографии. Основной задачей специалиста криптоаналитика является поиск ключа. Ему могут представиться следующие возможности для атаки:

- получен лишь зашифрованный текст  $C = E_k(m)$ ;
- известны незашифрованный и зашифрованный тексты;
- имеется возможность выбрать пространство сообщений  $\{m\}$  и пространство шифротекстов  $\{C\}$ , т. е. иметь пару  $\{m, C\}$ .

Криптоанализ и криптография развиваются параллельно. Криптографы пытаются создать такую криптосистему, которая была бы стойкой ко всем известным в данный момент методам криптоанализа.

Эффективность шифрования зависит от сохранения тайны ключа и криптостойкости шифра.

*Определение 6.11.* Криптостойкостью называется характеристика шифра, определяющая его стойкость криптоанализу (к дешифрованию) без знания ключа.

Имеется несколько основных показателей криптостойкости, среди которых:

- количество всех возможных ключей;
- среднее время, необходимое для криптоанализа.

В практических применениях ограничиваются алгоритмами, обеспечивающими вычислительную стойкость, т. е. такими алгоритмами, которые теоретически раскрываемы, но требуют для осуществления криптоанализа значительных вычислительных затрат, например, работы 10 млн. компьютеров в течение 10000 лет.

История развития криптологии имеет большую продолжительность. При

археологических раскопках в Месопотамии был найден, относящийся к 20 веку до н. э. один из самых древних шифротекстов. Он был написан клинописью на глиняной дощечке и содержал коммерческую тайну: рецепт глазури для покрытия гончарных изделий. В 17 веке кардинал Ришелье создал первую в мире шифрослужбу. Задачами криптологии занимались такие известные ученые как Ньютон, Лейбниц, Эйлер, Гаусс и др. В развитии криптологии принято выделять три этапа.

Первый этап – с древних времен до 1949 года, характеризовался частными, узкоспециальными и вычислительно простыми алгоритмами криптографии и криптоанализа. Этот этап называют этапом докомпьютерной криптологии.

Второй этап – с 1949, когда К. Шеннон опубликовал работу «Теория связи в секретных системах», до 1976 года. В этот период проводились большие исследования с использованием компьютеров. Основным потребителем результатов криптологии являлась связь для военных и дипломатических организаций, поэтому криптология была закрытой наукой.

Третий этап – с 1976 года, когда была опубликована работа «Новые направления в криптографии» американских математиков У. Диффи (W. Diffie) и М. Хеллмана (M. Hellman). В этой работе показано, что секретная передача информации возможна без предварительной передачи ключа. Особенностью этого этапа стало применение криптографии в банковском деле, компьютерных сетях и др. приложениях. В развитие криптологии вкладываются значительные государственные средства. Например, в США ежегодные расходы на криптологию составляют порядка 18 – 20 млрд. долларов.

Криптология строится на базе таких дисциплин как теория вероятностей, математическая статистика, алгебра, теория чисел, теория алгоритмов и сложность вычислений. Процесс шифрования осуществляется на специализированных компьютерах.

## **6.6. Алгоритмы блочного шифрования**

Блочное шифрование используется в симметричных криптосистемах. Блочный шифр обрабатывает блок открытого текста фиксированной длины. Процесс шифрования текста  $m$  записывается как

$$C = E_k(m)$$

где  $C$  – это блок шифротекста,  $k$  – секретный ключ, шифрующая функция.

В этом алгоритме зашифрованный первый блок сообщения далее используется для шифрования следующего. Шифрующие процедуры одного типа чередуются с процедурами другого типа. В качестве простых шифров могут быть использованы подстановки  $S$ , перестановки  $L$  и линейные преобразования  $T$ . Секретный ключ может использоваться при осуществлении всех процедур. Блочные шифры используют многократное повторение операций преобразования, называемое раундом шифрования.

На этом принципе работает известный стандарт шифрования данных DES и его модификация AES. DES использует ключи размером 64 бит с эффективной длиной 56 бит. Всего создается  $K_{DES} = 2^{56} \approx 7,2 \times 10^{16}$  ключей. Система AES имеет три варианта размеров ключей 128, 192 и 256 битов. Длина ключа 128 бита позволяет формировать  $K_{AES} = 2^{128} = 3,4 \times 10^{38}$  ключей. Система AES имеет в  $\left(\frac{K_{AES}}{K_{DES}}\right) \approx 10^{21}$  раз больше ключей, чем DES. Если предположить, что можно проверить все ключи DES за одну секунду, то при такой скорости  $R = 2^{56} \frac{\text{бит}}{\text{с}}$ , для тестирования всех ключей блочного шифрования AES потребовалось 149 триллионов лет (возраст вселенной 13,7 миллиарда лет).

Недостатком блочного шифрования является обмен ключами между отправителем и получателем. Передача ключей в практическом аспекте уязвима для перехвата.

## 6.7. Ассиметричные алгоритмы шифрования

Теоретико-числовые алгоритмы являются основой современной криптографии и криптографических систем с открытым ключом. В этом случае программное обеспечение, программный интерфейс криптографических систем с шифрующими алгоритмами строится на базе конечных алгебраических структур: групп, колец, полей.

Первой известной криптографической системой с открытым ключом является система, созданная в Массачусетском технологическом институте в 1978 году. Система названа по фамилиям авторов (R. L. Rivest, A. Shamir, L. Alldean) как RSA-криптосистема. Особенностью математического алгоритма системы является то, что криптосистему создает не отправитель сообщения, а получатель. Алгоритм шифрования основывается на задаче RSA.

### 6.7.1. Задача RSA

Предположим, что произвольный получатель А информации разрешает всем желающим передавать ему секретные сообщения. Т. е. он выступает в качестве получателя сообщения. Получатель А случайным образом выбирает два больших простых числа  $p$  и  $q$ , причем  $p \neq q$ . Числа  $p$  и  $q$  выбираются порядка не менее чем  $2^{256}$ . Эти числа являются секретными.

Получатель А вычисляет число  $N = p \cdot q$ . Число  $N$  называется модулем алгоритма и является несекретным. Получатель А открыто публикует число  $N$ . Таким образом, всем пользователем системы известно число  $N$ , но не известны сомножители –  $p$  и  $q$ .

Решение задачи RSA (дешифрации) сведется к поиску простых делителей  $p$  и  $q$  числа  $N$ . Криптостойкость системы обосновывается сложностью решения задачи факторизации очень больших чисел в произведение простых чисел.

Алгоритм RSA использует понятие функции Эйлера числа  $N$  и теорему Эйлера.

**Определение 6.12.** Количество положительных целых чисел меньших  $M$  и взаимно простых с  $M$  называется функцией Эйлера  $\varphi(M)$ , или тождественной функцией Эйлера. Функция Эйлера  $\varphi(M)$  – это количество вычетов по модулю  $M$ .

**Теорема 6.1.** Если  $p$  – простое число, то  $\varphi(p) = p - 1$ .

Так как пара простых чисел  $p$  и  $q$  известна получателю А, используя свойство мультипликативности функции Эйлера, получатель А секретной информации легко может вычислить значение функции  $\varphi(N)$ :

$$\varphi(N) = \varphi(p \cdot q) = \varphi(p)\varphi(q) = (p - 1)(q - 1).$$

Получатель А публикует также число  $E$ , т. е.  $E$  является несекретным. Число  $E$  выступает в качестве открытого ключа криптосистемы RSA и используется для шифрования данных. Число  $E$  называется шифрующей экспонентой. Выбор получателем А числа  $E$  должен удовлетворять двум условиям:

$$1 < E \leq \varphi(N) = (p - 1)(q - 1); \quad (6.1)$$

$$\text{НОД}(E, \varphi(N)) = \text{НОД}(E, (p - 1), (q - 1)) = 1. \quad (6.2)$$

Следовательно, число  $E$  и число  $\varphi(N)$  должны быть взаимно простыми. Число  $E$  выбирается случайным образом, но часто  $E$  равно числу Ферма

$$E = 2^{2^t} + 1, t \in \mathbb{Z}_N.$$

Например, открытый ключ  $E$  может быть равен таким числам Ферма:

$$5, 17, 257, 65537, \dots$$

**Определение 6.13.** Пара  $(E, N)$  называется открытым ключом RSA (RSA public key).

**Теорема 6.2.** (Теорема Эйлера). Если  $\alpha$  и  $M$  – взаимно простые числа, т. е.  $\text{НОД}(\alpha, M) = 1$ , то

$$\alpha^{\varphi(M)} \equiv 1 \pmod{M}. \quad (6.3)$$

Получатель А пересылает отправителю В пару чисел  $(E, N)$  по открытому (незащищенному) каналу. Таким образом, всем желающим передавать получателю А секретную информацию доступна пара  $(E, N)$ .

Исходный текст  $m$  переводится в числовую форму (шифруется) по формуле

$$m^E \equiv C \pmod{N}. \quad (6.4)$$

Шифруемый текст представляется криптограммой  $C = Z_N \in \{1, 2, \dots, N - 1\}$  в виде одного большого числа. Затем это число разбивается на блоки так, что каждый из них представляется в виде числа  $m_i \in \{0, 1, 2, \dots, N - 1\}$ . Формула (6.4) шифрования (кодирования) текста в числовую форму является несекретной.

Так как по задаче RSA число  $E$  и число  $\varphi(N)$  должны быть взаимно простыми, по теореме Эйлера справедливо выражение

$$E^{\varphi(N)} \equiv 1 \pmod{\varphi(N)} = E^{\varphi((p-1)(q-1))} \equiv 1 \pmod{(p-1)(q-1)}. \quad (6.5)$$

Обозначим функцию Эйлера  $\varphi(\varphi(N)) = \varphi((p-1)(q-1))$  через  $x$ , тогда выражение (6.5) запишем в виде

$$E^x \equiv 1 \pmod{(p-1)(q-1)}. \quad (6.6)$$

Выражение (6.6) можно записать в виде

$$E \cdot E^{x-1} \equiv 1 \pmod{(p-1)(q-1)}. \quad (6.7)$$

Обозначим  $E^{x-1} = d$ . С учетом этого получаем

$$E \cdot d \equiv 1 \pmod{(p-1)(q-1)}, \quad (6.8)$$

где  $d$  – это секретный ключ.

Секретный ключ  $d$ , применяется для дешифрования криптограммы по формуле

$$m \equiv C^d \pmod{N}. \quad (6.9)$$

Таким образом, секретными данными RSA системы является тройка чисел  $(d, p, q)$ .

Нахождение  $d$  получателем А для дешифрации криптограммы сводится к вычислению числа обратному числу  $E$  по  $\pmod{(p-1)(q-1)}$ . Необходимо найти такое число  $d \in \{1, 2, \dots, N - 1\}$  для которого выполняется сравнение (6.8). Так как получателю А известны числа  $p, q, E$ , сравнение разрешимо единственным образом, поскольку  $\text{НОД}(E, (p-1)(q-1)) = 1$ .

*Замечание.* Число  $d$  можно легко вычислить, используя алгоритм Евклида или другие быстрые алгоритмы нахождения обратных чисел.

### 6.7.2. Алгоритм вычисления обратных чисел по теореме Эйлера

Используя формулу (6.3), можно записать

$$\alpha^{-1} \cdot \alpha^{\varphi(M)} \equiv \alpha^{-1} \cdot 1 \pmod{M}.$$

Тогда число, обратное числу  $\alpha$  равно

$$\alpha^{-1} \equiv \alpha^{\varphi(M)-1}. \quad (6.10)$$

Эффективное применение выражения (6.10) требует нахождения значения порядка  $n$  элемента  $\alpha$ , т. е. когда выполняется условие

$$\alpha^n \equiv 1 \pmod{M}.$$

**Теорема 6.3.** Порядок  $n$  числа  $\alpha$  должен быть делителем функции Эйлера  $\varphi(M)$ .

*Пример 6.1.* Пусть  $\alpha = 5$ ,  $M = 31$ . Найти:

- обратное число  $\alpha^{-1}$ ;
- порядок числа 5;
- число, обратное числу  $5^{-1}$  по mod 31.

Решение.

1. Так как  $M$  – простое число,  $\varphi(M) = M - 1 = 30$ . Обратное число

$$\alpha^{-1} = ((\alpha^{\varphi(M)-1})) = ((\alpha^{(M-1)-1})) = ((\alpha^{30-1})) = \alpha^{29}.$$

2. Находим порядок числа 5:

$$5^3 = 125 \equiv 1 \pmod{31}, n = 3.$$

Число 3 является делителем числа  $\varphi(31) = 30$ .

3. Обратное число

$$5^{-1} = 5^{29} = 5^{27} \cdot 5^2 = ((5^3)^9 \cdot 5^2) = 25.$$

Действительно,

$$5 \cdot 5^{-1} = 5 \cdot 25 \equiv 1 \pmod{31}.$$

*Пример 6.2.* Пусть  $\alpha = 6$ ,  $M = 31$ . Найти  $6^{-1}$ .

Решение. Функция Эйлера  $\varphi(M) = M - 1 = 30$ .

Число  $6^{-1} = ((\alpha^{30-1})) = 6^{29}$ .

Хотя ближайшие числа 2, 3, 5, делят число 30, они не являются порядком  $n$  числа 6. Только число 6 является порядком  $n$  числа 6. Действительно

$$6^6 = 46656 \equiv 1 \pmod{31}.$$

Находим число:

$$6^{-1} = ((6^{29})) = ((6^{24} \cdot 6^5)) = ((6^6)^4 \cdot 6^5) = 6^5 = 7776 \equiv 26 \pmod{31}.$$

Действительно,

$$6 \cdot 26 \equiv 1 \pmod{31}.$$

### 6.7.3. Последовательность шагов криптоалгоритма RSA

*Пример 6.3.* Зашифровать слово РУХ с помощью алгоритма RSA.

*Решение. Действия получателя А шифрованной информации.*

1. Выбираем  $p = 3$  и  $q = 7$ .
2. Вычисляем модуль  $N = pq = 3 \cdot 7 = 21$ .
3. Вычисляем значение функции Эйлера для  $N = 21$ ,

$$\varphi(N) = \varphi(21) = (p - 1)(q - 1) = 2 \cdot 6 = 12.$$

4. Выбираем в качестве открытого ключа  $E$  произвольное число с учетом выполнения условий:

$$1 < E \leq \varphi(N); \text{НОД}(E, \varphi(N)) = 1, \text{НОД}(E, 12) = 1. \text{ Пусть } E = 11.$$

5. Из выражения (6.8) вычисляем значение секретного ключа  $d$ , используя алгоритм нахождения обратных чисел по теореме Эйлера:

$$E \cdot d \equiv 1 \pmod{\varphi(N)},$$

$$11 \cdot d \equiv 1 \pmod{12}.$$

$$d = E^{-1} \equiv 11^{-1} \pmod{12}.$$

Число 2 является порядком  $n = 2$  числа  $\alpha = 11$ , так как соблюдается условие:

$$\begin{aligned} \alpha^n &\equiv 1 \pmod{M}; \\ 11^2 &= 121 \equiv 1 \pmod{12}. \end{aligned}$$

Из алгоритма нахождения обратных чисел по теореме Эйлера

$$\alpha^{-1} = ((\alpha^{\varphi(M)-1}))$$

находим  $\varphi(12) = 4$ .

Число, обратное  $E$  равно

$$\begin{aligned} d &= E^{-1} = ((E^{\varphi(12)-1})) = ((E^3)), \\ 11^{-1} &= ((11^3)) = ((11^2 \cdot 11)) = 11. \end{aligned}$$

Проверим правильность полученного значения  $d$ :

$$E \cdot d \equiv 1 \pmod{12} = 11 \cdot 11 \equiv 1 \pmod{12}.$$

6. Получатель А шифрованной информации пересылает отправителю пару открытых чисел ( $N = 21, E = 11$ ).



### *Действия отправителя шифрованной информации*

7. Представляем шифруемое сообщение РУХ в виде последовательности целых чисел в диапазоне  $M = 0, 1, \dots, N - 1$ .

Пусть буква Р записывается числом 1, буква У – числом 2, буква Х – числом 3. Сообщению РУХ соответствует последовательность (блоки) чисел 123. Текст сообщения в виде блоков записывается как  $m_1 m_2 m_3$ , где

$$m_1 = 1, m_2 = 2, m_3 = 3.$$

8. Шифруем текст, используя ключ  $E = 11$  и  $N = 21$  по формуле (6.4)

$$C_i \equiv m_i^E \bmod N \equiv m_i^{11} \bmod 21.$$

Шифротексту соответствуют следующие числа:

$$C_1 = 1^{11} \bmod 21 \equiv 1;$$

$$C_2 \equiv 2^{11} \bmod 21 \equiv 2048 \bmod 21 \equiv 11;$$

$$C_3 \equiv 3^{11} \bmod 21 \equiv 12.$$

Криптограмма имеет вид

$$C_1 C_2 C_3 = Z_N = 11112.$$

После разбиения на блоки она пересылается.

### *Действия получателя по дешифрации криптограммы*

9. Дешифрация криптограммы  $C_1 C_2 C_3 = 1, 11, 12$  производится с использованием секретного ключа  $d = 11$  по формуле

$$m_i = C_i^d \bmod N = C_i^{11} \bmod 21.$$

В результате получаем:

$$m_1 = 1^{11} \bmod 21 \equiv ((1));$$

$$m_2 = 11^{11} \bmod 21 \equiv ((2));$$

$$m_3 = 12^{11} \bmod 21 \equiv ((3)).$$

Исходное сообщение: РУХ.

