Министерство образования и науки РФ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Санкт-Петербургский политехнический университет»

Г. Н. Солопченко

ТЕОРИЯ ИНФОРМАЦИИ

Учебное пособие

Издание третье
переработанное и дополненное

Санкт-Петербург
Издательство Политехнического университета
2015

ОГЛАВЛЕНИЕ

1. ПРЕДМЕТ ТЕОРИИ ИНФОРМАЦИИ

Философское понятие “информация” является очень обширным. В общем случае это понятие характеризует внутреннюю организованность любой материальной системы. С этой точки зрения информация существует вне зависимости от того, воспринимается она или нет.

В бытовом смысле мы воспринимаем многообразную информацию о красоте природы или рукотворных объектов, о вкусе пищи, о холоде или тепле, о приятных или неприятных ощущениях. Огромное количество информации мы получаем из художественной и технической литературы, спектаклей и кинофильмов, телевизионных передач, путеводителей и рассказов экскурсоводов. Числовую форму приобретает метеоинформация или информация о расписании занятий и о движении транспорта. Наиболее естественную числовую форму имеет информация об объеме продукции, выпускаемой предприятием, ее стоимости, о движении денежных средств на предприятии или в банке, о котировках на бирже, а также измерительная информация, то есть результаты измерений, которые в соответствии с непременным метрологическим требованием должны снабжаться сообщением об их погрешности.

Теория информации, как формализованная математическая теория, оперирует только такой информацией, которая может быть выражена в числовой форме и проявляется в виде сигналов. Информацию, представленную в формализованном виде, позволяющем осуществить ее обработку с помощью технических средств, называют *данными*. После того как формализация осуществлена, сведения об авторстве, виде и важности информации, то есть потребительские сведения утрачиваются и остаются лишь абстрактные данные о количестве информации. Именно с подобного рода

данными оперирует теория информации независимо от их характера, содержания и конкретной ценности. Такое абстрагирование дает возможность применять результаты, полученные в теории информации, в различных прикладных областях.

Предметом теории информации является исследование и разработка общих теоретических вопросов в области:

получения и обработки информации;

кодирования, передачи и надежного восстановления информации;

хранения информации.

Обязательной подсистемой любой информационной системы является подсистема передачи и хранения данных. После этапа первичной обработки и подготовки к передаче, который носит на себе отпечаток специфики вида информации, на вход подсистемы передачи поступают данные, выраженные в математически абстрактной цифровой форме, поэтому анализ таких подсистем не зависит от специфического вида информации. Диапазон расстояний, на которые передается информация в этих подсистемах, очень широк: от единиц и даже долей метра до тысяч километров, как это происходит, например, в распределенных информационных системах со спутниковыми каналами связи. Распределенные информационные системы создаются на основе компьютерных сетей.

2. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

2.1. Разновидности информационных систем

Технические средства, которые обеспечивают получение, кодирование, передачу, восстановление, обработку и хранение информации – *средства информационной техники*.

Совокупность средств информационной техники и людей, объединенных для достижения определенных целей, называется *информационной системой*. Наиболее распространенными информационными системами являются:

измерительные информационные системы;

информационные системы управления и сигнализации;

информационные системы цифровой связи;

навигационные информационные системы;

банковские информационные системы;

справочные информационные системы;

другие виды информационных систем.

Возможны комбинированные информационные системы, в которых обращается информация разного вида. Например, измерительная информационная система может быть подсистемой управления и выполнять некоторые справочные функции.

В измерительных информационных системах, являющихся разновидностью информационных систем, устройства восприятия данных (data aquisition devices) – технические средства, а именно, средства измерений, приведенные в физическое взаимодействие с объектом и получающие объективную информацию в результате этого взаимодействия. Для преобразования этой информации в цифровую форму, как правило, применяют-

ся аналого-цифровые преобразователи (АЦП). Подготовка информации в данном случае заключается в нормализации данных, их предварительной обработке и преобразовании к виду (формату), удобному для передачи, обработки и хранения. В других системах, например банковских, функцию восприятия и подготовки информации исполняет человек, который эту информацию вводит в систему (в компьютер) с клавиатуры или сканера.

Воздействие на объект предусматривается в информационных системах управления и в некоторых измерительных информационных системах,

2.2. Структура систем получения и передачи информации

В теории информации рассматриваются информационные системы, состав и обобщенная структура которых представлена на рис. 1. На этом рисунке пунктиром выделены составные части информационных систем: *источник* информации (сообщений), *передающее устройство*, *приемное устройство* и *канал связи*.

Источник информации (сообщений) – совокупность первичного источника непрерывной или дискретной информации и преобразователя, служащего для представления информации в форме данных, удобных для

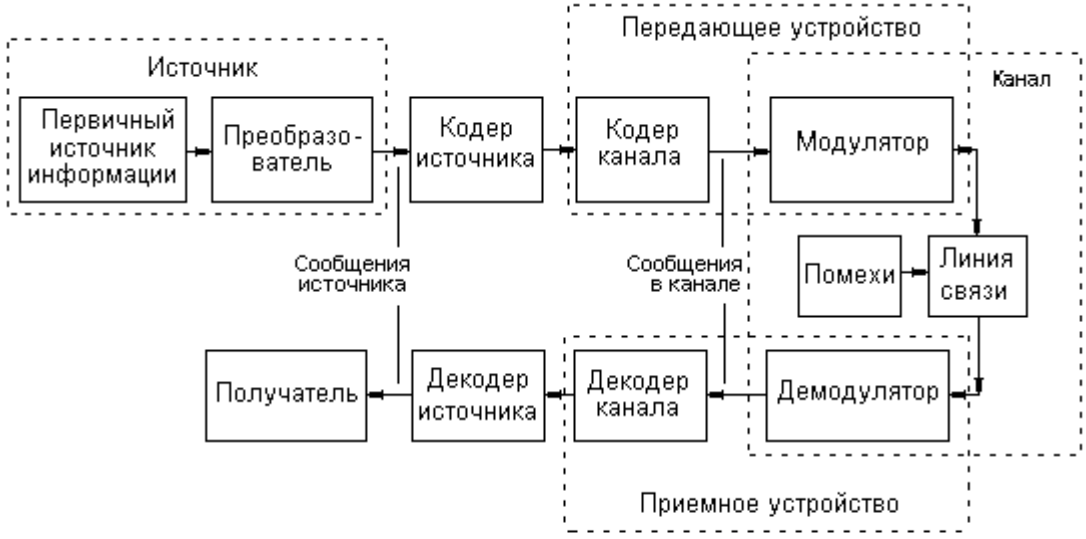

Рис. 1. Структурная схема системы передачи информации

дальнейшего преобразования. Сообщениями первичного источника являются различимые состояния или свойства объекта, которые содержат ин-

формацию, подлежащую последующим преобразованиям и передаче по линии связи.

Применительно к измерительным информационным системам (ИИС) первичным источником информации является рукотворный или природный объект исследований, с которым взаимодействует чувствительный элемент системы. Различимые состояния или свойства объекта оцифровываются, благодаря чему на выходе источника появляются сообщения источника.

Кроме того, первичным источником информации может быть, например, речь человека, печатный текст или изображение. Преобразователями в этих случаях являются микрофон или видеокамера с последующей оцифровкой. Это преобразование должно быть взаимнооднозначным.

Подготовленные таким образом данные или сообщения источника кодируются кодером источника таким образом, чтобы между сообщением первичного источника и *кодовой последовательностью* (*кодовыми словами*) на выходе кодера было обеспечено взаимнооднозначное соответствие. В силу взаимной однозначности понятие "сообщение источника" иногда относят и к кодовым последовательностям на выходе кодера, которые соответствуют сообщениям источника. По этой же причине символы, из которых состоят кодовые последовательности на выходе кодера источника, являются *информационными символами.*

Следующим элементом структурной схемы информационной системы является кодер канала, который входит в состав передающего устройства. Его задача состоит в том, чтобы обеспечить надежную передачу закодированного сообщения источника по *линии связи*, в которой из-за действия помех и других факторов возможны ошибки в передаче, а именно искажения отдельных символов кодовых последовательностей и, в том числе, стирания. Для обнаружения и(или) исправления этих ошибок кодер канала по специальным алгоритмам добавляет к информационным символам избыточные символы таким образом, чтобы декодер канала, стоящий в приемном устройстве, смог выполнить исправление ошибок и с большой вероятностью получить на выходе неискаженную кодовую последовательность, соответствующую передаваемому сообщению источника. При вза-

имно однозначном кодировании сообщения источника декодер канала предъявит получателю это неискаженное сообщение.

Кроме перечисленных элементов системы передачи информации упомянем представленные на рис. 1 *модулятор, линию связи* и *демодулятор*.

Модулятор преобразует кодовые символы в физические сигналы, удобные для передачи по *линии связи*. Модуляция называется *посимвольной*, если сигнал, соответствующий кодовому слову, представляет собой последовательность *элементарных сигналов*, соответствующих отдельным символам кодового слова.

В качестве линии связи может служить любая физическая среда (воздушная среда, металл, магнитная лента, магнитные диски, оптическая линия связи и др.). На сигналы в линии связи в большинстве случаев действуют помехи. В радиоканалах возможны затухания. Обычно эти помехи аналоговые и могут быть как аддитивными, так и мультипликативными.

На выходе линии связи осуществляется обратное преобразование полученного сигнала в кодовую последовательность. Это преобразование осуществляет *демодулятор* приемного устройства.

2.3. Информационные модели составных частей информационных систем

2.3.1. Информационная модель источника сообщений, марковские, стационарные и эргодические источники

В теории информации используются следующие характеристики источника сообщений.

1. Множество $\boldsymbol{N}$ возможных сообщений первичного источника и их количество K. Эти сообщения формулируются на языке соответствующих состояний или свойств объекта – первичного источника информации. Если информационной системой является измерительная система, то первичный источник информации – это физический объект, а сообщениями являются различимые значения физических параметров этого объекта. Если первич-

ный источник – газетный текст, то сообщениями первичного источника являются буквы, слова или фразы на языке, принятом для данной газеты.

2. Последующее преобразование сообщения первичного источника заключается в представлении этого сообщения с помощью каких – либо абстрактных формальных символов (например, цифр, букв или иных символов) и это его представление также именуется сообщением источника. Совокупность $\mathbf{X} = \{x_1,\ x_2,\ ...,\ x_M\}$ абстрактных символов $x_i \in \mathbf{X}$, которыми в результате преобразования представляется любое сообщение первичного источника, в теории информации называется *алфавитом источника*. Количество M всех его элементов называется *объемом алфавита источника* и обозначается $|\mathbf{X}| = M$. В дальнейшем применительно к этим элементам алфавита мы будем применять единый термин “буква”. Каждое j – е сообщение первичного источника в общем случае преобразуется в *последовательность s букв алфавита, то есть в* вектор размерности s*:*

$$\mathbf{x}_j = (x_1^{(j)},\ x_2^{(j)},\ ...,\ x_s^{(j)}) \in \mathrm{X}^s,\ \ j = 1, 2, 3, \ldots, K,$$

где X^s – множество слов в алфавите источника, представляющее собой декартово произведение s алфавитов X.

В частных случаях при определенном выборе алфавита этот вектор может вырождаться в скаляр, то есть сообщением источника, поступающим на вход кодера источника может оказаться один символ (буква). Верхний индекс у составляющих вектора $\mathbf{x}_j$ есть номер последовательности (сообщения), а нижний индекс означает только место буквы в сообщении, но не ее вид. В одном сообщении источника одна и та же буква может появляться не один раз. Примерами таких сообщений могут служить слова «сказка» и «рассказ».

Как уже было отмечено, каждое сообщение $\mathbf{x}_j$ может состоять из одной буквы. Сообщение длиной s = 1, содержащее один символ, есть *элементарное сообщение*. Общим свойством элементарного сообщения является его неделимость на более мелкие сообщения.

3. *Скорость создания информации* источником – *наименьшее количество знаков (букв), необходимое для взаимно однозначного представления*

сообщения первичного источника в алфавите **X**. Скорость создания информации обозначается *H*.

4. Мы будем рассматривать *дискретные источники* информации, то есть такие источники, которые в каждую единицу времени порождают на входе кодера источника *только одну букву сообщения*.

5. В теории информации принято, что первичный источник выбирает то или иное сообщение случайным образом. Это означает, что на выходе источника и на входе кодера источника та или иная буква алфавита появляется случайно. Поскольку каждое сообщение первичного источника формулируется преобразователем в буквах алфавита **X**, последовательность сообщений на входе кодера источника фактически представляет собой случайную последовательность букв этого алфавита. Источник *считается заданным*, если для любого i, $i = 1, 2,...$ и для любых последовательностей $\mathbf{x} = (x_i, x_{i+1},..., x_{i+t-1})$ длиной t, составленных из букв алфавита **X** и начинающихся в момент времени i, заданы совместные вероятности $p(\mathbf{x}) = p(x_i, x_{i+1}, ..., x_{i+t-1})$ появления этих последовательностей на выходе источника.

6. Источник называется *источником без памяти*, если два следующих друг за другом сообщения и (или) две следующих друг за другом буквы появляются на выходе источника независимо друг от друга. В этом случае

$$p(\mathbf{x}) = p(x_i, x_{i+1},..., x_{i+t-1}) = \prod_{j=0}^{t-1} p(x_{i+j}).$$

7. Сообщения *марковского источника* не являются независимыми, и вероятность появления каждого из них является условной вероятностью.

Вероятность появления сообщения $\mathbf{x}_j$ на выходе марковского источника порядка s зависит только от предыдущих s сообщений:

$$p(\mathbf{x}_j / \mathbf{x}_{j-1},..., \mathbf{x}_{j-i}).$$

8. Источник называется *стационарным*, если для любых i и j вероятности двух последовательностей сообщений одинаковой длины, одна из которых начинается в момент времени i, а другая – в момент времени j, совпадают. Марковский источник может быть стационарным.

9. Специальное свойство *эргодического источника* заключается в том, что вероятность любого сообщения этого источника может быть экспериментально определена по относительной частоте появления этого сообщения в одной достаточно протяженной во времени последовательности сообщений.

10. *Любой дискретный стационарный источник без памяти является эргодическим*.

В дальнейшем мы будем рассматривать исключительно *дискретные стационарные источники без памяти*.

2.3.2. Информационная модель кодера и декодера источника

Кодер источника – устройство, предназначенное для отображения множества сообщений источника на множество кодовых слов. Эта процедура называется *кодированием источника*.

Закодированные кодером сообщения источника в силу взаимной однозначности этого кодирования также считаются сообщениями источника.

Кодовое слово – последовательность *кодовых символов*, возникающая при кодировании одного сообщения источника. Семейство всех кодовых символов кодера источника есть алфавит кода. При двоичном кодировании, которое принято в настоящем курсе, алфавит кода А состоит из двух символов 0 и 1, то есть объем алфавита $L = |\mathrm{A}| = 2$. Семейство всех кодовых слов $\mathbf{m} = (m_1, m_2, ..., m_k)$, где $m_i \in \mathrm{A}$, на выходе кодера источника называется *кодом источника*. Каждое кодовое слово является элементом k – мерного пространства A^k, которое есть k – кратное декартово произведение алфавита кода. Количество кодовых слов, то есть количество элементов пространства A^k – *объем кода источника*, $|\mathrm{A}^k| = 2^k$. Объем кода источника не должен быть меньше количества сообщений источника: $|\mathrm{A}^k| \geq K$. Кодирование источников может быть *побуквенным* или *блоковым*. Все символы кода источника являются *информативными символами*.

При побуквенном кодировании длина каждого сообщения, поступающего на вход кодера источника, равна $s = 1$, элементарным сообщением является одна буква. Взаимно однозначное кодирование каждой буквы со-

общения первичного источника с помощью только двух символов "0" и "1" двоичного кода возможно с помощью нескольких символов этого кода. Например, если сообщение первичного источника выражено в десятичном коде, буквами которого являются цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, то для кодирования каждой этой буквы потребуется 4 двоичных символа. Так например, цифра 5 в двоичном коде выражается, как 0101, цифра 9 – как 1001.

Всякое слово первичного источника, то есть последовательность букв представляется вектором и может рассматриваться, как *блок* букв, всякая фраза есть последовательность слов и также может рассматриваться, как *блок* слов. При блоковом кодировании кодируются блоки букв или слов, для чего кодеру источника потребуется применить, возможно, более длинные двоичные последовательности. Например, при двоичном кодировании каждой цифры десятичного алфавита требуется четыре символа двоичного кода.

В теории информации приняты следующие характеристики кодера источника.

1. *Длина кодовых слов k*. Длина кодового слова кодера источника равна количеству *информационных символов* в кодовом слове, необходимых для кодирования буквы, слова или фразы первичного источника. При одинаковой длине кодовых слов код называется *равномерным*, а соответствующий процесс кодирования – *равномерным кодированием*. В противном случае код и кодирование называются *неравномерными*.

2. При длине двоичных кодовых слов, равной k, количество этих слов есть 2^k. Поскольку количество кодовых слов (или объем кода) не должно быть меньше количества сообщений источника, длина двоичного кодового слова должна удовлетворять неравенству $K \le 2^k$.

3. *Скорость кодирования источника* $R_{\text{ист}}$ измеряется в относительных единицах, а именно, в количестве кодовых символов кодера источника , которые необходимы для кодирования одной буквы сообщения первичного источника, возникающей на входе кодера. При двоичном кодировании сообщений первичного источника длиной s букв, скорость кодирования источника

$$R_{\text{ист}} = \frac{\log_2 K}{s} \text{ бит/буква.} \qquad (1)$$

В тех случаях, когда $\log_2 K = k$, то есть, когда $K = 2^k$,

$$R_{\text{ист}} = k/s. \qquad (2)$$

При *побуквенном кодировании*, когда каждая буква сообщения кодируется собственным кодовым словом, тогда $s = 1$ и скорость кодирования источника равна $R_{\text{ист}} = k$. В недавно приведенном примере двоичного кодирования десятичных цифр скорость кодирования равна 4.

При неравномерном кодировании кодовые слова имеют неодинаковую длину. В этих случаях естественно определить среднюю скорость кодирования источника, применив для этого вероятность появления j – го кодового слова p_j:

$$R_{\text{ист}} = \frac{k_{\text{ср}}}{s} = \frac{1}{s}\sum_{j=1}^{K} p_j k_j , \qquad (3)$$

где $k_{\text{ср}}$ – средняя длина кода, k_j – длина j – го кодового слова s – длина сообщений первичного источника.

Декодер источника выполняет функцию, обратную функции кодера источника. При двоичном кодировании, принятом в настоящем пособии, алфавитом кодовых слов на входе декодера является двоичный алфавит объема $L = 2$ с символами 0 и 1. Для обеспечения взаимно однозначного кодирования и декодирования кодер и декодер проектируются совместно. Поэтому характеристики декодера источника сопряжены с характеристиками кодера источника. Важной технической характеристикой декодеров является вероятность правильного декодирования получаемых сообщений.

2.3.3. Информационная модель кодера и декодера канала

Из-за действия в канале помех возможны искажения или стирания символов, передаваемое сообщение может быть искажено. Для обеспечения возможности обнаружения или исправления этих искажений в отличие от кодера источника кодер канала вводит в код передаваемых сообщений r избыточных символов по специальному алгоритму. Поэтому длина

кода в канале n превышает длину кода источника k: $n = k + r > k$. Алфавит А кода, передаваемого по каналу связи, как и алфавит кода источника – двоичный, его символы – 0 и 1, объем алфавита $L = |\mathrm{A}| = 2$. Все возможные кодовые последовательности (коды) длины n являются элементами n − мерного пространства A^n, являющегося n – кратным декартовым произведением множества А. Объем пространства $|\mathrm{A}^n| = 2^n$.

В теории информации приняты следующие характеристики кодера канала.

1. Длина кодового слова n – количество символов в кодовом слове.

2. *Скорость кодирования в канале* $R_{\text{к}}$ представляет собой относительное количество кодовых символов кода канала, необходимых для кодирования и передачи одного символа кода источника:

$$R_{\text{к}} = \frac{\log_2 K}{n} \text{ бит/символ.} \qquad (4)$$

Другими словами, скорость кодирования в канале – относительное информационное содержание каждого символа кода канала.

Когда избыточности нет, и длина кода n равна $\log_2 K$, $K = 2^n$, скорость кодирования в канале равна 1. В реальных ситуациях, когда для обнаружения и (или) исправления ошибок в кодовые слова вводят избыточные символы, длина кодового слова увеличивается, $n > \log_2 K$, скорость кодирования уменьшается: $R_{\text{к}} < 1$.

Декодер канала выполняет функцию, обратную функции кодера канала. Основными характеристиками декодера канала являются вероятность обнаружения ошибки и вероятность исправления ошибки передачи.

В завершение описания математических моделей отдельных компонентов информационных систем, ответственных за получение и подготовку информации с целью ее безошибочной передачи по линиям связи, на рис. 2 приводится обобщенная схема, иллюстрирующая основное содержание информационных моделей источника информации, кодера источника и кодера канала при взаимно однозначном побуквенном кодировании сообщений источника. Количество букв в слове (сообщении первичного источника) равно s. Каждая буква x_j сообщения источника кодируется k

информационными символами. Затем для того, чтобы обеспечить надежную передачу этих кодов по каналу связи для обнаружения и исправления возможных ошибок декодером канала на приемном конце, в код каждой буквы сообщения источника кодер канала вводит r избыточных символов. Таким образом формируются кодовые слова длиной $n = k + r$, являющиеся элементами n - мерного пространства кодовых последовательностей A^n.

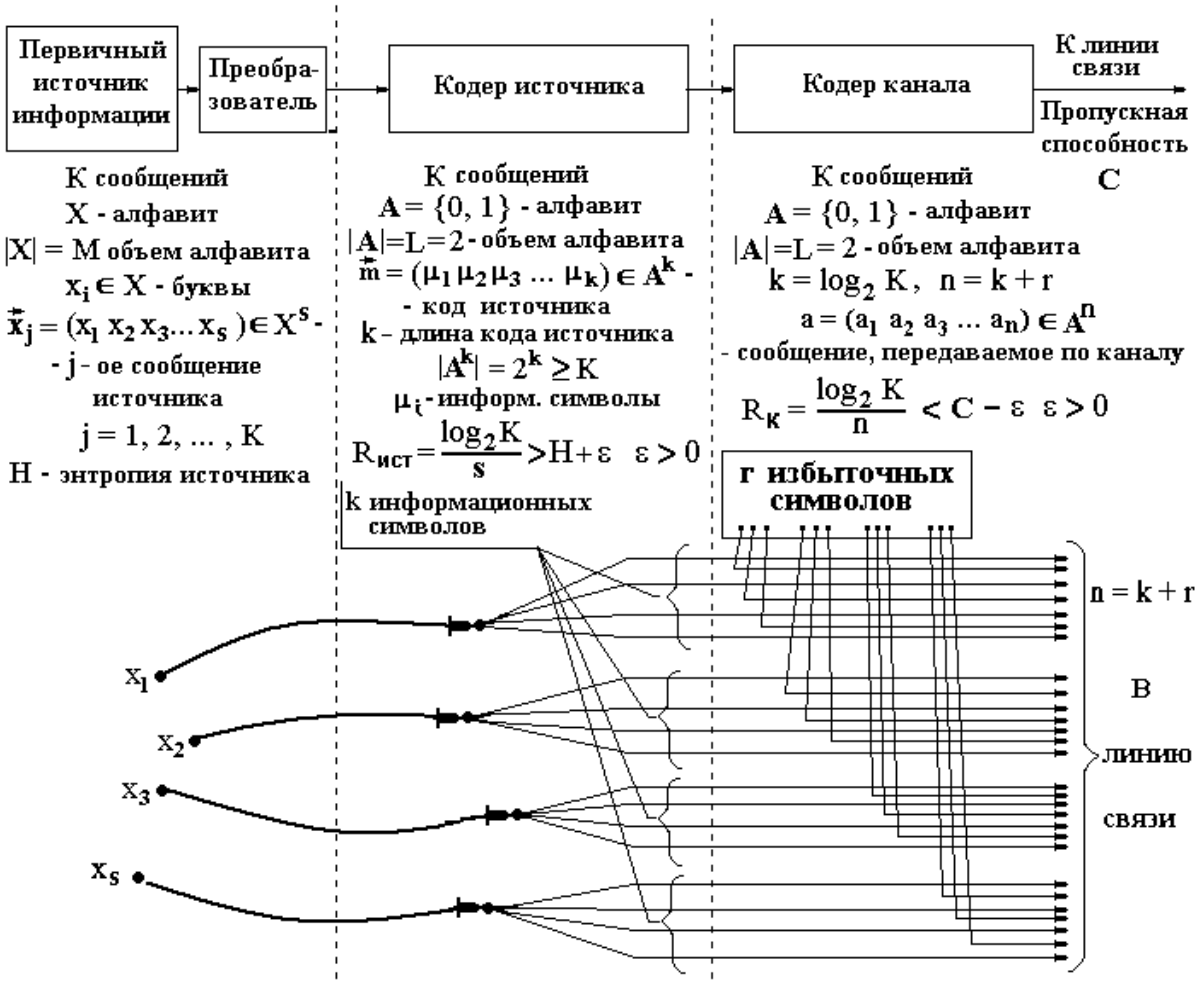

Рис. 2. Обобщенная схема получения и кодирования информации в информационных системах

2.3.4. Информационная модель канала связи

Информационная модель канала связи относится к каналу в целом, состоящего из модулятора, линии связи и демодулятора (см. рис. 2).

При дискретной передаче по каналу связи двоичных символов 0 и 1 от источника "без памяти" основными характеристиками канала связи являются условные вероятности ошибочной передачи символов:

$p(1/0)$ – вероятность искажения переданного символа 0,

$p(0/1)$ – вероятность искажения переданного символа 1.

Если эти вероятности не зависят от исхода передачи предыдущих символов, такие каналы называются каналами "без памяти".

Если $p(1/0) = p(0/1) = p,$ каналы называются симметричными.

В дальнейшем будут в основном рассматриваться дискретные двоичные симметричные каналы без памяти.

Обобщенной характеристикой каналов связи является *показатель информационной пропускной способности канала C.*

2.4. Примеры кодирования источника сообщений

В первом примере в качестве источника сообщений принята телеграмма и рассматриваются два способа кодирования телеграмм. Второй пример относится к методам кодирования измерительной информации в многоканальных измерительных информационных системах.

П р и м е р 1. Кодирование телеграмм.

При кодировании каждой буквы любой телеграммы (при *побуквенном кодировании*) алфавитом **X** источника является совокупность букв русского и латинского алфавитов, включая знаки препинания, знак переноса и пробел. Допустим, что объем такого алфавита $|\mathbf{X}| = M = 64$.

Кодер источника отображает множество букв этого алфавита на множество двоичных кодовых слов, представляющих собой последовательности кодовых символов 0 или 1. При побуквенном двоичном кодировании телеграммы для представления каждой буквы предполагаемого алфавита **X**, содержащего 64 русские и латинские буквы а также знаки препинания, пробелы, скобки, кавычки, требуется $k = \log_2 64 = 6$ двоичных разрядов.

Это длина кодового слова, кодирующего одну букву, и, следовательно, $s = 1$, значит скорость кодирования составляет $R_{\text{ист}} = k = 6$ бит/буква.

Пусть средняя длина слова телеграммы равна 8 буквам. Тогда среднее количество двоичных символов, затрачиваемых на одно слово, равно $6 \times 8 = 48$ символов. Если принять, что средняя длина телеграммы составляет 20 слов, то для кодирования одной телеграммы потребуется в среднем $6 \times 8 \times 20 = 960$ символов двоичного алфавита. Такое кодирование неэкономно, но обеспечивает однозначное кодирование и декодирование, поскольку таким образом можно однозначно закодировать, а по полученным двоичным последовательностям однозначно декодировать любую телеграмму.

Рассмотрим иной способ кодирования телеграмм. Пусть в результате статистического анализа совокупности телеграмм установлено, что подавляющее большинство телеграмм может быть передано некоторым набором из 8192 типичных слов. В этом случае элементами (буквами) алфавита источника и его элементарными сообщениями служат все типичные слова, поэтому количество элементарных сообщений первичного источника $K = 8192 = 2^{13}$. Кодируется каждая такая "буква" отдельным кодовым словом. Такое кодирование называется *блоковым*. При блоковом кодировании элементарными сообщениями первичного источника являются блоки, количество которых равно количеству типичных слов (сообщений), а это значит, что объем алфавита первичного источника возрастает до 8192, и для кодирования любого слова из предусмотренного набора необходимо не менее, чем $k = \log_2 8192 = 13$ двоичных символов.

Если средняя длина слова составляет 8 букв, то среднее количество кодовых символов, приходящееся на одну букву телеграммы, а значит, и скорость кодирования телеграммы (то есть первичного источника) составляет $R_{\text{ист}} = 13/8$ бит/буква исходного текста сообщения. Количество кодовых символов на одно слово (13 символов) существенно меньше, чем 48 символов при побуквенном кодировании. По этой причине для блочного кодирования телеграммы потребуется в среднем $13 \times 20 = 260$ двоичных символов, а не 960, как при побуквенном кодировании. Скорость блокового кодирования существенно меньше. Однако однозначность кодиро-

вания и декодирования утрачивается, поскольку существует отличная от нуля вероятность появления нетипичного слова или бессмысленное сочетание букв, для которого не предусмотрена никакая кодовая последовательность, и поэтому оно будет восприниматься как "ошибка". В подобных случаях множество возможных сообщений распадается на два непересекающихся подмножества. Первое из них – подмножество сообщений, которым взаимно однозначно сопоставлены кодовые слова. Оно называется *множеством однозначно кодируемых и декодируемых сообщений*. Второе подмножество – это подмножество сообщений, которым не соответствует ни одно из кодовых слов. Все сообщения из второго подмножества кодируются только одним словом (например, “ошибка”), и эти сообщения образуют в совокупности *множество неоднозначно кодируемых и декодируемых сообщений*. Объем этого множества гораздо больше объема множества однозначно кодируемых и декодируемых сообщений, он равен бесконечности.

Хотя свойство однозначного кодирования и декодирования является важным свойством кода источника, на практике для повышения эффективности кодирования/декодирования иногда допускают возможность ошибки, стремясь сделать ее достаточно маловероятной.

П р и м е р 2. Пример кодирования сообщений в многоканальной измерительной информационной системе (ИИС).

Как уже упоминалось, первичным источником информации для ИИС является объект измерений. Преобразователями этой информации являются датчики, фильтры, усилители, линеаризаторы, аналоговый коммутатор, масштабные преобразователи и аналого-цифровой преобразователь или цифровой вольтметр.

Вначале допустим, что с целью измерения параметров объекта применяется цифровой измерительный прибор. В режиме устойчивого функционирования объекта можно считать, что источник является стационарным и эргодическим. Цифровой измерительный прибор представляет сообщения источника, то есть результаты измерений в десятичном коде. Пусть количество разрядов этого кода равно 4. В этом случае полное количество различаемых сообщений первичного источника $K = 10^4$. Алфа-

вит источника **X** = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Буквами (знаками) этих сообщений являются цифры от 0 до 9 включительно, поэтому объем алфавита источника |**X**| = 10. Распределение вероятностей выбора источником букв алфавита будем считать равномерным. С целью подготовки такой информации к передаче по каналу кодер источника должен закодировать эти сообщения двоичным кодом. Как правило, в реальных ИИС выполняется побуквенное кодирование. Для обеспечения взаимно однозначного кодирования и декодирования каждой буквы (знака, символа, цифры) в двоичном коде требуется четыре символа 0 или 1 (табл. 1).

В этом случае элементарным сообщением источника является одно десятичное число (буква), то есть s = 1, на его кодирование следует истратить 4 символа (буквы, знака) кода. Это означает, что скорость кодирования $R_{\text{ист}}$ = 4 бит / буква.

Таблица 1

Побуквенное кодирование показаний цифрового прибора

Элемент алфавита источника	Двоичное представление	Элемент алфавита источника	Двоичное представление	Элемент алфавита источника	Двоичное представление
0	0000	3	0011	6	0110
1	0001	4	0100	7	0111
2	0010	5	0101	8	1000
				9	1001

Пусть, например, сообщение источника – десятичное число x = 3941. Тогда при подобном кодировании на выходе кодера источника получим последовательность

0011100101000001.

В данной ситуации для кодирования всего сообщения необходимо 16 символов. На каждую букву сообщения источника приходится 4 двоичных символа, что соответствует скорости кодирования источника, равной 4 бит/букву.

На практике в современных ИИС чаще всего используются двоичные АЦП. В этом случае кодируются не отдельные буквы, но все сообщение в

целом. Длина кодового слова k кода источника не должна быть меньше, чем $\log_2 K$, то есть $k \geq \log_2 10^4 = 4\log_2 10 = 4 \cdot 3.322 = 13.288$, или k = 14.

При таком способе кодирования скорость кодирования составит

$$R_{\text{ист}} = 14/4 = 3{,}5 \text{ бит / буква.}$$

Взаимно однозначное кодирование и декодирование в этом случае достижимо.

Из приведенных примеров видно, что отказ от побуквенного кодирования и переход к блоковому кодированию сообщений позволяет сократить в среднем длину кодового слова кода источника.

2.5. Условия Шеннона достоверной передачи информации. Вопросы теории информации и кодирования, рассматриваемые в настоящем пособии

Условия достоверной передачи информации с заданной малой вероятностью искажений в информационных системах заключаются в согласовании

информационной производительности (скорости создания информации) источника и скорости кодирования источника;

скорости кодирования в канале и информационной пропускной способности канала.

Указанные условия выражаются следующими утверждениями, основанными на двух фундаментальных теоремах Шеннона.

1. Д л я и с т о ч н и к а. Объективно существует показатель информационной производительности H источника, такой, что если скорость кодирования источника

$$R_{\text{ист}} \geq H + \varepsilon ,$$

где ε - сколь угодно малое положительное число, то сообщение может быть закодировано так, что вероятность ошибки кодирования и декодирования будет произвольно малой: $P_{\text{ош}} \leq \delta, \quad \delta > 0$. Это содержание прямой теоремы Шеннона об источнике.

Обратная теорема гласит: если скорость кодирования источника $R_{\text{ист}} \leq H$, то не может быть найден код, обеспечивающий кодирование и декодирование сообщений источника с произвольно малой ошибкой, бо-

лее того, вероятность ошибки кодирования и декодирования может даже стремиться к единице при увеличении длины блоков.

2. Для канала связи. Объективно существует показатель информационной пропускной способности канала С, такой, что если скорость кодирования в канале

$$R_{\text{к}} \le C - \varepsilon, \quad \varepsilon > 0 \ ,$$

где ε - сколь угодно малое положительное число, то существует такой способ кодирования в канале, который позволяет обеспечить передачу информации источника по каналу со сколь угодно малой вероятностью ошибки.

И напротив, если скорость кодирования в канале $R_{\text{к}} > C$, то не существует способа кодирования, позволяющего вести передачу информации по каналу связи со сколь угодно малой вероятностью ошибки.

В данном пособии излагаются начальные сведения из теории информации, необходимые для создания избыточных кодов, а также для построения простейших кодеров и декодеров, которые обеспечивают обнаружение и коррекцию ошибок, возникающих при передаче закодированных сообщений по линиям связи.

3. ЭНТРОПИЯ ВЕРОЯТНОСТНОЙ СХЕМЫ. ЭНТРОПИЯ И КОЛИЧЕСТВО ИНФОРМАЦИИ

3.1. Энтропия источников, дискретный источник без памяти

Рассмотрим элементарные сообщения, считая, что они могут представлять собой отдельные символы. Важно то, что источник из множества сообщений **N** в каждый момент времени выбирает одно i – е сообщение $\mathbf{x}_i = x_i$ в виде одной буквы, i= 1,2, **...,** K, c вероятностью $p(\mathbf{x}_i) = p(x_i)$ независимо от предыдущего $(i - 1)$ – го сообщения. Как это уже было отмечено, такой источник называется *дискретным источником без памяти.* Выбор сообщения из полного множества сообщений – случайное событие. В совокупности эти события составляют полную группу событий, поэтому сумма их вероятностей равна 1, а совокупность $\{p(x_i)\}$ этих вероятностей есть распределение вероятностей на множестве сообщений. Если каждое сообщение есть одна буква из алфавита **X**, то $|\mathbf{X}| = |\mathbf{N}| = L = K$ и

$$x_i \in \mathbf{N},\ i = 1, 2, ..., \mathrm{K},\ \bigcup_{i=1}^{K} x_i = \mathbf{N},\quad p(x_i) = p_i,\ \sum_{i=1}^{K} p_i = 1\ .$$

К.Шеннон ввел следующее определение *количества собственной информации* $I(x_j)$, содержащегося в элементе x_i:

$$I(x_i) = -\log p(x_i) = -\log p_i\ . \qquad (5)$$

Кроме того, К. Шеннон ввел определение *средней собственной информации*, которая вычисляется по всему ансамблю **X** с весами, равными вероятностям соответствующих элементов:

$$H(\mathbf{X}) = \sum_{i=1}^{K} p_i I(x_i) = -\sum_{i=1}^{K} p_i \log p_i\ .$$

Эта усредненная по ансамблю собственная информация всех его элементов называется *энтропией дискретного ансамбля*. Предложенная мера

энтропией названа не случайно. Структура выражения энтропии источника информации совпадает с выражением Больцмана для энтропии физической системы, соответствующей второму закону термодинамики:

$$H = -\sum_{i=1}^{N_m} \frac{m_i}{N_m} \ln \frac{m_i}{N_m},$$

где N_m – число молекул в данном замкнутом пространстве, m_i – число молекул, обладающих скоростью из интервала $(v_i, v_i + \Delta v)$, m_i / N_m – вероятность того, что молекула обладает скоростью из интервала $(v_i, v_i + \Delta v)$.

При равномерном распределении вероятностей выбора элементов из алфавита **X**, то есть при $p(x_i) = 1/K$ энтропия

$$H(\mathbf{X}) = -\sum_{i=1}^{K} \frac{1}{K} \log_2 \frac{1}{K} = \log_2 k .$$

Если для измерения энтропии мы выберем двоичные единицы (*bit*), то выражение для энтропии алфавита источника примет вид:

$$H(\mathbf{X}) = -\sum_{i=1}^{K} p_i \log_2 p_i , \qquad (6)$$

в котором, подразумевая далее только двоичное исчисление, мы иногда будем опускать обозначение основания логарифма.

3.2. Формальные свойства энтропии, аксиомы Хинчина и Фаддеева

1. Энтропия является вещественным неотрицательным функционалом распределения вероятностей $\{p_i\}$, поскольку при изменении p_i в пределах $0 < p_i \le 1$ величина $-\log p_i \ge 0$, и значит, $-p_i \log p_i \ge 0$.

2. Энтропия – ограниченный и непрерывный функционал от распределения вероятностей $\{p_i\}$. В самом деле при изменении p_i в пределах $0 < p_i \le 1$ ограниченность и непрерывность слагаемых $-p_i \log p_i$ очевидна. Остается определить предел $\lim\limits_{p_i \to 0} (-p_i \log p_i)$.

В качестве основания логарифма примем 2, и воспользовавшись правилом Лопиталя, получим:

$$\lim_{p_i \to 0} \left(- p_i \log_2 p_i \right) = \lim_{p_i \to 0} \frac{\log_2 (1/p_i)}{1/p_i} = \log_2 e \lim_{p_i \to 0} \left(\frac{p_i}{1 \cdot p_i^2} \cdot \frac{p_i^2}{1} \right) = 0 . \qquad (7)$$

Равенство (7) позволяет доопределить выражение $- p_i \log_2 p_i$ по непрерывности в нуле, как 0 при $p_i \to 0$. Из этого свойства следует аксиома Хинчина о том, что добавление к ансамблю сообщения с нулевой вероятностью не изменяет энтропии этого ансамбля.

3. Энтропия обращается в нуль лишь тогда, когда вероятность одного сообщения равна 1. В этом случае говорят о полностью детерминированном источнике и об отсутствии неопределенности в нем, так как наблюдатель знает о сообщении источника до момента его наблюдения.

4. Энтропия максимальна, когда все сообщения источника равновероятны. Докажем это положение двумя способами.

Первый способ – нахождение условного максимума с помощью неопределенных множителей Лагранжа.

Условие задачи: найти $\max_{p_i} \left(- \sum_{i=1}^{K} p_i \log_2 p_i \right)$ при условии $\sum_{i=1}^{K} p_i = 1$.

В соответствии с принятым способом эта задача переформулируется в задачу нахождения безусловного экстремума:

$$\max_{p_i, \lambda} \left(- \sum_{i=1}^{K} p_i \log_2 p_i + \lambda \left(\sum_{i=1}^{K} p_i - 1 \right) \right) = \max_{p_i, \lambda} \left(- \log_2 e \sum_{i=1}^{K} p_i \ln p_i + \lambda \left(\sum_{i=1}^{k} p_i - 1 \right) \right).$$

Экстремум достигается в точках, где производная этого выражения обращается в ноль одновременно для каждого значения p_i, $i = 1,2,...,K$, и по λ

производные по p_i: $\quad - \log_2 e \quad (\ln p_i + 1) + \lambda = 0 ,$

производные по λ: $\quad \sum_{i=1}^{K} p_i - 1 = 0 .$

Из первого равенства следует, что $\ln p_i = \lambda \ln 2 - 1$, то есть все значения p_i одинаковы, а из второго равенства следует, что эти вероятности равны $p_i = \frac{1}{K}$, что и требовалось доказать.

Второй способ основан на том, что для натуральных логарифмов справедливо неравенство $\ln x \le x - 1$, равенство имеет место только при $x = 1$.

Это неравенство непосредственно следует из графика рис.3.

Докажем справедливость неравенства $H(\mathbf{X}) \le \log_2 K$, или $H(\mathbf{X}) - \log_2 K \le 0$, что то же самое.

Доказательство. Заметим, что $\log_2 x = (\log_2 e)\ln x$. Поэтому

$$H(\mathbf{X}) - \log_2 K = -\sum_{i=1}^{K} p_i \log_2 p_i - \sum_{i=1}^{K} p_i \log_2 K = -\log_2 e \sum_{i=1}^{K} p_i \ln(p_i K) =$$

$$= \log_2 e \sum_{i=1}^{K} p_i \ln \frac{1}{p_i K} \le \log_2 e \sum_{i=1}^{K} p_i \frac{1 - p_i K}{p_i K} = \log_2 e \sum_{i=1}^{K} \left(\frac{1}{K} - p_i \right) = \log_2 e \left[\sum_{i=1}^{K} \frac{1}{K} - \sum_{i=1}^{K} p_i \right] = 0,$$

что и требовалось доказать.

Равенство здесь, как и ранее, достигается при $p_i = \frac{1}{K}$.

Приведенные приемы доказательства в дальнейшем будут использоваться неоднократно.

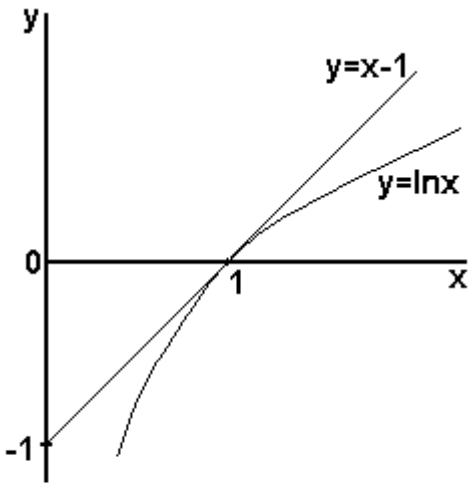

Рис. 3. К доказательству свойства 4 энтропии

5. Энтропия двоичного источника изменяется от нуля до единицы, достигая максимума, равного *1*, при равенстве вероятностей сообщений.

Если алфавит двоичного источника состоит из двух букв (символов) $x_1 = 0, \quad x_2 = 1, \quad x_1 \bigcup x_2 = \mathbf{A}$, тогда $p_1 = 1 - p_2$, и наоборот $p_2 = 1 - p_1$, то есть если $p_1 = p$, то $p_2 = 1 - p$.

Энтропия этого ансамбля есть следующая функция от p :

$$H(\mathbf{X}) = -\left(p \log_2 p + (1 - p)\log_2 (1 - p)\right). \qquad (8)$$

График этой функции представлен на рис. 4. При $p = 0$ и $p = 1$ по ранее доказанному (7) энтропия равна 0. При $p = 0.5$ энтропия достигает максимума, $H(\mathbf{A}) = 1$. Последнее равенство соответствует свойству 4, поскольку здесь $L = 2$, и энтропия равна $\log_2 L = 1$. Кроме того, график еще раз свидетельствует о непрерывной зависимости энтропии от вероятности p.

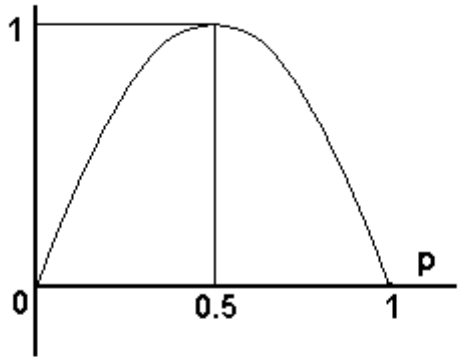

Рис. 4. Зависимость энтропии ансамбля из двух сообщений от вероятности p

6. Энтропия нескольких статистически независимых ансамблей равна сумме энтропий этих ансамблей (свойство аддитивности функционала энтропии).

Сначала рассмотрим два ансамбля (два источника), **U** и **V.** Обозначим элементы первого из них u_i, i=1, 2, ...,K, элементы второго – v_j, j=1, 2, ..., N. Множество **UV**, содержащее все пары ($u_i \cap v_j$), – декартово произведение двух множеств (ансамблей) **U** и **V**. Пусть совместные распределения вероятностей этих пар будут $p(u_i, v_j)$. Тогда энтропия объединения этих ансамблей, то есть совокупности пар ($u_i \cap v_j$) выражается равенством

$$H(\mathbf{UV}) = -\sum_{j=1}^{N}\sum_{i=1}^{K} p(u_i v_j)\log_2 p(u_i v_j). \qquad (9)$$

В силу статистической независимости ансамблей $p(u_i, v_j) = p(u_i)p(v_j)$, и поэтому

$$H(\mathbf{UV}) = -\sum_{j=1}^{N}\sum_{i=1}^{K} p(u_i)p(v_j)\log_2\left(p(u_i)p(v_j)\right) =$$

$$= -\sum_{i=1}^{K} p(u_i)\log_2 p(u_i)\sum_{j=1}^{N} p(v_j) - \sum_{j=1}^{N} p(v_j)\log_2 p(v_j)\sum_{i=1}^{K} p(u_i).$$

Но поскольку $\sum_{i=1}^{K} p(u_i) = 1$ и $\sum_{j=1}^{N} p(v_j) = 1$,

$$H(\mathbf{UV}) = H(\mathbf{U}) + H(\mathbf{V}) = H(\mathbf{VU}). \qquad (10)$$

Эти же преобразования совершенно аналогично выполняются и для трех, и для любого другого количества ансамблей, то есть

$$H(\mathbf{UV...\ Z}) = H(\mathbf{U}) + H(\mathbf{V}) + \ldots + H(\mathbf{Z}). \qquad (11)$$

Перечисленные выше свойства энтропии некоторые авторы называют аксиомами Хинчина. Аксиомы, равносильные по отношению к этим аксиомам, сформулировал Д. К. Фаддеев и опубликовал их в одном номере журнала с А. Я. Хинчиным (Успехи математических наук, 1956. Т. 11, №1). Однако, вообще говоря, данные утверждения считать аксиомами затруднительно, поскольку все они доказываются.

Поскольку здесь и далее в пособии рассматриваются только двоичные ансамбли и логарифмы с основанием 2, мы будем опускать это обозначения, считая основания всех логарифмов равным 2.

3.3. Условная энтропия и ее свойства, третья аксиома Хинчина

Если два ансамбля **U** и **V** статистически зависимы, то совместное распределение на ансамбле пар ($u_i \cap v_j$) может быть представлено в виде:

$$p(u_i, v_j) = p(u_i)\,p(v_j / u_i) = p(v_j)\,p(u_i / v_j).$$

В этом случае энтропия объединения ансамблей принимает вид

$$H(\mathbf{UV}) = -\sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i)\,p(v_j / u_i)\log\left(p(u_i)\,p(v_j / u_i)\right) =$$

$$= -\sum_{i=1}^{K} p(u_i)\log p(u_i)\sum_{j=1}^{N}\frac{p(u_i v_j)}{p(u_i)} - \sum_{i=1}^{K} p(u_i)\sum_{j=1}^{N} p(v_j / u_i)\log p(v_j / u_i).$$

В этом выражении

$$\sum_{j=1}^{N}\frac{p(u_i v_j)}{p(u_i)} = \sum_{j=1}^{N} p(v_j / u_i) = \frac{1}{p(u_i)}\sum_{j=1}^{N} p(u_i v_j) = \frac{p(u_i)}{p(u_i)} = 1.$$

Поэтому

$$H(\mathbf{UV}) = -\sum_{i=1}^{K} p(u_i)\log p(u_i) - \sum_{i=1}^{K} p(u_i)\sum_{j=1}^{N} p(v_j / u_i)\log p(v_j / u_i).$$

Первое слагаемое в правой части есть знакомая нам энтропия ансамбля **U**. Второе слагаемое представляет собой усредненное по ансамблю **U** выражение $-\sum_{j=1}^{N} p(v_j / u_i)\log p(v_j / u_i)$. Конструкция этого выражения аналогична конструкции функционала энтропии, но вместо безусловных вероятностей используются условные вероятности. Это дает нам право считать, что выделенное выражение – энтропия ансамбля **V** при условии, что известно одно сообщение u_i. Это выражение называется *частной условной энтропией* ансамбля **V** при условии, что известно сообщение u_i из ансамбля **U,** который статистически связан с **V**. Обозначим эту частную условную энтропию $H(\mathbf{V} / u_i)$. Усреднение частной условной энтропии по ансамблю **U** дает *среднюю условную энтропию*

$$H(\mathbf{V} / \mathbf{U}) = -\sum_{i=1}^{K} p(u_i)\sum_{j=1}^{N} p(v_j / u_i)\log p(v_j / u_i) = \sum_{i=1}^{K} p(u_i) H(\mathbf{V} / u_i). \quad (12)$$

С другой стороны, из (12) следует, что

$$H(\mathbf{V} / \mathbf{U}) = -\sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i v_j)\log p(u_i / v_j).$$

Окончательно получим выражение для энтропии объединения статистически связанных ансамблей:

$$H(\mathbf{UV}) = H(\mathbf{U}) + H(\mathbf{V} / \mathbf{U}). \quad (13)$$

Аналогично выводится следующее выражение, симметричное (13):

$$H(\mathbf{UV}) = H(\mathbf{VU}) = H(\mathbf{V}) + H(\mathbf{U}/\mathbf{V}). \quad (14)$$

Несмотря на то, что соотношение (14) получено в результате вывода, некоторые авторы называют его третьей аксиомой Хинчина.

При наличии третьего ансамбля *W*, статистически связанного с первыми двумя, получим

$$H(\mathbf{UVW}) = H(\mathbf{UV}) + H(\mathbf{W}/\mathbf{UV}) = H(\mathbf{U}) + H(\mathbf{V}/\mathbf{U}) + H(\mathbf{W}/\mathbf{UV}).$$

Это выражение легко расширяется на объединение произвольного количества статистически связанных ансамблей:

$$H(\mathbf{UVW...Z})=H(\mathbf{U})+H(\mathbf{V/U})+H(\mathbf{W/UV})+\ldots+H(\mathbf{Z/UVW}\ldots).$$

Докажем несколько свойств средней условной энтропии.

1. Наличие статистической взаимозависимости между ансамблями приводит к уменьшению общей энтропии этих двух ансамблей. Последнее утверждение равносильно следующему: безусловная энтропия больше условной. Математическая запись этого утверждения:

$$H(\mathbf{V}/\mathbf{U}) \le H(\mathbf{V}). \qquad (15)$$

Доказательство.

Перепишем неравенство (15) в виде $H(\mathbf{V/U}) - H(\mathbf{V}) \le 0$.

При доказательстве (15) будем пользоваться неравенством $\ln x \le x - 1$, а в качестве основания логарифма для определенности, как и раньше, примем 2.

$$H(\mathbf{V}/\mathbf{U}) - H(\mathbf{V}) = -\sum_{i=1}^{K} p(u_i) \sum_{j=1}^{N} p(v_j/u_i) \log_2 p(v_j/u_i) + \sum_{j=1}^{N} p(v_j) \log_2 p(v_j) =$$

$$= -\sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i v_j) log_2\, p(v_j/u_i) + \sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i v_j) log_2\, p(v_j) =$$

$$= \sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i v_j) \log_2 \frac{p(u_i)p(v_j)}{p(u_i v_j)} \le \log_2 e \sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i v_j) \left[\frac{p(u_i)p(v_j)}{p(u_i v_j)} - 1 \right] =$$

$$= \log_2 e \sum_{i=1}^{K}\sum_{j=1}^{N} \left[p(u_i)p(v_j) - p(u_i v_j) \right] = \log_2 e [1-1] = 0,$$

что и требовалось доказать.

Равенство $H(\mathbf{V}) = H(\mathbf{V}/\mathbf{U})$ достигается только тогда, когда $p(u_i)p(v_j) = p(u_i v_j)$, то есть, когда ансамбли независимы.

Из доказанного, а также из (10) и (13) следует, что в общем случае

$$H(\mathbf{UV}) \le H(\mathbf{U}) + H(\mathbf{V}). \qquad (16)$$

Полученный результат очевидным образом обобщается для произвольного количества ансамблей:

$$H(\mathbf{V}) \ge H(\mathbf{V}/\mathbf{U}) \ge H(\mathbf{V}/\mathbf{UW}) \ge \ldots \ge H(\mathbf{V}/\mathbf{UW}\ldots\mathbf{Z}). \quad (17)$$

Эта цепочка неравенств очевидна, ибо увеличение количества условий обязательно ведет к уменьшению энтропии.

2. Если **U** и **V** независимы, то

$$H(\mathbf{U}/\mathbf{V})=H(\mathbf{U}), \qquad H(\mathbf{V}/\mathbf{U})=H(\mathbf{V}). \qquad (18)$$

Справедливость этого утверждения следует из сопоставления (10) с (13), а также из предыдущего доказательства.

3. Если **U** и **V** связаны взаимно однозначно, то есть $v = \varphi(u)$, где функция $\varphi(u)$ монотонна, то

$$H(\mathbf{U}/\mathbf{V}) = 0, \qquad H(\mathbf{V}/\mathbf{U})=0. \qquad (19)$$

В силу монотонности функции $\varphi(u)$ она обратима, то есть для каждого значения $v \in \mathbf{V}$ существует единственное значение $u = \varphi^{-1}(v)$

При взаимно однозначной зависимости **U** и **V** количество элементов одинаково, $K = M$, и

$$p(u_i) = p\left(\varphi^{-1}(v_i)\right) = p(v_i)\ , \qquad p(u_i/v_i) = p\left(\varphi^{-1}(v_i)/v_i\right) = 1$$

$$p(v_i/u_i) = p\left(v_i/\varphi^{-1}(v_i)\right) = 1, \qquad p(u_i/v_j)_{i\neq j} = 0, \quad p(u_i v_j)_{i\neq j} = 0,$$

$$p(u_i v_i) = p(u_i) = p(v_i).$$

В этой ситуации

$$H(\mathbf{U}/\mathbf{V}) = -\sum_{i=1}^{K}\sum_{j=1}^{K} p(u_i v_j)\log_2 p(u_i/v_j) = 0. \qquad (20)$$

Точно так же доказывается и второе равенство в (19).

3.4. Собственная информация элемента ансамбля

Пусть имеется ансамбль (алфавит, сообщения) **U**: $u_i \in (\mathbf{U}, p(u_i))$.

Количество собственной информации, содержащейся в элементе ансамбля (букве, сообщении) есть, по определению,

$$I(u_i) = -\log p(u_i),\ i = 1,2,\ldots,K.$$

Свойства собственной информации.

1. Если ансамбль детерминирован, то есть когда вероятность выбора одного i – го элемента равна 1, выбор других элементов ансамбля невозможен. В этой ситуации, если $p(u_i) = 1$, то $I(u_i) = 0$.

2. Пусть существуют два ансамбля **U** и **V**, совместное распределение их элементов описывается совместными вероятностями $p(u_i, v_j)$. Если **U** и **V** независимы, то $p(u_i, v_j) = p(u_i)p(v_j)$. Тогда

$$I(u_i, v_j) = -\log p(u_i, v_j) = -\log p(u_i) - \log p(v_j) = I(u_i) + I(v_j).$$

Таким образом, установлено свойство аддитивности информации.

Усредняя собственное количество информации элементов ансамбля по ансамблю **U**, получим, что

$$\bar{I}(\mathbf{U}) = -\sum_{i=1}^{K} p(u_i)\log p(u_i) = H(\mathbf{U}). \qquad (21)$$

Это означает, что энтропия ансамбля – не что иное, как среднее значение собственной информации элементов этого ансамбля.

3.5. Взаимная информация и ее свойства

Количество взаимной информации между элементами $u_i \in \mathbf{U}$ и $v_j \in \mathbf{V}$ определено, как

$$I(u_i; v_j) = \log \frac{p(u_i v_j)}{p(u_i)p(v_j)} = \log \frac{p(u_i / v_j)}{p(u_i)} = \log \frac{p(v_j / u_i)}{p(v_j)} = I(v_j; u_i). \qquad (22)$$

Из этого выражения сразу же вытекает, что если ансамбли независимы, то есть если $p(u_i)p(v_j) = p(u_i v_j)$ для всех *i, j*, то $I(u_i; v_j) = I(v_j; u_i) = 0$, чего и следовало ожидать.

Количество информации, определенное в (21), измеряется в двоичных единицах - “бит”, и в дальнейшем мы будем опускать обозначение основания логарифма, подразумевая его равным 2.

Усредняя это частное количество информации по ансамблям **U** и **V**, получим среднее количество взаимной информации об ансамбле **U**, содержащееся в ансамбле **V**:

$$I(\mathbf{U};\mathbf{V}) = \sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i, v_j)\log \frac{p(u_i, v_j)}{p(u_i)p(v_j)} =$$

$$= \sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i, v_j)\log p(u_i, v_j) - \sum_{i=1}^{K}\sum_{j=1}^{N} p(u_i, v_j)\log p(u_i)p(v_j) =$$

$$= -H(\mathbf{U},\mathbf{V}) - \sum_{i=1}^{K} \log p(u_i) \sum_{j=1}^{N} p(u_i, v_j) - \sum_{j=1}^{N} \log p(v_j) \sum_{i=1}^{K} p(u_i, v_j). \qquad (23)$$

Однако $\sum_{j=1}^{N} p(u_i, v_j) = p(u_i)$ и $\sum_{i=1}^{K} p(u_i, v_j) = p(v_j)$, поэтому

$$I(\mathbf{U},\mathbf{V}) = -H(\mathbf{U},\mathbf{V}) - \sum_{i=1}^{K} p(u_i) \log p(u_i) - \sum_{j=1}^{N} p(v_j) \log p(v_j) = H(\mathbf{U}) + H(\mathbf{V}) - H(\mathbf{U},\mathbf{V}).$$

Воспользовавшись выражением (13), получим, что среднее количество взаимной информации об ансамбле **V**, содержащееся в ансамбле **U**, или, что то же самое, количество информации об ансамбле **U**, содержащееся в **V**, равно

$$I(\mathbf{U},\mathbf{V}) = H(\mathbf{U}) + H(\mathbf{V}) - H(\mathbf{U}) - H(\mathbf{V}/\mathbf{U}) = H(\mathbf{V}) - H(\mathbf{V}/\mathbf{U}). \qquad (24)$$

Воспользовавшись формулой (14), получим симметричное выражение:

$$I(\mathbf{U},\mathbf{V}) = H(\mathbf{U}) + H(\mathbf{V}) - H(\mathbf{V}) - H(\mathbf{U}/\mathbf{V}) = H(\mathbf{U}) - H(\mathbf{U}/\mathbf{V}). \qquad (25)$$

Поскольку $H(\mathbf{U}) \geq H(\mathbf{U}/\mathbf{V})$ и $H(\mathbf{V}) \geq H(\mathbf{V}/\mathbf{U})$, количество средней взаимной информации всегда неотрицательно:

$$I(\mathbf{U};\mathbf{V}) \geq 0, \qquad (26)$$

и равенство имеет место лишь тогда, когда ансамбли **U** и **V** независимы.

При взаимно однозначной зависимости между ансамблями, как это следует из (19), $H(\mathbf{V}/\mathbf{U}) = H(\mathbf{U}/\mathbf{V}) = 0$, а это значит, что в этом случае

$$I(\mathbf{U};\mathbf{V}) = H(\mathbf{U}) = H(\mathbf{V}) = \max.$$

Полученные результаты позволяют сделать следующие выводы.

Если принять ансамбль **U** в качестве ансамбля передаваемых сообщений, то энтропия ансамбля **U** есть не что иное, как все количество информации, содержащееся в этом ансамбле. Из этого же следует известная теорема Шеннона, которая гласит: *ни при каком преобразовании информация возрасти не может, она никогда не будет превышать энтропию источника*.

При передаче информации потерь не будет, если передача будет выполнена взаимнооднозначно. В этом случае энтропия ансамбля принятых сообщений **V** окажется той же самой, что и энтропия ансамбля **U**.

В дополнение к первым двум замечаниям п. 2.1.1 снова отметим, что энтропия ансамбля **U** есть *средняя информационная производительность* ансамбля или *средняя скорость создания информации*, достигающая максимума при равновероятном выборе источником сообщений из этого ансамбля.

Реально никакая передача (или преобразование) информации не выполняется без потерь. Эти потери вызваны шумами и помехами в линии связи, сбоями и ошибками кодера и декодера и т.д. Поэтому всегда при передаче (преобразовании) информации остается неопределенность, которая при равномерном распределении сообщений источника выражается в виде условной энтропии $H(\mathbf{U}/\mathbf{V})$. Это означает, что средняя условная энтропия $H(\mathbf{U}/\mathbf{V})$ есть мера неопределенности, вносимой передающим каналом. В частном крайнем случае, когда шумы и помехи в канале связи таковы, что $H(\mathbf{U}/\mathbf{V}) = H(\mathbf{U})$, то $I(\mathbf{U};\mathbf{V}) = H(\mathbf{U}) - H(\mathbf{U}/\mathbf{V}) = H(\mathbf{U}) - H(\mathbf{U}) = 0$.

В приложении 2 рассмотрены ситуации, в которых средняя условная энтропия $H(e)$ аддитивных помех в канале принимает экстремально высокие значения при ограничениях на амплитуду и на дисперсию помех.

3.6. Информационная дивергенция, граница Симмонса

Соломон Кульбак ввел в свое время функционал информации, который представляет собой информационную меру расхождения между двумя распределениями, отвечающими двум выборкам. Данная информационная мера, называемая информацией по Кульбаку, играет существенную роль в математической статистике при проверке статистических гипотез. В литературе, посвященной криптографии, информация по Кульбаку трактуется, как информационная дивергенция и выражается в виде информации для различения в пользу гипотезы H_1 против H_2, усредненной по той вероятностной мере, которая порождает плотность вероятности $f_1(x)$:

$$I(1:2) = \int_X f_1(x) \log \frac{f_1(x)}{f_2(x)} dx .$$

Такая трактовка кульбаковской информации не будет использоваться в настоящем пособии.

Также в настоящем пособии не найдет применения граница Симмонса, которая устанавливается для определения достижимой имитостойкости множества допустимых криптограмм при произвольном распределении на множестве ключей шифров. Здесь определяющим является количество информации, которая содержится в множестве криптограмм относительно множества ключей. В общем случае не известно, при каких условиях существуют шрифты, обеспечивающие совершенную имитостойкость.

Поскольку криптография не является целью образования по настоящему курсу теории информации, мы ограничимся только теми сведениями о границе Симмонса, которые приведены в данном разделе.

4. КОДИРОВАНИЕ ИСТОЧНИКОВ

4.1. Равномерное кодирование

4.1.1. Высоковероятные множества сообщений источника

Дискретный стационарный источник без памяти создает последовательности (сообщения) $\mathbf{x}_j$ длиной s из букв, которые выбираются из алфавита (множества, ансамбля) **X** объема *L* с вероятностью $p(x_i)$ независимо от позиции этой буквы в последовательности $\mathbf{x}_j = (x_1^j\ x_2^j\ ...x_s^j)$. Случайное событие, которое заключается в независимом выборе *m* букв из алфавита **X** при создании последовательности $\mathbf{x}_j$ есть пересечение элементарных событий:

$$(x_1^j \in \mathbf{X}) \cap (x_2^j \in \mathbf{X}) \cap (x_3^j \in \mathbf{X}) \cap \ldots \cap (x_s^j \in \mathbf{X}).$$

Это событие есть не что иное, как событие $\mathbf{x}_j \in \mathbf{X}^s$, где $\mathbf{X}^s$ есть *s* – кратное декартово произведение множеств **X**: $\mathbf{X}^s = \mathbf{X} \times \mathbf{X} \times \mathbf{...} \times \mathbf{X}$.

В силу независимости выбора источником букв из алфавита (множества) **X** вероятность выбора *s* букв в последовательности длиной *s* есть произведение вероятностей $p(x_i)$ выбора букв сообщения $\mathbf{x}_j$**:**

$$P(\mathbf{x}_j) = \prod_{i=1}^{s} p(x_i^j),\ \ x_i^j \in \mathbf{X}. \qquad (27)$$

$P(\mathbf{x}_j)$ – распределение вероятностей выбора источником сообщения $\mathbf{x}_j$ длиной *s* из множества $\mathbf{X}^s$. Энтропия множества $\mathbf{X}^s$ есть совместная энтропия множеств **X**, а именно $H(\mathbf{X}^s) = H(\mathbf{XXX...X})$. В силу независимости

выбора источником букв из алфавита **X** эта энтропия в соответствии с (10) и (11) равна

$$H(\mathbf{X}^s) = H(\mathbf{XXX...X}) = sH(\mathbf{X})$$

С другой стороны, собственная информация, содержащаяся в каждой последовательности $\mathbf{x}_j \in \mathbf{X}^s$,

$$I(\mathbf{x}_j) = -\log P(\mathbf{x}_j),$$

а энтропия множества $\mathbf{X}^s$ равна средней собственной информации элементов этого множества:

$$H(\mathbf{X}^s) = \sum_{\mathbf{x}_j \in \mathrm{X}^s} P(\mathbf{x}_j) I(\mathbf{x}_j) = -\sum_{\mathbf{x}_j \in \mathbf{X}^s} P(\mathbf{x}_j) \log P(\mathbf{x}_j) = sH(\mathbf{X}), \qquad (28)$$

где суммирование выполняется по всем вариантам выбора любых комбинаций букв из множества $\mathbf{X}^s$.

Равенство (28) – не что иное, как формальное определение математического ожидания величины $-\log P(\mathbf{x}_j)$, то есть собственной информации, содержащейся в каждой последовательности $\mathbf{x}_j$**:**

$$H(\mathbf{X}^s) = M\left[I(\mathbf{x}_j)\right] = -\sum_{\mathbf{x}_j \in \mathbf{X}^s} P(\mathbf{x}_j) \log P(\mathbf{x}_j) = sH(\mathbf{X}). \qquad (29)$$

В соответствии с законом больших чисел оценка вероятности отклонения любой из усредняемых величин от своего математического ожидания на величину ε выражается равносильными неравенствами, вытекающими из неравенства Чебышева:

$$P\left\{ \left| H(\mathbf{X}) - \frac{I(\mathbf{x}_j)}{s} \right| \le \varepsilon \right\} \ge 1 - \delta$$

или

$$P\left\{ \left| sH(\mathbf{X}) - I(\mathbf{x}_j) \right| \le s\varepsilon \right\} \ge 1 - \delta, \qquad (30)$$

где $\delta = \left(\frac{C}{s\varepsilon}\right)^2$, $\varepsilon > 0$, $C > 0$ - некоторая постоянная для множества **X**.

Этим неравенством посредством задания произвольного сколь угодно малого положительного значения ε с ростом s до значения, при котором δ становится достаточно малым, определяется *высоковероятное множество*

$W(\varepsilon, \delta, s) \subset \mathbf{X}^s$ последовательностей $\mathbf{x}_j$ на выходе *дискретного источника без памяти*, такое, что, начиная с некоторого s > $M(\varepsilon, \delta)$ для каждой последовательности $\mathbf{x}_j \in \mathbf{X}^s$ длины, не меньшей s ,

$$1 - \delta < P\{\mathbf{x}_j \in W(\varepsilon, \delta, s)\} < 1 \ , \ j = 1, 2, \ldots \qquad (31)$$

Вероятность $P\{\mathbf{x}_j \in W(\varepsilon, \delta, s)\}$ – это вероятность того, что из множества $W(\varepsilon, \delta, s)$ источник выбирает или сообщение $\mathbf{x}_1$, или сообщение $\mathbf{x}_2$, или любое иное сообщение $\mathbf{x}_j$. Так описывается объединение событий. По этой причине, а также потому, что эти события не пересекаются, мы можем применить аксиому Колмогорова о счетной аддитивности вероятностной меры непересекающихся множеств, и переписать неравенство (31) следующим образом:

$$1 - \delta < \sum_{\mathbf{x}_j \in W(\varepsilon, \delta, s)} P(\mathbf{x}_j) < 1, \qquad (32)$$

Высоковероятное множество $W(\varepsilon, \delta, s)$ иногда называется *множеством типичных последовательностей*, которые однозначно кодируются и декодируются. Все последовательности, которые не попадают в это множество, кодируются каким-либо одним словом и декодируются также одним словом, например, словом «ошибка».

Из выражения (30) следует, что границы для множества $W(\varepsilon, \delta, m)$ определяются неравенством

$$W(\varepsilon, \delta, s) = \{\mathbf{x}_j : s(H(\mathbf{X}) - \varepsilon) \le I(\mathbf{x}_j) \le s(H(\mathbf{X}) + \varepsilon)\},$$

а это означает, что для $\mathbf{x}_j \in W(\varepsilon, \delta, s)$ с вероятностью, не меньшей $1 - \delta$:

$$s[H(\mathbf{X}) - \varepsilon] \le I(\mathbf{x}_j) \le s[H(\mathbf{X}) + \varepsilon].$$

Если считать, что последовательности $\mathbf{x}_j$ составлены из букв двоичного алфавита, то, учитывая, что $I(\mathbf{x}_j) = -\log_2 P(\mathbf{x}_j)$, и что логарифм по основанию 2 – функция монотонная, для всех последовательностей, принадлежащих множеству $W(\varepsilon, \delta, s)$, после простых преобразований получим:

$$s[H(\mathbf{X}) - \varepsilon] \le -\log_2 P(\mathbf{x}_j) \le s[H(\mathbf{X}) + \varepsilon],$$
$$-s[H(\mathbf{X}) + \varepsilon] \le \log_2 P(\mathbf{x}_j) \le -s[H(\mathbf{X}) - \varepsilon],$$

$$2^{-s[H(\mathbf{X})+\varepsilon]} \le P\left(\mathbf{x}_j\right) \le 2^{-s[H(\mathbf{X})-\varepsilon]}. \qquad (33)$$

Обозначим количество элементов множества последовательностей $W(\varepsilon, \delta, s)$, как $\left|W\left(\varepsilon,\delta,s\right)\right|$. Теперь просуммируем все части неравенства (33) по всем непересекающимся последовательностям из высоковероятного множества $W(\varepsilon, \delta, s)$ и объединим его с (32):

$$1-\delta < \left|W\left(\varepsilon,\delta,s\right)\right|2^{-s[H(\mathbf{X})+\varepsilon]} \le \sum_{x_j \in W(\varepsilon,\delta,s)} P\left(\mathbf{x}_j\right) \le \left|W\left(\varepsilon,\delta,s\right)\right|2^{-s[H(\mathbf{X})-\varepsilon]} < 1. \qquad (34)$$

С одной стороны, из неравенств (34) можно вычленить отдельные компоненты, из сравнения которых следует, что при больших m для всех элементов (последовательностей) множества $W(\varepsilon, \delta, m)$ справедливо соотношение:

$$\left|W\left(\varepsilon,\delta,s\right)\right|2^{-s[H(\mathbf{X})+\varepsilon]} \le \sum_{x_j \in W(\varepsilon,\delta,s)} P\left(\mathbf{x}_j\right) \le 1,$$

откуда после умножения на $2^{s[H(\mathbf{X})+\varepsilon]}$ и исключения средней части, получим:

$$\left|W\left(\varepsilon,\delta,s\right)\right| \le 2^{s[H(\mathbf{X})+\varepsilon]}. \qquad (35)$$

Таким образом получена верхняя оценка количества последовательностей, входящих в множество $W(\varepsilon, \delta, s)$ и обладающих свойством однозначного кодирования и декодирования.

С другой стороны, совмещая левую часть неравенства (34) с фрагментом правой части этого же неравенства, обнаружим, что

$$1-\delta < \sum_{\mathbf{x}_j \in W(\varepsilon,\ \delta,\ s)} P\left(\mathbf{x}_j\right) \le \left|W\left(\varepsilon,\delta,s\right)\right|2^{-s[H(\mathbf{X})-\varepsilon]},$$

откуда следует

$$\left(1-\delta\right) < \left|W\left(\varepsilon,\delta,s\right)\right|2^{-s[H(\mathbf{X})-\varepsilon]},$$

и, наконец,

$$\left(1-\delta\right)2^{s[H(\mathbf{X})-\varepsilon]} < \left|W\left(\varepsilon,\delta,s\right)\right|. \qquad (36)$$

4.1.2. Прямая и обратная теоремы Шеннона о кодировании источника

Теорема 1 (прямая теорема). Пусть $R_{\text{ист}} > H(\mathbf{X})$, тогда для любого положительного δ существует равномерный код со скоростью $R_{\text{ист}}$, который кодирует дискретный источник без памяти с вероятностью ошибки, не превышающей δ.

Доказательство.

Из неравенств (31) и (32) следует, что для любых положительных ε и δ найдется такое $M(\varepsilon, \delta, s)$, что для любого $s > M(\varepsilon, \delta, s)$ вероятность появления на выходе источника последовательности **x** длины s, которая лежит вне множества $W(\varepsilon, \delta, s)$, не превосходит δ. Это следует также из неравенств (32) и (36). По этой причине и в силу высокой вероятности множества $W(\varepsilon, \delta, s)$ выбор его в качестве множества однозначно кодируемых и декодируемых последовательностей (сообщений источника) является естественным.

Если количество сообщений источника K выражается точной степенью двойки, то есть $K = 2^k$, то скорость $R_{\text{ист}}$ равномерного кодирования последовательности **x** в соответствии с (1) и (2) равна $R_{\text{ист}} = k/s$. В этих условиях длина кодового слова равна $k = s\,R_{\text{ист}}$, а количество всех двоичных кодовых слов равно $2^k = 2^{sR_{ист}}$.

Для однозначного кодирования и декодирования с вероятностью, не меньшей $1-\delta$, количество всех возможных кодовых слов должно быть не меньше, чем число элементов в множестве $W(\varepsilon, \delta, s)$. Для выражения этого обстоятельства в виде неравенства воспользуемся сравнением количества сообщений (кодовых слов источника) с верхней границей (35):

$$2^{s[H(\mathbf{X})+\varepsilon]} \le 2^{sR_{ист}}$$

откуда следует, что

$$R_{\text{ист}} \ge H(\mathbf{X}) + \varepsilon. \qquad (37)$$

Теорема доказана.

Теорема 2 (обратная теорема). Пусть $R_{\text{ист}} < H(\mathbf{X})$, тогда для каждого равномерного кода, кодирующего сообщения дискретного источника без памяти длиной s со скоростью $R_{\text{ист}}$, вероятность ошибки p_{es} такова, что

$$\lim_{s\to\infty} p_{es} = 1. \qquad (38)$$

Доказательство.

Докажем вначале справедливость равенства (38).

В соответствии с условием теоремы $2\varepsilon = H(\mathbf{X}) - R_{\text{ист}} > 0$ или $R_{\text{ист}} < H(\mathbf{X}) - 2\varepsilon$.

Образуем последовательность множеств $W(\varepsilon,\delta,s) \in \mathbf{X}^s$, которые, начиная с некоторого s = $M(\varepsilon,\delta)$, удовлетворяют условиям п. 4.1.1 и являются высоковероятными. Кроме того при каждом значении s однозначно определяется множество $W_S \in \mathbf{X}^s$ однозначно кодируемых и декодируемых последовательностей сообщений, количество которых выражается степенью двойки, то есть $M = |W_s| = 2^{sR}$. Тогда вероятностная мера каждого такого множества W_S есть вероятность $1 - p_{es}$ безошибочного кодирования и может быть оценена сверху следующим образом:

$$P(W_s) = 1 - p_{es} = P[W_S \cap W(\varepsilon,\delta,s)] + P[W_S \cap \overline{W}(\varepsilon,\delta,s)] \le$$
$$\le P[W_S \cap W(\varepsilon,\delta,s)] + P[\overline{W}(\varepsilon,\delta,s)], \qquad (39)$$

где $\overline{W}(\varepsilon,\delta,s)$ – дополнение множества $W(\varepsilon,\delta,s)$ до $\mathbf{X}^s$, то есть

$$W(\varepsilon,\delta,s) \cap \overline{W}(\varepsilon,\delta,s) = \mathbf{X}^s.$$

Поскольку $W(\varepsilon,\delta,s)$ - высоковероятное множество,

$$\lim_{s\to\infty} P[\overline{W}(\varepsilon,\delta,s)] = 0,\ \varepsilon > 0. \qquad (40)$$

Любое слово $\mathbf{x}_j \in W_S \cap W(\varepsilon,\delta,s)$ принадлежит также множеству $W(\varepsilon,\delta,s)$, поэтому из (33) следует, что $P(\mathbf{x}_j) \le 2^{-s[H(\mathbf{X})-\varepsilon]}$. Кроме того, количество элементов в множестве $W_S \cap W(\varepsilon,\delta,s)$ не превосходит $|W_s| = 2^{sR_{\text{ист}}}$. Из этих двух фактов следует, что

$$P[W_S \cap W(\varepsilon,\delta,s)] = \sum_{W_S \cap W(\varepsilon,\delta,s)} P(\mathbf{x}_j) \le 2^{sR_{ист}} 2^{-s(H(\mathbf{X})-\varepsilon)} = 2^{-s\varepsilon}, \quad (41)$$

поскольку $s\,R_{\text{ист}}$ - $s(H(\mathbf{X})\text{-}\varepsilon) = s(H(\mathbf{X})\text{ -}2\varepsilon)$ - $s(H(\mathbf{X})\text{-}\varepsilon)$ = - $s\varepsilon$.

Используя (40) и (41) в (39) , получим

$$\lim_{k\to\infty}\left[1 - p_{es}\right] \le \lim_{s\to\infty} 2^{-s\varepsilon} = 0\,,$$

а это свидетельствует о справедливости (38).

Содержательный смысл доказанных теорем состоит в следующем.

На выходе двоичного кодера источника в среднем на каждое элементарное сообщение будет затрачиваться $R_{\text{ист}}$ кодовых символов, в том числе, в той последовательности, которая содержится в $\overline{W_s}$. Эффективность кодирования источника возрастает при уменьшении разности $\varepsilon = R_{\text{ист}} - H(\mathbf{X})$, если только $\varepsilon > 0$. Следовательно, *при двоичном кодировании энтропия источника (в бит) – не что иное, как наименьшее количество двоичных символов на элементарное сообщение, которое появляется на выходе наилучшего кодера для этого источника при условии, что сообщения могут быть восстановлены сколь угодно точно*. Так, для примера из разд. 2.4 при побуквенном кодировании телеграмм энтропия источника при равновероятном выборе букв равна $H = 6.0$ бит, но на самом деле, из-за фактического неравновероятного выбора букв и значительной памяти в естественном языке $H \approx 4.0$ бит. Скорость побуквенного кодирования телеграммы $R_{\text{ист}} = 6$ симв/буква, что обеспечивает однозначное кодирование и декодирование, поскольку $R_{\text{ист}} > H$. При втором (блочном) способе кодирования телеграмм скорость кодирования $R_{\text{ист}} = 13/8$ симв/буква, то есть $R_{\text{ист}} < H$, а потому в этом случае, как уже было отмечено, однозначного декодирования телеграммы не произойдет.

В примере, связанном с получением информации в ИИС, объем алфавита источника 10. При равновероятном выборе букв этого алфавита его энтропия равна $H(\mathbf{X}) = log_2 10 \approx 3{,}32193$ бит. Скорость десятичного кодирования источника с помощью цифрового прибора и следующего за ним равномерного двоичного кодера $R_{\text{ист}} = 4$ бит/буква. Разность $R_{\text{ист}} - H(\mathbf{X}) = 0{,}7807$ бит. Фактическая скорость блочного двоичного равномерного кодирования того же источника равна отношению длины последовательности (14 двоичных символов), которой кодируется инфор-

мация источника, к количеству десятичных символов (букв), которыми представляются сообщения источника, то есть $R_{\text{ист}} = 14/4 = 3.5$ бит/буква. В этом случае $R_{\text{ист}} - H(\mathbf{X}) = 0{,}2807$ бит/буква. Из этого сопоставления мы видим, что второй способ кодирования эффективнее первого, и оба способа обеспечивают однозначное кодирование и декодирование.

Изложенные выше две теоремы в совокупности свидетельствуют о том, что скорость создания информации при равномерном кодировании сообщений источника без памяти должна превышать энтропию алфавита источника или, что то же самое, энтропию множества его элементарных сообщений хотя бы не на много.

Подобные теоремы имеют место и для любых других источников и кодеров, в том числе, для кодеров, реализующих неравномерное кодирование. Свойства неравномерных кодов рассматривается в следующем разделе.

4.2. Неравномерное кодирование

4.2.1. Условия однозначного кодирования и декодирования, префиксные коды

Снова обратимся к примеру кодирования телеграммы, приведенному в разделе 2.4. В этом примере было представлено равномерное кодирование источника: в первом случае – побуквенное, а во втором – блочное. Скорость побуквенного кодирования составила 6 двоичных символов на букву. При использовании второго метода скорость кодирования гораздо меньше, что облегчает работу аппаратуры канала связи, но кодирование и декодирование телеграмм будет неоднозначным. С другой стороны, скорость побуквенного кодирования слишком велика, и в связи с этим побуквенное кодирование приводит к неоправданной перегрузке канала связи.

Значительного снижения скорости кодирования источника без нарушения условий однозначного декодирования можно достигнуть, применяя неравномерное кодирование, а именно выбирая кодовые слова, длина которых зависит от частоты использования кодируемых букв (символов, со-

общений, блоков) источника: чем чаще используется кодируемое сообщение, тем короче кодовое слово, и наоборот. Такое кодирование позволит однозначно кодировать и декодировать любое сообщение и существенно снизить общую длину передаваемого текста.

В дальнейшем при изложении методов кодирования источника мы будем рассматривать побуквенное кодирование. В этой ситуации для обозначения элементов алфавита источника нет необходимости в индексе, указывающем позицию буквы в кодовом слове (последовательности). Поэтому здесь компоненты вектора **x** мы будем снабжать только одним нижним индексом, указывающим вид элемента алфавита (буквы), а верхний индекс будем опускать.

Таблица 2

Равномерное и неравномерное кодирование

Сообщения x_j	Вероятности p_j	Равномерный код	Неравномерный код	Длина кодового слова неравномерного кода k_j
x_1	1/4	0 0 0	0 0	2
x_2	1/4	0 0 1	0 1	2
x_3	1/8	0 1 0	1 0 0	3
x_4	1/8	0 1 1	1 0 1	3
x_5	1/16	1 0 0	1 1 0 0	4
x_6	1/16	1 0 1	1 1 0 1	4
x_7	1/16	1 1 0	1 1 1 0	4
x_8	1/16	1 1 1	1 1 1 1	4

Пусть источник в каждый момент времени с тактом τ порождает одно из восьми сообщений с вероятностями, указанными в табл. 2. Эти восемь сообщений – буквы x_i, их полный набор – алфавит **X** источника, объем алфавита $|\mathbf{X}| = M = 8$, длина кодируемого элементарного сообщения источника, то есть буквы, s = 1. Поскольку каждое *j* – ое сообщение есть одна буква, то в этом случае *K* = *M*, *i*= *j*, а вероятности $p(x_i) = p(x_j) = p_j$.