### *Упражнения*

6.1. Пусть  $\alpha = 2$ ,  $M = 9$ . Найти порядок элемента  $\alpha$ .

6.2. Пусть  $\alpha = 4$ ,  $M = 17$ . Найти порядок элемента  $\alpha$ .

6.3. Пусть  $\alpha = 7$ ,  $M = 17$ . Найти  $7^{-1}$ .

6.4. Пусть  $\alpha = 10$ ,  $M = 23$ . Найти  $10^{-1}$ .

6.5. Создайте RSA-криптосистему, используя  $N = 2 \cdot 13$ . Вычислить публичный ключ  $(N, E)$  и секретный ключ  $d$ . Используйте эти ключи, для кодирования сообщения «МИНСК».

6.6. Создайте RSA-криптосистему, используя  $N = 3 \cdot 17$ . Вычислить публичный ключ  $(N, E)$  и секретный ключ  $(d, N)$ . Используйте эти ключи, для кодирования сообщения «ИТ»

## ЧАСТЬ 2. ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

### 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

#### 1.1. Основная теорема Шеннона кодирования для канала с шумом (вторая теорема Шеннона)

**Теорема 1.1.** Пусть  $C$  – пропускная способность дискретного канала без памяти, источник характеризуется энтропией  $H$ . Если  $H < C$ , тогда для любого  $\varepsilon > 0$ , существует метод кодирования информации  $[n, k, d]$ - двоичным кодом, вероятность ошибки декодирования которого  $P_{ош} < \varepsilon$ .

Теорема Шеннона состоит из двух утверждений.

*Прямое утверждение.* При любой производительности источника сообщений, меньшей, чем пропускная способность канала, существует такой способ кодирования, который обеспечивает передачу информации со сколь угодно малой вероятностью ошибки.

*Замечания:*

1. Под производительностью источника сообщений понимают количество информации, вырабатываемой источником в единицу времени.

2. Пропускная способность канала – это предельная скорость передачи информации, при которой еще возможна передача со сколь угодно малой вероятностью ошибки.

3. Пропускная способность  $C$  канала – это максимальная взаимная информация  $I(X; Y)_{\max}$ , которая может быть достигнута в канале с матрицей  $\mathbf{P}$  переходных вероятностей (см. *Опр. 5.5*).

*Обратное утверждение.* Не существует способа кодирования, позволяющего вести передачу информации со сколь угодно малой вероятностью ошибки, если производительность источника сообщений больше пропускной способности канала.

Теорема с одной стороны – фундаментальна, а с другой стороны – некон-

структивна.

Фундаментальность – устанавливается теоретический предел эффективности системы при достоверной передаче информации. При этом:

- помехи в канале не ограничивают точность (достоверность) передачи;
- помехи ограничивают скорость передачи информации, при которой может быть достигнута сколь угодно высокая достоверность передачи;
- при любой конечной скорости передачи информации, вплоть до пропускной способности, сколь угодно малая вероятность ошибки достигается лишь при увеличении длительности кодовых последовательностей и использовании большого ансамбля кодовых слов.

Неконструктивность теоремы заключается в том, что в ней не затрагиваются пути построения кодов, методы обработки, обеспечивающие безошибочную передачу информации, а лишь утверждается их существование.

## **1.2. Возможность исправления ошибок помехоустойчивым кодом**

### **1.2.1. Параметры кодов**

*Определение 1.1.* Код – это множество дискретных последовательностей, разрешенное для передачи сообщений.

Коды характеризуются следующими параметрами.

1. Размерность  $q$  кодового алфавита – число различных элементов алфавита, выбранное для построения кода.

В качестве кодового алфавита могут использоваться символы двоичного  $\{0,1\}$  или бинарного алфавита  $\{1,-1\}$ . Например, слово  $x = (x_1 x_2 \dots x_7) = (-1 - 1 - 1 1 1 - 1 1)$  представлено символами бинарного алфавита источника,  $q = 2$ .

*Определение 1.2.* Длина  $n$  кода (значность) – число символов кодового слова. Последовательности символов называются кодовыми словами или кодовыми векторами.

Параметр  $n$  определяет следующие особенности кодов. Коды бывают:

- равномерные (блоковые),  $n = \text{const}$ ;
- неравномерные,  $n = \text{var}$ ;
- бесконечные,  $n = \infty$ .

*Определение 1.3.* Размерность  $k$  кода – число информационных элементов (позиций) кодового слова.

*Определение 1.4.* Мощность  $M = q^k$  кода – это число различных кодовых последовательностей (комбинаций), разрешенных для кодирования.

Различают  $M_{\max} = q^n$  максимальное число кодовых слов при заданных  $q$  и  $n$ . Например, для  $q = 3, n = 6$  имеем  $M_{\max} = q^n = 729$  слов. Код, у которого используются все комбинации, т. е. его мощность  $M_{\max}$ , называется полным (безизбыточным). Для него  $k = n$ .

Если код определяется числом  $M_{\max} > M$ , то код называется избыточным.

*Пример 1.1.* Код с параметрами:  $q = 2, n = 5, M = 4$  является избыточным. Кодовые слова записаны в виде матрицы

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

*Определение 1.5.* Число проверочных (избыточных)  $r = (n - k)$  позиций кодового слова.

Пусть  $n = 7, q = 2, k = 4$ . Тогда на длине слова из семи символов три избыточных.

*Определение 1.6.* Скорость  $R = \frac{k}{n}$  передачи кода. Для приведенного выше примера  $R = \frac{4}{7}$ .

*Определение 1.7.* Кратность ошибки  $t$ . Параметр  $t$  указывает, что все конфигурации с  $t$  или менее ошибками в любом кодовом слове могут быть исправлены и (или) обнаружены.

Предположим, что по каналу передается или хранится в памяти двоичный вектор  $\mathbf{x} = (x_0 x_1 \dots x_{n-1})$ , а принимается, или считывается из памяти вектор  $\mathbf{y} = (y_0 y_1 \dots y_{n-1})$ . Тогда вектор  $\mathbf{y} - \mathbf{x} = \mathbf{e} = (e_0 e_1 \dots e_{n-1})$  называется вектором ошибок, где  $e_i \in \{0, 1\}$ . Если ошибок не произошло, то все  $e_i = 0$ . Вектор ошибок указывает место и значение ошибок.

*Определение 1.8.* Расстояние Хэмминга  $d_x$  между двумя векторами (степень удаленности любых кодовых последовательностей друг от друга). Если  $\mathbf{x} = (x_0 x_1 \dots x_{n-1})$  и  $\mathbf{y} = (y_0 y_1 \dots y_{n-1})$  кодовые векторы, то расстояние Хэмминга равно числу позиций, в которых они различаются.

Расстояние Хэмминга может обозначаться и как  $\text{dist}(\mathbf{x}, \mathbf{y})$ . Например,

$$\text{dist}((a b b c b), (c b c a a)) = 4,$$

$$\text{dist}((0 1 2 2), (2 2 1 2)) = 3.$$

*Замечание.* С позиции теории кодирования  $d_x$  показывает, сколько символов в слове надо исказить, чтобы перевести одно кодовое слово в другое.

*Определение 1.9.* Наименьшее значение расстояния Хэмминга для всех

пар кодовых последовательностей кода  $\mathbf{G}$  называют кодовым расстоянием  $d$  (минимальное расстояние кода),

$$d = \min \{\text{dist}(\mathbf{x}, \mathbf{y})\}, \text{ где } \mathbf{x} \in \mathbf{G}, \mathbf{y} \in \mathbf{G}, \mathbf{x} \neq \mathbf{y}.$$

Кодовое расстояние  $d$  характеризует корректирующую способность кода  $t = f(d)$ .

*Определение 1.10.* Код значностью  $n$ , размерностью  $k$  и расстоянием  $d$  называется  $[n, k, d]$ -кодом.

*Пример 1.2.* Можно построить следующий  $[5, 2, 2]$ -код:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Данный код можно использовать для кодирования 2-х битовых двоичных чисел, используя следующее (произвольное) соответствие:

$$00 \leftrightarrow 00000;$$

$$10 \leftrightarrow 10110;$$

$$01 \leftrightarrow 01010;$$

$$11 \leftrightarrow 11100.$$

Найдем кодовое расстояние кода:

$$\text{dist}((10110), (01010)) = 3;$$

$$\text{dist}((10110), (11100)) = 2;$$

$$\text{dist}((01010), (11100)) = 2.$$

Для этого кода  $d = \min\{\text{dist}(\mathbf{x}, \mathbf{y})\} = 2$ .

*Определение 1.11.* Вес Хэмминга вектора  $\mathbf{x} = (x_0 x_1 \dots x_{n-1})$  равен числу ненулевых позиций этого вектора; обозначается  $\text{wt}(\mathbf{x})$ .

Например,  $\text{wt}(1230430) = 5$ . Используя определение веса Хэмминга, получим очевидное выражение

$$\text{dist}(\mathbf{x}, \mathbf{y}) = \text{wt}(\mathbf{x} - \mathbf{y}). \quad (1.1)$$

*Пример 1.3.* Найти  $d_x$  векторов  $\mathbf{x} = (1202)^T$  и  $\mathbf{y} = (2012)^T$ .

Решение.  $\text{dist}(\mathbf{x}, \mathbf{y}) = \text{wt}\left(\begin{bmatrix} 1 \\ 2 \\ 0 \\ 2 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \\ 1 \\ 2 \end{bmatrix}\right) = \text{wt}\begin{bmatrix} 2 \\ 2 \\ 2 \\ 0 \end{bmatrix} = 3.$

Из выражения (1.1) следует, что минимальное расстояние Хэмминга равно

$$d = \min\{\text{dist}(\mathbf{x}, \mathbf{y})\} = \min\{\text{wt}(\mathbf{x} - \mathbf{y}), \text{ где } \mathbf{x} \in \mathbf{G}, \mathbf{y} \in \mathbf{G}, \mathbf{x} \neq \mathbf{y}.$$

*Замечание.* Для нахождения минимального расстояния линейного кода не обязательно сравнивать все возможные пары кодовых слов. Если  $\mathbf{x}$  и  $\mathbf{y}$  принадлежат линейному коду  $\mathbf{G}$ , то  $\mathbf{x} - \mathbf{y} = \mathbf{u}$  также является кодовым словом кода  $\mathbf{G}$ . Такой код является аддитивной группой (определена операция сложения), следовательно,

$$d = \min\{\text{dist}(\mathbf{x}, \mathbf{y})\} = \min \text{wt}(\mathbf{u}),$$

где  $\mathbf{u} \in \mathbf{G}$ ,  $\mathbf{u} \neq 0$ .

**Теорема 1.2.** Минимальное расстояние линейного кода равно минимальному весу ненулевых кодовых слов.

Так как  $t = f(d)$ , то возникает вопрос о величине  $d$ , такой, чтобы код обеспечивал контроль ошибок, т. е. обнаружение и исправление ошибок.

### 1.2.2. Блочные коды

У блочных кодов поток данных разделяется на блоки по  $k$  информационных символов, и далее они кодируются  $n$ -символьными кодовыми словами. На рис. 1.1 показана структура кодирования блочным кодом.

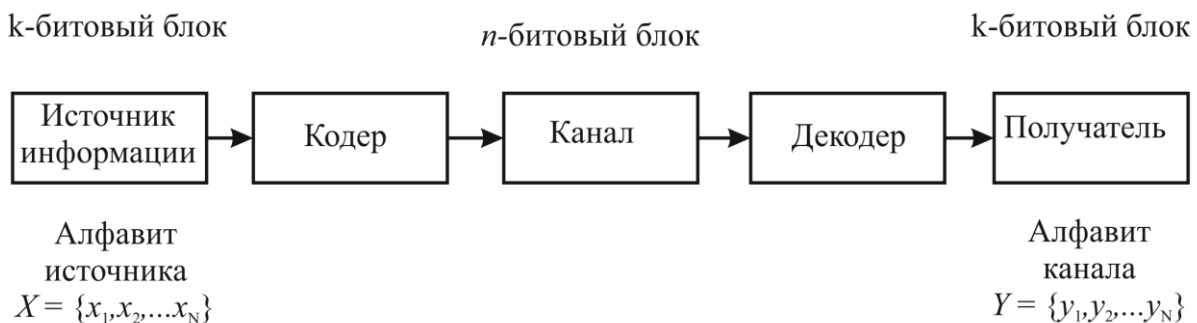


Рис 1.1. Блочное кодирование

### 1.2.3. Ошибки, их разновидности

Ошибки бывают:

- случайные (независимые);
- зависимые.

Случайные ошибки возникают независимо друг от друга в передаваемом сообщении. При этом говорят о кратности ошибок  $t$

Зависимые ошибки делятся на:

- пакетные;
- модульные.

Пакет ошибок описывается длиной пакета  $p$ . Он может находиться в произвольном месте потока передаваемых (записываемых) символов. Говорят, граница (фаза) пакета неизвестна.

Например, в ЗУ длиной 16 следует записать информацию

0001 0110 1111 1010.

Произошла пакетная ошибка длиной  $p = 4$ . В ЗУ запишется информация

0001 0101 0011 1010.

Кратность пакета ошибок обозначается через  $q$ . Пакет ошибок описывается вектором ошибки  $\mathbf{e} = (e_0 e_1 \dots e_{p-1})$ . Для двоичных кодов  $e_i = \{0, 1\}$ . В приведенном примере  $\mathbf{e} = (1111)$ ,  $q = 1$ .

Модульная ошибка – это частный случай пакетной ошибки. В литературе по кодированию ее называют и байтовой ошибкой. Модуль ошибок – это фазированный пакет ошибок. В этом случае граница (фаза) пакета известна. Длина модуля ошибок и кратность модуля ошибок обозначаются соответственно  $b$  и  $q$ . Число возможных конфигураций векторов ошибок равно  $2^b - 1$ . Для  $b = 4$  конфигурации ошибок образуют следующее множество векторов ошибок:

$$\mathbf{e} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Пусть двоичная информация

0001 0110 1111 1010

записывается в ЗУ с модульной ошибкой. Вектор  $\mathbf{e} = (1010)$ ,  $q = 1$ . Тогда на выходе ЗУ поток представляется в виде

1011 0110 1111 1010.

В реальных системах модульные и пакетные ошибки встречаются чаще, чем отдельно расположенные ошибки. Зависимые ошибки могут вызываться источником периодического шума, например, расположенным поблизости радиолокатором или каким-то вращающимся электромеханизмом, замираниями в

линии связи и пр. В этом случае необходимо применение кодов, исправляющих пакетные и модульные ошибки. Заметим, что при наличии таких ошибок может оказаться более правильным использовать процедуру перемежения порядка символов в закодированной последовательности перед передачей, и восстановления исходного порядка символов после приема с тем, чтобы рандомизировать ошибки, объединенные в пакеты.

#### 1.2.4. Контроль ошибок

Кодовое слово можно представить в виде вектора с координатами в  $n$ -мерном векторном пространстве. Например, для  $n = 3$  вектор  $\mathbf{x} = (x_0 x_1 x_2)$  находится в трёхмерном евклидовом пространстве, рис. 1.2. Разрешенными словами для передачи выбраны векторы  $\mathbf{x}_1 = (0\ 0\ 0)$  и  $\mathbf{x}_2 = (1\ 1\ 1)$ .

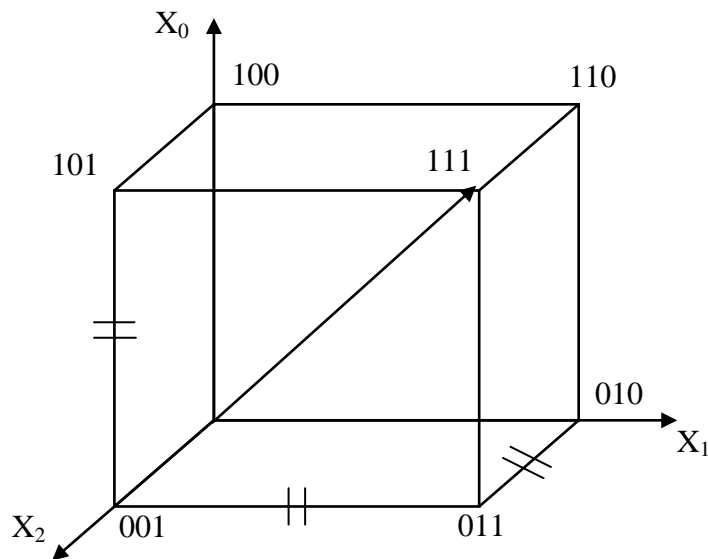


Рис. 1.2. Трёхмерный куб

Рис. 1.2 наглядно дает алгебраическую интерпретацию понятия «мощность кода».

1. Кодовые слова полного кода определяют  $n$ -мерное пространство, состоящее из  $q^n$  последовательностей. При  $q = 2$  трёхмерное пространство включает восемь последовательностей полного двоичного кода.

2. Кодовые слова избыточного кода определяют подпространство (подмножество)  $n$ -мерного пространства. При  $q = 2$  трёхмерное подпространство включает  $q^k$  последовательностей двоичного кода.

Под воздействием помех происходит искажение отдельных разрядов вектора разрешенного подмножества. В результате разрешённые для передачи кодовые векторы переходят в другие векторы (с иными координатами) – запрещённые. Факт перехода разрешённого слова в запрещённое слово можно использовать для контроля над ошибками.

Возможна ситуация, когда разрешённый вектор переходит в другой раз-



решённый кодовый вектор:  $(0\ 0\ 0) \Leftrightarrow (1\ 1\ 1)$ . В этом случае ошибки не обнаруживаются, и контроль становится неэффективным.

Из рассмотренной модели можно сделать следующий очевидный, но важный вывод. Для того, чтобы передаваемые векторы можно было бы отличать друг от друга при наличии помех, необходимо располагать эти векторы в  $n$ -мерном пространстве как можно дальше друг от друга.

Из этой же модели следует геометрическая интерпретация расстояния Хэмминга  $d_x$  – это число рёбер, которые нужно пройти, чтобы перевести один вектор в другой, т. е. попасть из вершины одного вектора в вершину другого.

## 1.2.5. Кодовое расстояние кода и его связь с корректирующей способностью

### 1.2.5.1. Обнаружение ошибок

Стратегия обнаружения ошибок в переданном слове заключается в следующем. Можно обнаружить ошибку, если установить, что переданным словом было ближайшее по расстоянию Хэмминга к принятому слову. Покажем применение этого утверждения.

*Пример 1.3.* Пусть  $n = 3, q = 2$ . Разрешенным для передачи является множество кодовых слов:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Очевидно, что код  $\mathbf{G}$  имеет  $d = 1$ . Любая одиночная ошибка трансформирует слово кода в другое разрешенное для передачи слово. Это случай, когда выбранное для передачи информации множество слов не обладает корректирующей способностью.

*Пример 1.4.* Пусть теперь подмножество  $\mathbf{G}$  разрешённых для передачи кодовых слов представлено в виде двоичных векторов с чётным весом (чётным числом единиц).

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Заданный код  $\mathbf{G}$  имеет  $d = 2$ . Запрещенные кодовые слова представлены в виде подмножества

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Если  $d = 2$ , то ни одно из разрешенных кодовых слов (т. е. кода  $G$ ) при одиночной ошибке не переходит в другое разрешённое слово этого же кода.

Таким образом, код  $G$  обнаруживает:

- одиночные ошибки;
- ошибки нечетной кратности (для  $n = 3$  – тройные).

Например, тройная ошибка кодового слова  $x = (110)$ ;  $e = (111)$ , переводит его в запрещенный вектор  $y = x \oplus e = (001)$ .

Рассмотрим процесс декодирования в режиме обнаружения.

Допустим, передавалось кодовое слово  $x = (101)$ . Возникла одиночная ошибка, вектор которой  $e = (100)$ . На вход декодера поступил вектор  $y = x \oplus e = (001)$ . Данный вектор принадлежит запрещенному подмножеству  $A$ . Следовательно, декодер устанавливает, что при передаче информации произошла ошибка. Таким образом, осуществляется контроль ошибок.

Еще раз рассмотрим утверждение: переданным словом было то, которое является ближайшим по расстоянию Хэмминга к принятому. Вычислим расстояния Хэмминга для всех слов разрешенного подмножества и входным вектором.

$$\begin{aligned} \text{dist}_1(000, 001) &= 1, \\ \text{dist}_2(101, 001) &= 1, \\ \text{dist}_3(011, 001) &= 1, \\ \text{dist}_4(110, 001) &= 3. \end{aligned}$$

Наиболее вероятно, что передаваемыми могли быть следующие слова :

$$x_1 = (000), x_2 = (101), x_3 = (011).$$

*Вывод.* В общем случае, при необходимости обнаруживать ошибки кратностью  $t$  кодовое расстояние кода должно удовлетворять выражению

$$d \geq t + 1. \quad (1.2)$$

### 1.2.5.2. Исправление ошибок

*Пример 1.5.* Пусть  $n = 3$ ;  $q = 2$ ; код  $G$  задан векторами  $x_1 = (000)$  и  $x_2 = (111)$ . При возникновении одиночных ошибок или множества векторов

$$e = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

кодovому слову  $x_1 = (000)$  соответствует следующее запрещенное подмножест-

во:

$$A = \{x_1 \oplus e\} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Кодовому слову  $x_2 = (111)$  соответствует следующее запрещенное подмножество

$$L = \{x_2 \oplus e\} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Таким образом, коду  $G$ , разрешенному для передачи подмножеств векторов  $x_1, x_2$ , соответствуют два запрещенных подмножества векторов  $\{A\}$  и  $\{L\}$ :

$$G = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \xrightarrow{\begin{matrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = A \\ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = L \end{matrix}}.$$

Стратегия исправления ошибок заключается в следующем:

- каждая из одиночных ошибок приводит к запрещенному кодовому слову того или иного запрещенного подмножества ( $L$  и  $A$ );
- структура кодового запрещенного подмножества, относящаяся к соответствующему исходному разрешенному подмножеству, позволяет определить местоположение ошибки, т. е. исправить ошибку.

Допустим, передавалось кодовое слово  $x_2 = (111)$ . Возникла одиночная ошибка, вектор которой  $e = (100)$ . На вход декодера поступил вектор

$$y = x \oplus e = (011).$$

Вычислим расстояния Хэмминга для всех слов разрешенного подмножества и входным вектором:

$$\begin{aligned} \text{dist}_1(000, 011) &= 2, \\ \text{dist}_2(111, 011) &= 1. \end{aligned}$$

Наиболее вероятно, что передавалось слово

$$x_2 = (111).$$

Для исправления ошибок кратностью  $t$  кодовое расстояние должно удовлетворять соотношению

$$d \geq 2t + 1. \quad (1.3)$$

Используя эту формулу, можно записать

$$t = \lceil (d-1)/2 \rceil,$$

где  $\lceil l \rceil$  обозначает целую часть числа  $l$ .

## 2. АЛГЕБРАИЧЕСКИЕ СТРУКТУРЫ

Основой построения наиболее важных известных кодов, корректирующих ошибки, является их алгебраическая структура. Существование особых структурных закономерностей в строении таких кодов желательно по двум причинам:

- они облегчают изучение свойств кода;
- обеспечивают возможность практической реализации кодов на аппаратно – программном уровне.

Теория групп, колец, полей – это те понятия конечной алгебры, которые являются основой для поиска хороших кодов, контролирующих ошибки. Эти же алгебраические структуры используются при технической реализации методов криптографической защиты информации, алгоритмов криптоанализа. Математические основы названных понятий широко применяются в алгоритмах дискретных преобразований цифровой обработки сигналов и изображений.

### 2.1. Группы

#### 2.1.1. Арифметика конечных групп

*Определение 2.1.* Группа – это алгебраическая структура или множество элементов  $G$  с заданной на нем основной алгебраической операцией.

*Замечания:*

1. В большей части это те же операции, которые применимы к числовым системам;
2. Для теории кодирования наибольшее значение имеют бинарные операции.

Элементы множества произвольной природы обозначают как  $G = \{a, b, c\}$ , а результат операции символически записывают в виде  $c = f(a, b)$ . Говорят: на множестве  $G$  задана бинарная операция  $f$ . Чтобы подчеркнуть абстрактность операции применяют символы  $*$  или  $\circ$ . При операции сложения пишут  $c = a + b$ , а при операции умножения  $c = a \cdot b$ .

*Замечание.* Операции вычитания и деления это неосновные операции, так как они являются обратными для сложения и умножения.

Алгебраические операции могут задаваться таблицами Кэли. Первые строка и столбец таблицы состоят из элементов множества  $G$ . Результат операции записывается в ячейке таблицы с координатами, задаваемыми элементами

множества. Например, для множества элементов  $G = \{a, b\}$  операцию можно представить в виде таблицы

$f$	$a$	$b$
$a$	$a$	$b$
$b$	$b$	$a$

В группе должны выполняться следующие аксиомы:

- 1) замкнутость – любой паре  $a, b$  элементов из множества  $G$  ставится в однозначное соответствие третий элемент  $c \in G$ , т. е.  $a*b = c$ ;
- 2) ассоциативность – для всех элементов множества  $G = \{a, b, c\}$  выполняется

$$a*(b*c) = (a*b)*c = a*b*c;$$

- 3) существование во множестве  $G$  нейтрального элемента  $e$ , такого, что выполняется

$$a*e = e*a = a \in G;$$

- 4) существование для каждого элемента  $a \in G$  обратного  $\hat{a}$  или  $a^{-1} \in G$ , такого, что  $a * \hat{a} = \hat{a} * a = e$  или  $a * a^{-1} = a^{-1} * a = e$ ;

- 5) если в группе выполняется аксиома коммутативности, т. е. для любых  $a$  и  $b$  из группы  $a*b = b*a$ , то группа называется коммутативной или абелевой (N. Abel (1802 – 1829), норвежский математик).

**Определение 2.2.** Если группа содержит конечное число элементов, то она называется конечной группой, а число элементов в группе называется порядком группы. Если порядок группы бесконечен, то она называется бесконечной.

*Замечания.*

1. Аддитивная абелева группа в качестве нейтрального элемента  $e$  имеет  $0$ ,  $a + \hat{a} = e = 0$ ;
2. Мультипликативная абелева группа в качестве нейтрального элемента  $e$  имеет  $1$ ,  $a \cdot \hat{a} = e = 1$ .

**Теорема 2.1.** Единичный элемент в группе является единственным. Для каждого элемента группы обратный элемент также является единственным.

**Примеры абелевых групп**

- 2.1. Рациональные числа относительно операции сложения  $\langle \mathbb{R}; + \rangle$ .
- 2.2. Натуральные числа относительно операции умножения  $\langle \mathbb{N}; \cdot \rangle$ .
- 2.3. Совокупность действительных чисел относительно операции сложения  $\langle \mathbb{R}; + \rangle$ .
- 2.4. Группа из одного элемента (единичного)  $\langle e; * \rangle$ , например,  $\langle 0; + \rangle$  – аддитивная,  $\langle 1; \cdot \rangle$  – мультипликативная.
- 2.5. Группа второго порядка. Ее таблица Кэли имеет вид

$f$	$e$	$a$
-----	-----	-----

$e$	$e$	$b$
$a$	$b$	$e$

Здесь  $a * \hat{a} = e$ . Выражение  $(a * a)$  не может быть равным  $a$ , так как в группе может быть только один нейтральный элемент.

На множестве из двух элементов групповая операция задается не более чем одним способом. Например, аддитивная абелева группа второго порядка  $\langle \{0,1\}; +; 0 \rangle$  задается таблицей Кэли

+	0	1
0	0	1
1	1	0

Для элемента 0 обратным является 0; для элемента 1 обратным является 1, так как в группе может быть только один нейтральный элемент, который получается из определения  $a * \hat{a} = e$ .

Мультипликативная абелева группа второго порядка  $\langle \{1, -1\}; \cdot; 1 \rangle$  задается таблицей Кэли

$\cdot$	1	-1
1	1	-1
-1	-1	1

Для элемента 1 обратным является 1; для элемента  $(-1)$  обратным является  $(-1)$ , так как в группе может быть только один нейтральный элемент, который получается из определения  $a * \hat{a} = e$ .

2.6. На множестве из трех элементов групповая операция задается не более чем одним способом. Например, группа третьего порядка  $\langle \{e, a, b\}; +; e \rangle$  задается таблицей Кэли

Таблица 2.1

+	$e$	$a$	$b$
$e$	$e$	$a$	$b$
$a$	$a$	$b$	$e$
$b$	$b$	$e$	$a$

Табл. 2.1 обладает следующими свойствами:

- 1) из однозначности разрешимости уравнения  $a * x = e$  (у каждого элемента  $a$  из множества  $G$  существует единственный обратный элемент) следует, что в каждой строке (в каждом столбце) содержатся все элементы группы по одному разу;
- 2) строки и столбцы ее являются перестановками последовательности  $(e, a, b)$ ;
- 3) все строки (столбцы) различны.

Другой пример группы  $\langle \{0, 1, 2\}; +; 0 \rangle$  на множестве из трех элементов. Операция в группе задается таблицей Кэли

Таблица 2.2

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Обратные элементы

этой группы:

- для 1 обратным является элемент 2, т. к.  $1 + \hat{1} = 1 + 2 = 0$ ;
- для 2 обратным является элемент 1, т. к.  $2 + \hat{2} = 2 + 1 = 0$ .

*Замечание.* С точки зрения понятий конечной алгебры, рассмотренные группы (табл. 2.1 и табл. 2.2) следует считать однотипными. Этот факт приводит к очень важному понятию алгебры – изоморфизма. Из этого понятия строятся изоморфные коды.

2.7. Совокупность двоичных  $n$ -разрядных чисел с операцией сложения по модулю 2 ( $\text{mod } 2$ ) образует группу. Например, в группе

$$\langle \{a = (1110010), b = (1100101)\}; \oplus; 0 \rangle$$

операция

$$a \oplus b = (0010111).$$

В такой группе  $e = (0000000)$ . Обратный элемент равен самому себе так как  $a \oplus \hat{a} = e = (0000000)$ . Для  $a = (1110010)$  обратный элемент  $\hat{a} = (1110010)$ .

2.8. Пример неабелевой группы. Совокупность невырожденных матриц порядка  $n$  относительно матричного умножения  $A \cdot B \neq B \cdot A$ .

*Замечание.* Существует матрица порядка  $n$ , которая коммутирует с любой матрицей такого же порядка. При умножении на нее действие аналогично умножению на нейтральный элемент (единичный элемент), т. е.  $A \cdot I = I \cdot A = A$ . Это будет иметь место в том случае, когда  $I$  будет единичной матрицей. Например, для  $n = 4$  матрица  $I$  имеет вид

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

### 2.1.2. Особые свойства мультипликативной группы

Для мультипликативной группы  $\langle G; \cdot \rangle$  вводится понятие натуральной степени элемента группы

$$\alpha^0, \alpha, \alpha\alpha, \dots, \alpha\alpha\alpha = \alpha^0, \alpha, \alpha^2, \dots, \alpha^n.$$

где  $n$  – ноль или  $n \in \mathbb{N}$ .

**Определение 2.3.** Группа называется циклической, порожденной элементом  $\alpha$ , если каждый элемент группы есть некоторая степень  $\alpha^j$ .

**Теорема 2.2.** Пусть  $\langle G; \cdot \rangle$  – группа и элемент  $\alpha \in G$  такой, что  $\alpha^s = 1$  для некоторого целого  $s$ . Если  $N$  – наименьшее положительное целое число такое, что  $\alpha^N = 1$ , то  $N|s$  ( $N$  делит  $s$ ). Целое число  $N$  называется порядком  $\alpha$ .

**Пример 2.9.** Задана мультипликативная группа  $\langle \{1, -1, j, -j\}; \cdot; 1 \rangle$ . Каков порядок элемента  $j$ ? Каков порядок  $(-1)$ ? Какой элемент может быть использован в качестве  $\alpha$ ?

**Решение.** Находим степени элемента  $j$ :  $j^1 = j, j^2 = -1, j^3 = -j, j^4 = 1$ . Порядок элемента  $j$  равен 4. Элемент  $(-1)$  имеет порядок равный 2, т. к.  $(-1)^1 = -1, (-1)^2 = 1$ . Порядок элемента  $(-j)$  также равен 4. Данная мультипликативная группа является циклической, порожденная элементом  $j$ .

**Определение 2.4.** Группа называется бесконечной, если все натуральные степени порождающего элемента  $\alpha$  различны, т. е.  $\alpha^n \neq \alpha^m$  при  $m \neq n$ . В этом случае говорят, что элемент  $\alpha$  имеет бесконечный порядок.

**Определение 2.5.** Циклическая группа конечна и имеет порядок  $N$ , если для  $N$  выполняется равенство  $\alpha^N = 1$  и  $N$  – наименьшее положительное число с таким свойством. Обозначение циклической группы

$$G = \langle \{\alpha^0 = 1, \alpha^1, \alpha^2, \dots, \alpha^N\}; \cdot \rangle,$$

где  $N$  – порядок элемента  $\alpha$  (число элементов группы).

**Определение 2.6.** Натуральное число  $p > 1$  называется простым, если оно делится только на 1 и на себя. Натуральное число, большее 1, называется составным, если оно не является простым.

**Замечание.** По определению целое число 1 не является ни простым, ни составным. Число 2 – единственное четное простое число.

**Теорема 2.3.** Всякая группа простого порядка  $p$  ( $p$  – простое число) является циклической. Всякая циклическая группа является абелевой, при этом

$$\alpha^m \alpha^n = \alpha^{m+n}; (\alpha^m)^n = \alpha^{mn}.$$

**Определение 2.7.** Если все элементы циклической группы  $G$  могут быть представлены в виде натуральных степеней некоторого элемента  $\alpha \in G$ , то такой элемент называется примитивным.

В примере 2.9 элемент  $j$  – примитивный.



**Определение 2.8.** Целые числа  $a$  и  $b$  называются взаимно-простыми, если наибольший общий делитель (НОД)  $d = (a, b) = 1$ .

**Теорема 2.4.** (Эйлера, (Леонард Эйлер (1707–1783), швейц. математик). Если  $\alpha$  и  $M$  взаимно-простые числа, то

$$\alpha^{\varphi(M)} \equiv 1 \pmod{M},$$

где  $\varphi(M)$  – функция Эйлера, равная количеству всех натуральных чисел меньших  $M$  и взаимно-простых с  $M$ .