Энтропия источника

$$H(\mathbf{A}) = -\sum_{i=1}^{8} p_j \log_2 p_j = \frac{2}{4}\log_2 4 + \frac{2}{8}\log_2 8 + \frac{4}{16}\log_2 16 = 2{,}75 \text{ бит.}$$

Длина кодовых слов равномерного кода k = 3, отсюда скорость равномерного кода равна 3 бит/буква.

Поскольку в нашем примере вероятности p_j равны степеням двойки, длину кодовых слов неравномерного кода определим, как отрицательное значение вероятности кодируемого элемента алфавита источника, то есть $k_j = \mathrm{Ent}\lfloor -\log_2 p_j \rfloor + 1$. Средняя длина кодовых слов неравномерного кода

$$k_{\text{ср}} = \sum_{j=1}^{K} p_j k_j \ . \qquad (42)$$

Скорость кодирования - средняя скорость, которая в общем случае кодирования блоков длиной s выражается формулой:

$$R_{\text{ист}} = \frac{k_{\text{ср}}}{s} \text{ бит/буква.} \qquad (43)$$

В приведенном примере кодирование побуквенное, длина каждого сообщения, то есть буквы, равна s = 1,

$$k_{\text{ср}} = \sum_{j=1}^{K} k_j p_j = \frac{2}{4}2 + \frac{2}{8}3 + \frac{4}{16}4 = 2{,}75 \text{ бит,} \qquad (44)$$

и средняя скорость кодирования в соответствии с (43) равна

$$R_{\text{ист}} = 2{,}75 \text{ бит/буква.}$$

Мы видим, что средняя скорость кодирования равна энтропии источника. Это не случайно, поскольку в данном примере длина кодовых слов выбрана пропорциональной модулю логарифма вероятности кодируемых букв, то есть $k_j = \left|\log_2 p_j\right|$, и средняя длина кодовых слов $k_{\text{ср}} = \sum_{j=1}^{K} p_j \left|\log_2 p_j\right|$. В нашем примере

$$k_{\text{ср}} = \sum_{j=1}^{K} p_j \left|\log_2 p_j\right| = -\sum_{j=1}^{K} p_j \log_2 p_j = H(\mathbf{X}).$$

В реальной ситуации так не бывает, поэтому в качестве длины кодовых слов приходится выбирать целые числа с округлением в сторону возрастания, от этого средняя длина слов увеличится, и скорость кодирования

будет немного больше $H(\mathbf{X})$, что и требуется для однозначного кодирования и декодирования.

Отмеченное преимущество неравномерного кодирования перед равномерным сопровождается следующими отрицательными обстоятельствами.

В о – п е р в ы х, априорные сведения о вероятностях сообщений, как правило, отсутствуют.

Данное обстоятельство устраняется статистическим оцениванием вероятностей сообщений источника. С этой целью сообщения необходимо аккумулировать в буферном устройстве кодера, оценить вероятности этих сообщений, и затем кодировать их кодовыми словами, длина которых с точностью до целых чисел пропорциональна модулям логарифма этих оценок. Это приводит к задержке кодирования и, значит, к задержке передачи сообщений.

В о – в т о р ы х, при неравномерном кодировании возможны ситуации, когда отдельные кодовые слова оказываются неотделимыми при декодировании в силу различия их длины. Приведем пример.

Пусть четыре элементарных сообщения источника суть x_1, x_2, x_3, x_4. Они кодируются неравномерным двоичным кодом: $x_1 \to 0$, $x_2 \to 01$, $x_3 \to 10$, $x_4 \to 011$. Пусть на выходе источника появилось сообщение $x_2\, x_3\, x_3\, x_1$. На выходе кодера будет последовательность: 0110100. Разделительного знака нет, иначе пришлось бы для кодирования использовать троичный код, где этот знак был бы третьим элементом алфавита. Не имея в двоичном коде разделительного знака, декодер может расшифровать полученную последовательность как $x_4 x_2 x_1 x_1$ или как $x_4 x_1 x_3 x_1$, что неверно.

Для устранения этого негативного обстоятельства необходимо кодировать сообщения источника такими кодовыми словами, чтобы никакое из них не было бы началом другого слова. Такие коды называются *префиксными*. Для существования префиксного кода с длинами слов $k_1, k_2, \ldots, k_K$, где K – количество сообщений источника, необходимо и достаточно, чтобы выполнялось неравенство Крафта:

$$\sum_{j=1}^{K} 2^{-k_j} \le 1.$$

Доказательство этого утверждения приводится в п. 4.2.2.

Для доказательства того, что в качестве числа H выступает энтропия источника, необходимо, как и ранее, доказывать прямую и обратную теоремы кодирования, утверждения которых распространяются и на случаи неравномерного кодирования.

4.2.2. Кодовые деревья. Неравенство Крафта

Удобным средством описания префиксных кодов являются *кодовые деревья*.

Двоичным кодовым деревом называется *граф*, то есть такая система *узлов и связывающих их ребер*, в котором нет *петель* или *замкнутых путей* и в котором из каждого узла выходит не более двух ребер и в каждый узел, кроме одного (*корня дерева*), входит точно одно ребро. Узел, из которого не выходит ни одного ребра, называется *концевым*. Каждому из ребер, выходящему из узла, сопоставляется один символ кодового алфавита, причем различным ребрам, выходящим из одного узла, сопоставляются различные символы.

Пример двоичного кодового дерева для кода, приведенного ранее (см. табл. 1), представлен на рис. 5.

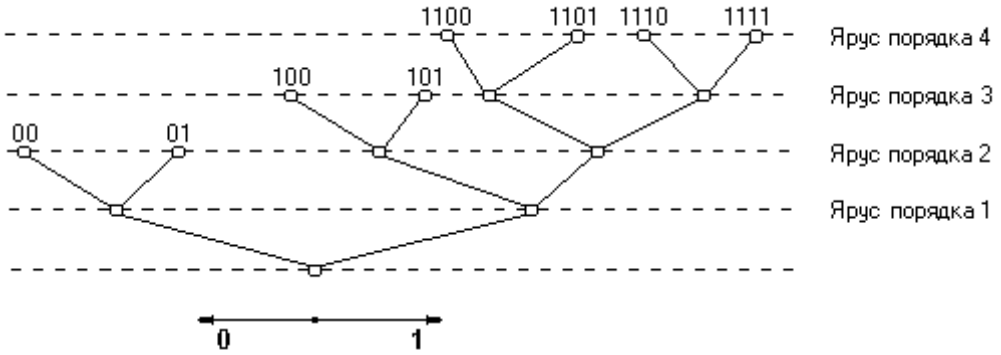

Рис. 5. Пример двоичного кодового дерева

Под кодовым деревом показано правило сопоставления ребер и двоичных символов: каждому правому ребру сопоставляется 1, каждому левому ребру – 0. Код является префиксным, если кодовые слова соответствуют только концевым узлам дерева. В противном случае код префиксным не является.

Узлы, отстоящие от корня на *i* ребер, образуют *ярус порядка i*. *Порядком узла* называется номер яруса, на котором находится этот узел.

Каждый концевой узел представляет кодовое слово кодера источника. Общее количество концевых узлов равно количеству сообщений источника *K*. Длина кодового слова k_j равна порядку яруса *s*, на котором находится концевой узел этого слова.

Теорема 3 (неравенство Крафта). Для того, чтобы существовал префиксный двоичный код с длинами кодовых слов $k_1, k_2, ..., k_K$, необходимо и достаточно, чтобы

$$\sum_{j=1}^{K} 2^{-k_j} \le 1. \qquad (45)$$

Доказательство необходимости. Заметим вначале, что порядок яруса равен длине кодовых слов, находящихся на этом ярусе. Это значит, что на ярусе s – ого порядка длина всех кодовых слов равна $k_j = s$, поэтому максимально возможное количество кодовых слов на ярусе порядка $s = k_j$ равно 2^{k_j}. Обозначим порядок яруса k_j. Пусть максимальная длина кодового слова или максимальный порядок яруса в кодовом дереве есть *k*. Рассмотрим один их концевых узлов на ярусе порядка k_j. Этот узел отстоит от яруса порядка *k* на $k - k_j$ ребер. Это значит, что вследствие того, что этот узел концевой, от него до яруса *k* не дошло 2^{k-k_j} узлов. От всех концевых узлов на ярусах порядка k_j до яруса k не дошло в общей сложности $\sum_{i=1}^{K} 2^{k-k_i}$ узлов. Но количество этих узлов, не дошедших до яруса *k* не может быть больше максимального количества узлов на этом ярусе, то есть

$$\sum_{j=1}^{K} 2^{k-k_j} \le 2^k .$$

Делением обеих частей этого равенства на 2^k, получим доказательство необходимости условия (45).

Доказательство достаточности. Необходимо доказать, что если выполняется неравенство (45), то этого и только этого достаточно для построения префиксного кода, то есть дерева с концевыми узлами порядков $k_1, k_2, ..., k_K$. Понятно, что если какой-либо s – й узел из этих возможных концевых узлов реализовался, то он реализовался α_s раз, s = 1, 2, ..., k. Поэтому неравенство (45) может быть переписано в виде :

$$\sum_{j=1}^{K} 2^{-k_j} = \alpha_1 2^{-k_1} + \alpha_2 2^{-k_2} + ... + \alpha_k 2^{-k} = \sum_{s=1}^{k} \alpha_s 2^{-s} \le 1. \qquad (46)$$

Из неравенства (46) выделим слагаемое, содержащее множитель α_i:

$$\sum_{s=1}^{i-1} \alpha_s 2^{-s} + \alpha_i 2^{-i} + \sum_{s=i+1}^{k} \alpha_s 2^{-s} \le 1.$$

Умножим обе части неравенства на 2^i, сделаем перестановку и перепишем это неравенство, усугубляя его:

$$\alpha_i \le 2^i - \sum_{s=1}^{i-1} \alpha_s 2^{i-s} - \sum_{s=i+1}^{k} \alpha_s 2^{i-s} \le 2^i - \sum_{s=1}^{i-1} \alpha_s 2^{i-s} . \qquad (47)$$

Теперь применим метод математической индукции. Начнем с доказательства того, что дерево, содержащее узлы на ярусе порядка 1 в количестве α_1 штук, может быть построено. Действительно, из (47) следует, что $\alpha_1 \le 2$. Так как максимально возможное количество концевых узлов порядка 1 есть 2, а $\alpha_1 \le 2$, то мы можем заключить, что дерево с α_1 концевыми узлами порядка 1 может быть построено.

В соответствии с методом математической индукции предположим, что дерево с α_s концевыми узлами порядка s, s = 1, 2, .. , i - 1 может быть построено. Докажем, что к этому дереву можно добавить еще α_i концевых узлов на ярусе порядка i. Из этого доказательства последует справедливость теоремы для любого i.

В самом деле, из верности предположения о построении дерева с α_{i-1} концевыми узлами следует, что до яруса порядка j не дошло $\sum_{s=1}^{i-1}\alpha_s 2^{i-s}$ возможных концевых узлов. Поскольку максимальное количество концевых узлов на ярусе порядка i равно 2^i, то $2^i - \sum_{s=1}^{i-1}\alpha_s 2^{i-s}$ – это количество свободных мест для возможных концевых узлов на этом ярусе. Но из (47) следует, что на этом ярусе может быть добавлено еще α_i узлов, количество которых не превосходит числа свободных мест для них. Таким образом, доказано, что при условии (45) дерево, и следовательно, префиксный код могут быть построены. Тем самым теорема, а с ней и неравенство Крафта доказаны.

4.2.3. Побуквенное неравномерное кодирование

Рассматриваем произвольный алфавит источника $\mathbf{X} = \{x_1, x_2, ..., x_M\}$, выбор букв из которого случаен, распределение вероятностей выбора букв $p(x_j) = p_j$. При побуквенном кодировании $K = M$, средняя длина кодовых слов, кодирующих буквы алфавита $k_{\text{ср}}(\mathbf{X}) = \sum_{j=1}^{K} p_j k_j$, k_j – длина кодового слова, кодирующего j – ю букву. При побуквенном кодировании, $m = 1$, $R_{\text{ист}} = k_{\text{ср}}(\mathbf{X})$. В этих условиях имеет место теорема

Теорема 4.

При побуквенном кодировании для любого неравномерного кода со свойством однозначного кодирования и декодирования справедливо неравенство

$$R_{\text{ист}} = k_{\text{ср}}(\mathbf{X}) \geq H(\mathbf{X}). \qquad (48)$$

Доказательство.

Средняя длина кодового слова

$$k_{\text{ср}} = \sum_{j=1}^{K} p_j k_j = \sum_{i=1}^{K} p_j \log_2 2^{k_j} .$$

Запишем разность

$$H(\mathbf{X}) - k_{\text{ср}}(\mathbf{X}) = -\sum_{j=1}^{K} p_j \log_2 p_j - \sum_{j=1}^{K} p_j \log_2 2^{k_j} = \sum_{j=1}^{K} p_j \log_2 \frac{2^{-k_j}}{p_j}.$$

Равенство $H(\mathbf{X}) = k_{\text{ср}}(\mathbf{X})$ возможно, когда $k_j = -\log_2 p_j = I(x_j)$.

Применим к последнему выражению неравенство $l\text{n}\, x \le x - 1$. В результате получим следующее выражение:

$$H(\mathbf{X}) - k_{\text{ср}}(\mathbf{X}) \le \log_2 e \sum_{j=1}^{K} p_j \left[\frac{2^{-k_j}}{p_j} - 1 \right] = \log_2 e \left[\sum_{j=1}^{K} 2^{-k_j} - 1 \right] \le 0,$$

которое справедливо в силу справедливости неравенства Крафта для однозначно кодируемых и декодируемых префиксных кодов.

Теорема доказана.

Двоичный неравномерный код будет иметь среднюю длину $k_{\text{ср}} = H(\mathbf{X})$ только в том случае, когда $k_j = -\log_2 p_j$, то есть, когда вероятности букв будут целыми отрицательными степенями двойки. С этим фактом мы столкнулись уже в примере п. 4.2.1.

Вообще коды источника, для которых средняя длина кодовых слов (и соответственно, скорость кодирования) равна наименьшему возможному значению, называются *оптимальными*.

Теорема 5.

Существует неравномерный двоичный код со свойством однозначного кодирования и декодирования, для которого

$$H(\mathbf{X}) \le k_{\text{ср}}(\mathbf{X}) \le H(\mathbf{X}) + 1. \qquad (49)$$

Доказательство.

Левое неравенство, а именно $H(\mathbf{X}) \le k_{\text{ср}}(\mathbf{X})$, следует из предыдущей теоремы.

При реальном кодировании равенство $k_j = -\log_2 p_j$ вряд ли достижимо. Поэтому для каждого $x_j \in \mathbf{X}$ может быть выполнено неравенство

$$-\log_2 p_j \le k'_j \le -\log_2 p_j + 1, \qquad (50)$$

где число k'_j есть не что иное, как длина кодового слова, кодирующего сообщение x_j.

Просуммируем неравенство (50) в целом с весами p_j:

$$-\sum_{j=1}^{K} p_j \log_2 p_j \le \sum_{j=1}^{K} p_j k'_j \le -\sum_{j=1}^{K} p_j \log_2 p_j + \sum_{j=1}^{K} p_j ,$$

откуда следует

$$H(\mathbf{X}) \le k_{\text{ср}} \le H(\mathbf{X}) + 1,$$

что и требовалось доказать.

От побуквенного кодирования теперь легко перейти к кодированию блоков длиной *s*. Для получения аналогичных результатов в пересчете на одну букву, из неравенств (48) и (49) в соответствии с теоремами 4, 5 получим, что для любого кода со свойством однозначного кодирования и декодирования

$$H(\mathbf{X}^s) \le k_{\text{ср}}(\mathbf{X}^s).$$

Кроме того, существует код со свойством однозначного кодирования и декодирования, для которого

$$k_{\text{ср}}(\mathbf{X}^s) \le H(\mathbf{X}^s) + 1.$$

Объединим эти два неравенства в одно, учитывая аддитивность функционала энтропии объединения независимых ансамблей:

$$sH(\mathbf{X}) \le k_{\text{ср}}(\mathbf{X}^s) \le sH(\mathbf{X}) + 1.$$

Поделив все части неравенства на s, получим:

$$H(\mathbf{X}) \le \frac{k_{\text{ср}}(\mathbf{X}^s)}{s} \le H(\mathbf{X}) + \frac{1}{s}.$$

Но отношение $\dfrac{k_{\text{ср}}(\mathbf{X}^s)}{s}$ есть средняя скорость кода на одну букву, поэтому

$$H(\mathbf{X}) \le R_{\text{ист}} \le H(\mathbf{X}) + \frac{1}{s}, \qquad (51)$$

и с увеличением длины блоков эти границы сближаются.

Тем самым теоремы 4, 5 свидетельствуют о том, что энтропия *H* алфавита источника играет определяющую роль в общей теории кодирования и декодирования сообщений источника и представляет собой не что иное, как *скорость создания информации. В точной формулировке скоростью создания информации дискретным источником при равномерном и*

неравномерном кодировании называется наименьшее число *H*, такое, что для любого $R_{\text{ист}} > H$ найдется длина кодируемых сообщений *m* и неравномерный код со скоростью кодирования $R_{\text{ист}}$, который допускает взаимно однозначное кодирование и декодирование. При $R_{\text{ист}} < H$, взаимно однозначное кодирование и декодирование недостижимо, и с увеличением длины кодируемых сообщений вероятность ошибочного кодирования и декодирования стремится к единице.

4.2.4. Код Шеннона

Пусть **X** = $\{x_1, x_2, ..., x_M\}$ - алфавит источника. Буквы в этом алфавите расставлены в порядке невозрастания их вероятностей. Если p_j - вероятность буквы x_j, то имеют место неравенства $p_1 \geq p_2 \geq ... \geq p_M$.Каждой букве сопоставляется вспомогательная величина q_i, вычисляемая по формуле:

$$q_1 = 0, \quad q_i = \sum_{j=1}^{i-1} p_j, \; i = 2,3,...,M \; .$$

Длина двоичного кодового слова, сопоставляемого букве x_j в коде К.Шеннона, равна $k_j = \text{Ent}\left[-\log_2 p_j\right] + 1$ разрядов после запятой в двоичном представлении числа q_i. Числа q_i суть правильные десятичные дроби. Если $\log_2 p_j$ – целое число, то $k_j = -\log_2 p_j$. Выражение Ent[•] означает целую часть числа, стоящего в скобках. Пример неравномерного побуквенного кодирования источника кодом Шеннона приведен в таблице 3.

Двоичное представление чисел q_j получается следующим образом.

Вначале записывается 0 с точкой. Затем число q_j умножается на два. Если результат умножения оказался равным или превышающим 1, после нуля записывается единица, и процесс заканчивается. Если результат умножения оказался меньше единицы, после нуля с точкой записывается 0. При превышении единицы, из результата умножения единица отнимается, и остаток умножается на два, после чего эта процедура продолжается.

Таблица 3

Пример побуквенного кодирования источника кодом Шеннона

j	Буква x_j	Вероятность p_j	$-\log_2 p_j$	Длина кодового слова $k_j = Ent\left[-\log_2 p_j\right]+1$	Число q_i	Двоичная запись q_i	Кодовое слово
1	а	0.3125	1.678072	2	0.0000	0.0000	00
2	б	0.1875	2.415037	3	0.3125	0.0101	010
3	в	0.1875	2.415037	3	0.5000	0.1000	100
4	г	0.1875	2.415037	3	0.6875	0.1011	101
5	д	0.0625	4.000000	4	0.8750	0.1110	1110
6	е	0.0625	4.000000	4	0.9375	0.1111	1111

В приведенном примере энтропия источника

$$H(\mathbf{X}) = -\sum_{j=1}^{6} p_j \log p_j = 0.3125 \cdot 1.678 + 3 \cdot 0.1875 \cdot 2{,}415 + 2 \cdot 0.0625 \cdot 4 = 2.3829 \text{ бит},$$

средняя длина кодовых слов

$$k_{\text{ср}}(\mathbf{X}) = \sum_{j=1}^{6} p_j k_j = 0.3125 \cdot 2 + 3 \cdot 0.1875 \cdot 3 + 2 \cdot 0.0625 \cdot 4 = 2.8125 \text{ бит}.$$

При равновероятном выборе букв было бы

$$H(\mathbf{X}) = \log_2 6 = 2.5849 \text{ бит}.$$

Исследуем свойства этого кода. Для любого слова длины k_i в соответствии с построением кода Шеннона можно записать

$$-\log_2 p_j \le k_j \le -\log_2 p_j + 1. \qquad (52)$$

Напомним, что при побуквенном кодировании К = М. Поэтому, усредняя неравенство (52) с вероятностями p_j, получим, что средняя длина кодовых слов подчиняется неравенствам:

$$-\sum_{j=1}^{K} p_j \log p_j \le \sum_{j=1}^{K} p_j k_j \le -\sum_{j=1}^{K} p_j \log p_j + 1$$

или

$$H(\mathbf{X}) \le k_{\text{ср}}(\mathbf{X}) \le H(\mathbf{X}) + 1, \qquad (53)$$

что совпадает с (49).

Перепишем левую часть неравенства (52) в показательном виде:

$$p_j \geq 2^{-k_j}$$

и просуммируем по j. Вследствие того, что $\sum_j p_j = 1$, получим неравенство Крафта, из которого следует, что код К.Шеннона префиксный.

Для декодирования префиксных кодов применяется следующий алгоритм.

Для всех значений длины слова k, начиная с 1 до максимальной длины, детектор определяет, совпадают ли первые символы декодируемой последовательности с каким-либо кодовым словом. Если совпадают, соответствующее сообщение выдается получателю. В противном случае контролируемая длина слова увеличивается на 1, и проверка повторяется.

Повышения эффективности кода Шеннона можно добиться путем объединения букв источника в последовательности длины $\boldsymbol{s}$ и применения кода Шеннона к этим последовательностям.

4.2.5. Код Хаффмена

Код Хаффмена обеспечивает минимально возможную среднюю длину кодовых слов, а значит, минимальную скорость кода при кодировании источника с заданными вероятностями букв. При этом свойство однозначного кодирования и декодирования кода не утрачивается. Для построения кода Хаффмена удобно применить кодовое дерево (см. рис. 6).

Для построения кода все сообщения (буквы) расставляются в порядке убывания вероятностей, как это было при построении кода Шеннона.

Находятся два сообщения с наименьшими вероятностями. Образуется новое множество сообщений, в котором два наименее вероятных сообщения заменяются одним, имеющим вероятность, равную сумме вероятностей заменяемых сообщений. В кодовом дереве вводится промежуточный узел и соединяется с узлами, соответствующими объединяемым сообщениям исходного множества. Эти же действия применяются к вновь построенному множеству. Процедура повторяется до тех пор, пока в множестве останется одно сообщение, вероятность которого равна 1.

Кодовые слова, которыми кодируются сообщения источника, формируются в соответствии с полученным кодовым деревом.

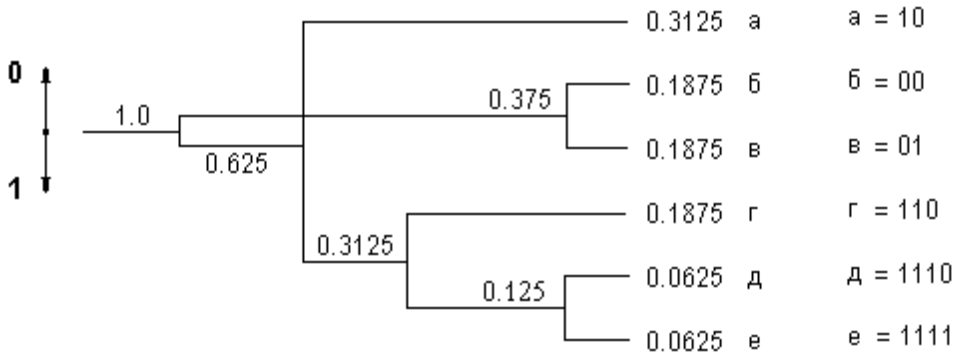

Рис. 6. Пример построения кодового дерева для кода Хаффмена

Пример построения кода Хаффмена приведен на рис. 6. Здесь используется то же множество сообщений, которое было использовано в примере разд. 4.2.4. Средняя длина кода, а следовательно, и скорость кодирования по Хаффмену составляет

$$k_{\text{ср}} = \sum_{j=1}^{6} p_j k_j = 0.3125 \cdot 2 + 2 \cdot 0.1875 \cdot 2 + 0.1875 \cdot 3 + 2 \cdot 0.0625 \cdot 4 = 2.4375 \text{ бит},$$

что меньше, чем тот же показатель кода Шеннона, и не на много больше энтропии источника, равной $H(\mathbf{X})$=2.3829 бит.

Для кода Хаффмена справедливо неравенство (52). Кроме того он оптимален. В статье, посвященной 25 – летию кода Хаффмена, Р. Галлагер получил оценку скорости кода Хаффмена в виде:

$$R_{\text{ист}} \le H(\mathbf{X}) + p_{\max} + 0.086, \qquad (54)$$

где $p_{\max}$ – наибольшая из вероятностей элементов ансамбля $\mathbf{X}$.

Сравнивая среднюю длину кодовых слов кода Хаффмена со средней длиной кодовых слов кода Шеннона, видим, что код Хаффмена эффективнее кода Шеннона, но сложнее его по построению.

Кодовые слова (последовательности) на выходе кодера источника, которыми кодируются сообщения источника $\mathbf{x}_j$, будем обозначать $\mathbf{m}_j$

5. КОДИРОВАНИЕ В КАНАЛЕ. ЛИНЕЙНЫЕ КОДЫ

5.1. Математическая модель канала связи, пропускная способность канала

Канал передачи информации был определен в разделе 1.2 как набор следующих технических средств: кодер канала, модулятор, линия связи, демодулятор, декодер канала. В математическую модель канала (эта модель является и информационной моделью) мы не будем включать модулятор и демодулятор, поскольку существует большое количество вариантов их технического воплощения (амплитудная, частотная, фазовая, импульсная, широтно-импульсная и другие виды модуляции), которые не влияют на излагаемую теорию. Несовершенство технических средств учитывается с помощью вероятностного описания ошибок и сбоев, которые вносят эти средства в передаваемые сообщения.

Здесь мы будем рассматривать канал связи, который представляет собой последовательное соединение трех компонентов: кодер канала, линия связи, декодер канала.

На вход кодера канала с выхода кодера источника поступают кодовые слова $\mathbf{m}$, все k символов которых являются информационными: $\mathbf{m} = (\mu_1, \mu_2, ..., \mu_{k-1}, \mu_k)$. Эти кодовые слова предназначены для обеспечения последующей передачи по каналу связи. При этом желательно обеспечить надежность передачи с исправлением возможных ошибок. С этой целью к информативным символам кодовых слов источника необходимо добавить избыточные символы, вследствие чего длина кода n на выходе кодера канала превышает k, то есть $n > k$.

В данном разделе рассматриваются исключительно каналы связи, на вход которых поступают кодовые последовательности с выхода кодера канала. В настоящей дисциплине рассматриваются только двоичные коды,

поэтому алфавитом входных последовательностей является множество **A**, состоящее из двух символов: 0 и 1. Таким же является алфавит **B** выходного кода. Это означает, что алфавит входных и выходных кодовых последовательностей, каждый из них, представляет собой конечное поле Галуа **GF(2)** с числом элементов, равным двум.

При передаче кодовых слов по каналу связи возможны ошибки. Если обозначить символы входного кода, как $0 \to a_0$ и $1 \to a_1$, а символы выходного кода, как $0 \to b_0$ и $1 \to b_1$, то вероятности ошибочной передачи символов можно записать через условные вероятности:

$$p(a_1 / b_0), p(a_0 / b_1).$$

Тогда вероятности безошибочной передачи

$$p(a_0/b_0) = 1 - p(a_1/b_0), \quad p(a_1/b_1) = 1 - p(a_0/b_1).$$

Передача двоичных символов и вероятности ошибок в рассматриваемых каналах могут быть представлены с помощью графа переходов (см. рис. 7). **A** – входной алфавит канала, **A**={0,1}, **B**={0,1} – выходной алфавит канала. Множества последовательностей длины *n*, образованных символами входного и выходного алфавитов, являются *n* – кратными декартовыми произведениями множеств **A** и **B**: $\mathbf{A}^n, \mathbf{B}^n$

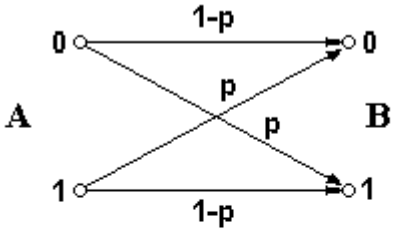

Рис. 7.
Граф двоичного симметричного канала с дискретным временем без памяти

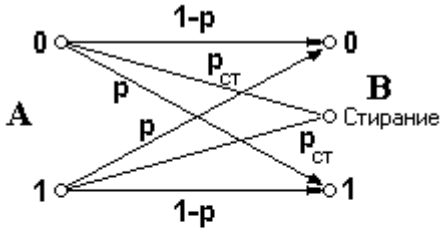

Рис. 8.
Граф двоичного симметричного канала без памяти со стиранием

В тех случаях, когда для связи используются коротковолновый радиодиапазон, и возможны затухания сигнала, или когда демодулятор и следующий за ним амплитудный дискриминатор имеет зону нечувствительности и в иных подобных случаях в канале возможны *стирания*. Модель двоичного канала со стиранием представлена графом на рис. 8.

Канал, у которого $p(a_1 / b_0) = p(a_0 / b_1) = p$, называется *симметричным каналом*. Если ошибка при передаче каждого символа происходит независимо от передаваемого сообщения и независимо от ошибки предыдущего символа, а кроме того вероятность ошибки символа не зависит от времени, то такой канал называется *стационарным каналом без памяти*, а возникающие при этом ошибки – *независимыми ошибками*. В подобных каналах *одиночные* ошибки более вероятны, чем многократные. В каналах с памятью преимущественный вид ошибок – это *серии* или *пачки ошибок*.

Итак, впредь, за редкими исключениями, мы будем рассматривать *двоичный симметричный стационарный канал без памяти с дискретным временем*. В дальнейшем, слова "стационарный" и "без памяти с дискретным временем" мы будем опускать, подразумевая, что рассматриваются только такие каналы. Примем также, что кодирование в канале равномерное, длина каждой кодовой последовательности (длина кода) постоянна и равна n.

Основной информационной характеристикой каналов передачи информации является *информационная емкость канала*. Информационной емкостью двоичного симметричного канала называется максимум средней взаимной информации

$$C^* = \max_{p(x)} I(\mathbf{A};\mathbf{B}) = \max_{p(x)} \left[H(\mathbf{A}) - H(\mathbf{A}/\mathbf{B}) \right] \text{ бит/симв}, \quad (55)$$

где максимум отыскивается по всем возможным распределениям $p(x)$ на множестве входных символов **A.** Для симметричного канала средняя условная энтропия $H(\mathbf{A}/\mathbf{B})$

$$H(\mathbf{A}/\mathbf{B}) = -\sum_{j=0}^{1} p(b_j) \sum_{i=0}^{1} p(a_i / b_j) \log\left(p(a_i / b_j)\right)$$

Если $i = j$, то $p(a_i / b_j) = 1 - p$, если же $i \neq j$, то $p(a_i / b_j) = p$. Поэтому последнее выражение можно переписать следующим образом:

$$H(\mathbf{A}/\mathbf{B}) = -p(b_0)\left[(1-p)\log(1-p) + p\log p\right] - p(b_1)\left[p\log p + (1-p)\log(1-p)\right] =$$
$$= -\left[p\log p + (1-p)\log(1-p)\right]\left[p(b_0) + p(b_1)\right] = -\left[p\log p + (1-p)\log(1-p)\right].$$

Очевидно, что максимум в (55) достигается при равномерном распределении сообщений из множества **A**. Поэтому $\max_{p(x)}[H(\mathbf{A})] = \log 2 = 1$.

Окончательно получим, что информационная емкость двоичного симметричного канала без памяти

$$C^* = 1 + p\log p + (1-p)\log(1-p) \quad (56)$$

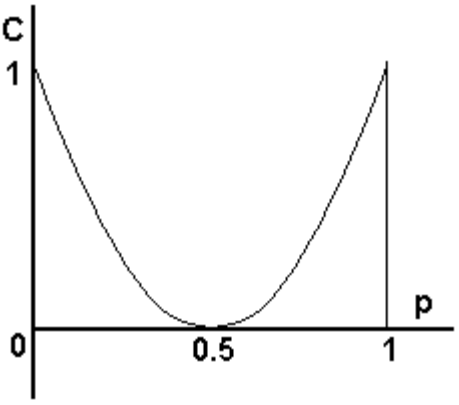

Рис. 9. Пропускная способность двоичного симметричного канала

График зависимости пропускной способности двоичного симметричного канала без памяти, построенный по зависимости (56), приведен на рис. 9. Из этого графика следует, что при $p = 0.5$ передача информации невозможна, а при $p > 0.5$ символы искажаются настолько, что вероятность искажения информации оказывается больше вероятности ее правильной передачи. В этом случае при декодировании сообщений следует принять стратегию: при получении символа y_j принимать решение в пользу $x_i, i \neq j$.

Информационная емкость канала со стиранием

$$C^* = -(1-p_{\text{СТ}})\log\frac{1-p_{\text{СТ}}}{2} + p\log p + (1-p-p_{\text{СТ}})\log(1-p-p_{\text{СТ}}) \text{ бит/симв.}$$

Если в таком канале отсутствуют ошибки, то есть $p = 0$, тогда

$$C^* = (1-p_{\text{СТ}}).$$

Теперь могут быть сформулированы прямая и обратная теоремы кодирования в дискретном канале связи. Доказательства не приводятся.

Теорема 6 (прямая теорема).

Пусть C^* – информационная емкость дискретного канала, $R_к$ – скорость кодирования в канале. При любом $R_К < C^*$ и любом положительном сколь угодно малом δ *существует код*, максимальная вероятность ошибки передачи которого по этому каналу не превосходит δ.

Теорема 7 (обратная теорема).

Пусть C^* – информационная емкость дискретного канала, $R_к$ – скорость кодирования в канале. Если $R_к = C^* + \varepsilon$, где ε – любое сколь угодно малое положительное число. Тогда существует положительное число δ, зависящее от $R_к$, такое, что вероятность ошибочной передачи любого кода по этому каналу будет не меньше δ. Это значит, что при $R_к = C^* + \varepsilon$ вероятность ошибки не может быть сделана произвольно малой.

Информационная емкость симметричных стационарных каналов без памяти называется *пропускной способностью* и обозначается С, то есть

$$C^* = \max_{p(x)} I(\mathbf{A},\mathbf{B}) = \max_{p(x)} \left[H(\mathbf{A}) - H(\mathbf{A}/\mathbf{B})\right] = C .$$

Из последних теорем 6 и 7 следует, что при постоянном канале безошибочная передача информации может быть достигнута путем вариации скорости кодирования, то есть увеличением длины кодовых последовательностей *n* при постоянстве их количества *K*, или уменьшением количества *K* кодовых последовательностей при постоянстве их длины *n*. Поскольку количество кодовых последовательностей определяется в конечном счете источником, мы выберем первый вариант.

5.2. Корректирующие свойства кодов, параметры кодов

В соответствии с прямой теоремой кодирования код может обеспечить сколь угодно малую вероятность ошибочной передачи кодовых слов по каналу, если его скорость

$$R_{\text{к}} = \frac{\log_2 K}{n} < C \le 1 . \qquad (57)$$

Это означает, что если длина кодовых слов источника одинакова и равна $k = \log_2 K$, то для обеспечения неравенства (57) кодовые слова канала должны содержать кроме *k* минимально необходимых информационных символов дополнительно $r = n - k$ символов, которые называются *избыточными символами кода* и придают коду канала корректирующие свойства.

Введем несколько понятий. Пусть $\mathbf{A}^n$ – множество двоичных последовательностей длиной *n* над полем **GF(2)** (полем Галуа)**,** свойства которого изложены в приложении 1. Пусть две последовательности принадлежат этому множеству: $\mathbf{a}_1, \mathbf{a}_2 \in \mathbf{A}^n$. *Расстояние Хемминга* $d(\mathbf{a}_1, \mathbf{a}_2)$ между двумя последовательностями – количество позиций, в которых эти последовательности различаются.

П р и м е р. Расстояние Хемминга между словами (кодовыми последовательностями) $\mathbf{a}_1 = 00110011$ и $\mathbf{a}_2 = 10101010$ равно *4*. Длина кодовых слов $n = 8$.

Пусть $\mathbf{a}_i, \mathbf{a}_j$- кодовые слова кода канала. Тогда *кодовое расстояние Хемминга* (или просто *расстояние кода*) – это минимальное расстояние между любыми различными словами кода $d = \min\limits_{i \neq j} d(\mathbf{a}_i, \mathbf{a}_j)$.

Вес кодового слова - количество единиц в кодовом слове, обозначается $w(\mathbf{a}_i)$. Вес слова $\mathbf{a}_1 = 00110011$ равен *4*. Расстояние $d(\mathbf{a}_i, \mathbf{a}_j) = w(\mathbf{a}_i - \mathbf{a}_j) = w(\mathbf{a}_i + \mathbf{a}_j)$, где сложение и вычитание выполняются по mod2. Эти операции по mod2 эквивалентны (см. также п. 3 Приложения 2).

В связи с этим справедливо равенство $d = \min\left[w(\mathbf{a}_i - \mathbf{a}_j)\right]$, $\mathbf{a}_i \neq \mathbf{a}_j$.

Расстояние Хемминга и вес кодовых слов обладают свойством расстояния, поскольку это неотрицательные величины, подчиняющиеся неравенству треугольника. Докажем это.

Пусть $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \in \mathbf{A}^n$ – произвольные двоичные кодовые слова над конечным двоичным полем **GF(2)**. Пусть a_{1i}, a_{2i}, a_{3i} – символы, стоящие на *i* - й позиции в каждом из них. Поскольку все кодовые символы могут принимать значения из поля **GF(2),** то есть 0 или 1, то расстояние между символами на *i* - й позиции

$$d(a_{1i}, a_{2i}) \leq d(a_{1i}, a_{3i}) + d(a_{2i}, a_{3i}).$$

Суммируя по всем *n* позициям, получим неравенство треугольника:

$$d(\mathbf{a}_1, \mathbf{a}_2) \leq d(\mathbf{a}_1, \mathbf{a}_3) + d(\mathbf{a}_2, \mathbf{a}_3).$$

Вес суммы двух слов, каждый символ которых также из **GF(2)**, не превышает суммы весов слагаемых:

$$w(\mathbf{a}_1 + \mathbf{a}_2) \le w(\mathbf{a}_1) + w(\mathbf{a}_2).$$

Это происходит потому, что при сложении символов по модулю 2 количество единиц в результате может разве только уменьшиться из-за того, что (1+1)mod 2 = 0. Равенство возможно лишь в том случае, когда единицы в этих словах – слагаемых расположены на разных позициях.

Код, образованный различными $N = 2^n$ кодовыми словами длиной *n*, имеет *d* = 1. Этот код не имеет избыточности и не предоставляет возможности обнаруживать или исправлять ошибки, потому что ошибка при передаче одного символа переводит одно слово этого кода в другое кодовое слово. Это наглядно показано на рис. 10, а , где жирными точками отмечены кодовые слова, отстоящие друг от друга на расстоянии, равном 1. Кодовые слова, расстояние между которыми равно 2, показаны жирными точками на рис. 10, б. Это *разрешенные кодовые слова*. Остальные точки изображают неразрешенные кодовые слова, отсутствующие в коде.

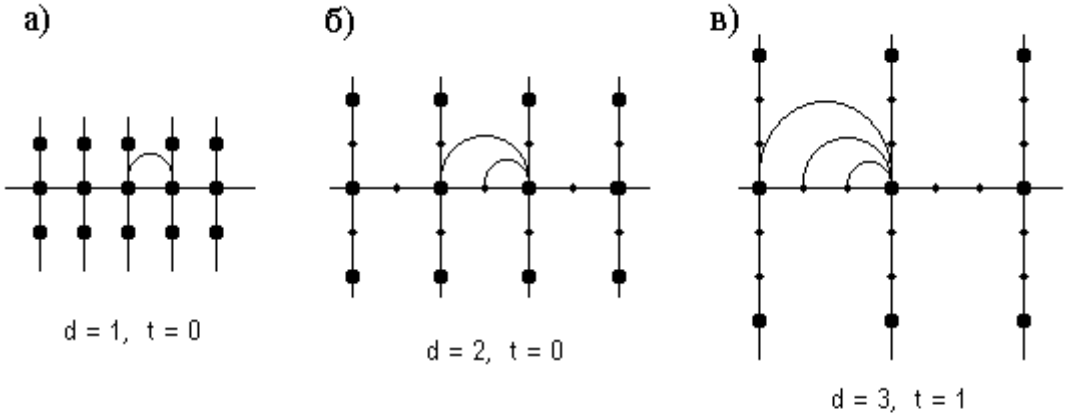

Рис. 10. Кодовое расстояние и ошибки в передаче кодовых слов

Из рисунка 10, б видно, что код с расстоянием 2 дает возможность обнаружить одну ошибку, поскольку одна ошибка приводит любое разрешенное кодовое слово к неразрешенному. Ошибки в двух позициях переводят кодовое слово в другое слово из этого же кода, и эти ошибки обнаружены быть не могут. На рис. 10, в изображены разрешенные кодовые слова с расстоянием между ними, равном 3. Из рисунка видно, что такой код допускает исправление одной ошибки и обнаружение двух ошибок. Для этого декодирование должно выполняться по принципу минимально-

го расстояния (ДМР), то есть при появлении на входе декодера неразрешенного кодового слова декодер должен отыскать разрешенное кодовое слово, ближайшее к принятому. Такое декодирование является также декодированием по принципу максимальной апостериорной вероятности. Результат этого декодирования является наиболее правдоподобным, что отвечает также принципу максимального правдоподобия.