*Примеры*

2.10. Пусть задано множество целых чисел  $G = \{1, 2, 3, 4, 5, 6\}$ . Найти  $\varphi(7)$ . Решение.  $\varphi(7) = 6$

2.11. Задано множество целых чисел  $G = \{1, 2, 3, 4, 5, 6, 7, 8\}$ . Найти  $\varphi(9)$ . Решение.  $\varphi(9) = 6$ .

2.12. Пусть задано множество целых чисел  $G = \{1, 2, 3, 4\}$ ,  $\varphi(5) = 4$ . Числа 2 и 5 являются взаимно-простыми так как

$$2^4 \equiv 1 \pmod{5}.$$

Если в циклической группе выполняется условие

$$\varphi(M) = N,$$

то для примитивного элемента циклической группы  $\alpha$  справедливо

$$\alpha^{\varphi(M)} \equiv 1 \pmod{M}, \quad \alpha^N \equiv 1. \quad (2.1)$$

Но  $\alpha$  не единственный примитивный элемент циклической группы. Примитивным всегда является элемент  $\alpha^{N-1}$ .

Доказательство. Если  $\varphi(M) = N$ , из (2.1) запишем

$$\alpha^N \equiv 1 \pmod{M}. \quad (2.2)$$

Умножим выражение (2.2) на элемент обратный примитивному  $\alpha$

$$\alpha^N \alpha^{-1} \equiv 1 \alpha^{-1} \pmod{M}.$$

Преобразуя последнее выражение, получим:

$$\frac{\alpha^{N-1}}{\alpha^{-1}} \equiv 1 \pmod{M};$$

$$\alpha^{N-1} \alpha \equiv 1 \pmod{M}.$$

Из определения обратного элемента мультипликативной группы ( $a * \hat{a} = e$ ) следует, что примитивным является элемент  $\hat{\alpha} = \alpha^{-1} = \alpha^{N-1}$ , обратный  $\alpha$ .

*Замечание.* Для циклической мультипликативной группы, числовых полей и колец обратные элементы можно найти:

- 1) перебором;
- 2) по теореме Эйлера;
- 3) по алгоритму Евклида.

Количество примитивных элементов определяется функцией Эйлера  $\varphi(M-1)$ .

*Пример 2.13.* Задана мультипликативная абелева группа

$$G = \langle \{a_0 = 1, a_1 = 2, a_2 = 3, a_3 = 4\}; ; 1 \rangle.$$

Построить возможные циклические группы. Определить число примитивных элементов.

*Решение.* Находим функцию Эйлера  $\varphi(5) = 4$ . Число примитивных элементов равно  $\varphi(4) = 2$ . Примитивным является элемент

$$a_1 = 2, \text{ т. к. } 2^4 = 16 \equiv 1 \pmod{5}.$$

Циклическую группу образует множество  $\{2^0, 2^1, 2^2, 2^3\}$ , где

$$2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = ((3)).$$

Примитивным будет также элемент

$$a_1^{N-1} = 2^3 = ((3)) = a_2.$$

Циклическую группу образует множество  $\{3^0, 3^1, 3^2, 3^3\}$ , где

$$3^0 = 1, 3^1 = 3, 3^2 = ((4))4, 3^3 = ((2)).$$

*Замечание.* Элемент  $a_3 = 4$  не является примитивным элементом заданной мультипликативной группы, так как имеет порядок 2,

$$\{4^0 = 1, 4^1 = 4, 4^2 \equiv ((1))\}.$$

### 2.1.3. Теорема Лагранжа

В группе множество  $G$  всех элементов по операции сложения образует циклическую аддитивную группу  $\langle \{G\}; +; 0 \rangle$ , а элемент 1 порождает множество вида  $\{1; 1 + 1; 1 + 1 + 1; \dots\}$ . Так как число элементов в группе конечно, то в этом ряду найдется сумма  $p$  единиц, равная нулевому элементу группы.

*Определение 2.9.* Порядок каждого элемента аддитивной группы  $a_j$  определяется по числу этих элементов в последовательности вида  $a, a + a, a + a + a, \dots, a + a + \dots = 0 = e$

**Пример 2.14.** Определить порядок каждого элемента аддитивной группы  $G = \langle \{0, 1, 2, 3, 4, 5\}; +; 0 \rangle$  порядка 6, задаваемой таблицей Кэли

Таблица 2.3

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

**Решение.** Используя *опр. 2.9*, представим все элементы группы как:

- 1) 0:  $0 = e$ , следовательно, порядок элемента 0 равен 1;
- 2) 1:  $1 + 1 + 1 + 1 + 1 + 1 = 6 \equiv 0 \pmod{6}$ , следовательно, порядок элемента 1 равен 6;
- 3) 2:  $2 + 2 + 2 = 6 \equiv 0 \pmod{6}$ , порядок элемента 2 равен 3;
- 4) 3:  $3 + 3 = 6 \equiv 0 \pmod{6}$ , порядок элемента 3 равен 2;
- 5) 4:  $4 + 4 + 4 = 12 \equiv 0 \pmod{6}$ , порядок элемента 4 равен 3;
- 6) 5:  $5 + 5 + 5 + 5 + 5 + 5 = 30 \equiv 0 \pmod{6}$ , порядок элемента 5 равен 6.

**Теорема 2.5.** (Лагранжа), (Ж. Л. Лагранж (1736 – 1813), франц. математик). Порядок любого элемента  $a_j$  произвольной конечной группы, а не только циклической является делителем порядка  $N$  группы (числа элементов группы).

*Следствие теоремы 2.5.* Порядок каждого элемента конечной подгруппы является делителем порядка группы.

#### 2.1.4. Подгруппа

**Определение 2.10.** Пусть  $\langle G; * \rangle$  – группа, а  $H \subseteq G$  – подмножество, являющееся группой относительно той же групповой операции. Тогда  $H$  называется подгруппой группы  $G$ .

Подгруппа  $H$  всегда содержит нейтральный элемент группы. Для того, чтобы определить, является ли  $H$  подгруппой, достаточно проверить выполнение аксиом замкнутости и существования обратных элементов.

*Примеры подгрупп*

2.15. Множество  $E$  всех четных чисел является подмножеством целых чисел  $\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$ . Тогда  $\langle E; \pm \rangle$  – подгруппа группы  $\langle \mathbb{Z}; \pm \rangle$ .

2.16. Множество  $\{0, 1, 2\}$  принадлежит подмножеству целых чисел  $\mathbb{Z}$ , но группа, задаваемая табл. 2.2, не является подгруппой группы  $\langle \mathbb{Z}; + \rangle$ , т. к. они определяются разными операциями.

2.17. Пусть  $G = \langle \{0, 1, 2, 3, 4\}; +; 0 \rangle$  – группа порядка 4, задаваемая табл. 2.4.

Таблица 2.4

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Тогда  $H = \langle \{0, 2\}; +; 0 \rangle$  – подгруппа группы  $G$  с операцией (табл. 2.5)

Таблица 2.5

+	0	2
0	0	2
2	2	0

**Теорема 2.6.** Пусть  $a$  – элемент порядка  $N$  группы  $G$ , тогда  $H = \langle \{a^1, a^2, \dots, a^{N-1}\}; \cdot; 1 \rangle$  – циклическая подгруппа.

*Пример 2.18.* Пусть  $G = \langle \{1, 2, 3, 4\}; \cdot; 1 \rangle$  – группа,  $H = \langle \{1, 4\}; \cdot; 1 \rangle$  – подмножество  $H$  образует циклическую подгруппу группы  $G$ . Порядок  $N$  элемента 4 равен 2.

*Следствие теоремы 2.6.* Во всякой мультипликативной группе натуральные степени  $a^j$  любого элемента  $a$  образуют подгруппу.

## 2.2. Разложение группы на смежные классы

### 2.2.1. Определение смежного класса по подгруппе

Для заданной конечной группы  $G$  и подгруппы  $H$  существует операция, которая устанавливает взаимосвязь между  $G$  и  $H$ . Эта операция называется разложением группы  $G$  на смежные классы по  $H$ . Обозначим элементы группы  $G$  через  $\{g_1 g_2 \dots g_N\}$ , а элементы подгруппы  $H$  этой группы через  $\{h_1 h_2 \dots h_M\}$ .

Рассмотрим таблицу, построенную следующим образом:

1) запишем элементы подгруппы  $H$  в строку с нейтральным элементом в качестве первого элемента строки;

2) выберем произвольным способом элемент группы  $g_i$  не принадлежащий подгруппе  $H$ , и запишем его первым элементом второй строки;

3) просуммировав  $g_i$  со всеми элементами  $H$ , получим вторую строку таблицы;

4) далее, выбрав произвольным способом элемент  $g_i$ , не принадлежащий ни первой, ни второй строке таблицы, и просуммировав его со всеми элементами  $H$ , получим третью строку и т. д. для всех элементов группы. Построение заканчивается тогда, когда после некоторой итерации оказывается, что каждый элемент группы  $G$  записан в некоторой ячейке таблицы. В результате получается таблица вида

Таблица 2.6

Таблица смежных классов

$h_1 = e$	$h_2$	$h_3$	...	$h_M$
$h_1 + g_1$	$h_2 + g_1$	$h_3 + g_1$	...	$h_M + g_1$
$h_1 + g_2$	$h_2 + g_2$	$h_3 + g_2$	...	$h_M + g_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$h_1 + g_i$	$h_2 + g_i$	$h_3 + g_i$	...	$h_M + g_i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$h_1 + g_j$	$h_2 + g_j$	$h_3 + g_j$	...	$h_M + g_j$

Подобным способом построенная таблица называется таблицей смежных классов. Строки таблицы называются смежными классами по подгруппе  $H$ . Элементы первого столбца называются образующими смежных классов (лидерами смежных классов).

*Пример 2.19.* Заданы группа  $G$  и подгруппа  $H$  группы  $G$ :  
 $G = \langle \{0, 1, 2, 3, 4, 5\}_6; + \rangle$ ;  $H = \langle \{0, 3\}_6; + \rangle$ . Построить таблицу смежных классов.

Решение. По определению смежного класса получим:

Таблица 2.7

0	3
1	4
2	5

**Теорема 2.7.** Каждый элемент группы принадлежит одному и только одному смежному классу.

Теорема используется для декодирования кодов по таблице смежных классов. Декодирование основывается на анализе стандартного расположения элементов таблицы.

*Утверждение 2.1.* Два смежных класса не пересекаются. Объединение всех смежных классов совпадает с множеством группы  $G$ .

Для примера 2.19 справедливы выражения:

$$(0 + H) \cap (1 + H) \cap (2 + H) = \emptyset;$$

$$(0 + H) \cup (1 + H) \cup (2 + H) = G,$$

где символ  $\emptyset$  обозначает пустое множество, а символы  $\cap$  и  $\cup$  – соответственно операции пересечения и объединения.

*Следствие из теоремы 2.7.* Если  $H$  – подгруппа конечной группы  $G$ , то число элементов в  $H$  делит число элементов в  $G$ . Доказательство следует из прямоугольности таблицы смежных классов. Следовательно, порядок  $G$  равен порядку  $H$  умноженному на число смежных классов разложения  $G$  по  $H$ .

### 2.3. Определение смежного класса кода

Пусть имеем код  $G$  мощностью  $M = q^k$ . Для произвольного вектора  $a$  группы  $\{F_n\}$  запишем выражение

$$a + G = \{a + x; x \in G\}.$$

*Определение 2.11.* Сумма вектора  $a$  со всеми векторами  $x$  множества  $G$  называется смежным классом кода  $G$ . Произвольный вектор  $b \in \{F_n\}$  находится в некотором смежном классе.

**Теорема 2.8.** Два вектора  $a$  и  $b$  лежат в одном и том же смежном классе тогда и только тогда, когда  $a - b \in G$ .

Действительно, если  $x \in G$ ,  $y \in G$  и  $a = x + e$ ,  $b = y + e$ ,

$$a - b = x + e - y - e = x - y \in G.$$

Множество всех векторов  $\{F_n\}$  может быть разбито на смежные классы кода  $G$ :

$$\{F_n\} = G \cup (a_1 + G) \cup \dots \cup (a_t + G), \quad (2.3)$$

где  $t = q^{n-k} - 1 = q^r - 1$ .

#### 2.3.1. Таблица стандартного расположения для кода

Первая строка таблицы состоит из всех кодовых слов кода  $x_1, x_2, \dots, x_M$  (включая нулевое слово). Другие строки – это смежные классы  $(a + G)$ , т. е.

$$\begin{aligned} & (a_1 + x_1), (a_1 + x_2), \dots, (a_1 + x_M); \\ & (a_2 + x_1), (a_2 + x_2), \dots, (a_2 + x_M); \\ & \dots \\ & (a_t + x_1), (a_t + x_2), \dots, (a_t + x_M). \end{aligned}$$

Напомним, что  $a_i$  не принадлежит коду. Стандартное расположение для кода приобретает вид, как показано в табл. 2.8.

Для  $q = 2$  таблица содержит:

- $2^r$  смежных классов (строк);
- каждый смежный класс содержит  $2^k$  векторов;
- $2^k$  столбцов;
- $2^r \cdot 2^k = 2^{r+k} = 2^n$  векторов длиной  $n$ .

Таблица 2.8

Стандартное расположение для кода

$x_1$	$x_2$	$x_3$	$\dots$	$x_M$
$a_1+x_1$	$a_1+x_2$	$a_1+x_3$	$\dots$	$a_1+x_M$
$a_2+x_1$	$a_2+x_2$	$a_2+x_3$	$\dots$	$a_2+x_M$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_i+x_1$	$a_i+x_2$	$a_i+x_3$	$\dots$	$a_i+x_M$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$a_t+x_1$	$a_t+x_2$	$a_t+x_3$	$\dots$	$a_t+x_M$

Например, таблица стандартного расположения для  $[7, 4, 3]$ -кода Хэмминга имеет размеры  $8 \times 16$  и содержит  $2^3 \cdot 2^4 = 128$  элементов (векторов).

*Пример 2.20.* Пусть  $[4, 2, 2]$ -код есть подгруппа некоторой двоичной группы  $F_n$ . Код предназначен для передачи сообщений:

$$\begin{aligned} u_0 &= (00) \rightarrow 0; \\ u_1 &= (10) \rightarrow 1; \\ u_2 &= (01) \rightarrow 2; \\ u_3 &= (11) \rightarrow 3. \end{aligned}$$

Сообщениям  $u_i$  соответствуют слова кода  $\mathbf{G}$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}. \quad (2.4)$$

Стандартное расположение элементов таблицы смежных классов кода  $\mathbf{G}$  показано ниже.

Таблица 2.9

Стандартное расположение для  $[4, 2, 2]$ -кода

0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

*Замечание.* Вектор [0001] не входит в образующие смежных классов.

Таблицы стандартного расположения для кода используются для декодирования помехоустойчивых кодов. Декодирование основывается на анализе стандартного расположения элементов таблицы.

*Упражнение 2.1.* Показать, что группа  $G = \langle \{0,1, 2, 3, 4, 5\}; +; 0 \rangle$  содержит подгруппы порядков: 1, 2, 3 и 6.

## 2.4. Кольцо

*Определение 2.12.* Кольцо – это алгебраическая структура или множество элементов  $R$ , в котором определены две основные операции (сложение и умножение) и операция, обратная первой из них (вычитание).

В кольце должны выполняться следующие аксиомы:

–  $\langle R; + \rangle$  – абелева группа;

–  $\langle R; \cdot \rangle$  – полугруппа;

– дистрибутивность:

для любых элементов множества  $R = \{ a, b, c \}$ ,  $a \cdot (b + c) = a \cdot b + a \cdot c$ .

*Определение 2.13.* Множество  $G$  с заданной на нем бинарной операцией и выполнением аксиом замкнутости и ассоциативности называется полугруппой.

### *Примеры полугрупп*

2.21.  $\langle \mathbb{N}; + \rangle$  – аддитивная полугруппа натуральных чисел.

2.22. Пусть  $q$  – это конечный набор символов (алфавит). Например,  $q$  – множество символов белорусского алфавита, или  $q$  двоичное множество  $\{0, 1\}$ . Слово символов из множества имеет вид  $a_1 a_2 \dots a_n$ , где  $a_i \in q$ . Пусть  $G$  обозначает множество слов алфавита  $q$ . Введем бинарную операцию  $\circ$  называемую конкатенацией над  $G$  следующим образом: если  $a_1 a_2 \dots a_n$  и  $b_1 b_2 \dots b_m \in G$ , то  $a_1 a_2 \dots a_n \circ b_1 b_2 \dots b_m = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$ .

Например, если  $q = \{0, 1\}$ , то  $11011 \circ 1010110 = 110111010110$ .

Пусть  $a_1 a_2 \dots a_n, b_1 b_2 \dots b_m$  и  $c_1 c_2 \dots c_t \in G$ . Тогда

$(a_1 a_2 \dots a_n \circ b_1 b_2 \dots b_m) \circ (c_1 c_2 \dots c_t) = (a_1 a_2 \dots a_n b_1 b_2 \dots b_m) \circ c_1 c_2 \dots c_t = a_1 a_2 \dots a_n b_1 b_2 \dots b_m c_1 c_2 \dots c_t$ . Бинарная операция конкатенации ассоциативна на множестве  $G$  и это множество вместе с операцией конкатенации образует полугруппу.

Кольцо можно определить и так: кольцо  $R$  является группой относительно операции сложения и полугруппой относительно операции умножения. В кольце для операции умножения могут не выполняться аксиомы существования нейтрального и обратного элементов группы.



### *Примеры колец*

2.23. Все действительные числа  $\mathbb{R}$  образуют кольцо относительно операций сложения и умножения.

2.24. Алгебраическая система целых чисел  $\langle \mathbb{Z}; +; \cdot \rangle$ .

2.25. Множество квадратных матриц размеров  $n \times n$ . Нейтральным элементом относительно операции умножения в кольце матриц является единичная матрица;

2.26. Кольцо целых чисел  $\mathbb{Z}$  по модулю  $M$ . Например,  $M = 14$ . Тогда система  $\langle \{0, 1, 2, \dots, 13\}; +; \cdot; 0; 1 \rangle$  – кольцо, в котором для элемента 2 не существует обратного элемента, т. к.  $2 \cdot 2^{-1} \neq 1 \pmod{14}$ .

### **2.4.1. Идеал кольца**

*Определение 2.14.* Пусть  $R$  – кольцо, а  $R'$  есть подкольцо – подмножество множества  $R$ . Подкольцо  $R'$  определяется операциями кольца.

### *Примеры подколец*

2.27. Целые числа  $\mathbb{Z}$  образуют подкольцо кольца рациональных чисел

$$R' = \langle \mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}; +; \cdot; 0; 1 \rangle.$$

2.28. Рациональные числа образуют подкольцо кольца действительных чисел  $\mathbb{R}$ .

2.29. Действительные числа  $\mathbb{R}$  образуют подкольцо кольца комплексных чисел;

2.30. Множество квадратных матриц размером  $n \times n$  с целыми значениями элементов образуют подкольцо кольца матриц размером  $n \times n$  с рациональными элементами.

*Определение 2.15.* Подмножество  $I$  элементов кольца  $R$  называется идеалом в  $R$ , если выполняются следующие условия:

–  $I$  является подкольцом кольца  $R$ , т. е. для любого множества  $\{a, b\} \in I$  выполняется  $(a + b) \in I$  и  $a \cdot b \in I$ ;

– для любого элемента  $a \in I$  и любого элемента  $r \in R$  произведения  $a \cdot r$  и  $r \cdot a$  принадлежат  $I$ .

*Упражнение 2.2.* Показать, что множество  $R' = \{0, 2, 4\}$  это подкольцо кольца  $\langle \mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}; +; \cdot; 0; 1 \rangle$ .

### **2.4.2. Главный идеал**

*Определение 2.15.* Пусть  $R$  – коммутативное кольцо. Идеал  $I$  кольца  $R$  называется главным идеалом, порожденным элементом  $a$  (обозначается  $\langle a \rangle$ ), если  $I$  состоит из всех произведений  $a$  на элементы кольца  $R$ , т. е.

$$I = \langle a \rangle = \{ a \cdot r : r \in R \}.$$

**Теорема 2.9.** Совокупность целых чисел  $\mathbb{Z}$  образует главный идеал тогда и только тогда, когда она состоит из всех чисел, кратных некоторому числу.

*Пример 2.31.* Множество целых чисел  $\mathbb{Z} = \{0, 1, 2, 3\}_4$  является коммутативным кольцом с единицей. Найти главный идеал кольца  $\mathbb{Z}$ .

Решение. Если  $a$  – минимальное целое число из  $I$ , то по определению идеала  $b = r \cdot a$ . Выберем в качестве  $a = 2$ , т. е.  $I = \langle 2 \rangle$ . Тогда элементы множества всех чисел, кратных  $a$ , запишем в виде  $I = \langle 2 \rangle$ :

$$0 \cdot 2 = 0; 1 \cdot 2 = 2; 2 \cdot 2 = ((0))_4; 3 \cdot 2 = ((2))_4.$$

Тогда главный идеал есть  $I = \langle 2 \rangle = \{0, 2\}$ .

*Пример 2.32.* Задано кольцо целых чисел  $\mathbb{Z}$ . Построить главный идеал, порожденный целым числом 8.

Решение.  $\langle 8 \rangle = \{8r : r \in \mathbb{Z}\} = \{\dots, -48, -40, -32, -24, -16, -8, 0, 8, 16, 24, 32, 40, 48, \dots\}$ .

*Замечание.* Если имеются главные идеалы  $\langle a \rangle$  и  $\langle b \rangle$  целых чисел  $\mathbb{Z}$ , то их пересечение  $\langle a \rangle \cap \langle b \rangle$  является наименьшим общим кратным (НОК  $(a, b)$ ) чисел  $a$  и  $b$ , т. е.

$$\langle a \rangle \cap \langle b \rangle = \langle \text{НОК}(a, b) \rangle.$$

*Упражнения*

2.3. Задано кольцо целых чисел  $\mathbb{Z}$ . Построить главный идеал, порожденный целым числом 12.

2.4. Найти главный идеал пересечения множеств  $\langle 8 \rangle \cap \langle 12 \rangle$ .

### 2.4.3. Кольцо полиномов

Рассмотрим кольцо вида  $R_n = \frac{R(x)}{x^n - 1}$ , состоящее из класса вычетов кольца полиномов  $R(x)$  по модулю полинома  $x^n - 1$ .

*Определение 2.16.* Идеалом  $I_n$  кольца  $R_n$  называется линейное подмножество полиномов от  $x$  такое, что если  $c(x) \in I_n$ , то  $r(x) \cdot c(x) \in I_n$ , для всех  $r(x) \in R_n$ .

*Пример 2.33.* Пусть  $n = 3$ . Подмножество полиномов вида  $I_n = \{0, (1 + x), (x + x^2), (1 + x^2)\}$  есть идеал в  $R_3$ . Действительно:

– подмножество замкнуто относительно сложения (линейно). Например,  $(1 + x) + (1 + x^2) = (x + x^2) \bmod (x^3 - 1)$ ;

– выполняется условие  $r(x) \cdot c(x) \in I_n$ . Например,  $x^2(1 + x^2) = (x^2 + x^4) \equiv (x + x^2) \bmod (x^3 - 1)$ .

## 2.5. Конечные поля

*Определение 2.17.* Полем называется коммутативное кольцо с единицей, каждый элемент которого имеет обратный элемент относительно умножения.

*Определение 2.18.* Поле – это множество элементов, над которыми заданы две бинарные операции (сложение, умножение) и для них существуют обратные операции (кроме деления на ноль), причем умножение дистрибутивно относительно сложения, т. е.

$$(a + b)c = ab + bc, \quad c(a + b) = ca + cb.$$

Поле – это алгебра  $\langle \{G\}; +; \cdot; -; :; 0; 1 \rangle$ .

*Примеры полей*

2.34. Множество рациональных чисел.

2.35. Множество действительных чисел  $\mathbb{R}$ .

2.36. Множество чисел с  $p$  элементами, где  $p$  – простое число.

При построении кодов различного назначения наибольший интерес представляют конечные алгебры – поля Галуа (Galois Feld –  $GF$ ). Эта алгебра названа в честь Эвариса Галуа ((1811 – 1832), франц. математик).

Конечные поля  $GF(p^m)$  имеют порядок поля  $p^m$ , где простое число  $p$  – это характеристика поля, натуральное число  $m$  – размерность поля. Порядок поля  $GF(p^m)$  равен числу элементов поля.

*Определение 2.19.* Поле  $GF(p)$  называется подполем поля  $GF(p^m)$ , если оно содержится в поле  $GF(p^m)$  и имеет те же операции. Поле  $GF(p^m)$  называется расширением поля  $GF(p)$  (простого поля).

Например, поле из двух элементов  $GF(2)$  – простое поле порядка 2, а  $GF(2^7) = GF(128)$  – его расширение; поле действительных чисел  $\mathbb{R}$  является расширением поля рациональных чисел.

*Замечания*

1. Алгебра (школьная) рациональных чисел – это пример бесконечного поля.

2. Целые числа  $\mathbb{Z}$  бесконечных множеств не образуют поле (результат операции деления двух целых чисел не обязательно является целым числом).

3. При  $m = 1$  имеем простое поле  $GF(p)$  с модулярными операциями сложения и умножения, т. е. операциями по  $\text{mod } p$ .

Для каждого простого  $p$  существует только одно поле, т. е. правила сложения и умножения, удовлетворяющие всем нужным аксиомам, можно задать

только одним способом. Для заданного простого поля  $p$  элементами поля являются числа  $0, 1, \dots, (p - 1)$ .

Наименьшее число элементов, образующих поле равно 2. Это является следствием того, что должны быть нейтральные элементы относительно операции сложения и умножения.

## 2.6. Представление элементов конечного поля Галуа $GF(p^m)$

Удобное описание многих вычислительных алгоритмов, процессов помехоустойчивого и криптографического кодирования и декодирования реализуется с помощью, степенного, полиномиального, векторного и логарифмического представления элементов расширенного поля Галуа.

Элементы поля  $GF(p^m)$  могут интерпретироваться как класс вычетов полиномов от  $x$  с коэффициентами из  $GF(p)$  по модулю неприводимого над полем  $GF(p)$  полинома степени  $m$ .

### 2.6.1. Арифметика полей Галуа

*Пример 2.37.* Определим основные операции в поле неприводимого над полем  $GF(2)$  полинома  $p(x) = 1 + x + x^2, m = 2$ .

Операция сложения в поле  $GF(2^2)$ , элементы которого записываются в виде полиномов, задается в виде

Таблица 2.10

Операция сложения в поле полиномов

+	0	1	$x$	$1 + x$
0	0	1	$x$	$1 + x$
1	1	0	$1 + x$	$x$
$x$	$x$	$1 + x$	0	1
$1 + x$	$1 + x$	$x$	1	0

Записывая коэффициенты полиномов над полем  $GF(2)$ , получим следующее соответствие между полиномами и двоичными векторами:

Таблица 2.11

0	0 0
1	1 0
$x$	0 1
$1 + x$	1 1

Таблица Кэли в поле  $GF(2^2)$ , элементы которого записываются в виде двоичных чисел, имеет вид

Таблица 2.12

Операция сложения в поле двоичных чисел

+	0 0	1 0	0 1	1 1
---	-----	-----	-----	-----

0 0	0 0	1 0	0 1	1 1
1 0	1 0	0 0	1 1	0 1
0 1	0 1	1 1	0 0	1 0
1 1	1 1	0 1	1 0	0 0

Аналогично построим таблицы умножения элементов поля  $GF(2^2)$ . Операция умножения в поле  $GF(2^2)$ , элементы которого записываются в виде полиномов, задается в виде

Таблица 2.13

Операция умножения в поле полиномов

$\times$	0	1	$x$	$1 + x$
0	0	0	0	0
1	0	1	$x$	$1 + x$
$x$	0	$x$	$1 + x$	1
$1 + x$	0	$1 + x$	1	$x$

Таблица умножения, представленная двоичными векторами, имеет вид

Таблица 2.12

Операция умножения в поле двоичных чисел

$\times$	0 0	1 0	0 1	1 1
0 0	0 0	0 0	0 0	0 0
1 0	0 0	1 0	0 1	1 1
0 1	0 0	0 1	1 1	1 0
1 1	0 0	1 1	1 0	0 1

Например, элемент таблицы 10 с координатами (01, 11) получен следующим образом.

1. Воспользуемся обычным умножением в «столбик» со старшим разрядом числа слева:

$$\begin{array}{r}
 1 \ 0 \\
 \underline{1 \ 1} \\
 1 \ 0 \\
 \underline{1 \ 0} \\
 1 \ 1 \ 0
 \end{array}$$

2. Результат умножения приведем по модулю полинома  $p(x)$  (вектора – коэффициентов  $p(x)$ ):

$$\begin{array}{r}
 \underline{1 \ 1 \ 0} | 1 \ 1 \ 1 \\
 \underline{1 \ 1 \ 1} \ 1 \\
 0 \ 0 \ 1
 \end{array}$$

3. Для записи двоичного числа со старшим разрядом числа справа

выполним инверсию:

0 0 1|1 0 0. Получим вектор 1 0.

**Теорема 2.10.** В расширенном поле полиномов  $GF(p^m)$  существует примитивный элемент  $\alpha$  порядка  $(p^m - 1)$ , т. е.

$$\alpha^{(p^m - 1)} = 1.$$

Каждый элемент  $\beta$  поля  $GF(p^m)$  может быть представлен как некоторая степень, т. е.  $\beta = \alpha^i$

*Определение 2.20.* Неприводимый над полем  $GF(p)$  полином степени  $m$  называется примитивным, если его корнем является примитивный элемент  $\alpha$  поля  $GF(p^m)$ .

*Пример 2.38.* Полином  $p(x) = 1 + x + x^2$  неприводим над полем  $GF(2)$ . Этот полином примитивный, так как примитивный элемент  $\alpha$  поля  $GF(p^m)$  является корнем полинома  $p(x)$ . Действительно,

$$p(\alpha) = 1 + \alpha + \alpha^2 = 0, \text{ так как } x^2 \equiv (1 + x) \pmod{(1 + x + x^2)}.$$

Тогда

$$p(\alpha) = 1 + \alpha + \alpha^2 = 1 + \alpha + 1 + \alpha = 0.$$

В табл. 2.13 приведены четыре формы представления элементов поля  $GF(2^2)$ . Поле образовано полиномами над полем  $GF(2)$  по модулю неприводимого полинома  $p(x) = 1 + x + x^2$ . Порядок элемента  $\alpha$  равен 3, так как  $\alpha^{(2^2 - 1)} = \alpha^3 \equiv 1 \pmod{(1 + \alpha + \alpha^2)}$ .

Таблица 2.13

Формы представления элементов поля  $GF(2^2)$

В виде степени примитивного элемента	В виде полинома	В виде двоичного числа	В виде логарифма
—	0	0 0	—
$\alpha^0$	1	1 0	0
$\alpha^1$	$\alpha$	0 1	1
$\alpha^2$	$1 + \alpha$	1 1	2
$\alpha^3 = \alpha^0$	1	1 0	0

Используя разные формы элементов поля можно эффективно производить алгебраические операции в поле  $GF(p^m)$ .

1. Умножение с представлением элементов  $\beta$  поля в виде степеней примитивного элемента  $\alpha$  выполняется следующим образом:

$$\beta_1 \cdot \beta_2 = \alpha^i \cdot \alpha^j = \alpha^{i+j} = ((\alpha^{\text{Rest}[\frac{i+j}{N}]})$$

где  $Rest \left[ \frac{i+j}{N} \right]$  – остаток от деления  $(i + j)$  на порядок  $N$  примитивного элемента  $\alpha$  поля  $GF(p^m)$ . Например, используя таблицу 2.13, имеем

$$\alpha^2 \cdot \alpha^2 = \alpha^4 \equiv \alpha^1 \pmod{3}.$$

2. Деление на элемент поля

**Теорема 2.11.** Если полином  $p(x)$  степени  $m$  неприводим над полем  $GF(p)$ , то каждый ненулевой полином  $c(\alpha)$  степени не более  $(m - 1)$  имеет единственный обратный полином  $c(\alpha)^{-1}$  такой, что

$$c(\alpha) \cdot c(\alpha)^{-1} \equiv 1 \pmod{p(\alpha)}.$$

Нахождение обратных элементов легко выполнять, если воспользоваться представлением элементов поля в виде степеней примитивного элемента или логарифмов.

Пусть  $c = c(\alpha)$  – произвольный элемент поля  $GF(p^m)$  с коэффициентами из поля  $GF(p)$ , т. е.

$$c = c(\alpha) = c_0 + c_1\alpha + c_2\alpha^2 + \dots + c_{m-1}\alpha^{m-1}.$$

Требуется разделить элемент  $c$  на элемент поля

$$b = b_0 + b_1\alpha + b_2\alpha^2 + \dots + b_{m-1}\alpha^{m-1}.$$

Для того чтобы найти  $c/b$ , вычислим обратный элемент  $b^{-1} = 1/b$ , а затем представим  $c/b$  как

$$c/b = c \cdot 1/b.$$

*Пример 2.39.* Поле образовано полиномами над полем  $GF(2)$  по модулю неприводимого полинома  $p(x) = 1 + x + x^2$ . Вычислить  $\alpha/(1 + \alpha)$ .

Решение. Полиному  $(1 + \alpha)$  соответствует  $\alpha^2$ . Тогда

$$\alpha/(1 + \alpha) = \frac{\alpha}{\alpha^2} = \alpha^{-1} = \alpha^3\alpha^{-1} = \alpha^2 = ((1 + \alpha)).$$

*Пример 2.40.* Поле образовано полиномами над полем  $GF(2)$  по модулю неприводимого полинома  $p(x) = 1 + x + x^3$ . Вычислить  $\sqrt{110}$ .

Решение. Вектору (110) соответствует элемент поля  $\alpha^3$ . Квадратный корень  $\sqrt{\alpha^3} = \sqrt{1\alpha^3} = \sqrt{\alpha^7\alpha^3} = \sqrt{\alpha^{10}} = \alpha^5 = ((111))$ .

*Упражнения*

2.5. Найти примитивные элементы расширенного поля Галуа  $GF(2^4)$ .