Понятно, что количество ошибок, исправляемых кодом, равно $t = (d-1)/2$, а количество обнаруживаемых ошибок равно $s = d - 1$. Иными словами, для исправления t ошибок необходимо обеспечить кодовое расстояние $d = 2t + 1$, а для обнаружения s ошибок нужно обеспечить кодовое расстояние $d = s + 1$. Если наименьшее количество информационных символов кода источника, необходимое для безизбыточной передачи сообщений источника, равно $k = \mathrm{Ent}[\log_2 K]+1$, и для исправления или обнаружения ошибок в код введено r избыточных символов, то длина кода будет $n = k + r$, а расстояние кода $d = r + 1$. Это объясняется тем, что расстояние безизбыточного кода равно 1, поэтому минимальное кодовое расстояние избыточного кода превышает количество избыточных символов по меньшей мере на единицу. Код длиной n, содержащий k информационных символов и $r = n - k$ избыточных, обозначается как (n, k) – код.

Возникает вопрос, какому условию должно удовлетворять количество избыточных символов, вводимых в код длины n, чтобы обеспечить передачу K сообщений источника и придать этому коду способность к исправлению одиночной ошибки.

Одиночные ошибки в кодовых словах длиной n могут возникать на любой из n позиций слова. Поэтому для распознавания и исправления такого количества вариантов одиночных ошибок необходимо предусматривать в кодовом слове такое количество ненулевых комбинаций избыточных символов, чтобы оно было не меньше, чем количество позиций появления ошибок, то есть n. Если количество избыточных символов равно $r = n - k$, то количество ненулевых двоичных комбинаций, составленных из этих символов, равно $2^{n-k} - 1$. Поэтому для кодов, исправляющих все одиночные ошибки на каждой из n позиций, необходимо составить из $r = n - k$ двоичных символов не менее n различных последовательностей.

Это значит, что $2^{n-k} - 1 \geq n$. Такое требование возникает потому, что для исправления одиночной ошибки нужно узнать место ее возникновения и затем добавить туда 1. Для двоичного представления номера места ошибки в (n, k) – коде необходимо $r = \log_2 n$ знаков.

И вообще, для исправления t ошибок при передаче кодового слова длиной n и кодовым расстоянием $d = 2t + 1$ нужно узнать и оцифровать места каждой ошибки, поэтому необходимо обеспечить, чтобы количество двоичных комбинаций из $n - k$ избыточных символов было не меньше, чем количество комбинаций, которые могут быть порождены одной, или двумя, или тремя, вплоть до t ошибками включительно, то есть

$$2^{n-k} \geq \sum_{i=0}^{t} C_n^i ,$$

где C_n^2 - количество ошибочных последовательностей, порождаемых двумя ошибками в одном слове длиной n, C_n^3 - количество ошибочных последовательностей, порождаемых тремя ошибками в одном слове длиной n, и так далее.

Видно, что при введении r избыточных символов и при неизменной длине кодового слова n количество передаваемых сообщений источника уменьшается. При желании сохранить количество K передаваемых сообщений с целью обеспечения необходимой *корректирующей способности кода* приходится увеличивать длину кодового слова на $r = n - k$ символов и тем самым уменьшать скорость кода. Коды максимального объема, обеспечивающие заданную корректирующую способность при минимально возможной избыточности, или, что то же самое, при минимальной длине кодовых слов, называются *оптимальными*.

5.3. Границы параметров корректирующих кодов

Скорость кода канала $R_к$ определена выше, как отношение логарифма числа кодовых слов K к длине кодового слова n. С целью упрощения выводов для числа кодовых слов будем строить асимптотические границы при $n \to \infty$, в зависимости от относительного расстояния кода и относительного количества исправляемых им ошибок. Границы мы будем стро-

ить при пропорциональном увеличении кодового расстояния d и количества исправляемых ошибок t, то есть

$$d = n\delta, \qquad t = n\tau, \qquad \tau \approx \delta / 2, \qquad n \to \infty,$$

где δ и τ суть относительные значения расстояния и числа исправляемых ошибок. Корректирующая способность кода задана соотношением между δ и τ.

Для дальнейшего анализа введем понятие шара Хемминга.

Пусть $\mathbf{a}$ – кодовое слово, $\mathbf{a} \in \mathbf{A}^n$. *Шаром Хемминга* $S_t(\mathbf{a})$ с центром в точке $\mathbf{a}$ и радиусом t называется множество слов, удаленных от центра $\mathbf{a}$ на расстояние t или меньше. Мы рассматриваем только двоичные кодовые слова. Число слов в таком шаре мы будем обозначать $|S_t(\mathbf{a})|$ и называть объемом шара. Объем шара Хемминга не зависит от выбора его центра и равен

$$|S_t(\mathbf{a})| = \sum_{i=0}^{t} C_n^i, \qquad (58)$$

где C_n^i – число сочетаний из n (длины слова, числа позиций в слове), по i (то есть по количеству позиций, в которых отличаются друг от друга кодовые слова, отстоящие от центра шара на расстоянии i).

Это в самом деле так, поскольку количество вариантов распределения позиций слова, в которых между словами имеется i различий, определяется именно числом сочетаний из n по i. По определению шара, внутри него находятся все слова, отличающиеся от **a** не более, чем в t позициях, в том числе и само слово **a**. Поэтому общее количество кодовых слов внутри и на поверхности шара определяется как сумма (58). В разделе 5.2 это количество было уже определено.

Теорема 8 (граница Хэмминга). Для любого двоичного кода длины n с минимальным кодовым расстоянием $d = 2t + 1$ число кодовых слов K удовлетворяет неравенству

$$K \le \frac{2^n}{|S_t(\mathbf{a})|} = \frac{2^n}{\sum_{i=0}^{t} C_n^i}. \qquad (59)$$

Доказательство.

Центры $\mathbf{a}$ этих шаров – кодовые слова избыточного кода. При $d = 2t + 1$ шары Хемминга не пересекаются. В противном случае при пересечении шаров $S_t(\mathbf{a}_1)$ и $S_t(\mathbf{a}_2)$ нашлось бы кодовое слово $\mathbf{a}_3$, принадлежащее обоим шарам, и поэтому $d(\mathbf{a}_1, \mathbf{a}_3) \le t$, и $d(\mathbf{a}_2, \mathbf{a}_3) \le t$. Но в силу неравенства треугольника

$$d(\mathbf{a}_1, \mathbf{a}_2) \le d(\mathbf{a}_1, \mathbf{a}_2) + d(\mathbf{a}_2, \mathbf{a}_3) \le 2t,$$

а это противоречит условию теоремы.

При такой конструкции шаров некоторые двоичные последовательности из множества $\mathbf{A}^n$ могут оказаться вне этих шаров. Поэтому общий объем всех шаров $K\sum_{i=0}^{t} C_n^i$ не может превысить числа 2^n всех возможных двоичных последовательностей длины n. Таким образом мы получили неравенство

$$K\sum_{i=0}^{t} C_n^i \le 2^n,$$

из которого следует утверждение теоремы.

Доказательство закончено.

Если количество сообщений источника – точная степень двойки, то из доказанной теоремы следует:

$$2^k \sum_{i=0}^{t} C_n^i \le 2^n, \text{ или } 2^{n-k} \ge \sum_{i=0}^{t} C_n^i.$$

Найдем асимптотические выражение для длины кода и его корректирующей способности при больших значениях n. Для этого будем считать, что в асимптотике число t растет пропорционально n с коэффициентом пропорциональности $\tau \le 1/2$. *Это означает, что количество исправляемых ошибок не может быть больше половины длины слова*, иначе расстояние кода $d = 2t + 1$ будет превышать длину кода, а это невозможно.

Построим асимптотическую оценку для $|S_t(\mathbf{a})|$ при $n \to \infty$ и $t = n\tau$:

$$|S_{n\tau}(\mathbf{a})| = \sum_{i=0}^{n\tau} C_n^i > C_n^{n\tau} = \frac{n!}{(n\tau)!\,(n - n\tau)!}.$$

Воспользуемся формулой Стирлинга

$$n! \approx n^n e^{-n} \sqrt{2\pi n}.$$

Тогда

$$C_n^{n\tau} \approx \frac{n^n e^{-n} \sqrt{2\pi n}}{(n\tau)^{n\tau} (n - n\tau)^{n-n\tau} e^{-n\tau} e^{-(n-n\tau)} 2\pi\sqrt{n\tau(n-n\tau)}} =$$

$$= \frac{n^n}{\sqrt{2\pi\tau}(n\tau)^{n\tau} (n - n\tau)^{n-n\tau} \sqrt{(n-n\tau)}}.$$

Пренебрежем квадратными корнями в знаменателе. Корень $\sqrt{2\pi\tau}$ – постоянное число, на асимптотику не влияет. Корень $\sqrt{n-n\tau}$ добавляет к показателю степени бинома (n - $n\tau$) только 1/2, что также на асимптотику не влияет, поскольку степень (n - $n\tau$) отличается от степени (n - $n\tau$+1/2) всего лишь на постоянную, роль которой при $n \to \infty$ незначительна и убывает.

Оставшееся выражение запишем в виде степени двойки:

$$= 2^{n\log n - n\tau \log n\tau - n(1-\tau)\log n(1-\tau)}.$$

Рассмотрим отдельно показатель степени, раскроем скобки и приведем подобные члены:

$$n[\log n - \tau(\log\tau + \log n) - (1-\tau)(\log(1-\tau) - \log n)] =$$

$$= n[-\tau\log\tau - (1-\tau)\log(1-\tau)] = nh(\tau),$$

где выражение $[-\tau\log\tau - (1-\tau)\log(1-\tau)]$ обозначено, как $h(\tau)$, а структура этого выражения напоминает структуру формулы для энтропии.

Таким образом для больших n мы получили асимптотическое неравенство:

$$|S_t(\mathbf{a})| > 2^{nh(\tau)}.$$

Подставим последнее выражение вместо знаменателя в неравенство (59) и усугубим его:

$$K < \frac{2^n}{2^{nh(\tau)}} = 2^{n(1-h(\tau))}.$$

Логарифмируя последнее неравенство $log_2 K < n(1-h(\tau))$ и разделив обе его части на n, получим:

$$R_{\text{к}} = \frac{\log_2 K}{n} < 1 - h(\tau) \text{ бит/симв},$$

где $\tau = t / n$.

Кодовое расстояние d при больших n также растет пропорционально n: $d = \delta n$. Тогда с учетом асимптотики $2\tau \approx \delta$ или $\tau \approx \delta / 2$ получим:

$$R_{\text{к}} \le 1 - h(\delta / 2) \qquad (60)$$

Выражение (60) есть верхняя *граница Хемминга* для относительной доли избыточных символов и скорости корректирующего кода или, что то же самое - для длины кода и его корректирующей способности.

На границе Хемминга лежат коды, для которых в (60) достигается равенство. Эти коды называются *плотно упакованными* или *совершенными*.

Теорема 9 (граница Варшамова-Гилберта). Существует двоичный корректирующий код длины n с кодовым расстоянием d, объем которого K удовлетворяет неравенству:

$$K \ge \frac{2^n}{|S_{d-1}(\mathbf{a})|} = \frac{2^n}{\sum_{i=0}^{d-1} C_n^i}, \qquad (61)$$

где $S_{d-1}(\mathbf{a})$ - шары Хемминга радиуса d - 1, с центрами в точках $\mathbf{a}$.

Доказательство.

Шары Хемминга, построенные для доказательства теоремы 8, не обязательно содержат в себе все без исключения двоичные последовательности из $\mathbf{A}^n$. Некоторые из этих последовательностей могут оказаться не попавшими ни в один шар. Именно по этой причине далеко не все коды являются плотно упакованными.

Для доказательства настоящей теоремы мы построим шары Хемминга так, чтобы каждая последовательность из $\mathbf{A}^n$ входила, по крайней мере, хотя бы в один шар и вне шаров не оставалось ни одной последовательности.

Будем строить двоичный код длины n с минимальным расстоянием d следующим образом. На множестве $\mathbf{A}^n$ всех двоичных последовательностей длины n последовательно строим шары Хемминга $S_{d-1}(\mathbf{a})$, центры которых – кодовые слова. Диаметр этих шаров равен d – 1. При постро-

нии этих шаров в $\mathbf{A}^n$ мы будем выбирать центр каждого следующего шара так, чтобы он не входил во все ранее построенные. Пусть K - номер последнего построенного шара, после которого множество $\mathbf{A}^n$ оказалось исчерпанным. По построению, совокупность точек, в которых оказались центры построенных таким образом шаров, есть двоичный код, минимальное расстояние его есть d, а объем кода – K. Каждое слово из $\mathbf{A}^n$ входит хотя бы в один шар, поэтому объем всех построенных шаров, равный $K\left|S_{d-1}(\mathbf{a})\right|$, не может быть меньше объема всего множества $\mathbf{A}^n$, равного 2^n, то есть

$$K\left|S_{d-1}(\mathbf{a})\right| \geq 2^n. \qquad (62)$$

Принимая во внимание, что $d - 1 = 2t$, и подставляя в (58) d - 1 вместо t, получим

$$\left|S_{d-1}(\mathbf{a})\right| = \sum_{i=0}^{d-1} C_n^i, \qquad (63)$$

откуда и из неравенства (62) непосредственно следует справедливость утверждения теоремы. Доказательство завершено.

Найдем асимптотическое выражение для неравенства (61) при $n \to \infty$.

Вначале получим оценку сверху для объема шара $S_{d-1}(\mathbf{a})$ при $d \leq n/2$. Слагаемые суммы, стоящей в правой части (63), вначале возрастают. Наибольшего значения достигают слагаемые, номера которых равны Ent$[n/2]$ или Ent$[n/2] + 1$. Если, по нашему условию, $d \leq n/2$, то в сумме (63) наибольшим будет последнее слагаемое. Поэтому d таких слагаемых будет больше суммы

$$\left|S_{d-1}(\mathbf{a})\right| = \sum_{i=0}^{d-1} C_n^i \leq dC_n^{d-1}.$$

По аналогии с предыдущим будем считать, что в асимптотике число d растет пропорционально n с коэффициентом пропорциональности $\delta \leq 1/2$, то есть $d = \delta n$. Тогда

$$\left|S_{d-1}(\mathbf{a})\right| = \sum_{i=0}^{d-1} C_n^i < n\delta\, C_n^{\mathrm{Ent}[n\,\delta]}.$$

Воспользуемся формулой Стирлинга:

$$n\delta C_n^{n\delta} \approx \frac{n\delta n^n}{(n\delta)^{n\delta}(n-n\delta)^{n-n\delta}\sqrt{2\pi}\sqrt{n\delta(n-n\delta)}}.$$

Поскольку все сомножители, не зависящие от n, на асимптотику не влияют, выполняя преобразования, аналогичные предыдущим, получим

$$|S_{d-1}(\mathbf{a})| \le 2^{nh(\delta)},$$

где $\delta = (d-1)/n$.

Усугубим неравенство (61) следующим образом

$$K \ge \frac{2^n}{2^{nh(\delta)}} = 2^{n(1-h(\delta))}.$$

Путем логарифмирования этого неравенства и деления его на n получим асимптотическое выражение для нижней границы Варшамова - Гилберта:

$$R_к \ge 1 - h(\delta). \qquad (64)$$

Рис. 11. Асимптотические границы скорости корректирующих кодов

Асимптотические выражения для границ Хемминга и Варшамова-Гилберта представлены в графическом виде на рис. 11. Эти границы означают следующее. Из двоичных последовательностей могут быть построены коды с заданной корректирующей способностью, то есть с заданным расстоянием d. Этому расстоянию соответствует значение δ. Вертикальная линия, проведенная из этой точки оси абсцисс, пересечет обе границы в точках с ординатами $R_{\min}, R_{\max}$. Отрезку этой прямой, лежащему между этими точками, соответствуют коды, которые могут быть сконструированы, и скорость которых будет между $R_{\min}$ и $R_{\max}$. Точно так же, если задать некоторую скорость кода канала, например, R_0,

то могут быть сконструированы коды с этой скоростью и с корректирующей способностью, лежащей между построенными границами.

Полученные границы позволяют сделать вывод о том, что при постоянной корректирующей способности кода (*d = const, t = const*) уменьшение длины кодовых слов *n* позволяет увеличить скорость кода, но не больше, чем это установлено границей Хемминга. При постоянных относительных значениях кодового расстояния с увеличением длины *n* кодовых слов их количество, а значит, и количество сообщений уменьшается, но не может быть меньше, чем это соответствует границе Варшамова-Гилберта. Наибольшее количество кодовых слов при неизменной корректирующей способности может теоретически достигать значений, соответствующих границе Хемминга. Однако лишь некоторые коды приближаются к этой границе. Рецептов конструирования кодов, соответствующих верхней границе, пока не существует.

5.4. Некоторые примеры кодов, исправляющих ошибки. Коды Хемминга

Рассмотрим вначале коды с кодовым расстоянием, равным 3. Как было отмечено, эти коды способны исправлять одну ошибку. Ранее в разделе 5.2 было отмечено, что для исправления одиночной ошибки необходимо, чтобы выполнялось неравенство $2^{n-k} - 1 \geq n$. Это же неравенство следует из теоремы 8, как частный случай при $t = 1$. Коды, для которых $d = 3$ и последнее неравенство является равенством, были впервые описаны Р.Хеммингом в 1950 году. Примеры некоторых двоичных кодов Хемминга приведены в табл. 4.

Коды, исправляющие одну ошибку, для которых выполняется равенство $n = 2^{n-k} - 1$, называются *совершенными* или *плотно упакованными кодами*. Вообще все коды, лежащие на границе Хемминга, являются совершенными. В частности, коды Хемминга являются совершенными.

Таблица 4

Параметры некоторых плотно упакованных кодов Хемминга, исправляющих однократную ошибку

Обозначение кода	$n - k$	$n = 2^{n-k} - 1$	k	d	K
(3, 1)	2	3	1	3	2
(7, 4)	3	7	4	3	16
(15, 11)	4	15	11	3	2096
(31, 26)	5	31	26	3	2^{26}

Такое наименование эти коды получили вследствие того, что равенство (60) достигается в случае, когда шары Хемминга не пересекаются и вне этих шаров отсутствуют кодовые слова. Другими словами, все множество кодовых последовательностей полностью покрывается непересекающимися шарами Хемминга. В таблице 4 приведены примеры плотно упакованных кодов.

Другим примером кода, исправляющего ошибки, является расширенный код Хемминга. Параметры некоторых двоичных расширенных кодов Хемминга приведены в табл. 5. При одинаковом с кодами Хемминга количестве информационных символов расширенные коды Хемминга длиннее на 1 символ и обладают способностью к обнаружению на 1 большего количества ошибок.

Таблица 5

Расширенные коды Хемминга

Обозначение кода	$n - k$	$2^{n-k} - 1$	k	d
(4, 1)	3	7	1	4
(8, 4)	4	15	4	4
(16, 11)	5	31	11	4
(32, 26)	6	63	26	4

Здесь уместно еще раз напомнить, что в общем случае при кодовом расстоянии d и количестве k информационных символов кода в соответ-

ствии с теоремой 8 длина n кодового слова должна удовлетворять неравенству

$$2^{n-k} \geq \sum_{i=0}^{t} C_n^i, \quad d = 2t + 1, \tag{65}$$

где t – количество ошибок, которые данный код способен исправить.

5.5. Линейные коды

Существует несколько видов и способов задания линейных кодов:

линейные коды, задаваемые в векторных пространствах с помощью матриц;

линейные циклические коды, задаваемые многочленами с коэффициентами из конечных групп;

линейные групповые коды, образующие алгебраические структуры – группы.

Один и тот же код может относиться к каждой из перечисленных групп.

Две первые из указанных разновидностей кодов, методы их задания и декодирования в настоящем учебном пособии будут рассматриваться последовательно.

Начнем с рассмотрения простейших случаев задания линейных кодов в векторных пространствах.

5.5.1. Векторное представление линейных кодов

В векторном пространстве линейные коды определяются следующим образом.

Линейным двоичным кодом длины n с k информационными символами (*линейным* (*n,k*)-*кодом*) называется k-мерное линейное подпространство $\mathbf{A}_k^n$ n-мерного линейного векторного пространства $\mathbf{A}^n$ над конечным полем (полем Галуа) **GF(2)**. Длина кодовых слов равна n (определения линейных пространств, конечных групп и конечных полей приведены в приложении 1).

Линейные коды замкнуты относительно линейной комбинации, как это свойственно линейному подпространству. Любая линейная комбинация кодовых слов линейного кода, в том числе сумма и разность – также кодовое слово того же кода. То есть если $\mathbf{a}_i$ и $\mathbf{a}_j$ – различные кодовые слова линейного кода, то $\mathbf{a}_i - \mathbf{a}_j$ – также кодовое слово. Пусть $d(\mathbf{a}_i, \mathbf{a}_j) = \min = d$. Вычтем из обоих кодовых слов одно из них, например, $\mathbf{a}_i$. Тогда $\mathbf{a}_i - \mathbf{a}_i = \mathbf{0}$ и $\mathbf{a}_j - \mathbf{a}_i$ - также слова этого кода, и расстояние между ними есть $d(\mathbf{0}, \mathbf{a}_j - \mathbf{a}_i) = w(\mathbf{a}_i - \mathbf{a}_j)$. В силу произвольного выбора $\mathbf{a}_i$ и $\mathbf{a}_j$ это значит, что минимальное расстояние линейного кода равно минимальному весу ненулевых кодовых слов:

$$d = \min_i w(\mathbf{a}_i) = \min_i d(\mathbf{0}, \mathbf{a}_i). \qquad (66)$$

Далее, как любое векторное подпространство, линейный код задается его базисом, состоящим из *k* линейно независимых векторов над полем **GF(2).** Базисные векторы линейного кода будем обозначать $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k$. Образуем из этих векторов матрицу **G** размера $[k \times n]$ **:**

$$\mathbf{G} = \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \ldots. \\ \mathbf{g}_k \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ . & . & \cdots & . \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}. \qquad (67)$$

Эта матрица называется *порождающей матрицей* кода. Поскольку каждое кодовое слово является линейной комбинацией базисных векторов, то соотношение

$$\mathbf{a} = \mu_1 \mathbf{g}_1 + \mu_2 \mathbf{g}_2 + \ldots + \mu_k \mathbf{g}_k, \ \mu_1, \mu_2, \ldots, \mu_k \in \mathbf{GF(2)}.$$

можно рассматривать, как установление правила соответствия между сообщениями источника – векторами $\mathbf{m} = (\mu_1, \mu_2, \ldots, \mu_k)$ с коэффициентами из **GF(2)** и кодовыми словами $\mathbf{a}$ с коэффициентами также из **GF(2)**, то есть как правило кодирования. Вектор $\mathbf{m}$ представляет собой кодируемое сообщение, полученное кодером источника, а вектор $\mathbf{a}$ – кодовое слово на выходе кодера канала, предназначенное для последующей передачи по линии связи. Это соответствие удобно записывать, используя матричные обозначения и правила действий с матрицами:

$$\mathbf{a} = \mathbf{mG} = (\mu_1 \quad \mu_2 \quad \ldots \quad \mu_k) \cdot \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \ldots \\ \mathbf{g}_k \end{pmatrix} = (\mu_1 \quad \mu_2 \quad \ldots \quad \mu_k) \cdot \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ . & \ldots & . \\ g_{k1} & \cdots & g_{kn} \end{pmatrix}. \qquad (68)$$

Понятно, что нулевых строк в матрице **G** быть не должно.

Как мы выяснили в разделе 5.2, для придания коду способности к обнаружению и исправлению ошибок, в нем необходимо предусмотреть кроме k информационных символов определенное количество, а именно $r = n - k$ избыточных символов, обеспечивающих желаемое кодовое расстояние. Матрица **G** формируется так, чтобы это свойство выполнялось. С этой целью в матрице **G** должно быть два блока, один из которых является единичной матрицей размера $k \times k$, ответственной за передачу k информационных символов кода $\mathbf{m}$, а оставшиеся $r = n - k$ столбцов образуют второй блок размером $[k \times (n - k)]$, который порождает избыточные символы кода канала:

$$\mathbf{G} = \left(\mathbf{I}_\mathbf{k} \middle| \mathbf{G}_1\right).$$

В результате код, предназначенный для передачи по каналу, есть вектор, состоящий из двух частей: k-мерного вектора $\mathbf{m}$ и $(n - k)$-мерного вектора $\mathbf{mG}_1$

$$\mathbf{a} = \mathbf{m} \cdot \mathbf{G} = \mathbf{m}\left(\mathbf{I}_\mathbf{k} \middle| \mathbf{G}_1\right) = (\mathbf{mI}_\mathbf{k}, \mathbf{mG}_1) = (\mathbf{m}, \mathbf{mG}_1). \qquad (69)$$

Первая часть кодового слова $\mathbf{a}$ – это исходное сообщение, подлежащее передаче, вторая часть – проверочный вектор.

Приведенная форма порождающей матрицы является *левой канонической формой*. Если блоки матрицы **G** поменяются местами, получим *правую каноническую форму*. Кодирование, при котором k информационных символов кодируемого сообщения находятся на определенных k местах, называется *систематическим*.

Пусть $\mathbf{A}_k^n$ – линейный (n,k)-код над **GF(2)**, и $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_\mathbf{r}$ - базисные векторы ортогонального дополнения $\mathbf{A}_r^n$, $r = n - k$. Образуем из этих векторов матрицу **H** следующим образом

$$\mathbf{H} = \begin{pmatrix} \mathbf{h}_1 \\ \dots \\ \mathbf{h}_r \end{pmatrix} = \begin{pmatrix} h_{11} & \dots. & h_{1n} \\ . & \dots. & . \\ h_{r1} & \dots. & h_{rn} \end{pmatrix}.$$

Матрица **H,** по построению, обладает тем свойством, что для любого кодового слова $\mathbf{a} \in \mathbf{A}_k^n$

$$\mathbf{a}\mathbf{H}^{\mathrm{T}} = \mathbf{0}, \qquad \text{или} \qquad \mathbf{H}\mathbf{a}^{\mathrm{T}} = 0 \qquad (70)$$

Эта матрица называется проверочной матрицей для кода **A.** Найдем связь между порождающей и проверочной матрицами. Из (69) и (70) получим, что $\mathbf{m}\mathbf{G}\mathbf{H}^{\mathrm{T}} = \mathbf{0}$. Это значит, что

$$\mathbf{G}\mathbf{H}^{\mathrm{T}} = \mathbf{0} \qquad (71)$$

Если **G** имеет левую каноническую форму, то, используя (71), можно указать проверочную матрицу **H** в правой канонической форме: $\mathbf{H} = \left(\mathbf{H}_1 \middle| \mathbf{I}_\mathbf{r}\right)$:

$$\mathbf{G}\mathbf{H}^{\mathrm{T}} = \left(\mathbf{I}_\mathbf{k} \middle| \mathbf{G}_1\right)\begin{pmatrix} \mathbf{H}_1^{\mathrm{T}} \\ -- \\ \mathbf{I}_\mathbf{r} \end{pmatrix} = \mathbf{I}_\mathbf{k}\mathbf{H}_1^{\mathrm{T}} + \mathbf{G}_1\mathbf{I}_\mathbf{r} = \mathbf{H}_1^{\mathrm{T}} + \mathbf{G}_1 = \mathbf{0},$$

откуда сразу получаем

$$\mathbf{H}_1 = -\mathbf{G}_1^{\mathrm{T}}, \qquad \mathbf{G}_1 = -\mathbf{H}_1^{\mathrm{T}}.$$

В матрице **H** не должно быть нулевых столбцов.

Отметим важную теорему, которая дает возможность определить расстояние (*n, k*)-кода по свойству проверочной матрицы.

Теорема 10. Матрица **H** размера $r \times k$ над полем **GF(2)** является проверочной матрицей кода длины *n* с минимальным расстоянием *d тогда и только тогда*, когда любые *d* - 1 столбцов матрицы **H** линейно независимы, но найдутся *d* линейно зависимых столбцов.

Доказательство.

Пусть $\mathbf{a}_i = (a_{i1}, a_{i2}, \dots, a_{in})$ – кодовое слово линейного (*n,k*) - кода, обладающее минимальным весом, который равен *d*. Только на *d* позициях этого слова находятся единицы, на остальных – нули. Как видно из формулы (72) при умножении вектора $\mathbf{a}_i$ на матрицу **H** остаются только те

столбцы матрицы **H**, номера которых равны номерам ненулевых позиций кодового слова. Если в этом примере $\mathbf{a}_i = (0101010)$ – кодовое слово минимального веса, то минимальное расстояние кода равно $d = 3$, и сумма трех

$$\mathbf{H}\mathbf{a}_i^{\mathrm{T}} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} & h_{16} & h_{17} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} & h_{26} & h_{27} \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} & h_{36} & h_{37} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} h_{12} + h_{14} + h_{16} \\ h_{22} + h_{24} + h_{26} \\ h_{32} + h_{34} + h_{36} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad (72)$$

столбцов равна нулю. Это свидетельствует о линейной зависимости второго, четвертого и шестого столбцов матрицы **H**, выделенных ненулевыми компонентами кодового слова. Два линейно зависимых столбца в матрице **H** могут иметь место только в том случае, когда в каком либо кодовом слове кода единицы будут находиться только на двух позициях из n. Но это будет другой код с расстоянием 2. В общем случае, если расстояние кода равно d, и следовательно, в нем найдется кодовое слово с таким же минимальным весом, то в проверочной матрице **H** этого кода обязательно будет d линейно зависимых столбцов. Снижение веса кодового слова на единицу, то есть изъятие из результирующей суммы одного ненулевого столбца приведет к тому, что результат проверки не будет нулевым. А это свидетельствует о линейной независимости $d - 1$ столбцов. Что и требовалось доказать.

Существуют коды, у которых проверочная и порождающая матрицы меняются ролями. Такие коды называются *двойственными* или *дуальными кодами* по отношению к оригиналу.

5.5.2. Некоторые линейные коды в векторном представлении

П р и м е р 1. *Код с повторением*, $n = 5$, $k = 1$, то есть (5, 1)-код. Размерность кода источника равна 1, то есть длина кода источника равна 1.

Порождающая матрица этого кода состоит из матрицы $\mathbf{G}_1 = (1 \quad 1 \quad 1 \quad 1)$ и единичной матрицы, в данном случае $\mathbf{I}_k = 1$. Поэтому порождающая матрица кода состоит из одной строки: $\mathbf{G} = (1 \quad 1 \quad 1 \quad 1 \quad 1)$.

Кодер источника порождает кодовые слова длины k = 1: 0 и 1. Код канала состоит из двух слов, длина которых равна 5: $\mathbf{a}_1 = (00000)$, $\mathbf{a}_2 = (11111)$, то есть один символ повторяется пять раз. Проверочная матрица кода имеет вид:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = (\mathbf{H}_1 | \mathbf{I}_4) = (-\mathbf{G}_1^{\mathbf{T}} | \mathbf{I}_4).$$

Умножение на проверочную матрицу дает :

$$\mathbf{a}_1\mathbf{H}^{\mathbf{T}} = (00000) \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (0000), \quad \mathbf{a}_2\mathbf{H}^{\mathbf{T}} = (11111) \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (0000).$$

П р и м е р 2. *Коды с проверкой на четность,* (n, n - 1) - коды. Длина кодов n – произвольная. Количество информационных символов $k = n$ - 1. Минимальное расстояние $d = n - k + 1 = n - n + 1 + 1 = 2$. Размер проверочной матрицы этих кодов [1 $\times n$]. Поскольку проверка на четность осуществляется суммированием символов кодового слова, проверочная матрица **H** имеет вид

$$\mathbf{H} = (1 \quad 1 \quad .... \quad 1).$$

Порождающая матрица

$$\mathbf{G} = (\mathbf{I}_\mathbf{k} | \mathbf{G}_1) = \begin{pmatrix} 1 & 0 & .... & 0 & 1 \\ 0 & 1 & .... & 0 & 1 \\ . & . & .... & . & . \\ 0 & 0 & .... & 1 & 1 \end{pmatrix}, \quad \mathbf{G}_1 = -\mathbf{H}^{\mathbf{T}} = \begin{pmatrix} 1 \\ 1 \\ . \\ 1 \end{pmatrix}.$$

Последний столбец $\mathbf{G}_1$ матрицы **G** формирует проверочные символы. При четном весе кодового слова источника $\mathbf{m}$ на последней позиции кодо-

вого слова $\mathbf{a} = (\mathbf{m} \mid \mathbf{mG}_1)$ будет 0. Напротив, при нечетном весе кодового слова $\mathbf{m}$ на последней позиции кодового слова $\mathbf{a}$ будет 1. Таким образом, любое слово $\mathbf{m}$ будет кодироваться кодовым словом $\mathbf{a}$ с четным весом. Поэтому при условии правильной передачи произведение $\mathbf{aH}^{\mathrm{T}} = \alpha_1 + \alpha_2 + ... + \alpha_n = 0$, поскольку количество единиц в каждом неискаженном слове $\mathbf{a}$ четно.

П р и м е р 3. Линейные коды с минимальным расстоянием 3. В качестве примера приведем (7, 4) – код Хемминга (см. раздел 4.4 и таблицу 3).

Порождающая матрица кода есть

$$\mathbf{G} = \left(\mathbf{I_k} \middle| -\mathbf{H}_1^{\mathrm{T}}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Канонический вид его проверочной матрицы

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Пусть на выходе кодера источника появилось слово $\mathbf{m} = (0110)$. Тогда кодер канала выдаст слово

$$\mathbf{a} = \mathbf{mG} = (0110110).$$

Умножение этого слова на проверочную матрицу

$$\mathbf{aH}^{\mathrm{T}} = (000).$$

П р и м е р 4. Код из примера 3. В проверочной матрице переставлены столбцы, форма матрицы отличается от канонической:

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

В порождающей матрице соответствующие столбцы переставляются точно так же, и эта матрица теряет каноническую форму:

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

Сообщение $\mathbf{m} = (0110)$ кодируется словом: $\mathbf{a} = (0111100)$. Умножение этого слова на проверочную матрицу дает нулевой результат:

$$\mathbf{a}\mathbf{H}^{\mathrm{T}} = (000).$$

Поскольку длина кода и количество информационных символов остались без изменений, кодовое расстояние также не изменилось, выполненные перестановки столбцов матриц **H** и **G** не изменили код. Это все тот же (7, 4) – код Хемминга.

5.5.3. Схемы кодирования в канале

Как уже упоминалось, кодирование задается соотношением $\mathbf{a} = \mathbf{m}\mathbf{G}$, где **G** – порождающая матрица кода, $\mathbf{m}$ – кодовое слово источника, подлежащее кодированию для канала, $\vec{\mathbf{a}}$ – кодовое слово, подлежащее передаче. Будем считать, что задан (7, 4) – код Хемминга, с порождающей и проверочной матрицами

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}, \qquad \mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

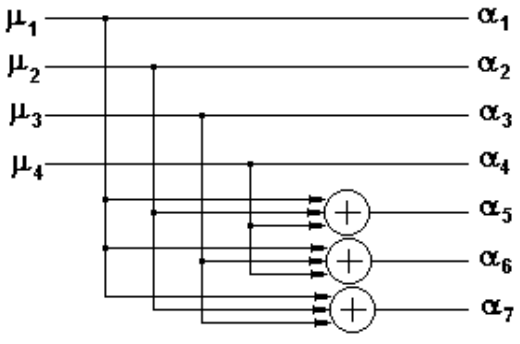

Рис. 12. Кодер (7, 4) - кода

Структурная схема кодера, выполняющего перемножение кода источника на порождающую матрицу, приведена на рис. 12. Суммирующий элемент – исключающее “или”, выполняет суммирование по модулю 2. Коды – параллельные.

При такой матрице G и приведенной схеме кодирова-

ния код канала **a** есть последовательность длиной 7:

$$\mathbf{a} = (\mu_1, \mu_2, \mu_3, \mu_4, (\mu_1 + \mu_2 + \mu_4), (\mu_1 + \mu_3 + \mu_4), (\mu_1 + \mu_2 + \mu_3)).$$

5.5.4. Синдромное декодирование линейных кодов, допускающих исправление однократных ошибок

Пусть **H** - проверочная матрица линейного (*n, k*) - кода. *Синдромом* двоичной последовательности $\mathbf{a} \in \mathbf{A}^n$ называется вектор $\mathbf{S_x} = \mathbf{aH}^\mathbf{T}$, количество составляющих которого равно количеству строк матрицы **H**, то есть количеству избыточных символов $r = n - k$.

Очевидно, для любого кодового слова $\mathbf{a}$ синдром $S_a = 0$. На этом свойстве основаны алгоритмы обнаружения ошибок. Если синдром полученного сообщения отличен от нуля, это значит, что при передаче произошли ошибки.

Пусть вектор ошибок есть $\mathbf{e} = (e_1, e_2, ..., e_n)$. Ошибке на позиции *j* соответствует 1 на этой же позиции. Количество ошибок равно весу двоичной последовательности, соответствующей вектору **e**. Если передавалось слово $\mathbf{a} = (a_1, a_2, ..., a_n)$ кода **A** с проверочной матрицей **H**, а на приемной стороне возникло сообщение $\mathbf{y} = \mathbf{a} + \mathbf{e}$, то синдром этого сообщения

$$\mathbf{S_y} = \mathbf{yH}^\mathbf{T} = (\mathbf{a} + \mathbf{e})\mathbf{H}^\mathbf{T} = \mathbf{aH}^\mathbf{T} + \mathbf{eH}^\mathbf{T} = \mathbf{eH}^\mathbf{T} = \mathbf{S}_e. \qquad (73)$$

Если при передаче кодового слова в некоторых его позициях с номерами *a, b, c* произошли ошибки, то синдром будет представлять собой сумму

$$\mathbf{S_y} = \mathbf{eH}^\mathbf{T} = \sum_{i=a,b,c} e_i \mathbf{H}_i, \qquad (74)$$

где $\mathbf{H}_i$ – *i* – ый столбец матрицы $\mathbf{H}^\mathbf{T}$.

Это означает, что при передаче по линии связи линейного кода синдром любого двоичного сообщения определяется лишь ошибкой передачи и не зависит от передаваемого кодового слова.

Синдромы различных векторов ошибок, которые могут быть исправлены, то есть такие, вес которых не превышает $t = (d - 1)/2$, различаются.

В самом деле, пусть $\mathbf{e}_1, \mathbf{e}_2$ – произвольные векторы исправляемых ошибок

$$\mathbf{e}_1 \neq \mathbf{e}_2, w(\mathbf{e}_1) \leq (d-1)/2, w(\mathbf{e}_2) \leq (d-1)/2.$$

Предположим, что $\mathbf{S}_{e1} = \mathbf{S}_{e2}$. Тогда, по определению, и в соответствии с равенством (74) $(\mathbf{e}_1 - \mathbf{e}_2)\mathbf{H}^{\mathrm{T}} = \mathbf{0}$, а это означает, что вектор $\mathbf{e}_1 - \mathbf{e}_2$ является кодовым словом. С другой стороны,

$$w(\mathbf{e}_1 - \mathbf{e}_2) = d(\mathbf{e}_1, \mathbf{e}_2) \leq d(0, \mathbf{e}_1) + d(0, \mathbf{e}_2) = w(\mathbf{e}_1) + w(\mathbf{e}_2) \leq d - 1,$$

что невозможно, если $\mathbf{e}_1 - \mathbf{e}_2$ – кодовое слово, потому что вес кодовых слов не должен быть меньше кодового расстояния, ибо ранее показано, что кодовое расстояние есть минимальный вес кодового слова.

Поскольку различные векторы ошибок веса, не превосходящего (d - 1)/2, имеют различные синдромы, то, зная синдром, можно однозначно определить вектор ошибок и исправить их.

Синдромное декодирование выполняется следующим образом.

Получив сообщение по линии связи, умножают его по mod2 на проверочную матрицу. Если результат равен 0, ошибок нет, или их такое количество, что они перевели одно кодовое слово в другое слово этого же кода. В противном случае получается синдром, по которому отыскивается вектор ошибок. Зная вектор ошибок, прибавляют этот вектор к принятому сообщению и тем самым получают переданное слово (если, правда, ошибок не настолько много, чтобы перевести сообщение в шар Хемминга с центром в другом кодовом слове).

Такое декодирование возможно тогда, когда заранее составлена таблица соответствия между векторами ошибок и их синдромами. Поиск, который нужно производить по этой таблице, занимает довольно много времени. Время поиска может быть уменьшено двумя способами.

П е р в ы й с п о с о б исправления одиночной ошибки проиллюстрируем на примере (7, 4) – кода Хемминга.

Переставим столбцы проверочной матрицы так, как это сделано в примере 4 пункта 5.5.2, и соответственно переставим столбцы в порождающей матрице. Заметим, что в порождающей матрице столбцы расположены в порядке возрастания числа, которое представляет каждый столбец

в двоичном представлении от 1 до 7. В этом случае синдром каждой одиночной ошибки численно равен номеру в двоичном представлении той позиции, в которой произошла ошибка. Это легко проверить непосредственной подстановкой. В этом случае синдромное декодирование заключается в вычислении синдрома и в изменении кодового символа полученного сообщения на той позиции, номер которой в двоичном представлении равен величине синдрома.

В т о р о й с п о с о б исправления одиночной ошибки заключается в том, что таблица, отображающая соответствие синдрома и ошибки, заводится в память компьютера так, что вектор ошибки записывается в ячейку памяти, двоичный номер которой равен синдрому. Тогда при декодировании сообщения по полученному синдрому из соответствующей ячейки памяти извлекается вектор ошибок и складывается с этим сообщением. Пример соответствия синдрома и ошибки для (7, 4) – кода Хемминга приведен ниже в табл. 6.

В заключение отметим, что синдромное декодирование есть декодирование по минимальному расстоянию Хемминга (ДМР).

Таблица 6

Синдромы однократных ошибок

Вектор ошибок	Синдром	Вектор ошибок	Синдром
$\mathbf{e}=(e_0e_1e_2e_3e_4e_5e_6)$	$S_0S_1S_2$	$\mathbf{e}=(e_0e_1e_2e_3e_4e_5e_6)$	$S_0S_1S_2$
0 0 0 0 0 0 0	0 0 0	0 0 0 1 0 0 0	1 1 0
1 0 0 0 0 0 0	1 1 1	0 0 0 0 1 0 0	1 0 0
0 1 0 0 0 0 0	1 0 1	0 0 0 0 0 1 0	0 1 0
0 0 1 0 0 0 0	0 1 1	0 0 0 0 0 0 1	0 0 1

Если кодовое расстояние линейного кода равно 5 и более, то согласно вышеизложенному подобный линейный код способен исправлять две и более ошибок в соответствии со своим кодовым расстоянием. Если при передаче кодового слова появляется более одной ошибки, то синдром ошибочного слова также не будет равен нулю, и это обстоятельство позволит обнаружить обе ошибки, но для их исправления необходимо знать позиции, в которых они произошли. Однако обнаружить позиции, в которых

произошли ошибки, весьма затруднительно, и актуальной задачей современной теории кодирования является разработка приемлемых способов решения этой проблемы.

Один из возможных способов нахождения позиций ошибок и их значений может состоять в решении системы уравнений (74), неизвестными в которой являются значения ошибок e_i и номера i – ых столбцов H_i матрицы **H**. В общем случае решение этой системы может быть найдено простым перебором столбцов матрицы **H** и значений ошибок. Такой перебор занимает конечное время, поскольку множество ошибок и множество столбцов матрицы $\mathbf{H}^{\mathbf{T}}$ ограничены.

6. КОДИРОВАНИЕ В КАНАЛЕ. ЦИКЛИЧЕСКИЕ КОДЫ

Циклические коды являются разновидностью полиномиальных кодов. При полиномиальном представлении каждому кодовому слову ставится в соответствие полином (многочлен), коэффициенты которого при степенях аргумента, начиная со старшей степени, суть кодовые символы. В соответствии с этим *полиномиальный* (*n, k*)-код есть множество всех многочленов степени $n - 1$ над полем **GF(2)**, делящихся на многочлен *g(x)* также над полем **GF(2)**, называемый *порождающим* или *генераторным многочленом*. Для описания циклических кодов используется специальный математический аппарат, а именно аппарат многочленов над конечными полями.

Примеры кодирования и декодирования циклических кодов, приведенные в настоящем разделе, относятся к случаям применения кодов, исправляющих однократные ошибки. Проблемы исправления двукратных ошибок и ошибок большей кратности существенно сложнее. Из двоичных циклических кодов, обладающих возможностями исправления более одной ошибки, наилучшими практически реализуемыми методами декодирования обеспечены коды Боуза-Чоудхури-Хоквингема (коды БЧХ). Основные принципы построения БЧХ-кодов изложены в разделе 7. Подробнее ознакомиться с этими кодами заинтересованный читатель сможет по литературным источникам [4, 7-10].

6.1. Многочлены над конечными полями

Многочленом (полиномом) над конечным полем Галуа **GF(2)** называется формальное выражение вида $a(x) = a_{n-1}x^{n-1} + ... + a_1 x + a_0$, где $a_0, a_1, ..., a_{n-1}$ - элементы поля **GF(2)**; и x - формальная переменная из поля **GF(2).** Число $n - 1 \geq 0$ называется *степенью многочлена* $a(x)$ и

часто обозначается deg a (x). Два многочлена равны, если равны коэффициенты при одинаковых степенях x. Многочлен называется *нормированным*, если коэффициент при старшей степени x равен единице, то есть $a_{n-1}=1$. Количество коэффициентов такого многочлена равно n, и если их считать компонентами вектора, то эти векторы в совокупности образуют линейное векторное пространство $\mathbf{A}^n$, размерность которого равна n.

Над многочленами определены два действия: сложение и умножение. При сложении и умножении полиномов их коэффициенты складываются и умножаются в соответствии с правилами, установленными для элементов поля **GF(2)**, то есть по mod 2.

Например, пусть $a_1(x)=x^7+x+1$ и $a_2(x)=x^4+x^2+x+1$ - многочлены над полем **GF(2)**. Тогда

$$a_1(x)+a_2(x)=x^7+x^4+x^2,$$

$$a_1(x)\cdot a_2(x)=x^{11}+x^9+x^8+x^7+x^5+x^4+x^3+1.$$

В качестве примера деления многочленов в поле **GF(2)** попытаемся разделить многочлен $f(x)$ на $g(x)$, с применением правил, установленных в этом поле.

П р и м е р 1. Деление многочлена $a_1(x)$ на $a_2(x)$:

$$\begin{array}{l|l} x^7+x+1 & x^4+x^2+x+1 \\ \underline{x^7+x^5+x^4+x^3} & \underline{x^3+x+1=h(x)} \\ x^5+x^4+x^3+x+1 & \\ \underline{x^5+x^3+x^2+x} & \\ x^4+x^2+1 & \\ \underline{x^4+x^2+x+1} & \\ x \quad \text{– остаток.} & \end{array}$$

В этих действиях, а также при делении многочленов в соответствии с правилами, действующими в поле **GF(2),** 1 + 1 =0, поэтому - 1 = 1 . Деление на нулевой многочлен не определено.

Из примера 1 следует, что для любых двух многочленов $f(x)$ и $g(x)$ справедливо соотношение (алгоритм Эвклида)

$$a(x) = h(x)\, g(x) + R(x), \qquad (75)$$

в котором $deg\ R(x) < deg\ g(x)$, многочлен $h(x)$ является *частным*, а $R(x)$ – *остатком* от деления $a(x)$ на $g(x)$. Многочлен $g(x)$ называется *делителем* многочлена $a(x)$, если существует многочлен $h(x)$, такой, что $a(x) = h(x) \cdot g(x)$**.** Представление многочлена $a(x)$ в виде (75) является единственным. Если $R(x) = 0$, это значит, что многочлен $a(x)$ делится на многочлен $g(x)$ без остатка.

При делении многочленов над полем **GF(2)** применяются правила сравнения и деления многочленов по модулю многочлена – делителя (см. также п. 3 Приложения 1). В общем случае два многочлена $a(x)$ и $h(x)$ *сравнимы по модулю* $g(x)$, если их разность нацело делится на $g(x)$ или, другими словами, каждый из многочленов имеет одинаковый остаток при делении на $g(x)$.

В частности, два многочлена $a(x)$ и $R(x)$ из (75) сравнимы по модулю многочлена $g(x)$. Это обстоятельство записывается в виде

$$R(x)\ \mathrm{mod} g(x) = a(x)\ \mathrm{mod} g(x).$$

Поскольку степень многочлена $R(x)$ меньше степени многочлена $g(x)$, то $R(x)\mathrm{mod}g(x) = R(x)$, и последнее равенство записывается, как $R(x) = a(x)\ \mathrm{mod} g(x)$.

Многочлен неотрицательной степени $R(x)$ в этом равенстве и в (75) называется *вычетом многочлена* $a(x)$ по модулю многочлена $g(x)$. Заметим, что

$$a(x)\ \mathrm{mod} g(x) = 0$$

тогда и только тогда, когда $a(x)$ делится нацело на многочлен $g(x)$, то есть, когда $a(x) = h(x) \cdot g(x)$.

П р и м е р 2. Для примера 1

$$x^7 + x + 1 \bmod \left(x^4 + x^2 + x + 1\right) = x.$$

$$x^7 + 1 \bmod \left(x^4 + x^2 + x + 1\right) = 0.$$

Легко проверить, что если $R_1(x) = a_1(x) \bmod g(x)$ и $R_2(x) = a_2(x) \bmod g(x)$, то

$$R_1(x) + R_2(x) = \left(a_1(x) + a_2(x)\right) \bmod g(x),$$

$$R_1(x) R_2(x) = \left(a_1(x) a_2(x)\right) \bmod g(x).$$

Многочлен $a(x)$ с коэффициентами из поля **GF(2)** называется *неприводимым* над полем **GF(2)**, если он не имеет других делителей с коэффициентами из **GF(2)**, кроме себя и единицы. Например, многочлены $x^3 + x^2 + 1$ и $x^3 + x + 1$ неприводимы. Многочлен $x^2 + 1$ приводим, поскольку $x^2 + 1 = (x + 1)(x + 1)$. Приводимым, как это можно показать, является также многочлен $x^7 + 1$, то есть

$$x^7 + 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1). \qquad (76)$$

Поскольку в поле **GF(2)** 1 = - 1, выполняется равенство $x^7 + 1 = x^7 - 1$. В общем случае для многочленов над полем **GF(2)** при любом целом n $x^n + 1 = x^n - 1$.

6.2. Описание циклических кодов

Линейный полиномиальный (n, k) – код над полем **GF(2)** называется *циклическим*, если циклический сдвиг каждого кодового слова является словом этого же кода. Из этого определения следует, что вместе с каждым словом $\mathbf{a}$ в циклический код входят слова, получающиеся из $\mathbf{a}$ циклическим сдвигом вправо или влево на любое число разрядов.

Каждому слову $\mathbf{a} = (a_{n-1}, a_{n-2}, ..., a_1, a_0)$, $a_i \in \mathbf{GF(2)}$ поставим в соответствие многочлен $a(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_1x + a_0$ и в дальнейшем не будем различать слово и соответствующий ему многочлен.

Рассмотрим слово $\mathbf{a}' = (a_{n-2}, ..., a_0, a_{n-1})$, которое получается из $\mathbf{a}$ циклическим сдвигом на один разряд влево. Соответствующий многочлен имеет вид:

$$a'(x) = a_{n-2}x^{n-1} + ... + a_0x + a_{n-1} =$$

$$= a_{n-1}x^n + a_{n-2}x^{n-1} + ... + a_0x - a_{n-1}(x^n - 1) = xa(x) - a_{n-1}(x^n - 1).$$

Отсюда следует, что $xa(x) = a'(x) + a_{n-1}(x^n - 1)$.

Степень многочлена $a'(x)$ не превышает n – 1. Это означает, что результатом деления многочлена *xa(x)* на $(x^n - 1)$ оказывается целая часть a_{n-1} и остаток (вычет) $a'(x)$. Таким образом, $a'(x) = xa(x) \bmod (x^n - 1)$. И

вообще, при циклическом сдвиге влево на i разрядов будет получаться многочлен $x^i a(x) \bmod \left(x^n - 1\right)$. Поэтому, если $a(x)$ – кодовое слово, то n многочленов $a(x) mod \left(x^n - 1\right)$, $xa(x) mod \left(x^n - 1\right)$, $x^2 a(x) \bmod \left(x^n - 1\right)$, ….., $x^{n-1} a(x) \bmod \left(x^n - 1\right)$ – также кодовые слова. Если этот ряд продолжить и записать $(n + 1)$ – й многочлен в той же форме: $x^n a(x) = \left(x^n - 1\right) a(x) + a(x)$, то окажется, что, как и следовало ожидать при циклическом сдвиге, $x^n a(x) \bmod \left(x^n - 1\right) = a(x) \bmod \left(x^n - 1\right)$. Это значит, что все последующие многочлены будут повторять уже полученный ряд, в связи с чем можно утверждать, что многочлены (вычеты) $x^i a(x) \bmod \left(x^n - 1\right)$ образуют конечную группу, количество элементов в которой равно n.

Поскольку сумма и разность, а также любая линейная комбинация слов линейных кодов также являются словом того же кода, то не только многочлены вида $x^i a(x) \bmod \left(x^n - 1\right)$, $i = 0,..., n-1$, но и любая линейная комбинация этих многочленов, то есть многочлен вида

$$\sum_{i=0}^{n-1} \mu_i x^i \, a(x) \bmod \left(x^n - 1\right) = m(x) a(x) \bmod \left(x^n - 1\right) \qquad (77)$$

является кодовым словом при произвольном наборе коэффициентов $\mu_0, \mu_1, ..., \mu_{n-1} \in \mathbf{GF(2)}$. Это означает, что векторы, соответствующие многочленам $x^i a(x) \bmod \left(x^n - 1\right)$, $i = 0,..., n-1$, представляют собой в совокупности базис векторного n-мерного пространства $\mathbf{A}^n$. Общее количество многочленов, представленных в виде (77), равно 2^n.

Заметим, что все элементы пространства $\mathbf{A}^n$ и соответствующие им многочлены суть кодовые слова циклического кода, не способного обнаруживать, а тем более, исправлять даже одиночные ошибки, поскольку расстояние между этими кодовыми словами равно 1, и любая одиночная ошибка переводит любой код в другой разрешенный код. Корректирующие коды должны иметь расстояние между словами не меньше трех, поэтому они должны принадлежать разреженному подпространству пространства $\mathbf{A}^n$. Такое разрежение можно получить с помощью *порождающего,* или *генераторного, многочлена*, то есть ненулевого нормирован-

ного многочлена, который обозначается $g(x)$. Для того, чтобы образовать линейное подпространство $\mathrm{A}_k^n \subset \mathrm{A}^n$, которое станет корректирующим циклическим (n, k)-кодом, необходимо, чтобы элементы этого подпространства могли быть представлены единственным образом в виде

$$a(x) = m(x)g(x) \bmod \left(x^n - 1\right) = m(x)g(x), \qquad (78)$$

где $g(x)$ – порождающий многочлен кода. Если степень полинома $a(x) \in \mathrm{A}^n$ равна $n - 1$, а количество информационных символов, то есть количество коэффициентов полинома $m(x)$ равно k, то степень порождающего многочлена должна быть

$$\deg g(x) = n - 1 - k + 1 = n - k = r.$$

При неизменном многочлене $g(x)$ все возможные элементы $m(x)g(x)$ не исчерпывают все пространство $\mathbf{A}^n$, но являются его частью, то есть подпространством $\mathbf{A}_{\mathrm{k}}^{\mathrm{n}}$. Степень многочлена $\deg g(x) = r$ обеспечивает в этом подпространстве необходимое расстояние кода, равное r и соответствующую корректирующую способность циклического (n, k)-кода. Этим кодом можно закодировать для последующей передачи до 2^k сообщений источника, где k – размерность подпространства $\mathbf{A}_{\mathrm{k}}^{\mathrm{n}}$, как это было при векторном представлении кодов, $k = n - r$. Мало того, представление кодовых слов по (78) обеспечивает выполнение проверки безошибочной передачи путем деления многочлена, соответствующего принятой последовательности, на порождающий многочлен. Поэтому и сам многочлен $g(x)$ является кодовым словом, и степень этого многочлена минимальна среди всех многочленов, представляющих кодовые слова.

Из (77) и из определения многочлена $g(x)$ следует, что $m(x)g(x) \bmod \left(x^n - 1\right) = \sum_{j=1}^{k} \mu_{j-1} x^{j-1} g(x) \bmod \left(x^n - 1\right) \in \mathbf{A}_{\mathrm{k}}^{\mathrm{n}}$ – кодовое слово циклического (n, k)-кода при любом коде источника $m(x)$, а вычеты $x^{k-1} g(x) \bmod \left(x^n - 1\right), x^{k-2} g(x) \bmod \left(x^n - 1\right), \ldots, x^0 g(x) \bmod \left(x^n - 1\right)$ в совокупности суть базис подпространства $\mathbf{A}_{\mathrm{k}}^{\mathrm{n}}$. Поскольку суммарная степень многочленов $m(x)$ и $g(x)$ не превосходит $n - 1$, то вычет $m(x)g(x) \bmod \left(x^n - 1\right)$ сов-

падает с произведением $m(x)g(x)$. Поэтому, если выполнено условие $\deg a(x) = \deg m(x)+\deg g(x) \le n - 1$, то $m(x)g(x)$ - кодовое слово циклического кода.

Многочлен $g(x)$ может быть порождающим многочленом циклического кода, если он удовлетворяет условию, утверждаемому следующей теоремой.

Теорема 11. Пусть $g(x)$ – порождающий многочлен циклического (n, k)-кода. Тогда $g(x)$ - делитель многочлена $x^n - 1$.

Доказательство.

Пусть $\deg g(x) = r$. Так как $g(x)$ - нормированный многочлен, то найдется такой многочлен $R(x)$, что

$$x^{n-r} g(x) = (x^n - 1) + R(x), \deg R(x) \le n - 1. \qquad (79)$$

В этом равенстве, в соответствии с алгоритмом Эвклида (75) многочлен $x^{n-r} g(x)$ – делимое, $(x^n - 1)$ – делитель, частное от деления есть 1, $R(x)$ – остаток. Поэтому $R(x) = x^{n-r} g(x) - (x^n - 1) = x^{n-r} g(x) \bmod (x^n - 1)$, то есть многочлен $R(x) = x^{n-r} g(x) \bmod (x^n - 1)$ – кодовое слово. Видно, что этот многочлен делится на $g(x)$, следовательно, в связи с (79) и $(x^n - 1)$ должен делиться на $g(x)$, что и требовалось доказать.

Таким образом, циклический (n, k)-код – это множество всех многочленов вида $m(x)g(x) \bmod (x^n - 1) = m(x)g(x)$ над полем **GF(2)**, где $g(x)$ – порождающий многочлен – делитель двучлена $(x^n - 1)$. Степень многочленов $m(x)g(x) \bmod (x^n - 1)$ не превышает $(n-1)$, и все они принадлежат конечному полю $\mathbf{GF}(2^n)$, но не исчерпывают его.

Делителей двучлена $(x^n - 1)$ может быть несколько. Среди них существует многочлен, который делит двучлен $(x^n - 1)$ и не делит никакой другой двучлен, степень которого ниже n. Такой многочлен – делитель, при котором это свойство имеет место, играет особую роль в теории циклических кодов.

Говорят, что многочлен $g(x)$ *принадлежит показателю n*, если

$$\left(x^n - 1\right) \bmod g(x) = 0,$$

и n является наименьшим целым положительным числом, для которого это равенство имеет место. Принадлежность многочлена $g(x)$ показателю n позволяет определить минимальную длину кодовых слов кода, имеющего расстояние d, равное степени полинома $g(x)$, и обеспечивающего исправление $t = (d-1)/2$ ошибок.

6.3. Векторное представление циклических кодов

В настоящем разделе будет показана взаимная связь между полиномиальным описанием циклических кодов и их векторным представлением.

В разд. 5.5.2 (n, k) – код был определен, как подпространство $\mathbf{A}_{\mathrm{k}}^{\mathrm{n}}$ линейного пространства $\mathbf{A}^{\mathrm{n}}$. С этим определением линейного кода гармонирует введенное в разд. 6.2 подпространство $\mathbf{A}_{\mathrm{k}}^{\mathrm{n}}$ вычетов по модулю $\left(x^n - 1\right)$ с базисом $x^{j-1} g(x) \bmod \left(x^n - 1\right)$, где $j = 1, 2, \ldots, k$.

В этой ситуации порождающая матрица циклического кода в полиномиальном выражении будет иметь вид:

$$\mathbf{G} = \begin{pmatrix} x^{k-1} g(x) \bmod \left(x^n - 1\right) \\ x^{k-2} g(x) \bmod \left(x^n - 1\right) \\ \ldots\ldots \\ x g(x) \bmod \left(x^n - 1\right) \\ g(x) \bmod \left(x^n - 1\right) \end{pmatrix}.$$

Как установлено в разд. 6.2, элементами эквивалентной порождающей матрицы при векторном представлении кодов будут коэффициенты соответствующих многочленов, сдвинутых по отношению друг к другу. Количество строк в этой матрице равно k. Понятно, что вид такой матрицы каноническим не является:

$$\mathbf{G} = \begin{pmatrix} g_r & g_{r-1} & \ldots & 0 & 0 & \ldots & 0 & 0 \\ 0 & g_r & \ldots & 0 & 0 & \ldots & 0 & 0 \\ . & . & \ldots & . & . & \ldots & . & . \\ 0 & 0 & \ldots & g_r & g_{r-1} & \ldots & g_1 & g_0 \end{pmatrix}.$$

Для всякого слова циклического (n, k) - кода справедливы равенства

$$a(x) = m(x)g(x), \quad a(x) \bmod g(x) = 0,$$

которые следуют из (78). Тогда, используя полиномиальную форму записи, получим выражение для проверочной матрицы:

$$\mathbf{H} = \left(x^{n-1} \bmod g(x), \quad \ldots \quad x \bmod g(x), \quad x^0 \bmod g(x)\right), \qquad (80)$$

где каждый j – й элемент строки есть столбец, составленный из коэффициентов полиномов – остатков $x^j \bmod g(x)$.

Сопоставляя многочлен $a(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0$ с вектором $\mathbf{a} = (a_{n-1}, a_{n-2}, \ldots, a_1, a_0)$ и принимая во внимание, что этот вектор и многочлен представляют собой кодовое слово циклического (n, k)-кода, мы вправе ожидать, что в соответствии с матричным описанием кодовых слов должно выполняться равенство $\mathbf{aH}^{\mathbf{T}} = \mathbf{0}$. Покажем, что это равенство, в самом деле, выполняется.

При транспонировании матрицы **H** ее столбцы становятся строками, а строки – столбцами. Поэтому произведение $\mathbf{aH}^{\mathbf{T}}$ вычисляется, как результат перемножения строки $\mathbf{a} = (a_{n-1}, a_{n-2}, \ldots, a_1, a_0)$ на матрицу **H**:

$$\mathbf{aH}^{\mathbf{T}} = (a_{n-1}\ a_{n-2} \ldots a_1\ a_0)\left(x^{n-1} \bmod g(x)\ \ x^{n-2} \bmod g(x) \ldots \ \ x \bmod g(x)\ \ x^0 \bmod g(x)\right)^{\mathbf{T}} =$$

$$= \left(\sum_{i=0}^{n-1} a_i x^i\right) \bmod g(x) = a(x) \bmod g(x) = 0.$$

Таким образом, матрица **H** - это проверочная матрица в обычном смысле.

Пусть n=7, $g(x) = x^3 + x^2 + 1$. Из теоремы 11 следует, что $g(x)$ – делитель многочлена $x^7 - 1$, следовательно, $g(x)$ − это порождающий многочлен циклического (7, 4)-кода. Коэффициенты многочлена $g(x)$ у степеней x, начиная со старшей степени, равны (1 1 0 1)**.** Строки порождающей матрицы этого кода получаются сдвигами коэффициентов порождающего многочлена:

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \qquad (81)$$

Столбцы проверочной матрицы вычисляем, как вычеты, которые в векторном представлении являются столбцами. Начнем с последнего столбца: $x^0 \bmod\left(x^3+x^2+1\right)=1\Rightarrow 001$. Следующий за ним столбец $x^1 \bmod\left(x^3+x^2+1\right)=x\Rightarrow 010$. Все остальные столбцы вычисляются следующим образом:

$$x^2 \bmod\left(x^3+x^2+1\right)=x^2\Rightarrow 100, \qquad x^3 \bmod\left(x^3+x^2+1\right)=x^2+1\Rightarrow 101.$$

Поскольку

$$x^4=x\left(x^3+x^2+1\right)-\left(x^3+x\right)=x\left(x^3+x^2+1\right)-\left(x^3+x^2+1\right)+\left(x^2+x+1\right),$$

$$x^4 \bmod\left(x^3+x^2+1\right)=\left(x^3+x\right)\bmod\left(x^3+x^2+1\right)=x^2+x+1\Rightarrow 111.$$

Из $x^5=\left(x^2+x+1\right)\left(x^3+x^2+1\right)+x+1$ следует

$$x^5 \bmod\left(x^3+x^2+1\right)=x+1\Rightarrow 011.$$

Из $x^6=\left(x^3+x^2+x\right)\left(x^3+x^2+1\right)+x^2+x$ следует

$$x^6 \bmod\left(x^3+x^2+1\right)=x^2+x\Rightarrow 110.$$

В результате вычислений мы получили столбцы проверочной матрицы (в порядке вычисления):

0 0 1, 0 1 0, 1 0 0, 1 0 1, 1 1 1, 0 1 1, 1 1 0.

Из этих векторов формируется уже знакомая нам проверочная матрица плотно упакованного (7, 4)-кода Хемминга:

$$\mathbf{H}=\begin{pmatrix}1&0&1&1&1&0&0\\1&1&1&0&0&1&0\\0&1&1&1&0&0&1\end{pmatrix}. \qquad (82)$$

Дальнейшие вычисления остатков $x^7 \bmod g(x)$, $x^8 \bmod g(x)$, $x^9 \bmod g(x)$ и последующих степеней x дают следующие результаты:

$$x^7 \bmod\left(x^3+x^2+1\right)=\left(x^7-1\right)\bmod\left(x^3+x^2+1\right)+x^0 \bmod\left(x^3+x^2+1\right).$$

В силу делимости $\left(x^7-1\right)$ на $\left(x^3+x^2+1\right)$, получим

$$x^7=x^0 \bmod\left(x^3+x^2+1\right)=x^0\Rightarrow 001.$$

Остатки $x^8 \bmod\left(x^3+x^2+1\right)=x^1 \bmod\left(x^3+x^2+1\right)=x^1\Rightarrow 010$,

$x^9 \bmod\left(x^3+x^2+1\right)=x^2 \bmod\left(x^3+x^2+1\right)=x^2\Rightarrow 100$ и так далее.

Это означает, что множество вычетов $x^i \bmod \left(x^3 + x^2 + 1\right)$ с добавлением к этому множеству нулевого элемента вместе образуют конечное поле многочленов степени не выше 2 с количеством этих элементов $2^3 = 8$. Это поле также является полем Галуа и обозначается **GF(8)** или $\mathbf{GF}\left(\mathbf{2^3}\right)$.

6.4. Выбор порождающего многочлена

Выбор того или иного многочлена в качестве порождающего определяет возможности кода длины *n* в отношении объема передаваемых сообщений и количества исправляемых ошибок. В самом деле, если *n* - длина кода, *r* - степень порождающего многочлена, то есть количество избыточных символов, то *k* = *n* - *r*. Верхняя, возможно, недостижимая граница количества исправляемых ошибок составляет Ent[(*r-1*)/2]. Наилучший циклический (*n, k*) – код, способный исправлять ошибки, это код, у которого доля избыточных символов по отношению к *n* окажется минимальной. Такой код порождается неприводимым многочленом, который является делителем многочлена $\left(x^n - 1\right)$ и принадлежит показателю *n*. В таблице 7 приведены разложения некоторых двучленов на неприводимые сомножители, каждый из них может быть использован в качестве порождающего многочлена циклического кода. В специальной литературе по кодированию, например, в [7-9] приводятся обширные таблицы разложения двучленов высокой степени на неприводимые сомножители.

Таблица 7

Примеры разложения некоторых биномов на неприводимые многочлены

Двучлен	Длина кода	Порождающие многочлены
$x^5 - 1$	5	$(x+1)\left(x^4 + x^3 + x^2 + x + 1\right)$
$x^7 - 1$	7	$(x+1)\left(x^3 + x^2 + 1\right)\left(x^3 + x + 1\right)$
$x^{15} - 1$	15	$(x+1)\left(x^2 + x + 1\right)\left(x^4 + x^3 + x^2 + x + 1\right)\left(x^4 + x^3 + 1\right)\left(x^4 + x + 1\right)$
$x^{17} - 1$	17	$(x+1)\left(x^8 + x^5 + x^4 + x^3 + 1\right)\left(x^8 + x^7 + x^6 + x^4 + x^2 + x + 1\right)$

$x^{23}-1$	23	$(x+1)\left(x^{11}+x^{10}+x^6+x^5+x^4+x^2+1\right)\times$ $\times\left(x^{11}+x^9+x^7+x^6+x^5+x+1\right)$

Таким образом, при выборе подходящего порождающего многочлена следует иметь в виду, что при n = const увеличение степени r порождающего многочлена приводит:

к уменьшению в коде количества информационных символов;

к уменьшению объема кода $K = 2^k$, то есть к уменьшению количества передаваемых сообщений;

к уменьшению скорости кода $R = k/n$;

к увеличению минимального расстояния d, которое определяется числом избыточных символов $r = n - k$, то есть к увеличению корректирующей способности кода и уменьшению вероятности ошибочного декодирования.

При желании сохранить объем кода K и увеличить корректирующую способность кода за счет увеличения степени порождающего многочлена придется выбирать неприводимый порождающий полином более высокой степени и тем самым увеличивать длину кодовых слов $n = k + r$.

В качестве порождающего многочлена (7, 4)-кода Хемминга (плотно упакованного кода) может быть применен (см. табл. 7) один из неприводимых многочленов, принадлежащих показателю 7 и потому являющихся делителями многочлена $\left(x^7-1\right)$.

П р и м е р. Требуется составить код минимальной длины, обеспечивающий исправление одной ошибки. Для этого необходимо обеспечить, чтобы степень порождающего многочлена, и, значит, количество избыточных символов было не меньше 3. Таким многочленом, в частности, является неприводимый многочлен $g(x)=x^3+x+1$. Определение минимальной длины кода сводится к определению степени n, которой принадлежит данный многочлен. Для этого необходимо делить двучлены вида x^n-1 на многочлен $g(x)$ при последовательном возрастании n до тех пор, пока в остатке не появится 0. Понятно, что начинать нужно с n = 3. Более

рациональный метод заключается в делении x^n на производящий полином до получения в остатке 1:

$$x^3/\left(x^3+x+1\right)=x+1,\qquad x^4/\left(x^3+x+1\right)=x^2+x,\qquad x^5/\left(x^3+x+1\right)=x,$$
$$x^6/\left(x^3+x+1\right)=x^2+x+1,\qquad x^7/\left(x^3+x+1\right)=1.$$

Этого результата следовало ожидать, поскольку многочлен $g(x)=x^3+x+1$ является неприводимым сомножителем двучлена x^7+1 и принадлежит показателю степени 7. Количество информационных символов в таком коде не превысит $k = 7 - 3 = 4$. Это означает, что объем кода или количество передаваемых сообщений не будет больше $K = 16$.

При необходимости увеличить длину кода *n,* способного передавать *K* сообщений источника, с условием исправления одной ошибки, может быть использован следующий итерационный способ.

Одна ошибка будет исправлена, если количество проверочных символов *r*, то есть степень порождающего многочлена $r = \deg(g(x))$ будет не меньше 3. Для $n > r + \log_2 K$ найдем разложение x^n+1 на неприводимые сомножители и из них в качестве порождающего многочлена возмем многочлен минимальной степени не меньше 3. Полученные значения *n* и *r* проверим на предмет выполнения неравенства $n > r + \log_2 K$. Если оно не выполняется, увеличим *n,*снова отыщем разложение на неприводимые множители бинома x^n+1 и повторим процедуру. Наилучший код получится при достижении минимального положительного значения разности $n-\left(r+\log_2 K\right)$. Тем самым будет минимизировано относительное количество неинформационных проверочных символов кода при сохранении корректирующей способности и объема передаваемых сообщений *K*.

Увеличение корректирующей способности циклических кодов выполняется либо за счет выбора неприводимого порождающего многочлена более высокой степени, либо за счет использования следующего указания Хемминга.

Если известен порождающий многочлен $g(x)$ кода длины *r*, позволяющего обнаруживать *t* ошибок, и $g(x)\bmod\left(x^n+1\right)\neq 0$ то порождающий многочлен кода, обнаруживающего $t + 1$ ошибку, может быть получен с

помощью порождающего многочлена $(x + 1)\, g(x)$, который неприводимым уже не является. За счет увеличения показателя степени r порождающего многочлена на 1 избыточность кода и, следовательно, его минимальное расстояние также увеличивается на 1. Если желательно сохранить длину кода, это приведет к соответствующему сокращению объема кода источника, поскольку с увеличением степени r при неизменном n уменьшается длина сообщения: $k = n - r$, $K = 2^k$.

Отметим, что минимальной корректирующей способностью, которая состоит в обнаружении одиночной ошибки, обладает код с проверкой на четность, который был представлен в примере 2 разд. 5.5.2. Этот код может быть получен при использовании такого порождающего многочлена $g(x) = x + 1$, который, являясь неприводимым, входит в разложение любого двучлена $x^n + 1$

В этом случае степень полинома $a(x) = g(x)\, m(x)$ превышает степень исходного кода $m(x)$ всего на единицу, а соответствующие кодовые слова будут иметь лишь один проверочный символ.

6.5. Кодирование в циклическом коде

Отображение кодового слова (многочлена) $m(x)$ в слово циклического кода производится путем умножения этого многочлена на порождающий полином $g(x)$. При выполнении кодирования в случае двоичных циклических кодов нет необходимости держать в памяти всю порождающую матрицу. Достаточно хранить лишь одну ее последнюю строку – порождающий многочлен. По сравнению с общими линейными кодами это несомненное удобство, но платой за это удобство являются ограничения, накладываемые на код его цикличностью.

Кодирование сводится к перемножению двух многочленов: информационного $m(x) = \mu_{k-1}x^{k-1} + \mu_{k-2}x^{k-2} + \ldots + \mu_1 x + \mu_0$

и порождающего $g(x) = g_r x^r + g_{r-1}x^{r-1} + \ldots + g_1 x + g_0, r = n - k$.

Это действие выполняется с применением r – разрядного регистра сдвига, сумматоров и перемножителей элементов поля **GF(2)**. Структурная схема двух вариантов схемы перемножения двоичных многочленов пока-

зана на рис. 13, где квадратами обозначены ячейки s_i регистра сдвига, принимающие состояния 0 или 1. Круги с соответствующими знаками арифметических действий – сумматоры и перемножители по mod 2. В зависимости от значения коэффициента g_i (0 или 1) перемножитель существует или нет.

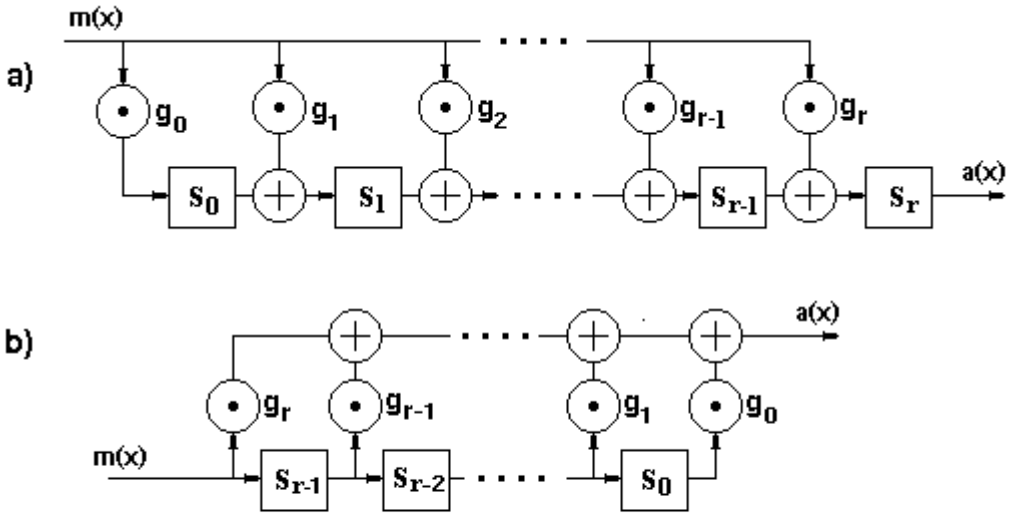

Рис. 13. Устройства умножения на многочлен g(x)

Для пояснения работы регистра в первом варианте (см. рис. 13, а) рассмотрим процесс перемножения многочленов $m(x)$ и $g(x)$ способом «в столбик» в следующем порядке: вначале умножим многочлен g(x) на старшее слагаемое многочлена $m(x)$, затем – на следующее слагаемое и так далее, вплоть до свободного члена $m(x)$.

$$g_r x^r + g_{r-1}x^{r-1} + g_{r-2}x^{r-2} + g_{r-3}x^{r-3} + \dots$$
$$\mu_{k-1}x^{k-1} + \mu_{k-2}x^{k-2} + \mu_{k-3}x^{k-3} + \mu_{k-4}x^{k-4} + \dots$$

$$\mu_{k-1}g_r x^{k+r-1} + \mu_{k-1}g_{r-1}x^{k+r-2} + \mu_{k-1}g_{r-2}x^{k+r-3} + \mu_{k-1}g_{r-3}x^{k+r-4} + \dots$$
$$\mu_{k-2}g_r x^{k+r-2} + \mu_{k-2}g_{r-1}x^{k+r-3} + \mu_{k-2}g_{r-2}x^{k+r-4} + \dots$$

$$\mu_{k-3} g_r x^{k+r-3} + \mu_{k-3} g_{r-1} x^{k+r-4} + \ldots$$

Конечный результат получается сложением частных результатов по столбцам, содержащим *x* в одинаковых степенях.

На вход устройства последовательно поступают кодовые символы, начиная со старших разрядов: $\mu_{k-1}, \mu_{k-2}, \ldots, \mu_0$. При поступлении первого символа μ_{k-1} в регистр заносятся коэффициенты многочлена $\mu_{k-1} g(x)$ – результата умножения этого символа на порождающий многочлен. Затем выполняется сдвиг, и первый символ $\alpha_{n-1} = \mu_{k-1} g_r$ кодового слова $a(x) = \alpha_{n-1} x^{n-1} + \ldots + \alpha_0$ считывается на выход кодера. При поступлении второго информационного символа μ_{k-2} к содержимому регистра после выполненного сдвига прибавляются коэффициенты многочлена $\mu_{k-2} g(x)$ – результата умножения поступившего символа на порождающий многочлен. В результате второй символ $\alpha_{n-1} = (\mu_{k-1} g_{r-1} + \mu_{k-2} g_r) \bmod 2$ кодового слова *a* (*x*) считывается на выход. Этот процесс повторяется в течение *n* тактов. Во время последних *n* – *k* тактов на вход регистра поступает *r* нулей.

Формально эта процедура записывается следующим образом. В начальном состоянии при *t* = 0 во всех ячейках регистра нули, $s_i = 0,\ i = 0,1,\ldots, r$. После появления на входе старшего коэффициента многочлена *m*(*x*) и умножения на него всех коэффициентов порождающего многочлена g(x) выполняется сдвиг, и далее при появлении на входе каждого следующего коэффициента этого многочлена состояния ячеек описываются соотношениями

$$s'_0 = \mu g_0, \qquad s'_1 = (s_0 + \mu g_1) \bmod 2,$$
$$s'_2 = (s_1 + \mu g_2) \bmod 2, \ \ldots, \ s'_r = (s_{r-1} + \mu g_r) \bmod 2, \quad (83)$$

где s_i – состояние *i*-й ячейки регистра сдвига в момент времени *t*, а s'_i – состояние той же ячейки в следующий момент времени *t* + 1 после сдвига и прихода очередного коэффициента μ многочлена *m*(*x*).

Последовательные состояния ячейки s_r, начиная с момента появления на входе старшего коэффициента (первого кодового символа), – коэффи-

циенты многочлена $a(x) = m(x)g(x)$, начиная с коэффициента при старшей степени x. После прихода последнего коэффициента μ_0 и следующих за ним r нулей регистр приобретает начальное состояние.

Схема второго варианта (см. рис. 13, б) работает несколько иначе. Снова поясним работу регистра сдвига перемножением многочленов $m(x)$ и $g(x)$ способом «в столбик», но при этом изменим последовательность действий на противоположную, а именно начнем умножение с перемножения всех слагаемых $m(x)$ вначале на старшее слагаемое $g(x)$, затем – на следующее слагаемое и так далее.

Конечный результат получается сложением частных результатов по столбцам, содержащим x в одинаковых степенях.

$$\begin{array}{llllllll}
\mu_{k-1}x^{k-1} & + & \mu_{k-2}x^{k-2} & + & \mu_{k-3}x^{k-3} & + & \mu_{k-4}x^{k-4} & +\ldots. \\
g_r x^r & + & g_{r-1}x^{r-1} & + & g_{r-2}x^{r-2} & + & g_{r-3}x^{r-3} & \ldots \\
\hline
g_r\mu_{k-1}x^{k+r-1} & + & g_r\mu_{k-2}x^{k+r-2} & + & g_r\mu_{k-3}x^{k+r-3} & + & g_r\mu_{k-4}x^{k+r-4} & \ldots\ldots \\
 & & g_{r-1}\mu_{k-1}x^{k+r-2} & + & g_{r-1}\mu_{k-2}x^{k+r-3} & + & g_{r-1}\mu_{k-3}x^{k+r-4} & \ldots. \\
 & & & & g_{r-2}\mu_{k-1}x^{k+r-3} & + & g_{r-3}\mu^{k+r-4} & \ldots.
\end{array}$$

В начальном состоянии при $t = 0$ во всех ячейках регистра нули, $s_i = 0, i = 0, 1, ..., r$ - 1. Старший коэффициент μ_{k-1} многочлена $m(x)$ умножается на g_r, и на выходе появляется первый символ циклического кода $\alpha_{n-1} = \mu_{k-1}g_r$, а ячейка s_{r-1} принимает состояние $\mu_{k-1} = 1$. Приход коэффициента μ_{k-2} вызывает сдвиг регистра вправо, передачу содержания ячейки s_{r-1} в ячейку s_{r-2} и в сумматор, и сумматоры складывают по модулю 2 результаты умножения: $\mu_{k-1}g_{r-1}$ и $\mu_{k-2}g_r$. Результаты суммируются, после чего на вход регистра и в ячейку s_{r-1} поступает коэффициент μ_{k-3}, регистр снова сдвигается, ячейка s_{r-2} принимает состояние μ_{k-2}, ячейка s_{r-3} принимает состояние $\mu_{k-1} = 1$ и передает это состояние в следующую ячейку. При этом сдвиге содержимое ячеек s_{r-1}, s_{r-2}, s_{r-3} пере-

дается в соответствующие сумматоры, и на выходе регистра появляется сумма $\alpha_{n-3} = \mu_{k-1} g_{r-2} + \mu_{r-2} g_{r-1} + \mu_{k-3} g_r$. Процедура повторяется вплоть до прихода последнего символа μ_0 входного кода. В заключение процедуры на вход поступает *r* нулей, и регистр приходит в начальное состояние.

Последующие состояния ячеек регистра сдвига в каждый момент, начиная от нулевого начального состояния, описываются через предыдущие состояния следующим образом:

$$s'_{r-1} = \mu, \quad s'_{r-2} = s_{r-1}, \quad s'_{r-3} = s_{r-2}, \ldots, s'_0 = s_1. \qquad (84)$$

П р и м е р. Пусть необходимо закодировать сообщения источника циклическим (7, 4)-кодом. В качестве порождающего многочлена применим один из неприводимых многочленов, на которые разлагается многочлен $x^7 - 1$, а именно, многочлен $g(x) = x^3 + x^2 + 1$ (см. табл. 7). Пусть на вход кодера канала поступает кодовое слово 1 0 0 1, которому соответствует многочлен $m(x) = x^3 + 1$. В этом случае мы должны получить многочлен $a(x) = x^6 + x^5 + x^2 + 1$, которому соответствует кодовое слово 1 1 0 0 1 0 1. Схема данного перемножителя построена, как частный случай второго варианта, представленного на рис. 13, б, и приведена на рис. 14.