2.6. Привести четыре формы представления элементов поля  $GF(2^4)$ . Поле образовано неприводимым над полем  $GF(2)$  полиномом  $p(x) = 1 + x + x^4$ .

2.7. Поле образовано полиномами над полем  $GF(2)$  по модулю неприводимого полинома  $p(x) = 1 + x + x^3$ . Вычислить:

- а) полином, обратный полиному  $a = 1 + \alpha + \alpha^2$ ;
- б) полином, обратный полиному  $b = \alpha + \alpha^2$ ;
- в)  $a/b$ ;
- г)  $\sqrt{b}$ ;
- д)  $(111)^{-1}$ .

## 2.7. Векторные пространства и подпространства

**Определение 2.21.** Векторное пространство  $V$  над полем  $F$  – это алгебраическая система, удовлетворяющая следующим аксиомам.

1.  $V$  является аддитивной группой.
2. Для любых двух элементов  $x \in V$  и  $c \in F$  определено произведение  $cx$ .
3. Выполняется аксиома ассоциативности для всех элементов  $x, y \in V$   
 $c(x + y) = cx + cy$ .
4. Если  $x \in V$  и  $c, b \in F$ , то  $(c + b)x = cx + bx$ .
5. Если  $x \in V$  и  $c, b \in F$ , то  $(cb)x = c(bx)$ .
6.  $1 \cdot x = x$ , где 1 – нейтральный элемент поля.

**Определение 2.22.** Подмножество  $W$  векторного пространства  $V$ , удовлетворяющее условиям:

- если  $w_1, w_2 \in W$ , то  $w_1 + w_2 \in W$ ;
- для  $c \in F$  и  $w \in W$ ,  $cw \in W$ .

**Замечание.** Кодовые векторы избыточного кода определяют подпространство векторного пространства.

### 2.7.1. Линейно зависимые и независимые векторы

**Определение 2.23.** Векторы  $v_0, v_1, \dots, v_{k-1}$  из  $V$  называются линейно зависимыми, если в  $F$  существуют такие элементы  $c_0, c_1, \dots, c_{k-1}$ , для которых

$$c_0 v_0 + c_1 v_1 + \dots + c_{k-1} v_{k-1} = 0,$$

и линейно независимыми, когда

$$c_0 v_0 + \dots + c_{k-1} v_{k-1} \neq 0.$$

Наглядно условие линейной независимости векторов над полем  $F$  можно изобразить как

$$c_0 \times \begin{array}{|c|} \hline \diagup \\ \hline \end{array} + c_1 \begin{array}{|c|} \hline \diagup \\ \hline \end{array} \dots + c_{k-1} \begin{array}{|c|} \hline \diagup \\ \hline \end{array} \neq \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline \cdot \\ \hline \cdot \\ \hline \cdot \\ \hline \end{array}$$



$$v_0 \quad v_1 \quad \dots \quad v_{k-1}$$

где  $c_i \in F$ . Для двоичных кодов  $c_i \in \{0,1\}$ , линейная независимость означает, что суммирование базисных двоичных векторов не образует нулевой вектор. Максимальное число линейно независимых векторов в  $V$  называется размерностью пространства  $V$  над полем  $F$ .

### 3. ЛИНЕЙНЫЕ КОДЫ

#### 3.1. Линейные коды, исправляющие ошибки: построение и основные свойства

*Определение 3.1.* Линейный  $[n, k, d]$ -код есть подпространство размерности  $k$  линейного  $n$ -мерного пространства над  $GF(q) = GF(p^m)$ . Подмножество, состоящее из  $q^k$  последовательностей длиной  $n$ , называется  $q$ -ичным блочным кодом длиной  $n$ .

*Замечания:*

1. Если  $q = 2$ , и в качестве кодового алфавита используются символы двоичного  $\{0, 1\}$  алфавита. Пространство кода над полем  $GF(2)$  образует аддитивную подгруппу группы всех двоичных последовательностей длиной  $n$ . Очевидно, что для этой подгруппы должны выполняться все аксиомы группы, и для нее определена одна основная операция – сложение.

2. Так как операция суммирования является линейной операцией, то и код называется линейным.

Примером линейного группового кода является двоичный код Хэмминга. Для каждого целого положительного числа  $m$  существует код Хэмминга с параметрами:

$$[2^m - 1, 2^m - 1 - m, 3]; R = (2^m - 1 - m) / (2^m - 1); r = m; t = 1.$$

Для больших значений  $m$  скорость кода  $R \approx 1$ . Например, для  $m = 8$  получаем величину

$$R = (2^8 - 1 - 8) / (2^8 - 1) = 247 / 255 = 0,968.$$

Линейные коды делятся на:

- систематические (разделимые);
- несистематические (неразделимые).

У систематического кода первые  $k$  символов кодового слова – информационные.

Несистематические – нет деления на информационные и проверочные символы (все символы являются кодовыми символами).

##### 3.1.1. Кодер систематического кода

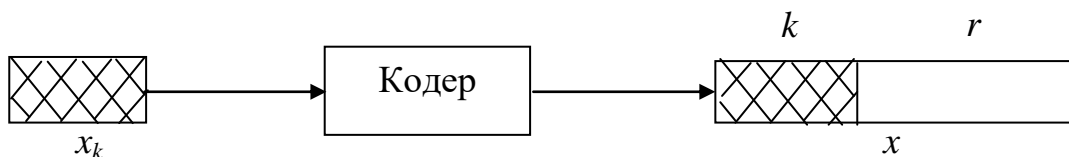


Рис. 3.1. Кодер систематического кода

Действия систематического кодера, рис. 3.1:

1) кодер разбивает входную информационную последовательность символов на блоки  $x_{k_l} = (x_0 x_1 \dots x_{k-1})$  длиной  $k$ ;

2) для каждого блока  $x_{k_l}$  находит слово  $x_l$ , первые  $k$  символов которого совпадают с  $x_{k_l}$ ;

3) кодер обрабатывает каждый поступающий блок независимо от других так, что каждое новое слово на его выходе оказывается не связанным с предыдущими кодовыми словами.

В качестве примера линейного кода приведем  $[7, 4, 3]$ -код Хэмминга мощностью  $M = 2^4 = 16$ . Ненулевые кодовые слова  $x_l = (x_0 x_1 \dots x_{n-1})$  кода Хэмминга  $[7, 4, 3]$  в порядке возрастания значения информационных весов  $\text{wt}\{x_k\}$  векторов кода записаны в табл. 3.

Таблица 3.1

Код Хэмминга  $[7, 4, 3]$

	$X_l$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	
$\text{wt}\{x_k\}=1$	$X_0$	1	0	0	0	1	1	0	$X_0$
	$X_1$	0	1	0	0	1	1	1	$X_1$
	$X_2$	0	0	1	0	0	1	1	$X_2$
	$X_3$	0	0	0	1	1	0	1	$X_3$
$\text{wt}\{x_k\}=2$	$X_4$	1	1	0	0	0	0	1	$X_0 + X_1$
	$X_5$	1	0	1	0	1	0	1	$X_0 + X_2$
	$X_6$	1	0	0	1	0	1	1	$X_0 + X_3$
	$X_7$	0	1	1	0	1	0	0	$X_1 + X_2$
	$X_8$	0	1	0	1	0	1	0	$X_1 + X_3$
	$X_9$	0	0	1	1	1	1	0	$X_2 + X_3$
$\text{wt}\{x_k\}=3$	$X_{10}$	1	1	1	0	0	1	0	$X_0 + X_1 + X_2$
	$X_{11}$	1	1	0	1	1	0	0	$X_0 + X_1 + X_3$
	$X_{12}$	1	0	1	1	0	0	0	$X_0 + X_2 + X_3$
	$X_{13}$	0	1	1	1	0	0	1	$X_1 + X_2 + X_3$
$\text{wt}\{x_k\}=4$	$X_{14}$	1	1	1	1	1	1	1	$X_0 + X_1 + X_2 + X_3$

Если информационный вес  $\text{wt}\{x_k\} = i$ , то число кодовых слов данного информационного веса определяется биномиальным коэффициентом  $C_k^i$ . Тогда  $[7, 4, 3]$ -код Хэмминга содержит  $(C_k^1 + C_k^2 + C_k^3 + C_k^4) = 15$  ненулевых слов. Обозначения  $x_0, x_1, x_2, x_3$  соответствуют информационным символам. Проверочные символы кода задаются следующими равенствами:

$$x_4 = x_0 + x_1 + x_3;$$

$$x_5 = x_0 + x_1 + x_2;$$

$$x_6 = x_1 + x_2 + x_3.$$

Все множество кодовых слов кода образуется путем суммирования первых четырех строк (базовых векторов) по 2, по 3, по 4, ..., по  $k$ , табл. 3.1. Таким образом, кодовые слова являются линейными комбинациями строк задающей код матрицы. С точки зрения алгебры все ненулевые слова кода образуют некоторое векторное пространство, базисом которого являются строки базовой (порождающей) матрицы.

### 3.2. Вектор ошибок

Пусть ДСК характеризуется вероятностью ошибки на символ  $p$ . Введем вектор ошибки  $\mathbf{e} = (e_1 \dots e_j \dots e_n)$ ;  $e_j = 1$  с вероятностью  $p$  и  $e_j = 0$  с вероятностью  $(1 - p)$ . Найдем вероятность возникновения нескольких конфигураций векторов  $\mathbf{e}$  ошибок для кодового слова длиной  $n = 5$ .

$\text{Prob}\{\mathbf{e} = (00000)\} = (1 - p)^5$  – есть вероятность правильного приема кодового слова длиной 5.

Вероятность возникновения конфигурации вектора ошибок вида  $\mathbf{e} = (10000)$  равна

$$\text{Prob}\{\mathbf{e} = (10000)\} = p(1 - p)^4.$$

Вероятность вектора ошибок вида  $\mathbf{e} = (10010)$  равна

$$\text{Prob}\{\mathbf{e} = (10010)\} = p^2(1 - p)^3.$$

В общем случае, вероятность возникновения вектора  $\mathbf{e}$  ошибок веса  $i$  запишется в виде

$$\text{Prob}\{\mathbf{e}_{\text{wt}(i)}\} = p^i(1 - p)^{n-i}. \quad (3.1)$$

Вероятность правильного приема кодового слова длиной  $n$  равна

$$\text{Prob}\{\mathbf{e}_{\text{wt}(0)}\} = (1 - p)^n.$$

*Пример 3.1.* Пусть  $p = 0,2$ ,  $n = 5$ . Рассмотрим возможные вероятности возникновения векторов ошибок.

1. Вероятность того, что не произошло ни одной ошибки на длине кодового слова

$$\text{Prob}\{\mathbf{e}_{\text{wt}(0)} = (00000)\} = (1 - 0,2)^5 \cong 0,32.$$

2. Вероятность того, что на длине кодового слова имеется ошибка единичного веса

$$\text{Prob}\{\mathbf{e}_{\text{wt}(1)} = (10000)\} = p(1 - p)^4 = 0,2(1 - 0,2)^4 \cong 0,081.$$

3. Вероятность того, что произошли две ошибки на длине кодового слова

$$\text{Prob}\{e_{\text{wt}(2)} = (10010)\} = p^2(1-p)^3 = 0,2^2(1-0,2)^3 \cong 0,01.$$

Из приведенного примера следуют очевидные выводы:

- вектор ошибок единичного веса более вероятен, чем вектор ошибок веса два и т. д.;
- ошибки малого веса необходимо обнаруживать и исправлять в первую очередь.

### 3.3. Порождающая и проверочная матрица систематического линейного кода

#### 3.3.1. Способы задания линейных кодов

Линейные коды задаются с помощью:

- порождающей матрицы  $\mathbf{G}$  размерностью  $k \times n$ ;
- проверочной матрицы  $\mathbf{H}$  размерностью  $r \times n$ .

Матрицы связаны основным уравнением кодирования

$$\mathbf{G} \times \mathbf{H}^T = 0 \quad (3.2)$$

Из (3.2) следует, что для всякой матрицы  $\mathbf{G}$  существует матрица  $\mathbf{H}$ , удовлетворяющая этому равенству. И наоборот, заданной матрице  $\mathbf{H}$  будет соответствовать только одна матрица  $\mathbf{G}$ . В качестве строк матрицы  $\mathbf{G}$  выбираются линейно-независимые слова длиной  $n$ , отстоящие друг от друга на заданное кодовое расстояние  $d$ .

Поскольку линейно независимые векторы могут быть выбраны произвольным образом, очевидно, можно построить множество матриц  $\mathbf{G}$  с одним и тем же кодовым расстоянием  $d$ . Например, второй вариант задания [7, 4, 3]-кода Хэмминга (см. табл. 3.1) в виде матрицы  $\mathbf{G}$  представляется как

$\mathbf{G}'$	$x_l$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
	$x_0$	1	0	0	0	1	0	1
	$x_1$	0	1	0	0	1	1	0
	$x_2$	0	0	1	0	1	1	1
	$x_3$	0	0	0	1	0	1	1

#### 3.3.2. Свойства линейных кодов

Линейно независимые векторы инвариантны относительно двух операций, при выполнении которых минимальное расстояние кода не изменяется. Справедливы такие операции:

- произвольные перестановки столбцов и строк матрицы  $\mathbf{G}$ ;
- элементарные операции (например, сложение) над строками матрицы  $\mathbf{G}$ .

*Замечание.* Перестановка символов кода эквивалентна перестановке

столбцов порождающей матрицы.

### 3.3.3. Эквивалентные коды

*Определение 3.2.* Два кода эквивалентны тогда, когда их порождающая матрица получается одна из другой на основе свойства инвариантности.

*Примеры*

3.2. Матрица  $G$

$$G = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

порождает эквивалентный код  $G'$

$$G' = \begin{pmatrix} X_0 + X_1 + X_2 \\ X_1 + X_2 \\ X_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

3.3. Эквивалентные коды Хэмминга  $[7, 4, 3]$  заданы матрицами  $G$  и  $G'$ .

$G$	$x_l$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
	$x_0$	1	0	0	0	1	1	0
	$x_1$	0	1	0	0	1	1	1
	$x_2$	0	0	1	0	0	1	1
	$x_3$	0	0	0	1	1	0	1

$G'$	$x_l$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
	$x_0$	1	0	0	0	1	0	1
	$x_1$	0	1	0	0	1	1	0
	$x_2$	0	0	1	0	1	1	1
	$x_3$	0	0	0	1	0	1	1

Следует различать эквивалентный и эквидистантный код.

### 3.3.4. Эквидистантные коды

*Определение 3.3.* Эквидистантный код – это множество слов, отстоящих друг от друга на одно и то же расстояние Хэмминга  $d_x$ .

Пример эквидистантного  $[7, 3, 4]$ -кода ( $m$ -код) представлен матрицей

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

### 3.3.5. Каноническая форма порождающей матрицы

Следствием свойства инвариантности векторов относительно выше названных операций является, каноническая (приведено ступенчатая) форма матриц  $\mathbf{G}$  и  $\mathbf{H}$ . Порождающая матрица кода в канонической форме записывается как

$$\mathbf{G} = [\mathbf{I}_k | \mathbf{G}^*] \quad (3.3)$$

где  $\mathbf{I}_k$  – единичная подматрица размером  $k \times k$ , а  $\mathbf{G}^*$  есть  $k \times (n - k)$  подматрица.

С учётом формы матрицы (3.3) уравнение кодирования представим как

$$\mathbf{GH}^T = [\mathbf{I}_k | \mathbf{G}^*] \begin{bmatrix} -\mathbf{G}^* \\ \mathbf{I}_r \end{bmatrix} = -\mathbf{G}^* + \mathbf{G}^* = 0.$$

Отсюда определяем проверочную матрицу  $\mathbf{H}$  в канонической форме

$$\mathbf{H} = [-\mathbf{G}^{*T} | \mathbf{I}_r]. \quad (3.4)$$

где  $\mathbf{I}_r$  единичная матрица размером  $r \times r$ .

В поле  $GF(2)$   $-\mathbf{G}^{*T} = \mathbf{G}^{*T}$ , поэтому

$$\mathbf{H} = [\mathbf{G}^{*T} | \mathbf{I}_r].$$

Матрицы (3.3) и (3.4) соответствуют систематическому линейному коду. Если известна проверочная матрица систематического кода

$$\mathbf{H} = [\mathbf{H}^* | \mathbf{I}_r], \quad (3.5)$$

то матрица  $\mathbf{G}$  записывается в виде

$$\mathbf{G} = [\mathbf{I}_k | \mathbf{H}^{*T}].$$

Таким образом, код можно задавать перечислением всех  $q^k$  разрешенных для передачи кодовых слов, или перечислением только базисных векторов кодового подпространства. Очевидно, второй способ гораздо компактнее и удобнее для описания кодов. Например, если  $q = 2$ ,  $R = \frac{1}{2}$ ,  $n = 256$ , число кодовых слов достигает порядка  $M = 2^{128}$ . Полная их запись требует  $2^{128} \cdot 2^8 = 2^{136}$  битов. Порождающая матрица этого же кода требует только  $128 \times 256 = 2^7 \cdot 2^8 = 2^{15} = 32768$  битов.