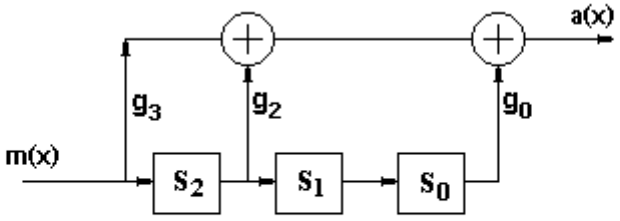

Рис. 14. Схема перемножения многочленов a(x) = m(x)g(x)

Схема работает следующим образом. Начальное состояние регистра 0 0 0. При поступлении первого символа слова *m*(*x*), то есть *1*, этот элемент тут же оказывается на выходе, и $a_6 = 1$. Ячейка s_2 приходит в состояние 1, и состояние ячеек регистра 1 0 0. Вторым символом слова *m*(*x*) приходит 0, выполняется сдвиг вправо, умножение, показанное на схеме, и суммирование результатов: $a_5 = 0 + 1 + 0 = 1$. Состояние регистра в результате

произошедшего сдвига 0 1 0. Эти состояния умножаются на коэффициенты полинома $g(x)$, затем суммируются с третьим символом входного кода, то есть с нулем, и в результате получаем на выходе следующий коэффициент многочлена циклического кода: $a_4 = 0 + 0 + 0 = 0$. После прихода очередного коэффициента информационного кода (многочлена) и очередного сдвига регистр переходит в состояние 0 0 1, выполняется перемножение на коэффициенты $g(x)$ и суммирование с последним коэффициентом многочлена $m(x)$, то есть с единицей. Получаем на выходе $a_3 = (1 + 0 + 1) \bmod 2 = 0$. Регистр переходит в состояние 1 0 0 .

Далее на вход регистра поступают нули, выполняются сдвиги и умножение, суммирование по модулю 2. В результате последовательно получаем: $a_2 = 0 + 1 + 0 = 1$, $a_1 = 0 + 0 + 0$, $a_0 = 0 + 0 + 1 = 1$. Таким образом на выходе этого перемножителя будет получено кодовое слово 1 1 0 0 1 0 1 , которому соответствует многочлен $a(x) = x^6 + x^5 + x^2 + 1$, что и следовало ожидать. Этот результат нетрудно проверить, выполнив перемножение многочленов вручную.

Описанное кодирование является несистематическим, то есть позиции, в которых стоят информационные и проверочные символы, распределены произвольно внутри каждого слова. Для восстановления переданного слова на приемной стороне необходимо выполнить обратное преобразование, то есть делить каждое полученное слово на порождающий многочлен.

6.6. Декодирование циклических кодов. Устройство деления многочленов

С целью получения сообщений источника на приемной стороне необходимо совершить операцию, обратную операции, выполненной при кодировании, а именно разделить полином, отображающий код канала, на порождающий многочлен и исправить возможные ошибки, возникшие из-за несовершенства канала связи.

Как и при умножении многочленов над полем **GF(2)**, деление многочленов выполняется на регистрах сдвига. Это оказывается возможным, поскольку в поле **GF(2)** вычитание эквивалентно сложению. Пояснение ра-

боты регистра сдвига, который делит многочлен *a(x)* на многочлен *g(x),* выполним на примере традиционного деления многочленов, предпринятого ранее в разделе 6.1. При этом имеем в виду, что оба многочлена нормированные.

$$\begin{array}{l|l} a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + a_{n-3}x^{n-3} + \ldots & g_r x^r + g_{r-1}x^{r-1} + g_{r-3}x^{r-2} + \ldots \\ g_n x^{n-1} + g_{r-1}x^{n-2} + g_{r-2}x^{n-3} + \ldots & x^{n-r-1} + x^{n-r-2} + \ldots\ldots \\ \hline \left(a_{n-2} + g_{r-1}\right)x^{n-2} + \left(a_{n-3} + g_{r-3}\right)x^{n-3} + \ldots & \end{array}$$

Приведенный пример показывает, что степень старшего слагаемого результата деления побирается так, чтобы $a_{n-1} - g_r = 0$ или, что то же самое, $a_{n-1} + g_r = 0$. Коэффициенты оставшегося многочлена представляют собой разности (суммы) $a_{n-2} + g_{r-1}$, $a_{n-3} + g_{r-3}$ и так далее. После деления этого остатка на $g(x)$ степень оставшегося многочлена снижается на единицу, а его коэффициенты суть подобные суммы по модулю 2. Деление продолжается до тех пор, пока степень остатка не окажется меньше r.

На рис. 15 приведена схема одного из вариантов регистра сдвига, выполняющего деление многочленов над полем **GF(2)**.

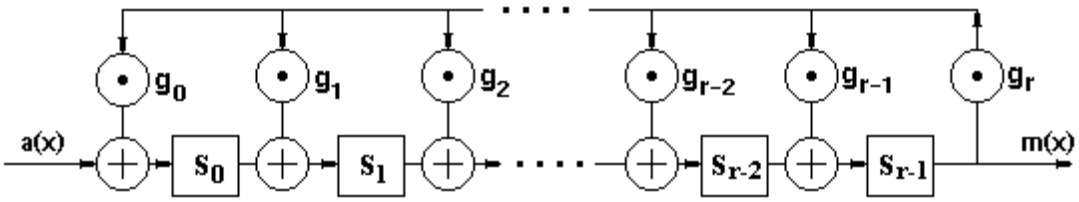

Рис. 15. Устройство деления на многочлен g(x)

В соответствии с приведенной схемой на выходе регистра втечение первых r сдвигов появляются r нулей. Первая единица в результате деления появляется только на месте $r + 1$, что соответствует степени $n - r - 1$ аргумента при старшем коэффициенте результата. Этот коэффициент равен 1. На него умножается многочлен *g(x),* то есть все его коэффициенты,

и результат этого перемножения вычитается (здесь – складывается) из делимого многочлена. Эта процедура была показана на примере деления "в столбик". Приход следующего коэффициента многочлена делимого вызывает сдвиг регистра, и если на выходе ячейки s_{r-1} снова оказывается единица, то многочлен *g(x)* умножается на нее и вычитается из получившегося первого остатка многочлена делимого, сдвинутого на один шаг вправо. Таким образом степень многочлена – частного, как и следовало ожидать, будет равна n – r – 1, а количество коэффициентов, то есть длина кодового слова $k = n - r$.

Принцип работы всех элементов регистра тот же, что и описанный выше в разделе 6.5. Здесь в качестве примера рассмотрим конкретный простой случай деления многочлена на многочлен.

П р и м е р. Многочлены для примера позаимствуем из раздела 6.5. Разделим многочлен $a(x) = x^6 + x^5 + x^2 + 1$ на многочлен $g(x) = x^3 + x^2 + 1$. Для этого нам понадобится регистр, схема которого представлена на рис. 16.

Многочлену $a(x)$ соответствует кодовое слово 1 1 0 0 1 0 1.

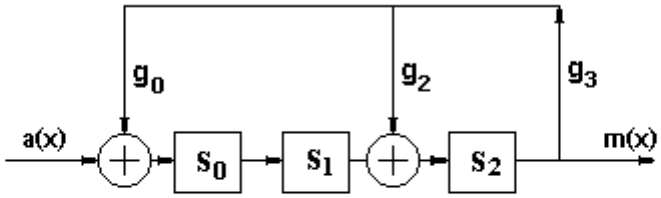

Рис. 16. Пример деления кодового слова на порождающий многочлен

Начальное состояние ячеек регистра нулевое, то есть $s_0 = s_1 = s_2 = 0$. Последовательные состояния регистра показаны в табл. 8.

Элементы многочлена – делимого $a(x)$ поступают на вход регистра, начиная с коэффициента α_{n-1} при старшем члене. В первый момент после начала изменяется только состояние ячейки s_0 с 0 на 1, и выполняется сдвиг на одну позицию вправо. Состояние ячейки s_2 не изменяется, и на выходе в результате сдвига появляется 0. В следующем такте на входе по-

является коэффициент α_{n-2} многочлена $a(x)$, и вновь выполняется сдвиг. После прихода на вход схемы следующего коэффициента – нуля на выходе также будет нуль, но ячейка s_2 наконец-то меняет свое состояние с 0 на 1. Приход коэффициента $\alpha_{n-4}=0$ возбуждает очередной сдвиг в регистре, ячейка s_2 вновь оказывается в состоянии 0, а на выходе регистра и в обратной связи появляется 1.

Таблица 8.

Кодовый символ кода $a(x)$	Состояние ячейки S_0	Состояние ячейки S_1	Состояние ячейки S_2	Кодовый символ кода $m(x)$
Начало	0	0	0	
1	1	0	0	0
1	1	1	0	0
0	0	1	1	0
0	1	0	0	1
1	1	1	0	0
0	0	1	1	0
1	0	0	0	1

Эта единица на входе складывается с коэффициентом $\alpha_{n-4}=0$, и ячейка s_0 снова приобретает состояние 1 (см. табл. 8). Приход следующего коэффициента вновь возбуждает сдвиг, и в ячейку s_0 записывается этот коэффициент, то есть 1. Первые n - k символов выходного кода суть нули. Результат деления есть кодовое слово 1 0 0 1, эквивалентный ему многочлен – $m(x)=x^3+1$. Состояние регистра на последнем шаге, как это будет показано в дальнейшем, – остаток от деления. В нашем случае последним состоянием регистра является 0 0 0, то есть деление выполнено без остатка, чего и следовало ожидать.

Описанный процесс может быть формализован через предыдущие s_i и последующие s_i' состояния ячеек регистра следующим образом.

$$s_2'=(s_1+s_2)\bmod 2, \qquad s_1'=s_0, \qquad s_0'=(s_2+\alpha)\bmod 2.$$

Эти уравнения в силу их линейности могут быть записаны в компактной матричной форме:

$$\mathbf{s}' = \mathbf{s} \cdot \mathbf{C} + \alpha_{n-i}\mathbf{b}\ , \qquad (85)$$

где вектор $\mathbf{s} = (s_2 \quad s_1 \quad s_0)$ и вектор $\mathbf{s}' = (s_2' \quad s_1' \quad s_0')$ – векторы, компоненты которых – предыдущие и последующие состояния ячеек регистра, α_{n-i} - n - i – й коэффициент полинома $a(x)$, поступающий на вход схемы в i - й момент времени, $i = 0, 1, ..., n$, вектор $\mathbf{b}$ в нашем случае равен $\mathbf{b} = (0\ 0\ 1)$, переходная матрица регистра имеет вид:

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \qquad (86)$$

Обозначим вектор $\mathbf{s}$ – $\mathbf{s}_i$, а вектор $\mathbf{s}'$ – $\mathbf{s}_{i+1}$. Тогда выражение (85) можно переписать в виде итерационной процедуры:

$$\mathbf{s}_{i+1} = \mathbf{s}_i \cdot \mathbf{C} + \alpha_{n-i} \cdot \mathbf{b},\ i = 0, 1, \ldots, n - 1. \qquad (87)$$

Пусть теперь в кодовом слове при передаче произошли ошибки, и на входе схемы рис. 16 появилось слово 1 1 0 0 1 1 0, которому соответствует многочлен $a(x) = x^6 + x^5 + x^2 + x$. Легко проверить непосредственным делением, что остаток от деления этого многочлена на производящий многочлен $g(x) = x^3 + x^2 + 1$ равен $x + 1$. При реализации этого деления с помощью схемы рис. 16 на последнем шаге состояние регистра будет:

$$s_2 = 0, \qquad s_1 = 1, \qquad s_0 = 1,$$

что соответствует многочлену $x + 1$.

6.7. Систематическое кодирование в циклическом коде

При систематическом кодировании информационные и избыточные символы располагаются на определенных позициях кодового слова, а именно, информационные символы – на первых k позициях, а избыточные символы – в конце слова. Перемещение информационных символов на первые позиции кодового слова выполняется путем умножения многочлена $m(x)$ на $x^{n-k} = x^r$. В результате степень многочлена $x^r m(x)$ будет равна

$n-1$, а формальная переменная в этом многочлене будет изменяться от $n-1$ до r.

Тогда кодовыми словами систематического циклического (n, k)-кода будут последовательности, выражаемые многочленами

$$a(x) = x^r m(x) + R(x), \qquad (88)$$

степень которых равна n - 1. Степень многочлена $R(x)$ не превышает r - 1. Для того, чтобы многочлены $a(x)$ из (88) были кодовыми словами, необходимо, чтобы они делились без остатка на порождающий многочлен $g(x)$, то есть

$$a(x)\mathrm{mod}g(x) = x^r m(x)\mathrm{mod}g(x) + R(x)\mathrm{mod}g(x) = 0,$$

откуда

$$R(x)\mathrm{mod}g(x) = -x^r m(x)\mathrm{mod}g(x). \qquad (89)$$

Поскольку при операциях в поле **GF(2)** различие между сложением и вычитанием не существует и поскольку степень многочлена $R(x)$ меньше степени многочлена $g(x)$, окончательно получим, что

$$R(x) = x^r m(x)\mathrm{mod}g(x). \qquad (90)$$

Это означает, что при систематическом кодировании на последних позициях кодового слова должны стоять проверочные символы, получаемые как коэффициенты многочленов $R(x)$ – остатков от деления многочлена $x^r m(x)$ на $g(x)$, то есть вычетов многочленов $x^r m(x)$ по модулю $g(x)$.

С учетом этого результата систематическое кодирование сообщения $m(x)$ заключается в следующем.

В старшие разряды слова $a(x)$ записывается сообщение $m(x)$. Затем в следующие $r = n - k$ разрядов записываются проверочные символы, которые получаются как остаток от деления многочлена $x^r m(x)$ на порождающий многочлен $g(x)$. В конечном итоге получаем кодовое слово в систематической форме $a(x) = (m(x), R(x))$.

В связи с изложенным с целью реализации систематического кодирования необходимо решить схемотехническую проблему деления многочленов вида $a(x) = b(x)\, m(x)$ на многочлен $g(x)$ и вычисления остатков от этого деления. Степени многочленов:

$$\deg a(x) = n - 1,\ \deg m(x) = k - 1,\ \deg b(x) = r = n - k.$$

Итак, для реализации систематического кодирования необходимо составить регистр сдвига, выполняющий умножение $m(x)$ на x^r, деление результата на $g(x)$ и вычисление остатка $R(x) = x^r m(x) \bmod g(x)$. Напомним, что степень многочлена $x^r m(x)$ равна n - 1.

Умножение многочлена $m(x)$ на x^r, последующее деление многочлена $x^r m(x)$ на $g(x)$ и вычисление остатка выполняется с помощью устройства, представленного на рис. 17. В этом устройстве умножение на x^r реализуется путем подачи кодового слова, эквивалентного многочлену $m(x)$, вперед на r – ю позицию.

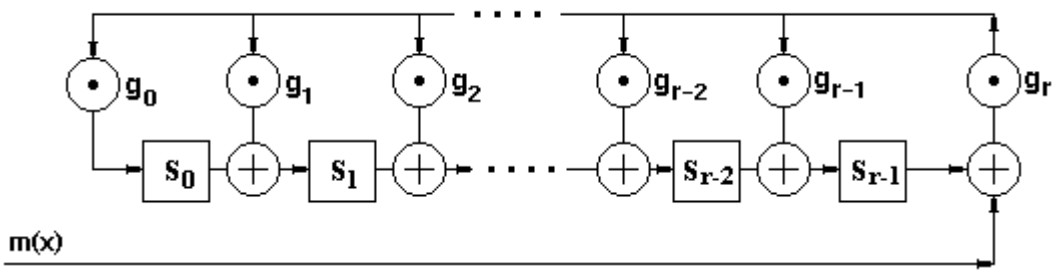

Рис. 17. Устройство деления многочлена $x^r m(x)$ на многочлен $g(x)$

Многочлен $m(x)$ степени $k - 1$ с k коэффициентами поступает на выход ячейки s_{r-1}. В результате выполнения каждого шага в ячейках $s_0, s_1, ..., s_{r-1}$ будет формироваться остаток от деления. По окончании последнего k – го шага в этих ячейках сформируется многочлен $R(x) = x^r m(x) \bmod g(x)$.

На рис. 18 приведены примеры подобного регистра, выполняющего деление многочлена $x^3(x^3 + 1)$ на многочлен $g(x) = x^3 + x^2 + 1$. Оба регистра сдвига эквивалентны, но второй из них удобнее для анализа.

Для схемы рис. 18 справедливы следующие соотношения между состояниями ячеек s_0', s_1', s_2', возникающими в момент прихода очередного символа кодового слова источника $m(x)$ и сдвига, и предыдущими состояниями тех же ячеек s_0, s_1, s_2:

$$s_0' = s_2 + \mu, \quad s_1' = s_0, \quad s_2' = s_1 + s_2 + \mu.$$

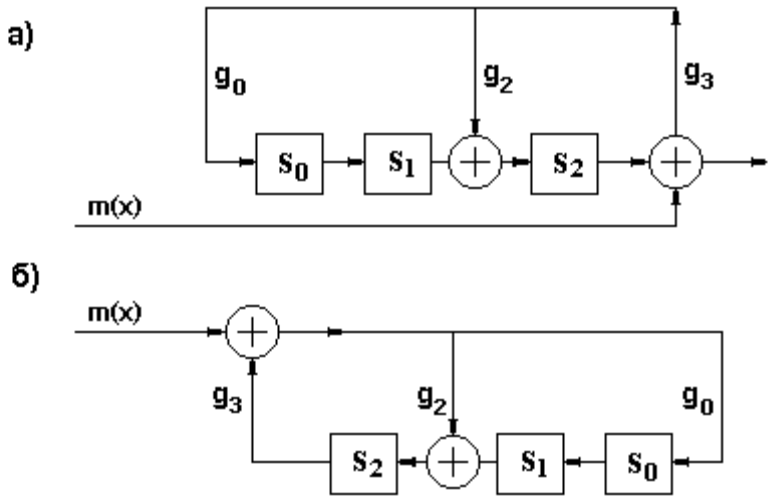

Рис. 18. Схемы вычисления остатков от деления многочлена $\mathbf{x}^3 \mathrm{m}(\mathbf{x})$ на многочлен $\mathrm{g}(\mathbf{x})$, $\mathrm{m}(\mathbf{x}) = \mathbf{x}^3 + 1$, $\mathrm{g}(\mathbf{x}) = \mathbf{x}^3 + \mathbf{x}^2 + 1$

Эти соотношения линейны, поэтому, как и в разд. 6.6, они могут быть выражены в матричном виде:

$$\mathbf{s}_{i+1} = \mathbf{s}_i \mathbf{C} + \mu_{k-i}\mathbf{b}, \quad i = 0,\ 1,\ 2,\ldots,\ n-1,$$

где μ_{k-i} - $(k - i)$ – й коэффициент многочлена $m(x)$, приходящий на вход схемы в момент времени i, вектор $\mathbf{b}$ = (1 0 1), $\mathbf{s}_i = (s_2\ s_1\ s_0)$, $\mathbf{s}_{i+1} = (s_2'\ s_2'\ s_0')$ - векторы предыдущих и последующих состояний системы, $\mathbf{C}$ – переходная матрица:

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Строки переходной матрицы суть коэффициенты деления многочленов x^{r-j+1} на порождающий многочлен $g(x)$:

$$\mathbf{C} = \begin{pmatrix} x^3 \bmod g(x) \\ x^2 \bmod g(x) \\ x \bmod g(x) \end{pmatrix}.$$

В самом деле, $x^3 \bmod g(x) = x^3 \bmod\left(x^3 + x^2 + 1\right) = x^2 + 1$, $x^2 \bmod g(x) = x^2 \bmod\left(x^3 + x^2 + 1\right) = x^2$, $x \bmod g(x) = x \bmod\left(x^3 + x^2 + 1\right) = x$.

Вектор **b** есть строка, состоящая из коэффициентов многочлена

$$b(x) = x^3 \bmod\left(x^3 + x^2 + 1\right) = x^2 + 1.$$

Этому многочлену соответствует вектор **b** = (1 0 1).

Последовательные состояния ячеек делителя, представленного на рис. 18, приведены в таблице 9.

Таблица 9

Кодовый символ кода $m(x)$	Состояние ячейки s_2	Состояние ячейки s_1	Состояние ячейки s_0
Начало	0	0	0
1	1	0	1
0	1	1	1
0	0	1	1
1	0	1	1

На четвертом шаге в ячейках оказалась комбинация 0 1 1, что соответствует многочлену $R(x) = x^3\left(x^3 + 1\right) \bmod\left(x^3 + x^2 + 1\right) = x + 1$. В этом можно убедиться непосредственным вычислением.

В конечном итоге можно предложить схему систематического кодера на регистре сдвига с обратными связями, представленную на рис. 19.

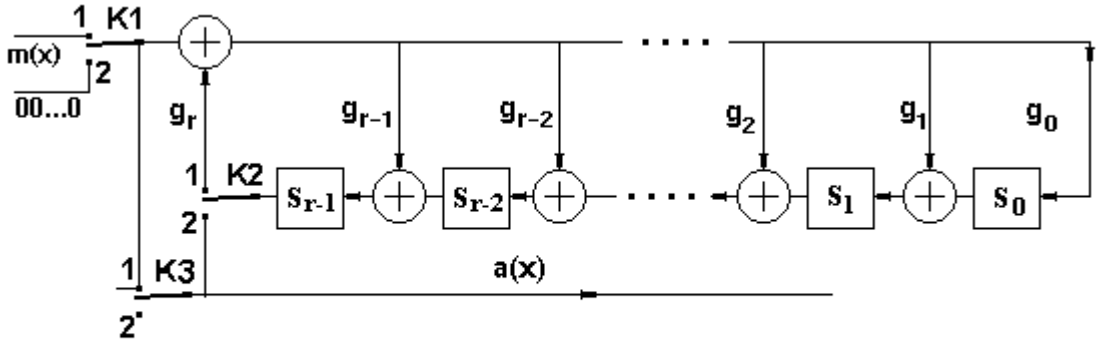

Рис.19. Систематический кодер для двоичного циклического кода

На вход кодера поступает кодовое слово (сообщение) источника $m(x)$, длина которого равна k . Во время первых k тактов ключи K1, K2 и K3

находятся в положениях 1. За это время первые k информационных символов кодового слова поступают на выход, заполняя первые k позиций выходного кодового слова канала. В это же самое время в регистре выполняется деление многочлена $x^r m(x)$ на порождающий многочлен $g(x)$, и в результате прихода k – го символа кода кодового слова $m(x)$ в ячейках $s_{r-1}, s_{r-2}, ..., s_0$ фиксируются коэффициенты полинома – остатка $R(x) = x^r m(x) \bmod g(x)$. Затем ключи К1, К2 и К3 переводятся в положение 2, на вход поступает r нулей, в регистре выполняются сдвиги, и на оставшиеся позиции кодового слова $a(x)$ из ячеек регистра считываются r символов остатка $R(x)$. В результате формируется кодовое слово $a(x) = (m(x), R(x))$, и на выходе кодера формируется кодовая последовательность длиной n, на первых k позициях которой находятся k информационных символов, вслед за ними идут $r = n - k$ проверочных.

Пример систематического декодирования в циклическом (7, 4)-коде приведен в разделе 6.9.

6.8. Синдромное декодирование систематического циклического кода

Поскольку циклические коды являются частным случаем линейных кодов, синдромное декодирование циклических кодов может быть выполнено с помощью приемов и схем, описанных выше в п. 5.5.4. Однако цикличность кодов позволяет существенно упростить схемы и процессы декодирования и вычисления синдромов. Для этого нужно допустить последовательный ввод данных в декодер и последовательный вывод результатов из декодера.

Пусть $a(x)$ – переданное слово циклического (n,k)-кода, порождающий многочлен которого $g(x)$. На приемной стороне получено слово $y(x) = y_{n-1}x^{n-1} + y_{n-2}x^{n-2} + ... + y_0$, которое может отличаться от переданного слова $a(x)$ вследствие ошибки $e(x)$, возникшей в линии связи: $y(x) = a(x) + e(x)$, где $e(x) = e_{n-1}x^{n-1} + e_{n-2}x^{n-2} + ... + e_0$.

Синдромом кодового слова $y(x)$ циклического кода называется многочлен $S(x)$, определяемый соотношением

$$S(x) = x^r y(x) \bmod g(x), \quad r = n - k. \qquad (91)$$

Это значит, что синдром слова $y(x)$ есть остаток от деления многочлена $x^r y(x)$ на порождающий многочлен кода. Синдром любой двоичной последовательности не зависит от передаваемого слова и определяется исключительно многочленом ошибок:

$$S(x) = x^r y(x) \bmod g(x) = \left(x^r a(x) + x^r e(x)\right) \bmod g(x) = x^r e(x) \bmod g(x),$$

поскольку $x^r a(x) \bmod g(x) = 0$.

Если минимальное расстояние кода равно d, то все многочлены ошибок кратности $(d - 1)/2$ и меньше имеют различные синдромы. Поэтому количество синдромов при стремлении исправить одну ошибку равно длине кодовых слов. Следовательно, как и ранее для линейных кодов вообще, зная синдром, можно однозначно восстановить переданное слово, если количество ошибок меньше $(d - 1)/2$.

Для вычисления синдрома применима схема сдвигового регистра, приведенная на рис. 17. В ячейках регистра на n – ом такте формируется вычет $x^r y(x) \bmod g(x) = S(x)$, то есть синдром принятого слова. Если этот синдром не равен нулю, то это означает, что передача произошла с ошибкой. В случае линейных кодов для исправления ошибок необходимо хранить таблицу соответствия синдромов и векторов (многочленов) ошибок. Но если коды циклические, для исправления ошибки эту таблицу можно сократить примерно в n раз, сохраняя в ней не все исправляемые комбинации ошибок, а только те, которые содержат ошибочный символ в старшем разряде. Для того чтобы обосновать это положение, заметим следующее.

Пусть ошибочное кодовое слово $y(x)$ имеет синдром $S(x)$, $y'(x)$ – многочлен, соответствующий циклическому сдвигу слова $y(x)$ в сторону старших разрядов на одну позицию, то есть, $y'(x) = xy(x) \, mod \, g(x)$. Тогда синдром $S'(x)$ слова $y'(x)$ также есть результат циклического сдвига многочлена *S(x)* в ту же сторону также на одну позицию:

$$S'(x) = x^r xy(x) \bmod g(x) = xS(x) \bmod g(x).$$

Другими словами, синдром циклически сдвинутого слова может быть получен в результате такого же циклического сдвига синдрома исходного

слова. Для двоичного кода, исправляющего однократные ошибки, достаточно заранее вычислить и использовать только один синдром кодового слова, ошибка передачи которого произошла в первом символе, то есть в коэффициенте при старшей степени *x* соответствующего многочлена. Вектор такой ошибки **e** = (1 0 0 0 0...), а соответствующий полином

$$e(x) = x^{n-1} + 0 \cdot x^{n-2} + 0 \cdot x^{n-3} + ... + 0 \cdot x + 0 \cdot 1. \qquad (92)$$

Синдром кодового слова, содержащего ошибку на j – й позиции, будет выражаться многочленом с единственным слагаемым

$$S(x) = x^r e(x) \bmod g(x) = x^r \left(0 \cdot x^{n-1} + ... + 0 \cdot x^{n-j+1} + x^{n-j} + 0 \cdot x^{n-j-1} + ... + 0\right) \bmod g(x) =$$

$$= x^r x^{n-j} \bmod g(x) = x^{r-j} x^n \bmod g(x).$$

В самом деле, по теореме 11 порождающий многочлен есть делитель двучлена $x^n - 1$ и принадлежит степени *n,* $\left(x^n - 1\right) \bmod g(x) = 0$. Тогда $x^n \bmod g(x) = 1$, а $x^{r-j} x^n \bmod g(x) = x^{r-j} = 0 \cdot x^{r-1} + ... + x^{r-j} + 0 \cdot x^{r-j-1} + ... + 0$. Это значит, что если при передаче кодового слова по линии связи возникла одиночная ошибка на j – й позиции этого слова, то в его синдроме символ 1 будет находиться точно на той же позиции, а на остальных позициях синдрома будут нули. Теперь для установления места ошибки и ее исправления достаточно выполнить столько сдвигов синдрома принятого слова, чтобы эта единственная 1 оказалась на одной позиции с позицией символа 1, заранее записанного в старший разряд логического устройства, которое вырабатывает 1 и посылает ее на обнаруженную позицию, благодаря чему приобретенная ошибка исправляется. На рис. 20 представлена схема декодера, в котором реализуется данный алгоритм.

В ячейки регистра логического устройства записывается многочлен $x^r x^{n-1} \bmod g(x) = x^{r-1} x^n \bmod g(x) = x^{r-1}$. Поскольку количество ячеек регистра логического устройства равно *r*, поэтому в старшем разряде логического устройства, который сравнивается с ячейкой s_{r-1}, оказывается и остается символ 1. В остальных разрядах – нули.

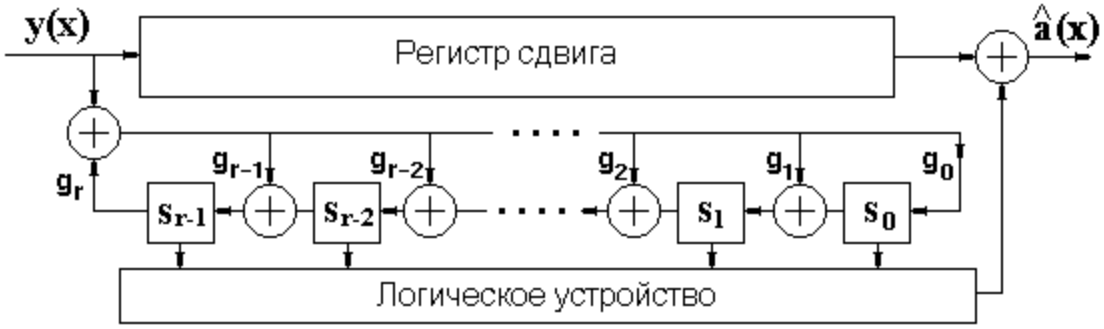

Рис. 20. Декодер циклического кода

В течение первых n тактов принятое кодовое слово запоминается в регистре сдвига, и одновременно вычисляется его синдром $S(x) = x^r\, y(x) \bmod g(x)$. В результате коэффициенты многочлена $S(x)$ оказываются в ячейках $s_{r-1}, s_{r-2}, ..., s_0$. Если при передаче кода по линии связи был искажен первый символ кода, то синдром $S(x)$ принятого кода совпадет с синдромом, записанным в логическое устройство. По этому признаку на выходе логического устройства вырабатывается 1, и за счет сдвига на выходе появляется первый символ принятого кода $y(x)$, который складывается с единицей и тем самым исправляется. Так выполняется первый такт исправления ошибки. Если упомянутого совпадения не происходит, на вход подаются нули, которые последовательно сдвигают на выход символы принятого кодового слова и инициируют дальнейшее вычисление синдромов ошибок, стоящих на следующих позициях. Логическое устройство, в котором заранее записан синдром ошибки в старшем разряде $e(x) = x^{n-1}$, на каждом такте сдвига сравнивает содержимое этих ячеек с синдромом. Так продолжается до тех пор, пока содержимое ячейки s_{r-1} вычислителя синдрома не совпадет с заранее записанным синдромом ошибки в старшем разряде $e(x) = x^{n-1}$. При совпадении на выходе логического устройства появляется 1 как раз в тот момент, когда на выходе регистра сдвига возникает ошибочный символ. При сложении с ним 1 ошибочный символ исправляется, и тем самым исправляется ошибочно переданное слово.

Если код способен к исправлению более, чем одной ошибки, алгоритм идентификации ошибок и их исправления существенно усложняется.

Ниже в следующем пункте приведены примеры кодирования и декодирования с исправлением одной ошибки кодовых слов (7, 4)-кода.

6.9. Пример систематического кодирования и декодирования в (7, 4) – коде

Рассмотрим систематическое кодирование для двоичного циклического (7, 4)-кода с порождающим многочленом $g(x) = x^3 + x^2 + 1$. Схема систематического кодера для этого случая приведена на рис. 21. В качестве сообщения источника примем, как и в разделе 6.5. кодовое слово $(\mu_3\mu_2\mu_1\mu_0)$ = 1 0 0 1, которому соответствует многочлен $m(x) = x^3 + 1$. В качестве вычислителя остатка $R(x) = x^r m(x) \bmod g(x)$ применяется делитель, представленный на рис. 18. В результате простого перемножения $m(x)$ и $g(x)$ получим многочлен $x^6 + x^5 + x^2 + 1$, которому соответствует несистематическое кодовое слово 1 1 0 0 1 0 1.

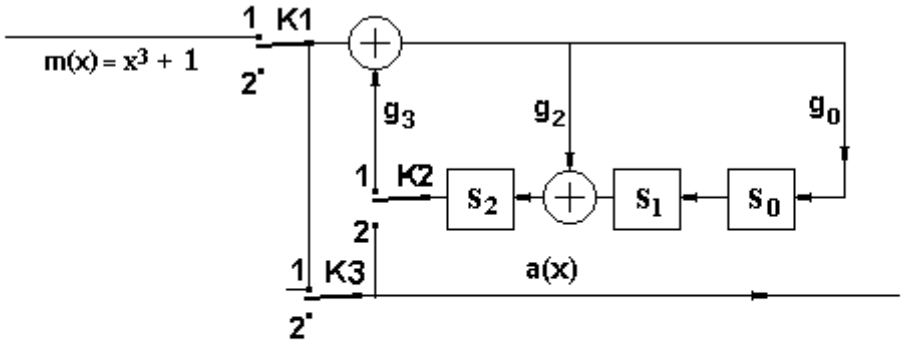

Рис. 21. Систематический кодер для циклического (7, 4) - кода, $g(x) = x^3 + x^2 + 1$

При систематическом кодировании на первом этапе ключи К1, К2 и К3 находятся в положении 1. На этом этапе выполняются k = 4 такта, и первыми символами выходного кодового слова, начиная со старших, будут 1 0 0 1, что соответствует многочлену $x^3 m(x) = x^6 + x^3$. В это же время

регистр с обратными связями (делитель) вычисляет остаток, соответствующий многочлену $x^2+1=x^3 \bmod g(x)$. Смежные состояния s'_j и s_j ячеек делителя приведены в п. 6.7. Там же в таблице 9 представлены последовательные состояния ячеек во время выполнения четырех шагов первого этапа. На четвертом такте в ячейках s_2, s_1, s_0 формируется остаток $x^3\left(x^3+1\right)\bmod g(x)$, который, как видно из таблицы 9, равен 0 1 1, что соответствует многочлену $x + 1$.

После завершения четвертого такта ключи К1, К2 и К3 переключаются в положение 2, и в результате трех сдвигов регистра, состоящего из ячеек s_1, s_2, s_3 к выходной комбинации 1 0 0 1 приписывается три символа остатка, то есть 0 1 1. В конечном итоге на выходе кодера формируется систематическое кодовое слово 1 0 0 1 0 1 1, которому соответствует многочлен $y(x)=x^6+x^3+x+1$. Как видно, это кодовое слово отличается от несистематического кодового слова 1 1 0 0 1 0 1, полученного ранее в результате непосредственного умножения многочлена $m(x)$ на порождающий многочлен. Отличие состоит в циклическом сдвиге на одну позицию влево.

Рассмотрим теперь процесс декодирования Для декодирования кодовых слов (7, 4)-кода с тем же самым порождающим многочленом $g(x)$ построим декодер в соответствии со схемой рис. 20. Схема декодера этого частного вида приведена на рис. 22. В регистр логического устройства записаны коэффициенты многочлена $x^r x^{n-1}\ mod\ g(x)$:

$$x^3 x^6 \bmod\left(x^3+x^2+1\right)=x^2 \bmod\left(x^3+x^2+1\right),$$

то есть слово 100, то есть синдром ошибки находится в старшем разряде логического устройства.

На вход данного декодера поступает принятая из линии связи последовательность $y(x)$, полученная в результате систематического кодирования кодового слова источника $m(x) = x^3+1$ и возможного влияния однократной ошибки. Предположим вначале, что ошибок нет, и в регистре

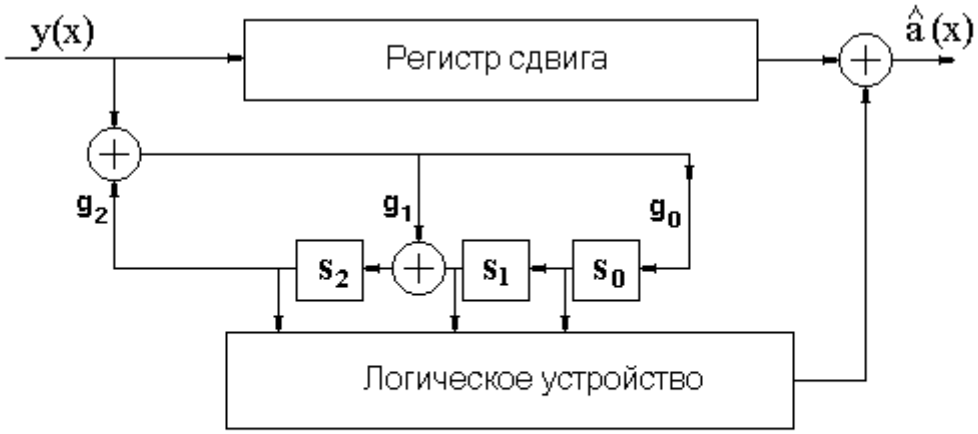

Рис. 22. Декодер двоичного циклического (7, 4) - кода с порождающим многочленом $g(x) = x^3 + x^2 + 1$

сдвига накапливается принятое кодовое слово 1 0 0 1 0 1 1. В этой схеме регистр, выполняющий деление, тот же, что и делитель систематического кодера, представленного на рис. 21.

Таблица 10

Кодовый символ $y(x)$	Состояние ячейки s_2	Состояние ячейки s_1	Состояние ячейки s_0
Начало	0	0	0
1	1	0	1
0	1	1	1
0	0	1	1
1	0	1	1
0	1	1	0
1	1	0	0
1	0	0	0

Состояния ячеек s_0, s_1, s_2 на всех тактах декодирования при поступлении на вход принятой последовательности представлены в табл. 10.

На k – ом такте деления в ячейках делителя формируется остаток от деления многочлена $x^3\left(x^6 + x^3 + x + 1\right)$ на порождающий многочлен $g(x)$ (см. разд. 6.7)

Поскольку ошибок в принятом слове нет, на последнем, седьмом такте остаток от деления на $g(x)$, который формируется в вычислителе синдрома, равен нулю. Поэтому из регистра сдвига считываются все символы принятого слова, к ним сумматор ничего не прибавляет, и на этом декодирование слова заканчивается.

При наличии ошибки, например, в третьем символе, когда вектор ошибки равен **e** = (0 0 1 0 0 0 0) и принятое кодовое слово есть 1011011, состояние ячеек будет меняться, как показано в табл. 11.

На седьмом такте вычисление синдрома завершается, и в ячейках s_2, s_1, s_0 образуется ненулевой синдром 0 0 1.

Таблица 11

Кодовый символ $y(x)$	Состояние ячейки s_2	Состояние ячейки s_1	Состояние ячейки s_0	Действие
Начало	0	0	0	
1	1	0	1	
0	1	1	1	
1	1	1	0	Вычисление
1	1	0	0	синдрома
0	1	0	1	
1	0	1	0	
1	0	0	1	
0	0	0	1	
0	0	1	0	Исправление
0	1	0	0	ошибки

После этого начинается процесс исправления ошибки. На первом такте этого процесса первый полученный синдром сравнивается с синдромом, записанным в старшем разряде логического устройства. Поскольку в логическом устройстве в соответствии с принципом декодирования, записан синдром слова с ошибкой в старшем разряде, то есть кодовое слово 1 0 0, на первом такте этого процесса синдром принятого слова ему не равен. Поэтому первый символ на выходе регистра сдвига не исправляется. После этого на вход вычислителя синдрома и регистра сдвига поступают ну-

ли, и синхронно с ними из регистра сдвига считываются последующие символы. Как только в ячейках s_2, s_1, s_0 в результате последовательных сдвигов возникает комбинация 1 0 0, логическая схема генерирует на выходе единицу, которая складывается по mod*2* с ошибочным символом и исправляет ошибку. В соответствии с табл. 11 это произойдет на третьем такте процесса исправления ошибок.

6.10. Кодирование и декодирование неравномерных кодовых слов источника

В разделе 3 была установлена высокая эффективность неравномерного кодирования источника информации. Однозначно кодируемыми и декодируемыми кодами являются префиксные коды, то есть коды, кодовые слова которых обладают следующим свойством: ни одно слово префиксного кода не должно являться началом другого слова этого же кода. В связи с этим кодирование и декодирование слов неравномерных префиксных кодов может быть осуществлено следующим образом.

Поскольку всякая закодированная информация источника представляет собой длинную последовательность кодовых символов (при двоичном кодировании – двоичных символов), то при неравномерном кодировании эта последовательность разбивается на блоки равной длины. Каждый такой блок может содержать начало и конец кодового слова источника в любом месте: в начале, в конце или где-то в середине. Длина k такого блока, количество избыточных символов r и длина кодового слова n выбираются в зависимости от модели канала, его пропускной способности, от вероятности искажения символов в канале и от требуемой вероятности исправления или обнаружения ошибок. Некоторые указания относительно выбора n, k и r были приведены в конце раздела 6.4. При этом полезно сопоставить полученный результат с границами Хемминга и Варшамова-Гилберта, полученными в разделе 5 для скорости и корректирующей способности кода.

Эти блоки длиной k символов кодируются в каком-либо циклическом коде, кодовые символы полученных кодовых слов длиной n модулируются и передаются по каналу связи, затем выполняется демодуляция принятых

кодовых слов. На выходе демодулятора мы получим непрерывную последовательность кодовых символов в циклическом коде, которая также должна быть разделена на блоки длиной n, которые декодируются одним из возможных методов.

На выходе декодера получим блоки информационных символов, длина каждого из них равна k. Эти блоки, идущие друг за другом непрерывно, без пропусков, образуют непрерывную последовательность. Если все ошибки, возникшие при передаче, исправлены, то эта последовательность в точности повторяет последовательность, полученную на выходе кодера источника. Теперь для правильного декодирования также необходимо знать только момент начала передачи. В силу того что код источника префиксный и начало каждого кодового слова не является началом никакого другого, декодер различает отдельные слова, декодирует их, и на выходе декодера абонент получает сообщение источника.

7. ЦИКЛИЧЕСКИЕ КОДЫ, ЗАДАВАЕМЫЕ КОРНЯМИ МНОГОЧЛЕНОВ. КОДЫ БОУЗА-ЧОУДХУРИ-ХОКВИНГЕМА

7.1. Дополнительные свойства многочленов и их корней

Из раздела 6 известно, что неразложимый многочлен *g(x)* является порождающим многочленом кода длины *n*, если он делит двучлен $x^n - 1$ без остатка, то есть если $\left(x^n - 1\right) \bmod g(x) = 0$. Для этого все корни многочлена *g(x)* должны быть корнями двучлена $x^n - 1$. И вообще, корни всех неразложимых многочленов, произведение которых есть двучлен $x^n - 1$, также должны быть корнями этого двучлена. Общее количество его корней должно быть равно степени этого двучлена. С другой стороны, поскольку любой многочлен *a(x)*, представляющий собой кодовое слово, должен делиться на порождающий многочлен *g(x)*, то все корни порождающего многочлена должны быть корнями многочлена *a(x)*. Но порождающий многочлен над полем **GF(2)** неразложим на сомножители, а это должно означать, что в поле **GF(2)** у порождающего многочлена корней нет. С подобным обстоятельством мы сталкивались при решении, например, квадратных уравнений с вещественными коэффициентами, у которых в вещественной области корней не было. В этих случаях приходилось расширять множество вещественных чисел до множества комплексных чисел и находить корни в этом расширенном множестве. В каком же множестве лежат все корни двучлена $x^n - 1$? Оказывается, что корнями этого двучлена являются элементы $\alpha_i = x^i \bmod g(x)$. Проверим это, подставляя корни α_i в выражение $x^n - 1 = 0$.

Прямой подстановкой убеждаемся в том, что $\alpha_0 = x^0 \bmod g(x) = 1$ есть корень двучлена $x^n - 1$. Второй корень $\alpha_1 = x^1 \bmod g(x) = x$, также является корнем этого двучлена. В самом деле, выполним подстановку: $(\alpha_1^n - 1) \bmod g(x) = (x^n - 1) \bmod g(x) = 0$. Проверим, является ли корнем этого двучлена $\alpha_2 = x^2 \bmod g(x)$. Для этого перепишем выражение $(x^n - 1) \bmod g(x)$ в эквивалентном виде $x^n \bmod g(x) = 1$ и подставим в него корень $\alpha_2 = x^2$. Получим $x^{2n} \bmod g(x) = x^n x^n \bmod g(x)$. Но мы знаем, что результат умножения любого многочлена $a(x) \bmod g(x)$ на x^n есть n – кратный циклический сдвиг его коэффициентов, который приводит к исходному многочлену: $x^n a(x) \bmod g(x) = a(x) \bmod g(x)$. Это значит, что $x^{2n} \bmod g(x) = x^n x^n \bmod g(x) = x^n \bmod g(x) = 1$, а потому $\alpha_2 = x^2$ - корень двучлена $(x^n - 1) \bmod g(x)$.

Повторяя эти рассуждения для $i = 3, 4, \ldots$, убедимся, что $\alpha_i = x^i \bmod g(x)$ – также корни двучлена $(x^n - 1) \bmod g(x)$..

В разд. 6.3 для частного примера показано, что элементы $x^i \bmod g(x)$ – корни двучлена $x^n - 1$, образуют полную систему вычетов по модулю $g(x)$. Если степень многочлена $g(x)$ равна r, то степень многочленов – остатков, то есть вычетов по модулю $g(x)$, не превышает $r - 1$, а количество коэффициентов этих многочленов не превышает r. Стало быть, общее количество ненулевых многочленов – вычетов равно $2^r - 1$. По коммутативному свойству остатков, приведенному в разд. 6.1, в этом множестве вычетов действует правило умножения элементов :

$$\alpha_i \cdot \alpha_j = x^i \bmod g(x) \cdot x^j \bmod g(x) = x^{i+j} \bmod g(x) = \alpha_{i+j}.$$

Из этого правила следует коммутативность элементов множества вычетов.

Для того, чтобы завершить операцию расширения поля **GF(2)**, дополним расширенное поле недостающим ему нулем. Это будет означать, что

расширенное таким образом поле $\mathbf{GF}\left(\mathbf{2}^{\mathbf{r}}\right)$ представляет собой *конечную совокупность корней двучлена* $x^{n+1} - x = x\left(x^{n} - 1\right)$.

Напомним, что элемент поля, такой, что все остальные элементы поля являются степенями этого элемента, называется *порождающим* или *примитивным элементом* поля. В нашем случае примитивным элементом является $x^{1} \; mod \; g(x) = x$. Количество ненулевых элементов поля есть порядок поля. Мультипликативным порядком примитивного элемента *x* поля называется минимальное число *n*, при котором $x^{n} = 1$. Здесь мультипликативный порядок примитивного элемента равен порядку поля.

Заметим здесь, что в векторном представлении все элементы расширенного поля $\mathbf{GF}\left(\mathbf{2}^{\mathbf{3}}\right)$ суть не что иное, как столбцы проверочной матрицы (7, 4) – кода, который мы рассматривали в примере 3 разд. 5.5.2 и в разд. 6.3.

Многочлен над полем **GF(2)** называется *примитивным многочленом*, если примитивный элемент является его корнем. Не все неприводимые многочлены примитивны. Существуют неприводимые многочлены, не являющиеся примитивными. Таким многочленом является, например, неприводимый многочлен $x^4 + x^3 + x^2 + x + 1$ из разложения двучлена $x^{15} - 1$, приведенного в табл. 7 разд. 6.3. Нахождение примитивного элемента поля $\mathbf{GF(2^r)}$ может оказаться нетривиальной задачей.

Для иллюстрации вышеизложенного снова воспользуемся результатами, приведенными в разд. 6.3. Покажем, что многочлен $g(x) = x^3 + x^2 + 1$ является примитивным, поскольку примитивный элемент поля $\mathbf{GF(2^3)}$ $\alpha = x \bmod g(x) = x$ есть корень этого многочлена. Как следует из материалов разд. 6.3,

$$\alpha^0 = 1 \bmod g(x) = 1, \;\; \alpha^2 = x^2 \bmod g(x) = x^2, \;\; \alpha^3 = x^3 \bmod g(x) = x^2 + 1,$$

поэтому $\alpha^3 + \alpha^2 + 1 = x^2 + 1 + x^2 + 1 = 0$.

Приведем без доказательств, но с иллюстрациями некоторые свойства многочленов и их корней, которые будут полезны при построении кодов по корням порождающих многочленов. При этом будем использовать как полиномиальное, так и векторное представление корней.

С в о й с т в о 1. $f^2(x) \bmod 2 = f\left(x^2\right) \bmod 2$, где $f(x)$ - произвольный многочлен над **GF(2)**.

В самом деле, пусть многочлен $f(x) = x^2 + 1$. Тогда

$$f^2(x) = \left(x^2 + 1\right)^2 \bmod 2 \ = (x^4 + x^2 + x^2 + 1) \bmod 2 = x^4 + 1.$$

Это частный вид соотношения Шенемана: $f^q(x) = f\left(x^q\right) \bmod q$, где $f(x)$ – произвольный многочлен над полем **GF(q)**.

С в о й с т в о 2. Если g*(x)* - неприводимый многочлен степени *r* над полем **GF(2)**, $\mathbf{GF}\left(2^{\mathbf{r}}\right)$ - конечное поле вычетов, построенное по mod(*g*(*x*)), то все корни многочлена *g*(*x*) лежат в поле $\mathbf{GF}\left(2^{\mathbf{r}}\right)$, и если $\alpha \in \mathbf{GF}\left(2^{\mathbf{r}}\right)$ - корень *g(x)*, то $\alpha^2, \alpha^4, \alpha^8$ – также корни этого многочлена. По свойству 1 все корни неприводимого многочлена g*(x)* могут быть найдены только по одному из его корней посредством последовательного возведения одного из корней в степень 2.

Выше в данном разделе показано, что элемент $\alpha = x$, то есть α = 010 есть корень многочлена $g(x) = x^3 + x^2 + 1$.

Покажем, что α^2 и α^4 – также корни этого же многочлена. Применим для этого векторное представление корней.

Из разд. 6.3 следует, что по модулю многочлена $g(x) = x^3 + x^2 + 1$:

$\alpha^2 = x^2 \Rightarrow 100$, $\alpha^3 = x^2 + 1 \Rightarrow 101$, $\alpha^4 = x^2 + x + 1 \Rightarrow 111$, $\alpha^6 = x^2 + x \Rightarrow 110$,

$\alpha^7 = \alpha^0 = 1 \Rightarrow 001$, $\alpha^8 = \alpha^1 \alpha^7 = \alpha^1 = x \Rightarrow 010$, $\alpha^{12} = \alpha^5 \alpha^7 = \alpha^5 = x + 1 \Rightarrow 011$.

Проверим корень α^2:

$\alpha^6 + \alpha^4 + 1 = 110 + 111 + 001 = 000$.

Проверим корень α^4:

$\alpha^{12} + \alpha^8 + 1 = 011 + 010 + 001 = 000$.

С другой стороны, поскольку α α^2, α^4 - корни многочлена g*(x)*, то по теореме Безу имеет место разложение его на множители:

$$g(x) = (x - \alpha)\left(x - \alpha^2\right)\left(x - \alpha^4\right) = x^3 - x^2\left(\alpha + \alpha^2 + \alpha^4\right) + x\left(\alpha^3 + \alpha^5 + \alpha^6\right) - \alpha^7 =$$

$= x^3 + x^2(010 + 100 + 111) + x(101 + 011 + 110) + 010 = x^3 + x^2 + x \cdot 0 + 1 = x^3 + x^2 + 1.$

Воспользуемся выражением (76) и табл. 7, в которых представлено разложение $x^7 + 1 = (x+1)(x^3 + x + 1)(x^3 + x^2 + 1)$. При иллюстрации свойства 2 было построено расширенное поле $\mathbf{GF(8)}$ (r = 3) с ненулевыми векторными элементами

$\alpha^0 = 001,\ \alpha^1 = 010,\ \alpha^2 = 100,\ \alpha^3 = 101,\ \alpha^4 = 111,\ \alpha^5 = 011,\ \alpha^6 = 110.$

В соответствии со свойствами конечного поля и мультипликативной группы все эти элементы – последовательные степени примитивного элемента α, по построению. Элементы $\alpha, \alpha^2, \alpha^4$ - корни многочлена $(x^3 + x^2 + 1)$. Элемент $\alpha^0 = 001$ - корень многочлена (x + 1). Непосредственной подстановкой читатель может убедиться в том, что элементы $\alpha^3,\ \alpha^6, \alpha^{12} = \alpha^5$ - корни многочлена $(x^3 + x + 1)$.

Мы видим, что степени элемента α распадаются на два непересекающиеся множества, которые называются *циклотомическими классами*.

Определение циклотомического класса.

Пусть α – корень многочлена $f(x)$ над полем **GF(2)**. Тогда циклотомическим классом будет множество элементов $\alpha,\ \alpha^2,\ \alpha^{2^2},\ \alpha^{2^3}, \ldots$, которые также являются корнями многочлена $f(x)$. Элемент α – *порождающий элемент* этого класса.

В приведенном примере в первом случае порождающий элемент циклотомического класса – корень α, во втором случае циклотомический класс порожден корнем α^3. В самом деле, $(\alpha^3)^2 = \alpha^6, (\alpha^3)^4 = \alpha^{12} = \alpha^5$.

Непременным элементом любого конечного поля является нулевой элемент. В связи с этим мы еще раз убеждаемся в том, что *все элементы поля* $\mathbf{GF}(2^r)$ *являются корнями многочлена* $x^{2^r} - x$.

С в о й с т в о 3. Поскольку все корни порождающего многочлена являются корнями многочлена $x^{2^r} - x$, в соответствии с теоремой Безу порождающий многочлен делит его без остатка.

С в о й с т в о 4. Многочлен $x^n - 1$ делится на $x^m - 1$ *в том и только в том случае*, если *n* делится на *m*. В самом деле, пусть $n = md$ и $x^m = y$. Тогда $x^n - 1 = y^d - 1$, а $y^d - 1$ делится на y - 1, поскольку $y = 1$ - корень многочлена $y^d - 1$. Подставляя $x^m = y$, получаем, что $x^{md} - 1$ делится на $x^m - 1$.

С в о й с т в о 5. Пусть *p(x)q(x)* – произвольный многочлен над полем **GF(2)** без кратных корней. Наименьшее значение *m*, при котором *p(x)q(x)* делит $x^m - 1$ без остатка, определяется как наименьшее общее кратное (НОК) порядков корней, задаваемых многочленами *p(x)* и *q(x)*.

Другими словами, многочлен *p(x)q(x)* принадлежит степени *m*, которая есть наименьшее общее кратное степеней, которым принадлежит каждый из многочленов – сомножителей *p(x)* и *q(x)*.

Напомним, что число ненулевых элементов поля – порядок поля, порядок корней – порядок поля, образованного этими корнями.

Пусть $g(x) = (x^3 + x + 1)(x^4 + x^3 + 1)$. Порядок корней первого из многочленов $2^3 - 1 = 7$, порядок корней второго многочлена $2^4 - 1 = 15$. Это эквивалентно тому, что многочлен $(x^3 + x + 1)$ принадлежит степени 7, и многочлен $(x^4 + x^3 + 1)$ принадлежит степени 15.

НОК(7, 15) = 105. Это значит, что многочлен *g*(*x*) делит без остатка многочлен $x^{105} - 1$, то есть многочлен *p(x)* принадлежит степени 105.

7.2. Построение кодов БЧХ, конструктивное расстояние кода БЧХ

Коды Боуза-Чоудхури-Хоквингема (коды БЧХ) являются линейными циклическими кодами и представляют собой обобщение циклических кодов. Эти коды позволяют исправлять многократные ошибки и пачки ошибок. БЧХ-коды задаются корнями порождающих многочленов. В качестве порождающих многочленов в этом случае служат приводимые многочлены *g*(*x*)=*p*(*x*)*q*(*x*), которые суть произведения неприводимых нормированных многочленов.

Циклический код длины $n = 2^r - 1$ над полем **GF(2)** называется *двоичным кодом БЧХ с конструктивным расстоянием* d_0, если для этого кода среди элементов $\alpha^{m_0}, \alpha^{m_0+1}, ..., \alpha^{m_0+d_0-2}$ расширения, построенного по порождающему многочлену $p(x)q(x)$, найдется $d_0 - 1$ элементов, которые являются корнями порождающего многочлена $p(x)q(x)$, и их степени последовательно увеличиваются на единицу. При этом m_0 – произвольное целое число. Чаще всего при задании кодов БЧХ принимают $m_0 = 1$. Если α – примитивный элемент конечного поля, код БЧХ называется *примитивным*.

Следующая теорема подтверждает приведенное определение кода БЧХ и позволяет дать нижнюю оценку для расстояния кодов БЧХ.

Теорема 12. (Граница БЧХ). Пусть A – циклический код с порождающим многочленом $g(x)=p(x)q(x)$ над полем **GF(2)** и α – корень многочлена $g(x)$, являющийся примитивным элементом расширения **GF(2^r)** по mod$g(x)$. Тогда если для некоторых целых чисел m_0 и d_0 выполняется равенство

$$g\left(\alpha^{m_0}\right) = g\left(\alpha^{m_0+1}\right) = g\left(\alpha^{m_0+2}\right) = ... = g\left(\alpha^{m_0+d_0-2}\right) = 0,$$

то минимальное расстояние кода не меньше d_0.

Расстояние d_0 называется *конструктивным расстоянием* кода A.

Текст теоремы говорит о том, что расстояние кода БЧХ будет не меньше d_0, если среди корней порождающего многочлена может быть выделена последовательность $d_0 - 1$ корней, степени которых последовательно увеличиваются на единицу. Эти корни являются корнями кода. При доказательстве настоящей теоремы используется утверждение теоремы 10 из разд. 5.5.1.

Доказательство.

Пусть $a(x) = x^{n-1} + a_{n-2}x^{n-2} + ... + a_1 x + a_0$ – кодовое слово и элементы $\alpha^{m_0}, \alpha^{m_0+1}, \alpha^{m_0+2}, ..., \alpha^{m_0+d_0-2}$ – корни кодового слова $a(x)$. Тогда должно выполняться равенство $\mathbf{aH}^{\mathbf{T}} = \mathbf{0}$, где $\mathbf{H}$ – проверочная матрица. Раскроем это равенство:

$$\left(a_{n-1} \quad a_{n-2} \quad \ldots \quad a_1 \quad a_0\right)\begin{pmatrix} \alpha^{m_0(n-1)} & \alpha^{(m_0+1)(n-1)} & \ldots & \alpha^{(m_0+d_0-3)(n-1)} & \alpha^{(m_0+d_0-2)(n-1)} \\ \alpha^{m_0(n-2)} & \alpha^{(m_0+1)(n-2)} & \ldots & \alpha^{(m_0+d_0-3)(n-2)} & \alpha^{(m_0+d_0-2)(n-2)} \\ . & . & . & . & . \\ \alpha^{m_0} & \alpha^{(m_0+1)} & \ldots & \alpha^{(m_0+d_0-3)} & \alpha^{(m_0+d_0-2)} \\ 1 & 1 & \ldots & 1 & 1 \end{pmatrix} = \left(0 \quad 0 \quad \ldots \quad 0 \quad 0\right).$$

Обозначим $\alpha^{n-i} = \beta_i$ и выберем из матрицы $\mathbf{H}^{\mathbf{T}}$ любые $d_0 - 1$ строк. Это действие равносильно тому, что из матрицы $\mathbf{H}$ выбираются любые $d_0 - 1$ столбцов. Тогда получившаяся матрица $\hat{\mathbf{H}}^{\mathbf{T}}$ будет квадратной, ее размер $(d_0 - 1)\times(d_0 - 1)$:

$$\hat{\mathbf{H}}^{\mathbf{T}} = \begin{pmatrix} \beta_1^{m_0} & \beta_1^{m_0+1} & \ldots & \beta_1^{m_0+d_0-3} & \beta_1^{m_0+d_0-2} \\ \beta_2^{m_0} & \beta_2^{m_0+1} & \ldots & \beta_2^{m_0+d_0-3} & \beta_2^{m_0+d_0-2} \\ . & . & \ldots & . & . \\ \beta_{d_0-2}^{m_0} & \beta_{d_0-2}^{m_0+1} & \ldots & \beta_{d_0-2}^{m_0+d_0-3} & \beta_{d_0-2}^{m_0+d_0-2} \\ \beta_{d_0-1}^{m_0} & \beta_{d_0-1}^{m_0+1} & \ldots & \beta_{d_0-1}^{m_0+d_0-3} & \beta_{d_0-1}^{m_0+d_0-2} \end{pmatrix}.$$

Для доказательства теоремы достаточно показать, что любые $d_0 - 1$ и меньше строк этой матрицы линейно независимы (см. также теорему 10 в п. 5.5.1, которая была сформулирована о столбцах матрицы **H**).

Какие бы $d_0 - 1$ строк этой матрицы мы не выбирали, из каждой i –й строки можно вынести общий множитель $\beta_i^{m_0}$. В результате при любом выборе строк матрицы $\mathbf{H}^{\mathbf{T}}$ получим общий множитель перед матрицей, который будет представлять собой произведение отличных от нуля сомножителей $\beta_i^{m_0}$ в количестве $d_0 - 1$ штук, а оставшаяся квадратная матрица приобретет вид:

$$\mathbf{V}=\begin{pmatrix} 1 & \beta_1 & \ldots & \beta_1^{d_0-3} & \beta_1^{d_0-2} \\ 1 & \beta_2 & \ldots & \beta_2^{d_0-3} & \beta_2^{d_0-2} \\ 1 & \beta_3 & \ldots & \beta_3^{d_0-3} & \beta_3^{d_0-2} \\ . & . & \ldots & . & . \\ 1 & \beta_{d_0-2} & \ldots & \beta_{d_0-2}^{d_0-3} & \beta_{d_0-2}^{d_0-2} \\ 1 & \beta_{d_0-1} & \ldots & \beta_{d_0-1}^{d_0-3} & \beta_{d_0-1}^{d_0-2} \end{pmatrix}$$

По условию теоремы, степени элементов β_i в каждой i – ой строке следуют друг за другом через единицу, поэтому определитель этой матрицы есть определитель Вандермонда. Этот определитель равен произведению разностей элементов второго столбца:

$$det\,\mathbf{V}=\prod_{i>j}\left(\beta_i-\beta_j\right).$$

Но поскольку все β_i - разные степени корней многочлена, степень которого n – 1, и $i\le n-1$, то все β_i различны. Поэтому определитель $\det\mathbf{V}\neq 0$, и значит, никакие d_0-1 строки не являются линейно зависимыми. Это значит, что расстояние кода не может быть меньше, чем d_0. Насколько больше будет расстояние кода, ответа не существует.

Доказанная теорема позволяет определить нижнюю оценку расстояние любого циклического кода посредством анализа элементов расширения $\mathbf{GF}\left(2^r\right)$, задаваемого его порождающим многочленом $g(x)$ степени r.

Для дальнейшего изложения введем понятие минимального многочлена.

Нормированный многочлен над полем **GF(2)** *наименьшей* возможной степени, корнем которого является элемент α^m расширенного поля, называется *минимальным многочленом (минимальной функцией)* элемента α^m. Степень m используется в качестве индекса минимального многочлена: $M_m(x)$.

Минимальный многочлен неприводим. Предположим противное, а именно, пусть минимальный многочлен может быть представлен произведением двух минимальных многочленов $M_m(x)=M_{1m}(x)M_{2m}(x)$, причем

степени обоих многочленов положительны. Тогда $M_{1m}(\alpha^m)M_{2m}(\alpha^m)=0$, а это значит, что либо $M_{1m}(\alpha^m)=0$, либо $M_{2m}(\alpha^m)=0$. Степень каждого из этих многочленов меньше степени многочлена $M_m(x)$. Это значит, что $M_m(x)$ – не может быть минимальным многочленом элемента α^m.

В соответствии со свойством 2 минимальный многочлен один и тот же для всех элементов некоторого циклотомического класса.

Из теоремы 11 и из свойства 5 разд. 7.1 следует, что код БЧХ с конструктивным расстоянием d_0 может быть задан порождающим многочленом, индекс которого представляет собой наименьшее общее кратное индексов таких минимальных многочленов, степени корней которых увеличиваются на единицу:

$$g(x)=\text{НОК}\left(M_{m_0}(x), M_{m_0+1}(x), ..., M_{m_0+d_0-2}(x)\right), \qquad (96)$$

где $M_{m_0}(x), M_{m_0+1}(x), ..., M_{m_0+d_0-2}(x)$ минимальные многочлены корней последовательных степеней $\alpha^{m_0}, \alpha^{m_0+1}, ..., \alpha^{m_0+d_0-2}$ соответственно, α - примитивный элемент конечного поля $\mathrm{GF}(2^r)$, где r – степень порождающего многочлена кода БЧХ. Понятно, что корнями БЧХ - кодов будут элементы α^{m_0}, α^{m_0+1}, ..., $\alpha^{m_0+d_0-2}$, поскольку, по построению, они же являются корнями порождающего многочлена. Кроме того, поскольку все эти корни являются корнями двучлена x^n-1, порождающий многочлен является его делителем, что свидетельствует о цикличности кода БЧХ.

В следующем пункте рассмотрим примеры построения минимальных многочленов и синтеза на их основе некоторых БЧХ-кодов при $m_0=1$.

7.3. Примеры минимальных многочленов

Приведем примеры минимальных многочленов для двоичных кодов при r = 2, 3, 4. Будем пользоваться свойствами, перечисленными выше в разд. 7.1.

П р и м е р 1. *r=2.* Длина кода $n=2^2-1=3$.

$$x^{2^2}+x=x^4+x=x(x+1)\left(x^2+x+1\right).$$

В этом случае расширенное поле есть $\mathbf{GF}\left(\mathbf{2^2}\right)$, оно порождено многочленом $x^2 + x + 1$ и содержит четыре элемента:

$$00, \qquad \alpha^0 = x^0 \bmod \left(x^2 + x + 1\right) = 1 \to \alpha = 01,$$

$$\alpha = x \bmod \left(x^2 + x + 1\right) = x \to \alpha = 10\ ,\ \alpha^2 = x^2\ mod\ \left(x^2 + x + 1\right) = x + 1 \to \alpha = 11.$$

Как и следовало ожидать, $\alpha^3 = x^3 \bmod \left(x^2 + x + 1\right) = 1 \to \alpha = 01$.

В приведенном разложении все многочлены неприводимы. Минимальные многочлены для всех корней:

Элемент	Минимальный многочлен
0	x
$\alpha^0 = 1$	$M_0(x) = x + 1$
$\alpha,\ \alpha^2$	$M_1(x) = M_2(x) = x^2 + x + 1$

П р и м е р 2. r = 3. Длина кода $n = 2^3 - 1 = 7$. Бином, делителем которого должны быть минимальные многочлены, это

$$x^{2^3} + x = x^8 + x = x(x+1)\left(x^3 + x + 1\right)\left(x^3 + x^2 + 1\right).$$

Все многочлены неприводимы, минимальные многочлены в расширении, образованном многочленом $\left(x^3 + x^2 + 1\right)$, имеют вид:

Элемент	Минимальный многочлен
0	x
$\alpha^0 = 1$	$M_0(x) = x + 1$
$\alpha, \alpha^2, \alpha^4$	$M_1(x) = M_2(x) = M_4(x) = x^3 + x^2 + 1$
$\alpha^3, \alpha^6, \alpha^{12} = \alpha^5$	$M_3(x) = M_6(x) = M_5(x) = x^3 + x + 1$

Из приведенных данных видно, что действительно, если расширение $\mathbf{GF}\left(\mathbf{2^3}\right)$ порождено многочленом $\left(x^3 + x^2 + 1\right)$, то для циклотомического класса, порожденного элементом α, получим:

$$M_1(x) = M_2(x) = M_4(x) = (x - \alpha)\left(x - \alpha^2\right)\left(x - \alpha^4\right) =$$

$$= x^3 + x^2\left(\alpha + \alpha^2 + \alpha^4\right) + x\left(\alpha^3 + \alpha^5 + \alpha^6\right) + \alpha^7 = x^3 + x^2 + 1.$$

Для циклотомического класса, порожденного элементом α^3, получим

$$M_3(x) = M_6(x) = M_5(x) = \left(x - \alpha^3\right)\left(x - \alpha^6\right)\left(x - \alpha^5\right) =$$
$$= x^3 + x^2\left(\alpha^3 + \alpha^6 + \alpha^5\right) + x\left(\alpha^9 + \alpha^8 + \alpha^{11}\right) + \alpha^{14} = x^3 + x + 1.$$

Все элементы расширения $\mathbf{GF}\left(2^3\right)$, использованные здесь, приведены выше, в разд.6.3 и при рассмотрении свойства 2 в разд. 7.1. Эти же элементы суть корни двучлена $x^8 + x$ по модулю многочлена $\left(x^3 + x + 1\right)$.

П р и м е р 3. $r = 4$. Длина кода равна $n = 2^4 - 1 = 15$. Количество корней, включая нулевой корень, равно 16. Бином, делителем которого должны быть минимальные многочлены, представляется в виде

$$x^{2^4} + x = x(x+1)\left(x^2 + x + 1\right)\left(x^4 + x + 1\right)\left(x^4 + x^3 + 1\right)\left(x^4 + x^3 + x^2 + x + 1\right).$$

Степени многочленов – сомножителей суть делители $r = 4$.

Если поле $\mathbf{GF}\left(2^4\right)$ задано многочленом $\left(x^4 + x + 1\right)$, то минимальные многочлены имеют вид:

Элемент	Минимальный многочлен
0	x
$\alpha^0 = 1$	$M_0(x) = x + 1$
$\alpha, \alpha^2, \alpha^4, \alpha^8$	$M_1(x) = M_2(x) = M_4(x) = M_8(x) = x^4 + x + 1$
$\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24} = \alpha^9$	$M_3(x) = M_6(x) = M_{12}(x) = M_9(x) = x^4 + x^3 + x^2 + x + 1$
α^5, α^{10}	$M_5(x) = M_{10}(x) = x^2 + x + 1$
$\alpha^7, \alpha^{13}, \alpha^{14}, \alpha^{56} = \alpha^{11}$	$M_7(x) = M_{14}(x) = M_{13}(x) = M_{11}(x) = x^4 + x^3 + 1$

Во всех приведенных примерах набор корней в каждой строке таблицы есть *циклотомический класс* корней двучлена $x^{2^r} + x$, что соответствует свойству 2, показанному ранее в разд. 7.1.

Многочлен, задающий поле, всегда может быть выбран примитивным. Для этого достаточно выбрать минимальный многочлен примитивного элемента. Однако задача определения, какой из неприводимых многочленов является примитивным, является весьма трудной.

7.4. Примеры синтеза кодов БЧХ

П р и м е р 1. Знакомый по предыдущим разделам (7, 4)-код Хемминга, являющийся частным случаем БЧХ-кода.

Степень порождающего многочлена (7, 4)-кода, исправляющего одну ошибку, как мы знаем, равна 3. Конечное расширенное поле **GF(8)** может быть задано одним из двух многочленов $g_1(x) = x^3 + x + 1$ или $g_1(x) = x^3 + x + 1$.

Выше в разд. 7.1 при иллюстрации свойства 2 было использовано поле **GF(8)**, построенное по многочлену $g_2(x)$. В состав этого поля входят элементы:

$$0 = 000,\ \alpha^0 = 001,\ \alpha^1 = 010,\ \alpha^2 = 100,\ \alpha^3 = 101,\ \alpha^4 = 111,\ \alpha^5 = 011,\ \alpha^6 = 110.$$

Примитивным элементом поля является $\alpha^1 = 010$, или в полиномиальном представлении, $\alpha^1 = x \bmod g_2(x)$. Минимальными многочленами для $\alpha, \alpha^2, \alpha^4$ являются многочлены $M_1(x) = M_2(x) = M_4(x) = g_2(x)$. Минимальными многочленами для $\alpha^3, \alpha^6, \alpha^5$ являются многочлены $M_3(x) = M_6(x) = M_5(x) = g_1(x)$ (см. пример 2**.** разд. 7.3)

Многочлен $g_1(x)$ задает иное поле **GF(8)**, состоящее из элементов:

$$0 = 000,\ \alpha^0 = 001,\ \alpha^1 = 010,\ \alpha^2 = 100,\ \alpha^3 = 011,\ \alpha^4 = 110,\ \alpha^5 = 111,$$
$$\alpha^6 = 101.$$

Примитивным элементом в этом поле является $\alpha^1 = 010$, или в полиномиальном представлении $\alpha^1 = x \bmod g_1(x)$. Минимальным многочленом

для $\alpha, \alpha^2, \alpha^4$ является многочлен $g_1(x)$. Для $\alpha^3, \alpha^6, \alpha^5$ минимальным многочленом является $g_2(x)$.

В соответствии с теоремой 12 разд. 7.1 показатели степени корней порождающего многочлена БЧХ-кода должны отличаться друг от друга на единицу. Количество корней в такой последовательности (то есть длина последовательности таких корней) должно быть не меньше, чем заданное кодовое расстояние минус единица .

Поскольку наиболее протяженная последовательность следующих друг за другом степеней элементов реализуется во втором случае, а именно α, α^2, порождающим многочленом кода является

$$g(x) = \text{НОК}[M_1(x), M_2(x)] = \text{НОК}\left[x^3 + x^2 + 1,\ x^3 + x^2 + 1\right] = x^3 + x^2 + 1.$$

В этом случае, когда $m_0 = 1$, из соотношения $m_0 + d_0 - 2 = 2$ получим расстояние кода $d_0 = 3$. Итак, длина кода равна $2^3 - 1 = 7$, количество избыточных символов равно степени порождающего многочлена $r = n - k = 3$, количество информационных символов равно $k = 7 - 3 = 4$. Количество передаваемых сообщений $2^4 = 16$.

Получен (7, 4)-код Хемминга, как частный случай БЧХ-кода.

П р и м е р 2. Построим код длиной 7, способный исправлять 2 ошибки. Для этого необходимо обеспечить кодовое расстояние, равное 5. Воспользуемся элементами поля **GF(8)**, построенного в примере 1.

Для исправления двух ошибок необходимо найти четыре следующих друг за другом корня, степени которых последовательно возрастают на единицу. Это корни $\alpha, \alpha^2, \alpha^3, \alpha^4$. Минимальным многочленом для $\alpha, \alpha^2, \alpha^4$ является многочлен $g_2(x)$, минимальным многочленом для α^3 является многочлен $g_1(x)$. Поскольку при $m_0 = 1$ из $m_0 + d_0 - 2 = 4$ следует $d_0 = 5$, порождающим многочленом будет

$$g(x) = \text{НОК}[M_1(x),\ M_2(x),\ M_3(x), M_4(x)] =$$

$$= \text{НОК}\left[x^3 + x^2 + 1,\ x^3 + x^2 + 1, x^3 + x + 1,\ x^3 + x^2 + 1\right] = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.$$

Таким образом получен (7, 1)-код. Число избыточных символов оказалось равным 6. Этот код способен к обнаружению 5 и к исправлению двух

ошибок. Однако его объем очень мал, он может передать лишь одно сообщение.

П р и м е р 3. Построим БЧХ-код длиной $n = 15$ и с расстоянием 5.

Воспользуемся примером в) из разд. 7.3.

Порождающий многочлен такого кода имеет вид:

$$g(x) = \text{НОК}[M_1(x), M_2(x), M_3(x), M_4(x)].$$

Мы остановились на минимальном многочлене $M_4(x)$, поскольку при $m_0 = 1$ из $m_0 + d_0 - 2 = 4$ следует $d_0 = 5$. Последовательность корней этого многочлена, степени которых возрастают на единицу, есть $\alpha, \alpha^2, \alpha^3, \alpha^4$.

Заимствуя минимальные многочлены с соответствующими номерами из примера 3 разд. 7.3, получим:

$$g(x) = \text{НОК}\left[x^4 + x + 1,\ x^4 + x + 1,\ x^4 + x^3 + x^2 + x + 1,\ x^4 + x + 1\right] =$$
$$= \left(x^4 + x + 1\right)\left(x^4 + x^3 + x^2 + x + 1\right) = x^8 + x^7 + x^6 + x^4 + 1.$$

Этот многочлен порождает БЧХ-код, число избыточных символов которого 8, длина $n = 15$, число информационных символов $k = 15 - 8 = 7$, конструктивное расстояние $d_0 = 5$. Этот код имеет параметры (15, 7), пригоден для кодирования $K = 2^7 = 128$ сообщений источника и способен исправить не менее 2 ошибок.

П р и м е р 4. Построим БЧХ-код длиной $n = 15$ и с расстоянием 7.

Как и раньше, воспользуемся примером 3 из разд. 7.3.

Порождающий многочлен такого кода имеет вид:

$$g(x) = \text{НОК}[M_1(x), M_2(x), M_3(x), M_4(x), M_5(x), M_6(x)].$$

В этот раз мы остановились на минимальном многочлене $M_6(x)$, поскольку при $m_0 = 1$ из $m_0 + d_0 - 2 = 6$ следует $d_0 = 7$.

Заимствуя минимальные многочлены с заданными номерами из примера в) разд. 7.3, получим:

$$g(x) = \text{НОК}\left[x^4 + x + 1,\ x^4 + x + 1,\ x^4 + x^3 + x^2 + x + 1,\ x^4 + x + 1, x^2 + x +,\ x^4 + x^3 + x^2 + x + 1\right] =$$
$$= \left(x^4 + x + 1\right) \cdot \left(x^4 + x^3 + x^2 + x + 1\right) \cdot \left(x^2 + x + 1\right) =$$
$$= x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1.$$

Мы получили (15, 5)-код с количеством избыточных символов 10, конструктивным расстоянием 7, способный кодировать $K = 2^5 = 32$ сообщения источника и исправлять 3 ошибки.

Конец примеров.

Отметим следующее. Расстояние БЧХ-кода, которое обеспечивается подобным построением и является конструктивным расстоянием, в ряде случаев оказывается меньше фактического расстояния построенного кода, как это мы видели в примерах 2–4. В общем случае фактическое расстояние можно определить путем перебора кодовых слов по признаку минимального их веса.

Обнаружение многократных ошибок при приеме БЧХ-кодов осуществляется путем деления кодовых слов на порождающий многочлен. Ненулевой остаток от такого деления свидетельствует о наличии ошибок. Однако исправление многократных ошибок вызывает большие затруднения, поскольку для идентификации каждой ошибки и позиции, в которой произошла каждая из них, приходится прибегать к весьма сложным алгоритмам. Здесь эти алгоритмы не рассматриваются. Интересующиеся читатели могут ознакомиться с алгоритмами декодирования кодов БЧХ и иных кодов, приведенными в технической литературе, например, в [4, 5, 7–10] .

Обширные таблицы неприводимых многочленов, на которые раскладываются биномы вида $x^{2^r-1}+1$, а также таблицы минимальных многочленов приведены в литературе [4, 7-10].

8. СВЕРТОЧНЫЕ КОДЫ

8.1. Общие свойства сверточных кодов

Алгебраическое описание, алгебраический инструментарий и методы кодирования и декодирования, описанные в разделах 3 – 7, относились к блоковым кодам. Входная кодовая последовательность разбивалась на блоки, которые кодировались независимо друг от друга. Таким образом закодированная выходная последовательность кодера состояла из блоков, содержащих всегда определенное количество информационных и проверочных символов. Блоковые коды имеют одинаковую длину, фиксированное минимальное расстояние, позволяющее обнаруживать и (или) исправлять определенное количество ошибок. Для этих кодов получены нижние и верхние оценки (Варшамова – Гилберта и Хемминга), которые дают возможность сопоставить эффективность каждого блокового кода с предельными возможностями. Кодирование и декодирование блоковых кодов осуществляется по жестким алгоритмам, реализуемым на сдвиговых регистрах.

Сверточные коды отличаются от блоковых кодов по всем изложенным позициям.

Удобными способами описания сверточных кодов являются древовидные структуры, решетчатые структуры или аппарат конечных автоматов. Кодирование в сверточных кодах сверточное, то есть *выходная последовательность сверточного кодера есть свертка входной последовательности с импульсной характеристикой кодера*. Длина сверточного кода теоретически бесконечна, и ее приходится искусственно ограничивать. В настоящее время не существует методов декодирования сверточных кодов, основанных на алгебраическом подходе, аналогичном изложенному в предыдущих разделах

Наиболее популярным алгоритмом декодирования сверточных кодов с исправлением ошибок является *алгоритм Витерби*, который по сути дела основан на применении метода максимального правдоподобия. Причиной популярности алгоритма Витерби является простота его реализации и значительный выигрыш от кодирования.

Достаточные сведения по описанию сверточных кодов а также методам их реализации и декодирования содержатся, например, в книге [10].

8.2. Двоичные сверточные кодеры

Простой пример сверточного кодера со скоростью $R = 1/2$ и кодовым ограничением 3 приведен на рис. 23. Кодовые слова на выходе сверточного кодера содержат в среднем одинаковое количество информационных и проверочных символов. Важным параметром сверточных кодов является *кодовое ограничение k*, равное количеству ячеек регистра сдвига, формирующего выходную последовательность. Информационные символы поступают в регистр сдвига слева, и для каждого информационного символа на выходах двух сумматоров по модулю 2 образуются два выходных символа. На данном рисунке ячейки регистра сдвига представляются порождающими многочленами: для верхнего сумматора $g_1(x) = 1 + x^2$, для нижнего сумматора $g_2(x) = 1 + x + x^2$. Степени здесь растут слева направо, и крайняя левая ячейка соответствует свободному члену.

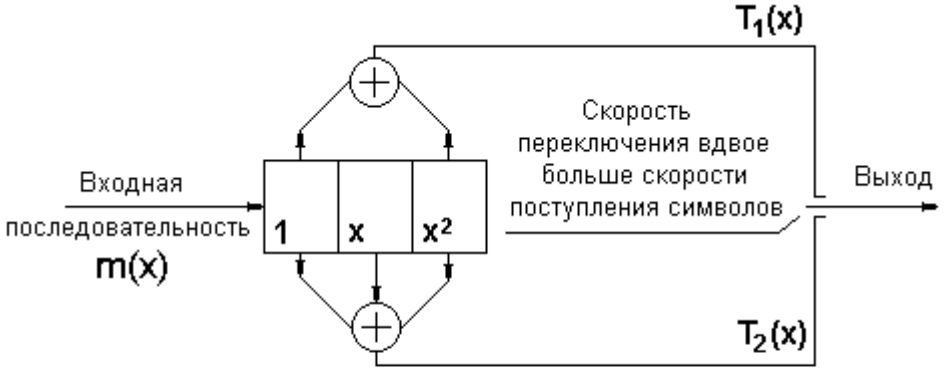

Рис. 23. Кодер для сверточного кода со скоростью 1/2
и с кодовым ограничением 3

Кодирование выполняется с тактом, который определяет тактовый генератор. На выходе в два раза быстрее чередуются кодовые символы двух последовательностей $T_1(x) = m(x)g_1(x)$ и $T_2(x) = m(x)g_2(x)$, где умножение выполняется по правилам, установленным для элементов поля **GF(2)**. В результате код получается несистематическим.

Можно говорить, что данный кодер обладает некоторой весовой функцией, и поэтому последовательность на выходе кодера можно рассматривать как свертку входной последовательности с весовой функцией кодера. Определить весовую функцию кодера нетрудно. Эта функция есть не что иное, как отклик на единичную импульсную функцию. Такой входной сигнал представляет собой единицу и последующие за ней нули. Откликом на этот сигнал в данном случае будет последовательность 1 1 0 1 1 1 0 0 0.... Это легко проверить по схеме рис. 23.