### 3.4. Кодирование линейным кодом

Кодирование представляет собой операцию умножения вектора  $\mathbf{u}$  сообщения на порождающую матрицу  $\mathbf{G}$ ,

$$\mathbf{x} = \mathbf{u}\mathbf{G}. \quad (3.6)$$

При умножении вектора  $\mathbf{u}$  на  $\mathbf{G}$  в форме (3.3) образуется систематический код, т. к. умножение  $\mathbf{u}\mathbf{I}_k = \mathbf{u}$  не изменяет входного сообщения. Первые  $k$  символов кодового слова равны соответствующим символам сообщения. Проверочные символы выбираются так, чтобы кодовые слова удовлетворяли основному уравнению кодирования

$$\mathbf{x}\mathbf{H}^T = (x_0x_1 \dots x_{k-1} \dots x_{n-1})\mathbf{H}^T = (00 \dots 0). \quad (3.7)$$

Запишем выражение (3.7) иначе

$$\mathbf{H}\mathbf{x}^T = \mathbf{0}. \quad (3.8)$$

С учетом выражения (3.5), формулу (3.8) представим в виде

$$[\mathbf{H}^* | \mathbf{I}_r] \begin{bmatrix} x_0 \\ \vdots \\ x_{k-1} \\ x_k \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Найдем из этого уравнения проверочные символы

$$\begin{bmatrix} x_k \\ \vdots \\ x_{n-1} \end{bmatrix} = -\mathbf{H}^* \begin{bmatrix} x_0 \\ \vdots \\ x_{k-1} \end{bmatrix}.$$

В поле  $GF(2)$   $(-\mathbf{H}^*) = \mathbf{H}^*$ . Поэтому перепишем последнее выражение как

$$\begin{bmatrix} x_k \\ \vdots \\ x_{n-1} \end{bmatrix} = \mathbf{H}^* \begin{bmatrix} x_0 \\ \vdots \\ x_{k-1} \end{bmatrix}. \quad (3.9)$$

Таким образом, в поле  $GF(2)$  проверочные символы записываются по формуле (3.9).

*Пример 3.4.* Пусть имеется матрица  $\mathbf{H}$  размерностью  $3 \times 6$ . Обозначим элементы матрицы  $\mathbf{H}$  через  $h_{ij}$ . Выражение (3.8) примет вид

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & 1 & 0 & 0 \\ h_{21} & h_{22} & h_{23} & 0 & 1 & 0 \\ h_{31} & h_{32} & h_{33} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Умножение матрицы **H** на вектор-столбец

$$\begin{bmatrix} h_{11}x_0 + h_{12}x_1 + h_{13}x_2 + x_3 \\ h_{21}x_0 + h_{22}x_1 + h_{23}x_2 + x_4 \\ h_{31}x_0 + h_{32}x_1 + h_{33}x_2 + x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

определяет уравнения для получения проверочных символов из выражений:

$$\begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}; \quad (3.10)$$

$$x_3 = h_{11}x_0 + h_{12}x_1 + h_{13}x_2;$$

$$x_4 = h_{21}x_0 + h_{22}x_1 + h_{23}x_2;$$

$$x_5 = h_{31}x_0 + h_{32}x_1 + h_{33}x_2.$$

Из формул (3.9, 3.10) следует, что для двоичных кодов:

- каждый проверочный символ есть сумма информационных символов;
- в вычислении  $i$ -го проверочного символа участвуют те информационные, которым соответствуют единицы в  $i$ -ой строке матрицы **H**;
- $i$ -ый проверочный символ определяется по  $i$ -му столбцу подматрицы **G**<sup>\*</sup>.

### 3.5. Линейный код Рида-Маллера

Как правило, в результате кодирования информации кодом Рида-Маллера (РМ-кодом) получается неразделимый код. При этом используется однородная и регулярная структура порождающей матрицы **G**, позволяющая упростить процедуры кодирования и декодирования.

Практическое применение этого кода осуществлено сравнительно давно в американской специальной системе скрытной связи «Диджиллок». В 1972 году РМ-коды использовались в американской космической программе «Маринер» по передаче изображений марсианской поверхности. РМ-код имеет следующие параметры.

1. Значность кода  $n = 2^m$ ,  $m \geq 2$ .
2. Кодовое расстояние  $d = 2^{m-l}$ .



3. Порядок кода  $l$ .

4. Размерность кода  $k = 1 + \sum_{i=1}^l C_m^i$ .

Порождающая матрица РМ-кода порядка  $l$  строится из определения операции пересечения двоичных векторов. Пусть заданы векторы

$$\mathbf{x} = (x_0 \ x_1 \dots x_{n-1}), \mathbf{y} = (y_0 \ y_1 \dots y_{n-1}).$$

Результат операции пересечения

$$\mathbf{x} \cdot \mathbf{y} = ((x_0 \cdot y_0)(x_1 \cdot y_1) \dots (x_{n-1} \cdot y_{n-1})).$$

При таком определении векторы порождающей матрицы РМ-кода образуют коммутативную группу.

Например, совокупность векторов, образованная пересечением трех векторов  $\mathbf{x}_i$  взятых по 2 имеет вид

$$\mathbf{A} = \left\{ \begin{matrix} \mathbf{x}_0 & \cdot & \mathbf{x}_1 \\ \mathbf{x}_0 & \cdot & \mathbf{x}_2 \\ \mathbf{x}_1 & \cdot & \mathbf{x}_2 \end{matrix} \right\}.$$

Код Рида-Маллера порядка  $l$  определяется как код, базисом которого являются векторы  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{m-1}$  и все векторные пересечения из  $m$  или меньшего числа этих векторов.

*Пример 3.5.* Построим код Рида-Маллера первого порядка  $l = 1$ ,  $m = 3$ .

Решение. Получаем РМ-код со следующими параметрами:

$$n = 2^3 = 8;$$

$$d = 2^{3-1} = 4;$$

$$k = 1 + \sum_{i=1}^1 C_3^i = 1 + C_3^1 = 4.$$

Имеем  $[8, 4, 4, ]$ -код. В общем случае кодовое расстояние РМ-кода первого порядка равно  $d = n / 2$ . Код Рида-Маллера первого порядка задается порождающей матрицей  $\mathbf{G}$ , первая строка которой состоит из  $n$  единиц. В качестве столбцов остальных  $m$  строк используются все двоичные числа длиной  $m$ . Порождающая матрица РМ-кода первого порядка при  $m = 3$  имеет вид

$$\mathbf{G} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (3.11)$$

Векторы  $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  линейно независимы.

*Пример 3.6.* Записать матрицу  $\mathbf{G}$  РМ-кода второго порядка для  $m = 3$ .

Код характеризуется параметрами:

$$n = 8;$$

$$k = 1 + \sum_{i=1}^2 C_3^i = 1 + C_3^1 + C_3^2 = 1 + 3 + 3 = 7;$$

$$d = 2^{m-l} = 2.$$

Порождающая матрица  $[8, 7, 2]$  РМ-кода 2-го порядка

$$\mathbf{G} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 = x_1 \cdot x_2 \\ x_5 = x_1 \cdot x_3 \\ x_6 = x_2 \cdot x_3 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

*Пример 3.7.* Закодировать сообщение  $\mathbf{u} = (1001)$  несистематическим  $[8, 4, 4]$  РМ-кодом.

Решение. Кодовое слово  $\mathbf{x} = \mathbf{uG} = (1\ 1\ 1\ 1\ 0000)$ .

*Пример 3.8.* Закодировать сообщение  $\mathbf{u} = (1001)$  систематическим  $[8, 4, 4]$  РМ-кодом.

Решение. Получим порождающую матрицу. Для этого:

– просуммируем все строки матрицы (3.11) и запишем суммарный вектор вместо 1-ой строки матрицы (3.11). В результате получим матрицу

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}; \quad (3.12)$$

– переставим столбцы единичного веса матрицы (3.12), приведя их к единичной матрице

$$\mathbf{G}_{8,4} = [\mathbf{I}_k | \mathbf{G}^*] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \quad (3.13)$$

Проверочная матрица систематического  $[8, 4, 4]$  РМ-кода имеет вид:

$$\mathbf{H}_{8,4} = [\mathbf{G}^{*T} | \mathbf{I}_r] = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.14)$$

Из (3.14) запишем уравнения для вычисления проверочных символов:

$$\begin{aligned}x_4 &= x_0 + x_1 + x_2; \\x_5 &= x_0 + x_1 + x_3; \\x_6 &= x_0 + x_2 + x_3; \\x_7 &= x_1 + x_2 + x_3.\end{aligned}$$

Кодовое слово, соответствующее сообщению  $\mathbf{u} = (1001)$ , есть

$$\mathbf{x} = (10011001).$$

*Пример 3.9.* Удовлетворяют ли матрицы канонического вида  $\mathbf{G}$  (3.13) и  $\mathbf{H}$  (3.14) основному уравнению кодирования? Для ответа на этот вопрос найдем произведение матриц  $\mathbf{G}$  и  $\mathbf{H}^T$ .

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Матрицы удовлетворяют основному уравнению кодирования.

#### Упражнения

3.1. Построить в канонической форме матрицы  $\mathbf{G}$  и  $\mathbf{H}$  кода с проверкой на четность  $[n, (n-1), 2], n = 8$ .

3.2. Показать, что матрицы  $\mathbf{G}$  и  $\mathbf{G}'$  порождают эквивалентные коды

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \mathbf{G}' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

3.3. Построить канонические матрицы  $\mathbf{G}$  и  $\mathbf{H}$  кода с повторением  $[n, 1, n], n = 6$ .

#### 3.6. Линейный код Хэмминга

Для практических целей желательно иметь код, который можно легко кодировать и декодировать. Важным семейством кодов, которые отвечают этому требованию, является семейство кодов Хэмминга, исправляющие одну ошибку.

Параметры кода Хэмминга:

- длина  $2^r - 1, r = 2, 3, \dots$ ;
- размерность  $k = 2^r - 1 - r$ ;
- кодовое расстояние  $d = 3$ ;
- распределение лидеров смежных классов веса  $a_{i=0} = 1, a_{i=1} = n$ .

В табл. 3.2 приведены значения некоторых параметров кодов Хэм-

минга.

Таблица 3.2

Параметры кодов Хэмминга

$r$	3	4	5	6	7	8
$n$	7	15	31	63	127	255
$k$	4	11	26	57	120	247

### 3.6.1. Формы представления проверочной матрицы кода Хэмминга

1. Классическая форма (несистематический код Хэмминга).

В матрице  $H$  каждый  $i$ -ый столбец равен двоичному представлению числа  $i \in \{1, 2, \dots, n\}$ . Например, матрица  $[7, 4, 3]$ -кода Хэмминга имеет вид

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (3.15)$$

2. Систематическая форма матрицы  $H$  кода Хэмминга.

Например, матрица  $[7, 4, 3]$ -кода Хэмминга имеет вид

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Из этой матрицы легко можно получить порождающую матрицу  $[7, 4, 3]$ -кода Хэмминга.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

### 3.6.2. Задание проверочной матрицы кода Хэмминга с помощью элементов расширенного поля Галуа $GF(2^m)$

Если  $\alpha$  – примитивный элемент поля  $GF(2^m)$ , то все элементы различны и представляются ненулевыми двоичными векторами. Это свойство используется для построения проверочной матрицы кода Хэмминга в форме

$$H = [1 \quad \alpha \quad \alpha^2 \quad \dots \quad \alpha^{2^m-2}].$$

*Пример 3.10.* Пусть  $\alpha \in GF(2^3)$  – корень уравнения  $\alpha^3 + \alpha + 1 = 0$ . Проверочная матрица

$$\mathbf{H} = [1 \quad \alpha \quad \alpha^2 \quad \alpha^3 \quad \alpha^4 \quad \alpha^5 \quad \alpha^6].$$

Все элементы поля могут быть представлены как различные ненулевые двоичные  $m$ -векторы. Проверочная матрица в виде векторов двоичного  $[7, 3, 4]$ -кода Хэмминга записывается как

$$\mathbf{H} = \begin{matrix} & 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \end{matrix}.$$

### 3.7. Совершенные и квазисовершенные коды

*Определение 3.4.* Код, исправляющий все конфигурации ошибок веса не более  $t$ , называется совершенным.

Для такого кода нет ни одной конфигурации ошибок веса большего, чем кратность исправляемых. Для совершенного кода число лидеров смежных классов веса  $i$  равно  $a_i = 0$  при условии

$$i > t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

*Определение 3.5.* Если число лидеров смежных классов веса  $i$  равно  $a_i = 0$  при условии

$$i > t + 1,$$

код называется квазисовершенным.

Квазисовершенный код может исправлять:

- все ошибки веса не более  $t$ ;
- некоторые ошибки веса  $t + 1$ ;
- не может исправлять ни одной ошибки веса больше, чем  $t + 1$ .

**Теорема 3.1.** Коды Хэмминга являются совершенными кодами.

### 3.8. Вычисление минимального веса кода по проверочной матрице кода

**Теорема 3.2.** Блочный код с порождающей матрицей  $\mathbf{G}$  имеет минимальный вес  $\min \text{wt}(\mathbf{x})$ , и, следовательно, кодовое расстояние

$$d = \min \text{wt}(\mathbf{x}), \mathbf{x} \in \mathbf{G}.$$

тогда и только тогда, когда любые  $d - 1$  столбцов матрицы  $\mathbf{H}$  будет линейно независимы, но найдутся  $d$  линейно зависимых столбцов.

*Пример. 3.11.* Пусть основному уравнению кодирования (3.2) соответствует матрицы:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \text{ и } \mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Очевидно, имеются совокупности, состоящие из 2-х линейно зависимых столбцов матрицы  $\mathbf{H}$ . Отсюда,  $\min \text{wt}(\mathbf{x}) = 2$ ,  $d = 2$ . Все множество  $\{\mathbf{x}\}$  ненулевых кодовых слов кода  $[5, 2, 2]$ -кода представляется в виде

$$\mathbf{G}_+ = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

## 4. МЕТОДЫ ДЕКОДИРОВАНИЯ ЛИНЕЙНЫХ КОДОВ

Известны и применяются четыре основных метода декодирования.

1. Декодирование по синдрому.
2. Декодирование на основе принципа максимального правдоподобия.
3. Спектральное декодирование.
4. Мажоритарное декодирование или декодирование по большинству проверок.

Первый и четвертый методы применяются для коррекции независимых, модульных и пакетных ошибок кратностью  $t = 1 \div 4$ . Второй и четвертый методы декодирования используются, как правило, в радиоэлектронных системах, работающих при низких отношениях сигнал/помеха на входе декодера, сложной помеховой обстановке.

### 4.1. Декодирование кода на основе принципа максимального правдоподобия

*Определение 4.1.* Декодирование кода на основе вычисления вектора ошибки  $\mathbf{e}$  с наименьшим весом, называется декодированием на основе принципа максимального правдоподобия или декодированием в ближайшее кодовое слово.

#### 4.1.1. Декодирование кода по таблице смежных классов

Если на вход декодера поступил вектор

$$\mathbf{y} = (\mathbf{a}_i + \mathbf{x}), \mathbf{x} \in G,$$

то он должен принадлежать некоторому смежному классу  $(\mathbf{a}_i + \mathbf{x})$ . Если было передано кодовое слово  $\mathbf{x}'$ , то вектор ошибок  $\mathbf{e}$  равен

$$\mathbf{e} = \mathbf{y} - \mathbf{x}' = (\mathbf{a}_i + \mathbf{x} - \mathbf{x}') = (\mathbf{a}_i + \mathbf{x}''), \mathbf{x}'' \in G. \quad (4.1)$$

Из (4.1) следует, что возможными векторами ошибок

$$\mathbf{e} = (\mathbf{a}_i + \mathbf{x}'') \in (\mathbf{a}_i + G).$$

являются все векторы из смежного класса, содержащего  $\mathbf{y}$ .

Стратегия декодирования кодов будет следующей:

– необходимо выбрать из смежного класса, содержащего  $y$ , вектор  $e$  с наименьшим весом;

– декодировать  $y$  как  $x = y - e$ .

*Замечания*

1. Вектор из смежного класса, имеющий минимальный вес, называется лидером смежного класса. Лидер смежного класса есть вектор ошибок  $e$ .

2. Если имеется более одного вектора с минимальным весом, то в качестве лидера смежных классов выбирается любой из таких векторов.

3. В формуле (4.1) лидерами смежных классов являются векторы  $a_i$ .

*Пример 4.1.* Декодировать входной вектор  $y = [0110]$  по табл. 2.9 (см. пример 2.20)

Таблица 2.9  
Стандартное расположение для  $[4, 2, 2]$ -кода

0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

Решение. Вектор  $[0110]$  принадлежит смежному классу – второй строке табл. 2.9. Этому классу соответствует лидер смежного класса  $e = [1000]$ . Тогда переданным кодовым словом является

$$x^T = y^T - e^T = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Из табл. 2.9 также видно, что векторы  $[0100]$  и  $[0001]$  принадлежат одному и тому же смежному классу (третья строка табл. 2.9) поскольку их разность

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = x_2 \in G$$

есть кодовый вектор кода. Если рассматривать эти векторы в качестве векторов ошибок, то соответствующие им конфигурации ошибок (с минимальным одинаковым весом) не могут быть исправленными. Обе конфигурации можно выбрать в качестве лидеров своего смежного класса. Ввиду неоднозначного определения лидера смежного класса невозможно исправление ошибок этих конфигураций. О корректирующей способности рассмотренного кода говорят, что он исправляет не все ошибки веса 1, а только однократные определенной конфи-



гурации. Это справедливо во всех случаях, когда используются кодовые слова веса 2. Поскольку только минимальный вес  $\min wt(x) = d = 3$  является необходимым и достаточным условием для исправления всех конфигураций одичных ошибок.

## 4.2. Декодирование по синдрому

*Определение 4.2.* Вектор  $s = Hy^T$  называется синдромом вектора  $y$ , где  $y$  – вектор на входе декодера.

### Свойства синдрома

1. Синдром  $s$  – представляет собой вектор-столбец размером  $(n - k) \times 1$ .
2. Синдром  $s$  равен нулю, если и только если  $y$  – кодовое слово кода.

Пусть  $y = x + e$ , где  $x \in G$ , тогда

$$s = Hy^T = H(x + e)^T = Hx^T + He^T = He^T. \quad (4.2)$$

3. Если в кодовом слове имеются ошибки на позициях с номерами  $a, b, c, \dots$  так, что  $e = [0 \dots \overset{a}{\underset{\sim}{1}} \dots \overset{b}{\underset{\sim}{1}} 00 \dots \overset{c}{\underset{\sim}{1}} \dots 00]$ , то из (4.2) получаем, что

$$s = \sum_i e_i H_i = aH_a + bH_b + cH_c + \dots,$$

где  $H_i$  – это  $i$ -й столбец матрицы  $H$ . Вычисление синдрома можно рассматривать как линейное преобразование вектора ошибок. В качестве ядра преобразования выступает матрица  $H$ .

**Теорема 4.1.** Для двоичного кода синдром равен сумме тех столбцов матрицы  $H$ , где произошли ошибки.

*Замечание.* Вектор  $s$  называется синдромом, так как выделяет совокупность ошибок (гр. Syndrome – стечение). Синдром кодового слова является индикатором вектора ошибок.

**Теорема 4.2.** Имеется взаимно однозначное соответствие между синдромами и смежными классами, а именно: два вектора находятся в одном и том же смежном классе кода  $G$ , если и только если имеют один и тот же синдром  $s$ .

Ранее (теорема 2.8) было показано, что если векторы  $a$  и  $b$  находятся в одном смежном классе, то разность этих векторов всегда дает вектор принадлежащий коду, т. е.

$$a - b = x \in G.$$

Умножая (слева) это равенство на транспонированную проверочную мат-

рицу кода  $H^T$  и используя основное уравнение кодирования  $H^T G = 0$ , можно записать

$$\begin{aligned} H^T(a - b) &= H^T x = s = 0, \\ H^T a &= H^T b = s. \end{aligned}$$

Пусть имеется код мощностью  $M = q^k$ . Кодовые слова множества

$$\{x\} = \{x_1, x_2, \dots, x_M\}$$

передаются по каналу с шумами. На вход декодера поступает множество векторов

$$\{y\} = \{x + e\}.$$

Таблица стандартного расположения для кода  $\{x\}$  вместе со столбцом синдромов будет иметь вид

Таблица 4.1

Таблица стандартного расположения для кода

$x_1$	$\dots$	$x_s$	$\dots$	$x_M$	$s_0$
$e_1 + x_1$	$\dots$	$e_1 + x_s$	$\dots$	$e_1 + x_M$	$s_1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$e_i + x_1$	$\dots$	$e_i + x_s$	$\dots$	$e_i + x_M$	$s_i$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$e_t + x_1$	$\dots$	$e_t + x_s$	$\dots$	$e_t + x_M$	$s_t$

Здесь обозначены:

$e_1$  и  $e_t$  – векторы соответственно однократных и  $t$ -кратных ошибок;  
 $s_1$  и  $s_t$  – синдромы соответственно однократных и  $t$ -кратных ошибок.

Если принимаются векторы одного смежного класса, т. е.

$$y_s = e_i + x_s \text{ и}$$

$$y_j = e_i + x_j,$$

то соответствующие им синдромы, равны

$$s_{i_s} = H y_s^T = H(e_i + x_s)^T = H e_i^T + H x_s^T = H e_i^T,$$

$$s_{i_j} = H y_j^T = H(e_i + x_j)^T = H e_i^T + H x_j^T = H e_i^T.$$

Таким образом,  $s_{i_s} = s_{i_j}$

*Пример 4.2.* Найти соответствие между синдромами и смежными клас-

сами [4, 2, 2]-кода (см. пример 4.1). После вычисления синдромов, таблица стандартного расположения для кода примет вид

Таблица 4.2

Таблица стандартного расположения для кода

0000	1011	0101	1110	$s_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
1000	0011	1101	0110	$s_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
0100	1111	0001	1010	$s_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
0010	1001	0111	1100	$s_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Алгоритм декодирования по синдрому состоит в следующем.

1. Синдром определяет, в каком смежном классе находится принятый вектор  $y$ .
2. Зная смежный класс, определяется вектор ошибок  $e$ .
3. Определяется искомое кодовое слово.
4. Определяется передаваемое сообщение.

Алгоритм декодирования включает в себя выполнение операций:

- 1)  $s_{i_s} = Hy_s^T$ ;
- 2)  $s_{i_s} \rightarrow e_i$  (лидер смежного класса);
- 3)  $x_s = y_s - e_i$ ;
- 4)  $x_s \rightarrow u_s$  (сообщение).

По свойству 2 синдрома декодирование состоит в сравнении синдромов с нулем. Для этого:

- а) вычисляются проверочные символы, используя принятые информационные;
- б) сравниваются полученные проверочные символы с принятыми проверочными.

*Замечание.* Следствием а) является то, что синдромный декодер содержит кодирующее устройство.

*Вывод.* Если в качестве образующих смежных классов выбраны векторы, имеющие минимальный вес в своем смежном классе, то декодирование по синдрому совпадает с декодированием по минимуму расстояния Хэмминга. В этом случае обеспечивается минимальная вероятность ошибки декодирования в ДСК.

#### 4.2.1. Эффективность синдромного декодирования

Тот факт, что все элементы одного и того же смежного класса имеют один и тот же синдром, позволяет упростить процедуру декодирования в сравнении с декодированием по таблице стандартного расположения для кода. Сравним техническую сложность декодирования по таблице смежных классов и синдромного декодирования с использованием таблицы синдромов.

Процесс декодирования по таблице стандартного расположения для кода требует проведения операций сравнения входного слова со всеми элементами таблицы размером  $(2^r \times 2^k)$ . Для декодирования слова значностью  $n$  необходима память объемом

$$V = n2^{k+r} = n2^n \text{ бит.}$$

Таблица стандартного расположения содержит  $2^r$  строк (смежных классов) и  $2^r$  разных синдромов. Для хранения всех векторов столбца лидеров смежных классов потребуется объем памяти величиной

$$V_s = n2^r \text{ бит.}$$

Выигрыш по объему памяти декодера составит значение

$$L = \frac{V}{V_s} = \frac{n2^n}{n2^r} = 2^k \text{ бит.}$$

#### 4.3. Декодирование кода Хэмминга

Согласно теореме 4.1, синдром принятого вектора равен сумме тех столбцов матрицы **H**, где произошли ошибки. Для того, чтобы построить код, исправляющий одну ошибку, необходимо выбрать столбцы матрицы **H** следующим образом:

- столбцы должны быть ненулевыми (иначе ошибка в этой позиции будет необнаруженной);
- столбцы должны быть различными (иначе, если два столбца матрицы **H** одинаковы, то ошибки в соответствующих двух позициях будут неразличимы).

Если матрица **H** имеет  $r$  строк, то существует только  $2^r - 1$  допустимых ненулевых двоичных векторов длиной  $r$ . Проверочная матрица кода Хэмминга содержит столбцы, которые являются двоичными представлениями чисел от 1 до  $r$ .

Используя такую матрицу, в которой  $i$ -ый столбец **H<sub>i</sub>** равен двоичному представлению числа  $i$ , по синдрому легко определяется номер ошибочного символа принятого слова.

*Замечание.* Более простой технической реализацией алгоритма синдромного декодирования, чем по матрице классического вида кода Хэмминга не существует.

## 4.4. Вычисление вероятности ошибки декодирования

### 4.4.1. Распределение весов ошибок

На длине  $n$  кодового слова возможны следующие ошибки:

- одиночные,  $t = 1$ ;
- двойные,  $t = 2$ ;
- трехкратные,  $t = 3$  и т.д.

На длине кодового слова возможны различные конфигурации ошибок – векторы ошибок различного веса и формы. Обозначим  $t_i$  – число всех конфигураций ошибок веса  $i$ . Это число определяется биномиальным коэффициентом

$$C_n^i = \frac{A_n^i}{P_i} = \frac{n(n-1)\dots(n-i+1)}{i!}.$$

*Пример 4.3.* Найти распределение весов ошибок на длине  $n = 5$  кодового слова. Число конфигураций ошибок равно:

$$t_1 = C_5^1 = 5;$$

$$t_2 = C_5^2 = \frac{A_5^2}{P_2} = \frac{n(n-1)}{2!} = \frac{5 \cdot 4}{2} = 10;$$

$$t_3 = C_5^3 = \frac{A_5^3}{P_3} = \frac{n(n-1)(n-2)}{3!} = \frac{5 \cdot 4 \cdot 3}{2 \cdot 3} = 10;$$

$$t_4 = 5, t_5 = 1.$$

Суммарное число возможных конфигураций ошибок на длине  $n$  составит величину

$$t_\Sigma = \sum_i^n C_n^i = \sum_{i=1}^n t_i = 2^n - 1.$$

Для приведенного примера

$$t_\Sigma = t_1 + t_2 + t_3 + t_4 + t_5 = 31.$$

### 4.4.2. Вероятность ошибки декодирования кода

Ранее (см. (3.1)) было введено понятие вероятности вектора  $\mathbf{e}$  ошибок веса  $i$

$$\text{Prob}\{\mathbf{e}_{\text{wt}(i)}\} = p^i(1-p)^{n-i}.$$

Вероятность того, что в слове длиной  $n$  содержится хотя бы одна однократная ошибка, определяется выражением

$$\text{Prob}\{e_{\text{wt}(1_n)}\} = C_n^1 p^1 (1-p)^{n-1}.$$

*Пример 4.4.* Пусть  $p = 0,2$ ,  $n = 5$ . Вычислим вероятности возникновения хотя бы одной конфигурации однократной и двукратной ошибки.

Решение. Вероятность возникновения хотя бы одной конфигурации однократной

$$\text{Prob}\{e_{\text{wt}(1_n)}\} = C_n^1 p^1 (1-p)^{n-1} \cong 5 \cdot 0,081 \cong 0,4.$$

Вероятность того, что имеется хотя бы одна конфигурация двукратной ошибки

$$\text{Prob}\{e_{\text{wt}(2_n)}\} = C_n^2 p^2 (1-p)^{n-2} \cong 10 \cdot 0,01 \cong 0,1.$$

Вновь подтверждается, что ошибки малого веса необходимо обнаруживать и исправлять в первую очередь.

В общем случае, вероятность того, что в слове длиной  $n$  содержится хотя бы одна ошибка кратностью  $t$  или величины веса  $i$  равна

$$\text{Prob}\{e_{\text{wt}(i_n)}\} = C_n^i p^i (1-p)^{n-i}. \quad (4.3)$$

*Определение 4.3.* Вероятностью ошибки декодирования кодового слова называется вероятность появления неправильного кодового слова на выходе декодера.

Пусть имеется код мощностью  $M$ . Слова кода  $X^1, X^2, \dots, X^s, \dots, X^M$  передаются по каналу с равной вероятностью. Тогда средняя вероятность ошибки декодирования на кодовое слово определяется выражением

$$P_f = \frac{1}{M} \sum_{s=1}^M \text{Prob}\{\text{слово на выходе декодера} \neq X^s | X^s \text{ было передано}\}.$$

Декодирование осуществляется с использованием таблицы стандартного расположения для кода. Напомним, что выбранный декодером вектор ошибки всегда есть один из лидеров смежных классов. Ошибка декодирования происходит тогда и только тогда, когда вектор ошибок не является лидером смежного класса, т. е.

$$P_f = \text{Prob}\{e \neq \text{лидер смежного класса}\}.$$

Предположим, что в таблице стандартного расположения имеется  $a_i$  лидеров смежных классов веса  $i$ , т. е. число векторов ошибок веса  $i$  равно  $a_i$ . Используя (4.3), найдем вероятность того, что вектор  $e$  ошибки является лидером смежного класса

$$\text{Prob}\{e = \text{лидер смежного класса}\} = \sum_{i=0}^n a_i p^i (1-p)^{n-i}. \quad (4.4)$$

Выражение (4.4) характеризует правильное декодирование. Вероятность не появления события (4.4) представляется как вероятность ошибки декодирования (вероятность появления неправильного кодового слова на выходе декодера):

$$P_f = 1 - \sum_{i=0}^n a_i p^i (1-p)^{n-i}. \quad (4.5)$$

Если код имеет кодовое расстояние  $d = 2t + 1$ , он исправляет все  $t$ -кратные конфигурации ошибок. В этом случае диапазон весов  $i$  исправляемых векторов ошибок лежит в пределах  $0 \leq i \leq t$ .

Тогда, каждый вектор ошибок веса не более  $t$  является лидером смежного класса. Число лидеров смежного класса веса  $i$  для кода с минимальным расстоянием  $d = 2t + 1$  находится их выражения, полученного ранее для возможного числа ошибок кратностью  $t$  на длине  $n$  кодового слова, т. е.

$$a_i = C_n^i.$$

Большинство известных кодов могут исправлять некоторые конфигурации ошибок для значений  $i > t$ . Вычисление числа лидеров смежных классов для значений  $i > t$  не простая задача. Эти числа известны только для немногих кодов. Но если вероятность ошибки  $p$  в канале такова, что

$$(1-p) \cong 1,$$

$$p^i (1-p)^{n-i} \gg p^{i+1} (1-p)^{n-i-1},$$

в этом случае в формуле (4.5) пренебрегают членами с большими значениями  $i$ . Тогда вероятность  $P_f$  ошибки декодирования кодового слова на основе таблицы стандартного расположения определяется по формуле

$$P_f \cong 1 - \sum_{i=0}^t C_n^i p^i (1-p)^{n-i}. \quad (4.6)$$

*Пример 4.5.* Определить ошибку декодирования кодового слова кода  $[4, 2, 2]$ , используя стандартное расположение для кода, табл. 4.2, вероятность ошибки на символ  $p = 0,2$ .

Решение. По таблице находим число лидеров смежных классов веса  $i$ :

$$a_{i=0} = 1, a_{i=1} = 3.$$

По формуле (4.5) получаем

$$\begin{aligned} P_f &= 1 - \sum_{i=0}^n a_i p^i (1-p)^{n-i} = 1 - \sum_{i=0}^4 a_i p^i (1-p)^{4-i} = \\ &= 1 - p^0 (1-p)^4 - 3p^1 (1-p)^3 = 1 - 0,8^4 - 3 \cdot 0,2 \cdot 0,8^3 = 0,2832. \end{aligned}$$

*Упражнение 4.1.* Определить ошибку декодирования кода  $[6, 3, 3]$ , используя стандартное расположение для кода, представленного матрицей

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix},$$

вероятность ошибки на символ  $p = 0,01$ .



## Литература

1. Кудряшов, Б. Д. Основы теории кодирования: учеб. пособие / Б. Д. Кудряшов. – СПб. : БХВ-Петербург, 2016.
2. Кудряшов, Б. Д. Теория информации: учебник для вузов / Б. Д. Кудряшов. – СПб. : Питер, 2009.
3. Теория прикладного кодирования: учеб. пособие. В 2 т. / В. К. Конопелько [и др.]. – Минск : БГУИР, 2004.
4. Ватолин, Д. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М. : Диалог-МИФИ, 2002.
5. Луенбергер, Д. Дж. Информатика / Д. Дж. Луенбергер. – М. : Техносфера, 2008.
6. Митюхин, А. И., Пачинин В. И. Элементы алгебраических структур теории кодирования: учеб. пособие / А. И. Митюхин, Пачинин В. И. – Минск : БГУИР, 2012.
7. Вернер М. Основы кодирования: учебник для вузов / М. Вернер. – М. : Техносфера, 2004.
8. Андерсон Дж. А. Дискретная математика и комбинаторика / Дж. А. Андерсон; пер. с англ. – М. : Вильямс, 2004.
9. Лидл Р., Нидеррайдер Г. Конечные поля. В 2 т. / Р. Лидл, Г. Нидеррайдер. – М. : Мир, 1988.
10. Хаггарты, Р. Дискретная математика для программистов / Р. Хаггарты. – М. : Техносфера, 2005.

*Учебное издание*

**Митюхин** Анатолий Иванович

## ТЕОРИЯ ИНФОРМАЦИИ

### УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор *М. А. Зайцева*

Корректор *Е. Н. Батурчик*

Компьютерная правка, оригинал-макет *В. М. Задоля*

Подписано в печать	Формат 60×84 1/16	Бумага офсетная. Гарнитура «Таймс»
Отпечатано на ризографе	Усл. печ. л.    Уч.-изд. л.    9.0	Тираж 50 экз.    Заказ 150

Издатель и полиграфическое исполнение: учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники».  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий №1/238 от 24.03.2014  
№2/113 от 07.04.2014, №3/615 от 07.04.2014  
ЛП №02330/264 от 14.04.2014.  
220013, Минск, П. Бровки, 6