Порождающая матрица кода, получаемого с помощью данной схемы, является теоретически полубесконечной и может быть представлена в виде

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & ... \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & ... \end{pmatrix}$$

Кодовое слово, соответствующее сообщению источника, может быть получено умножением входного кода (в данном случае – вектора) на матрицу **G**. В таком случае входная последовательность

$$\mathbf{m} = (1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ ...\ )$$

порождает на выходе кодера последовательность **a**, полученную покомпонентным суммированием по mod2 строк матрицы **G** :

$$\mathbf{a} = (1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ ....).$$

Для сверточных кодов, к сожалению, не существует каких-либо конструктивных указаний по выбору порождающих многочленов. Эти многочлены исследователь или проектировщик подбирает прямым перебором с последующим моделированием, руководствуясь интуицией и опытом.

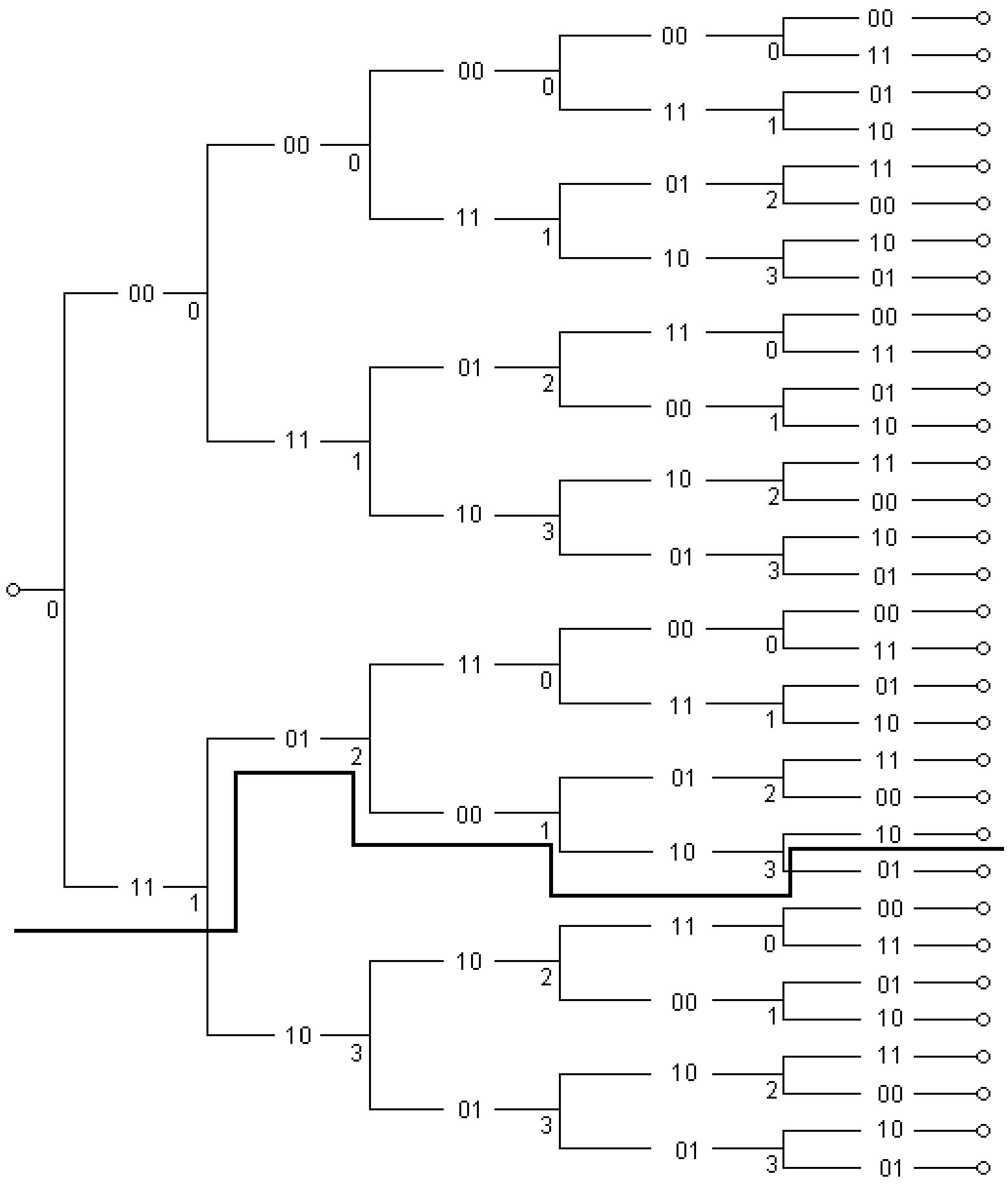

Рис. 24. Дерево для сверточного кода с кодовым ограничением 3.
В каждом узле верхняя ветвь соответствует 0, нижняя - 1

Другой удобный способ описания связей между входными и выходными кодовыми последовательностями состоит в использовании для этой

цели кодового дерева, подобно тому, как было описано в пп. 4.2.2 , 4.2.5. Пример дерева, пригодного для описания сверточного кода, приведен на рис. 24.

Из каждого узла выходят два ребра: одно идет наверх, это соответствует кодовому символу '0' входного кода (кода источника), второе ребро идет вниз, это соответствует кодовому символу 1 входного кода (кода источника). Таким образом каждая входная последовательность задает на дереве некоторый путь. В частности, входная последовательность 1 0 1 1 0 задает выходную последовательность 1 1 0 1 0 0 1 0 1 0, показанную на рисунке 24 жирной линией. Понятно, что при росте длины входной последовательности число возможных путей возрастает экспоненциально. Поэтому подобное представление сверточных кодов в виде дерева использовать на практике неудобно. Некоторое упрощение анализа сверточных кодов по их дереву возможно вследствие того, что уже на третьем ярусе два поддерева – нижнее и верхнее – становятся одинаковыми. Точно так же на четвертом ярусе одинаковыми являются уже четыре поддерева. Это открывает возможность отождествлять пути, приводящие к одинаковому состоянию. Такая возможность позволила перейти к решетчатому представлению сверточных кодов, которое будет рассмотрено в разд. 8.3.

8.3. Решетчатое представление сверточных кодов. Алгоритм декодирования Витерби

Решетчатое представление кодового дерева, изображенного на рис. 24, показано на рис. 25. Кодовым ограничением является количество ячеек в регистре сдвига на рис 23. В нашем случае кодовое ограничение $k = 3$. Число состояний решетки равно 2^{k-1}. В нашем случае число состояний решетки равно 4. Поэтому узлы решетки и состояния, соответствующие этим узлам, выражаются двумя двоичными разрядами. Это состояния, в которые переходят первые две ячейки регистра кодера, представленного на рис.23, при поступлении на вход левой ячейки кодового символа. Все ребра решетки обозначены двумя символами, которые получаются на выходе кодера при поступлении на вход (левую ячейку) кодового символа и переключении выходного переключателя. Как и раньше, любая входная

последовательность порождает выходную последовательность, которая соответствует некоторому пути на решетке. Например, последовательность 1 0 1 1 0, которая использовалась в разд. 8.2, дает выходную последовательность, совпадающую с результатом, полученным в упомянутом разд. 8.2:

1 1 0 1 0 0 1 0 1 0 .

Путь, соответствующий данной последовательности, нанесен на рис. 25 жирной линией.

Построение пути на решетке при входной последовательности 10110 выполняется следующим образом.

В первый момент на входе регистра сигнала нет, регистр находится в состоянии 00. При приходе первого символа 1 в левую ячейку две первые ячейки регистра находятся в состоянии 10, а на выходе генерируется последовательность, которая представляет собой две последовательные единицы, поскольку при переключении выходного ключа эти единицы поступают от левой ячейки.

Поэтому на первом такте выходная последовательность есть 11, а регистр находится в состоянии 10. Это ситуация после первого шага, она изображена на рис. 25.

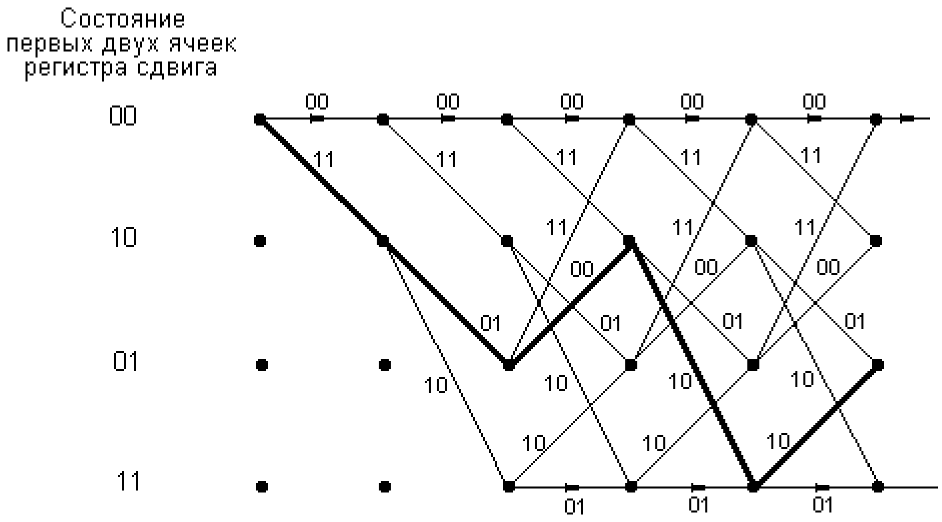

Рис. 25. Решетка для сверточного кода с кодовым ограничением 3

На втором такте на вход регистра приходит символ 0, выполняется сдвиг регистра, две левые ячейки регистра приходят в состояние 01, и при переключении выходного ключа на выходе образуется два символа: 0 и 1, поскольку только в средней ячейке регистра стоит единица, в остальных - нули, и единица поступает на выход из второго регистра. Таким образом, регистр оказывается в позиции 01, и на выходе к предыдущей последовательности добавляется последовательность символов 01. Это положение и переход изображены на рис. 25. Однако из этого же узла возможен переход в состояние 11, который возможен в том случае, когда на вход регистра поступит символ 1. Тогда две первые ячейки регистра окажутся в состоянии 11, в крайней правой ячейке в результате сдвига появится 1, и при переключении ключа на выходе образуются символы 10.

На третьем такте при поступлении единицы в регистре осуществляется очередной сдвиг, в левой ячейке оказывается эта единица, из второй ячейки единица перемещается в крайнюю правую ячейку, а в среднюю ячейку приходит нуль. Сложение двух единиц в обоих сумматорах по модулю 2 дает ноль, поэтому на выходе будет последовательность 00, а состояние двух левых ячеек 10. Описанный отрезок пути представлен на рис. 25, и теперь мы находимся в состоянии *10*. Если бы на вход регистра пришел нуль, то в двух левых ячейках были бы нули, что показано на рис. 25 ребром, выходящим из состояния 01 в состояние 00.

С приходом следующей единицы снова происходит сдвиг, две левые ячейки регистра приходят в состояние 11, в правую ячейку перемещается 0. Верхний сумматор дает единицу, из нижнего приходит 1 + 1 = 0. Последовательность символов на выходе будет 10. Позиция, в которую на этом такте переходят две левые ячейки регистра, есть 11, что отмечено на рис. 25.

Наконец, на вход регистра приходит 0. В результате сдвига регистра в двух левых ячейках оказывается 01, в правой – 1. Верхний сумматор дает 1, нижний – 0. Таким образом, на этом шаге состояние регистра – 01, на выходе последовательность 10, что и показано на рис. 25.

Таким образом выходной сверточный код 1 1 0 1 0 0 1 0 1 0, порожденный кодовым словом источника 10110, представляет собой последова-

тельность, образованную обозначениями переходов, из которых состоит весь путь, отмеченный жирной линией..

Отметим следующие свойства решетчатого представления сверточного кода:

в полном соответствии с деревом, представленным на рис. 24, начиная с третьего такта, все переходы между состояниями одинаковы;

два символа, стоящие у каждого ребра, обозначают переход между состояниями регистра и представляют собой фрагменты сверточного кода, порожденные на каждом такте символом кода источника;

при решетчатом представлении сверточного кода с ростом числа входных символов количество вершин в решетке не растет экспоненциально, как в дереве рис. 24, а остается постоянным, равным 2^{k-1}.

С ростом длины последовательностей количество путей растет экспоненциально, и задача декодирования сверточного кода на приемной стороне и исправления ошибок по методу максимального правдоподобия представляется, на первый взгляд, безнадежной. Но оказывается, что эта задача решается довольно просто с применением метрики Хемминга, вычисляемой для каждого пути на решетке. В этом состоит *алгоритм Витерби* декодирования сверточных кодов.

Он заключается в том, что, хотя на начальном отрезке путей на решетке в самом деле очень много, но в дальнейшем удается исключить некоторые пути, оставив сравнительно небольшой их список, содержащий наиболее правдоподобный путь. Еще раз отметим, что в решетке, уже начиная с третьего шага, в каждом узле решетки сходится по два пути. Поэтому декодер может выбрать из этих двух путей один, наиболее правдоподобный. Точно так же декодер может выбрать один из двух путей, сходящихся в каждом узле решетки на четвертом и на всех последующих шагах. Этим способом выбирается путь, который лучше всего согласуется с принятой последовательностью. Хотя этот способ не гарантирует минимума вероятности ошибки символа, но можно утверждать, что для всех кодов, кроме патологически плохих, малая вероятность последовательности приводит к малой вероятности ошибочного декодирования.

Выбор наиболее правдоподобного пути заключается в отклонении того пути, значение метрики которого оказалось больше. Оставшиеся пути с меньшим значением метрики Хемминга называются *выжившими*, а отклоненные – *погибшими* путями. Метрика Хемминга определяется как расстояние Хемминга между очередной парой символов кодового слова и обозначением возможных переходов в решетке на соответствующем такте.

Рассмотрим пример исправления ошибок с помощью алгоритма Витерби. Положим, что передавалась нулевая последовательность, в которой во время передачи были совершены две ошибки: 10 00 10 00 00 00...

На первом шаге при получении первых двух символов метрика Хемминга подсчитывается следующим образом.

Расстояние Хемминга между двумя первыми символами принятого слова, то есть 10, и обозначением перехода из состояния 00 в состояние 00 равно 1. Переход из состояния 00 в состояние 10 обозначен, как 11. Поэтому расстояние между двумя первыми символами принятого кода и обозначением перехода из состояния 00 в состояние 10 также равно 1. Других

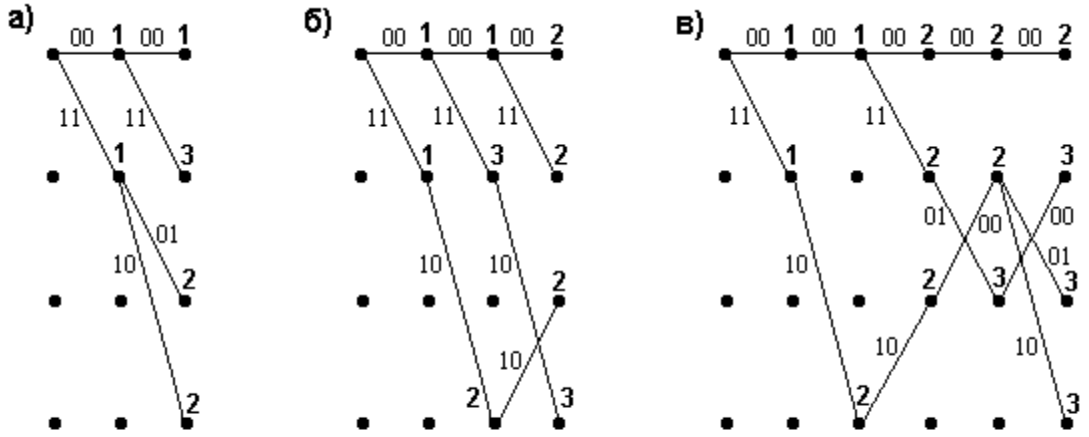

Рис. 26. Примеры декодирования с помощью алгоритма Витерби на втором, третьем и пятом шагах

путей на первом шаге быть не может (см. рис. 26 а). Значения накопленных значений метрики Хемминга нанесены на узлах решетки.

На втором шаге возможно большее количество путей (см. рис. 26 б). На этом шаге на вход декодера поступили символы 00. На переходе от состояния 00 к состоянию 00 не прибавилось ничего, поскольку расстояние

Хемминга между двумя пришедшими символами кода и обозначением перехода на этом отрезке пути равно 0. При переходе от состояния 00 к состоянию 10 расстояние Хемминга между обозначением этого пути (11) и принятыми двумя символами, равно 2, поэтому общая метрика двух отрезков пути 1 + 2 = 3. Обозначения двух путей из состояния 10 в состояния 01 и 11 удалены в смысле расстояния Хемминга от принятой пары символов 00 на единицу. Поэтому метрика принятой последовательности в узлах решетки на втором шаге равна: в состоянии 00 – 1, в состоянии 10 – 3. В состояниях 01 и 11 накопленная на этом пути метрика Хемминга 1 + 1 =2.

Метрика Хемминга на третьем шаге подсчитывается точно так же, как и на предыдущих. От принятых символов 10 на третьем шаге наибольшее расстояние 2 имеет путь от состояния 10 к состоянию 01. Это значение складывается с уже накопленным (2) и получается 4. Метрика другого пути к этому состоянию из состояния 11 не изменяется, и поэтому данный путь выживает, метрика состояния 01 равна 2. На пути из состояния 01 в состояние 00 добавляется 1 и в сумме метрика оказывается равной 3, а на альтернативном пути из состояния 00 в 00 метрика оказывается меньше, а именно, 2. Поэтому выживает альтернативный путь. На пути из состояния 01 в состояние 10 добавляется 1, метрика становится равной 3, и этот путь отмирает. Путь из состояния 10 в состояние 11 единственный. Его метрика не изменилась и осталась равной 3.

На рис. 26 в) показаны метрики состояний на пятом шаге. Мы видим, что количество путей, выживших от начала, уменьшилось. На девятом шаге (рис. 27) положение стало более определенным, и наконец, на десятом шаге (рис. 28) произошло слияние путей, и мы уже имеем возможность принять уверенное решение о переданной последовательности.

Глубина, на которой происходит слияние путей в алгоритме Витерби, не может быть вычислена заранее. Она является случайной величиной, которая зависит от характера ошибок. Также нельзя достоверно определить, как это было в алгебраических построениях, расстояние кода и количество исправляемых ошибок. Практические рекомендации состоят в том, чтобы глубину слияния устанавливать равной 4 – 5 кодовым ограничениям. Применительно к рассматриваемому примеру после достижения требуемой

глубины на вход подаются две комбинации нулей, и декодер возвращается в начальное состояние.

Реализация декодера состоит в следующем.

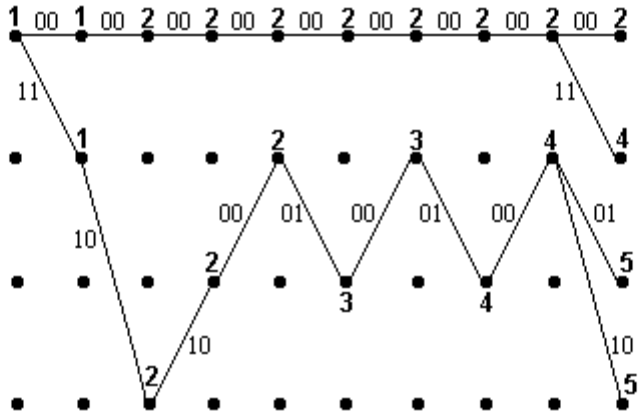

Рис. 27. Иллюстрация примера декодирования с помощью алгоритма Витерби на девятом шаге

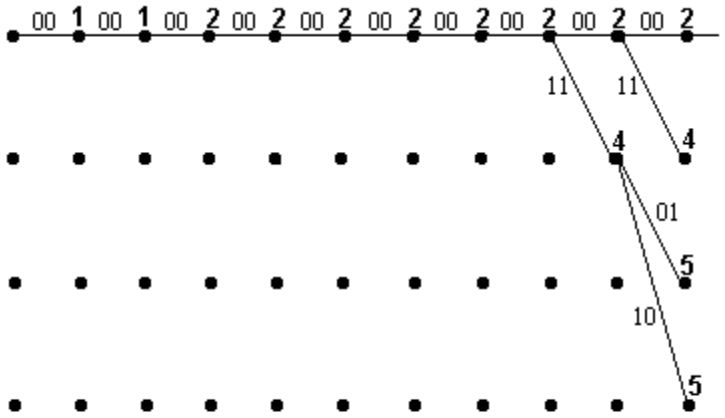

Рис. 28. Пример декодирования с помощью алгоритма Витерби на десятом шаге

Декодер состоит из регистров, количество которых $N = 2(k + 1)$, где k - кодовое ограничение. Каждая пара этих регистров последовательно накапливает значение метрики на фиксированном уровне: один из них - содержит значение метрики на i -ом шаге, второй – на $(i + 1)$-ом шаге. Запись нового значения метрики и идентификация регистра, куда должно

записываться новое значение метрики на каждом следующем шаге выполняется устройством управления в соответствии с решеткой, соответствующей кодеру.

В данном примере количество регистров может быть сокращено вдвое. Для этого нужно, чтобы устройство управления не переписывало из регистра в регистр новое значение метрики, а изменяло номера регистров, соответствующие состояниям, с одновременным обновлением значения метрики в них.

При большой глубине слияния в регистрах накапливаются большие значения метрик. Для хранения таких значений, объем регистров должен возрастать, в результате чего теряется мобильность. Во избежание этого на определенном шаге алгоритма из всех регистров вычитается некоторое постоянное число, не превосходящее минимального значения из всех накопленных метрик.

8.4. Сверточные коды со скоростью m/n

Структура кодера сверточного кода со скоростью 1/*n* остается такой же, как и структура кодера со скоростью 1/2. Отличие заключается только в том, что возрастает количество сумматоров по mod2, соответственно увеличивается число положений выходного переключателя и при появлении каждого входного символа на выходе появляется *n* символов. Решетка также не изменяется, только каждому ее ребру соответствует не 2 символа, а *n* символов. Кодовое ограничение или количество состояний решетки определяется, как и ранее, количеством разрядов единственного регистра.

При скоростях кодирования $R = m/n$ структура кодера изменяется. Кодер в этом случае должен иметь *m* входов (то есть *m* регистров) и *n* сумматоров. Пример такого кодера для кода со скоростью $R = 2/3$ показан на рис. 29. В такой кодер вводятся два символа, а на его выходе сумматоры вычисляют три символа для канала передачи сообщения. Код описывается шестью порождающими многочленами. Решетчатая структура декодера остается прежней с четырьмя состояниями, кодовое ограничение равно 3, но теперь каждому ребру соответствует три символа, а в каждую

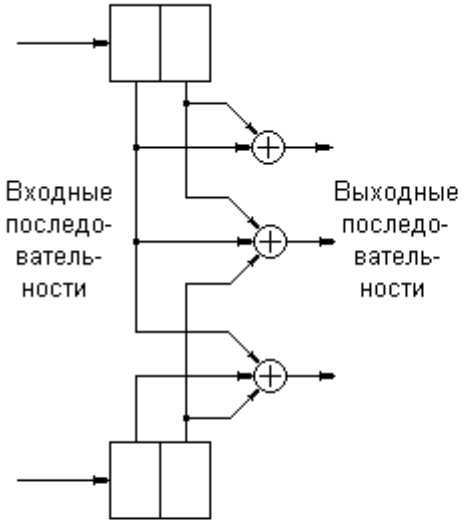

Рис. 29. Кодер для сверточного кода со скоростью 2/3

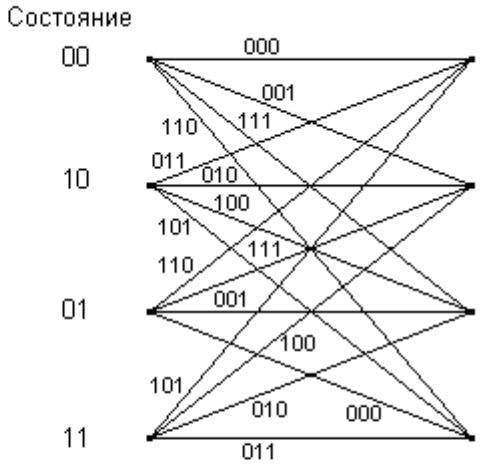

Рис. 30. Решетчатая структура для сверточного кода со скоростью 2/3

вершину входят четыре ребра. Решетчатая структура такого кода приведена на рис. 30. В соответствии с алгоритмом Витерби в этом случае необходимо в каждой вершине выбирать лучший путь из четырех, то есть произвести в два раза больше сравнений. В общем случае при $R = m/n$ в каждой вершине необходимо выполнить 2^m сравнений, что приводит к большим сложностям декодирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Евсеев Г. С.* Надежная передача и хранение информации в ЭВМ: учеб. пособие / Г. С. Евсеев, В. Д. Колесник, Е. А. Крук. – Л. : Изд-во ЛИАП, 1988. – 90 с.

2. *Колесник В. Д.* Курс теории информации / В. Д. Колесник, Г .Ш. Полтырев. – М. : Наука, 1982. – 416 с.

3. *Гельгор А. Л.* Основы теории информации /А. Л. Гельгор, Е. А. Попов. – СПб. : Изд-во Политехн. ун-та, 2008. – 78 с.

4. *Дмитриев В. И.* Прикладная теория информации / В. И. Дмитриев. – М. : Высш. шк., 1989. – 320 с.

5. *Колесник В. Д.* Декодирование циклических кодов / В. Д. Колесник, Е. Т. Мирончиков. – М. : Связь, 1968. – 251 с.

6. *Смирнов А. С.* Основы теории кодирования. Линейные групповые и циклические коды : учеб. пособие / А. С. Смирнов. – СПб. : Изд-во СПбГПУ, 1998. – 148 с.

7. *Заренин Ю. Г.* Корректирующие коды для передачи и переработки информации / Ю. Г. Заренин. – Киев : Техніка, 1965. – 170 с.

8. *Питерсон У.* Коды, исправляющие ошибки / У. Питерсон. – М. : Мир, 1964. – 338 с.

9. *Питерсон У.* Коды, исправляющие ошибки / У. Питерсон, Э. Уэлдон. – М. : Мир, 1976. – 594 с.

10. *Мак-Вильямс Ф. Дж.* Теория кодов, исправляющих ошибки / Ф. Дж. Мак-Вильямс. – М. : Связь, 1979. – 744 с.

11. *Кларк Дж. мл.* Кодирование с исправлением ошибок в системах цифровой связи / Кларк Дж. мл., Дж. Кейн. – М. : Радио и связь, 1987. – 391 с.

12. Основы криптографии : учеб. пособие / А. П. Алферов [и др.]. – М. : Гелиос АРВ, 2001. – 480 с.

13. *Кульбак С.* Теория информации и статистика / С. Кульбак. – М. : Наука, 1967. – 408 с.

Приложение 1

Необходимый математический аппарат

Группа

Группой **G** называется совокупность элементов, для которых определена операция * и выполняются следующие аксиомы.

1. Для любых $a \in \mathbf{G}$ и $b \in \mathbf{G}$, выполняется: $a*b \in \mathbf{G}$ – *замкнутость* относительно операции * .

2. Для любых $a \in \mathbf{G}$, $b \in \mathbf{G}$, $c \in \mathbf{G}$, выполняется $(a*(b*c)) = (a*b)*c$ – *ассоциативность*.

3. Для любого элемента $a \in \mathbf{G}$ существует *нейтральный* элемент e:

$$a*e = a.$$

В частности, если операция * – сложение, то *группа называется аддитивной*, нейтральный элемент называется *нулем* и обозначается через 0 . Если операция * – умножение, то *группа называется мультипликативной*, нейтральный элемент называется *единицей* и обозначается через 1.

4. Для любого элемента $a \in \mathbf{G}$ существует обратный элемент $b \in \mathbf{G}$:

$$a*b = e.$$

В частности, если операция * – сложение, то обратный элемент b обозначается - a. Если операция * – умножение, то обратный элемент b обозначается a^{-1}.

Если для любых a и b в группе $a*b = b*a$, то группа называется *коммутативной или абелевой*.

Пример абелевой группы – группа четных чисел, на которой определена обычная операция сложения. Единица в этой группе есть 0.

Циклическая группа – группа, каждый элемент которой может быть записан в виде счетного применения к элементу a операции *, то есть,

если $a \in \mathbf{G}$, то $g = a*a*a*.... \in \mathbf{G}$.

Элемент a называется *образующей* циклической группы.

Если операция * – умножение, то мультипликативная циклическая группа образуется степенями элемента *a*, начиная с нейтрального элемента группы – единицы:

если $a \in \mathbf{G}$, то $g = 1 \cdot a \cdot a^2 \cdot a^3 \cdot .... \in \mathbf{G}$.

Элемент *a* называется *порождающим элементом* группы.

Конечная группа - группа, число элементов которой конечно. Число элементов *n* в конечной группе называется *порядком группы*. *Порядком порождающего элемента a* группы называется наименьшее целое положительное число *n* такое, что $a^n = 1$. Порядок циклической группы равен порядку порождающего ее элемента. Все циклические группы абелевы.

Теорема. Если *a* – порождающий элемент циклической группы порядка *n*, то a^k – порождающий элемент той же группы, где *k* - число, взаимно простое с *n*.

Пусть *G* – конечная мультипликативная группа, $g_j = a^j$ – некоторый элемент этой группы. Для каждого элемента g_j группы в ней существует обратный элемент g^{n-j}, принадлежащий этой же группе.

Подгруппа **H** группы **G** – *подмножество* группы **G**, которое само является группой по отношению к операции, определенной на **G**.

Группа может быть разложена на смежные классы по подгруппе **H**.

Пусть в конечной группе **G** определена подгруппа

H : $\{e, h, h_1, h_2, ..., h_m\}$, *e* – нейтральный элемент подгруппы.

Пусть $g_1, g_2, \ldots, g_n$ – элементы группы, не вошедшие в подгруппу **H**.

В таблице

$h_1 = e$	h_2	h_3	. . . . .	h_m
$g_1 * e$	$g_1 * h_2$	$g_1 * h_3$	. . . .	$g_1 * h_m$
$g_2 * e$	$g_2 * h_2$	$g_2 * h_3$	. . . .	$g_2 * h_m$
.				
.				
$g_n * e$	$g_n * h_2$	$g_n * h_3$	. . . .	$g_n * h_m$

все строки, кроме первой, суть разложение группы на смежные классы по подгруппе **H**. Каждая *i* - ая строка $\{g_i H\}$ – левый смежный класс. Элемент g_i этого смежного класса называется его *образующим элементом*. Первый элемент в каждой строке – лидер соответствующего смежного класса.

Если у двух смежных классов оказался один элемент общий, то эти смежные классы совпадают.

Имеет место ***теорема Лагранжа***: Порядок подгруппы конечной группы является делителем порядка группы.

Кольцо

Кольцо **R** – множество элементов, на котором определены две операции: сложение и умножение, и обладающее следующими свойствами:

1. Множество **R** является аддитивной абелевой группой по сложению.

2. *Замкнутость кольца*. Для любых двух элементов из множества **R** определено произведение *a b***,** которое является элементом **R**.

3. *Ассоциативный закон.* Для любых трех элементов *a, b, c* из множества **R** справедливо: *a(bc)=(ab)c*.

4. *Дистрибутивный закон.* Для любых трех элементов *a, b, c* из множества **R** справедливы равенства: $a(b + c) = ab + ac$, $(b + c)a = ba + ca$.

Кольцо называется коммутативным, если $ab = ba$.

В кольце **R** имеется аддитивная единица. Мультипликативной единицы, вообще говоря, нет. Но если она есть, кольцо называется *кольцом с единицей*.

Обратные элементы по сложению имеются, по умножению – не обязательны, но в кольце с единицей могут быть и обозначаются a^{-1}.

Характеристика кольца – число целых чисел в кольце.

Непустое множество $\mathbf{R}'$ элементов кольца **R** называется *подкольцом кольца* **R,** если:

$\mathbf{R}'$ является подгруппой аддитивной группы кольца **R**;

$\mathbf{R}'$ замкнуто относительно операции умножения, заданной на **R**.

Подкольцо **J** кольца **R** называется *идеалом*, если для произвольного элемента *r* кольца **R** и элемента $a \in \mathbf{J}$ произведение $ar = ra \in \mathbf{J}$. Если

кольцо обладает единицей, идеал можно выразить как $ra + na = (r + n)a$, где n – любое целое число. Это значит, что идеал, порождаемый элементом $a \in \mathbf{R}$, состоит из всех элементов кольца **R**, кратных a. Идеал, порождаемый элементом a , называется *главным идеалом*. Здесь мы имеем дело только с главными идеалами, поэтому слово "главный" будем из определения опускать.

Имеет место ***теорема***: всякие идеалы кольца целых чисел и кольца полиномов являются главными.

Поле

Поле **F** – коммутативное кольцо с единичным элементом по умножению, в котором каждый ненулевой элемент обладает мультипликативным обратным элементом (то есть обратным по умножению).

Все ненулевые элементы поля удовлетворяют всем аксиомам группы и, следовательно, образуют мультипликативную группу, то есть группу по умножению.

Пример поля – поле всех вещественных чисел.

Элементы поля обладают следующими свойствами.

1. Поле – абелева группа по сложению.
2. Поле – абелева группа по умножению ненулевых элементов.
3. В поле действует дистрибутивный закон $(a + b)c = ac + bc$.
4. Единичный элемент поля по сложению – 0.
5. Единичный элемент поля по умножению – 1.
6. Обратный элемент поля по сложению для $a \in \mathbf{F}$ (аддитивный обратный элемент) – $(-a)$.
7. Мультипликативный обратный элемент любого ненулевого элемента a поля **F** – a^{-1}.
8. Минимальное количество элементов поля: 2.

Вычитание в поле **F** : $a + (-b)$.

Деление в поле **F** : $a/b = b^{-1}a$.

Примером поля может служить множество действительных чисел с определенными на нем операциями сложения и умножения.

Подполе – подмножество поля, на котором определены операции поля.

Конечное поле – поле с конечным числом элементов q, обозначается, как **GF(q)**. Количество элементов конечного поля называется *порядком поля*. Начало теории конечных полей положил Эварист Галуа, чьим именем конечные поля называются.

Каждое конечное поле имеет единичный и нулевой элементы, которые обозначаются 0 и 1 соответственно. Складывая последовательно единицы конечного поля, получим ряд: 1, 1+1, 1+1+1, ... В указанном ряду найдется сумма единиц, равная нулевому элементу, поскольку число элементов этого поля конечно. Наименьшее число, удовлетворяющее этому условию, является простым числом. Пусть $p = p_1 p_2$, где $p_1 < p$ и $p_2 < p$. Но тогда $p_1 p_2 = p = 0$, а это означает, что в конечном поле появились делители нуля, что невозможно. Поэтому поля могут быть конечными, только если q – простое число.

В конечном поле Галуа **GF(q)** определены арифметические операции: сложение и умножение по модулю q. Это означает, что числа – элементы поля складываются и умножаются, как обыкновенные целые числа, а полученное в результате число делится на q. Остаток от деления есть *вычет* и является результатом соответствующего арифметического действия в конечном поле. Множество таких остатков называется *полной системой вычетов* по mod q. Конечное поле является полной системой вычетов.

П р и м е р 1.

Конечное поле **GF(5)**, Элементами поля являются {0, 1, 2, 3, 4}. Сложим два элемента поля 3 и 4 по mod5: $3 + 4 = 7 = 5 + 2 = 2$. Умножим 3 на 4 по mod5: $3 \times 4 = 12 = 5 \times 2 + 2 = 2$.

Итак, в поле **GF(5)** $(3 + 4) \bmod 5 = 2$, $(3 \times 4) \bmod 5 = 2$.

Простое число p , обладающее свойством $p = 0 \bmod q$ в конечном поле **GF(q)**, называется *характеристикой* этого поля.

В конечном поле **GF(q)** каждый вычет имеет обратный элемент $x = a^{-1}$ в поле, поскольку можно показать, что по вычету $1 \bmod q$ из уравнения $ax = \bmod q$ можно найти обратный элемент, единственный в данном

поле. Общий рецепт отыскания обратного элемента указать невозможно, обычно этот элемент находят путем перебора.

Мультипликативным порядком (или просто *порядком*) ненулевого элемента g поля **GF(q)** называется наименьшее положительное число n, для которого $g^n = 1 \bmod q$ в поле **GF(q)**, или, в соответствии с правилами арифметических действий, установленных в этом поле по mod q, $g^n \bmod q = 1$.

П р и м е р 2. Проверим выполнение последнего соотношения для ненулевых элементов поля **GF(5)** при $n = q - 1$.

$1^4 = 1, \quad 2^4 = 1 \, mod \, 5, \quad 3^4 = 1 \, mod \, 5, \quad 4^4 = 1 \, mod \, 5.$

Мультипликативный порядок любого элемента этого поля равен 4.

Заметим, что если в последнем примере умножить обе части второго равенства на 2, то получим: $2^5 = 2 \bmod 5$. Этот факт может служить подтверждением так называемой ***малой теоремы Ферма***, а именно, для любого простого числа p и любого целого a имеет место равенство

$$a^p = a \bmod p.$$

Вообще в поле **GF(q)** каждый ненулевой элемент поля g удовлетворяет равенству $g^{q-1} = 1$ или $g^q = g$.

В конечном поле **GF(q)** существует по крайней мере один элемент, такой, что все элементы поля могут быть представлены степенью этого элемента, который называется *порождающим* или *примитивным* элементом. Мультипликативный порядок примитивного элемента в точности равен q - 1.

П р и м е р 3.

В поле **GF(5)** примитивными элементами являются *2* и *3*:

$$2^1 = 2 \, mod \, 5, \quad 2^2 = 4 \, mod \, 5, \quad 2^3 = 3 \, mod \, 5, \quad 2^4 = 1 \, mod \, 5,$$

$$3^1 = 3 \, mod \, 5, \quad 3^2 = 4 \, mod \, 5, \quad 3^3 = 2 \, mod \, 5, \quad 3^4 = 1 \, mod \, 5.$$

Число 4 примитивным элементом не является:

$$4^1 = 4 \, mod \, 5, \quad 4^2 = 1 \, mod \, 5, \quad 4^3 = 4 \, mod \, 5, \quad 4^4 = 1 \, mod \, 5.$$

Пусть α – примитивный элемент поля **GF(q)**. Тогда любой элемент поля выражается степенью примитивного элемента $g = \alpha^i$ и выполняется равенство $\alpha^{i(q-1)} \bmod q = 1$.

П р и м е р 4. Проверим, является ли число 8 порядком элементов *2* и *3*.

$2^8 \bmod 5 = 256 \bmod 5 = 1, \quad 3^8 \bmod 5 = 6561 \bmod 5 = 1.$

Такой же результат получается при всех порядках, делящихся на 4.

Для описания двоичных кодов и исследования их свойств применяется конечное поле, характеристика которого равна $q = 2$. Это поле является конечным полем вычетов по mod 2, содержит 2 числа 0 и 1 и обозначается **GF(2)**.

Проверим наличие перечисленных выше свойств у элементов поля **GF(2)**.

0+0=0 mod 2, 0+1=1 mod 2, 1+1 =0 mod 2, то есть первое свойство выполняется.

0·0=0 mod 2, 0·1=0 mod 2, 1·1=1 mod 2, то есть второе свойство выполняется.

Аддитивный единичный элемент 0 и мультипликативный единичный элемент 1 – элементы поля.

Аддитивные обратные элементы: для 0 есть -0, так как 0+0=0 mod 2**,** для 1 есть –1**,** так как 1+1= 0 mod 2**,** откуда следует, что –1=1 mod 2. Для любого элемента *g* поля **GF(2)** справедливо равенство $g - 1 = g + 1$.

Мультипликативный обратный элемент для 1 есть 1, так как 1·1=1 mod 2**.**

Ненулевой элемент поля удовлетворяет равенству x – 1, то есть 1. Мультипликативный порядок этого элемента q – 1. Этот же элемент есть примитивный элемент поля **GF(2)**.

В дальнейшем будем выполнять действия над элементами поля **GF(2)**, опуская запись mod 2**.**

Линейные пространства над полем GF(2)

Пусть **X = GF(2)** – двоичное конечное поле. Будем рассматривать множество X^n всех двоичных слов над алфавитом **X** как множество векторов длиной *n,* составляющие которого суть элементы поля **GF(2)**. Пусть над векторами из $\mathbf{X}^n$ определены операции умножения вектора на элемент поля (скаляр) и сложения векторов: если векторы $\mathbf{x} = (x_1, x_2, ..., x_n)$ и $\mathbf{y} = (y_1, y_2, ..., y_n)$ принадлежат $\mathbf{X}^n$ и $\alpha \in \mathbf{GF(2)}$, то

$$\alpha \cdot \mathbf{x} = (\alpha x_1, \alpha x_2, ..., \alpha x_n), \quad \mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, ..., x_n + y_n),$$

где умножение и сложение выполняются по модулю 2.

Множество $\mathbf{V}_n \in \mathbf{X}^n$ называется *линейным пространством* над полем **X** = **GF(2)**, если оно замкнуто относительно операций: умножения на скаляр из поля **GF(2)** и сложения, то есть, если для любых $\mathbf{x} \in \mathbf{X}^n$ и $\mathbf{y} \in \mathbf{X}^n$ и любого $\alpha \in \mathbf{X}$ имеют место включения: $\alpha\mathbf{x} \in \mathbf{V}_n$ и $\mathbf{x} + \mathbf{y} \in V_n$. Любой вектор $\mathbf{y} = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + ... + \alpha_k\mathbf{x}_k = \sum_{i=1}^{k} \alpha_i \mathbf{x}_i$, где элементы $\alpha_1, \alpha_2, ..., \alpha_k$ – элементы поля **GF(2)**, а $\mathbf{x_1, x_2, \ldots, x_k}$ - произвольные ненулевые векторы из $\mathbf{X}^n$, называется *линейной комбинацией* векторов $\mathbf{x_1, x_2, \ldots, x_k}$. Из определения линейного пространства $\mathbf{V}_n$ следует, что всякая линейная комбинация векторов из $\mathbf{V}_n$ принадлежит этому же пространству. Это верно и при $a_1 = a_2 = ... = a_k = 0$. Поэтому и нулевой вектор является элементом любого линейного пространства.

Подмножество элементов линейного пространства, которое удовлетворяет всем аксиомам векторного пространства, называется *подпространством*.

Векторы $\mathbf{x_1, x_2, ..., x_k}$ *линейно независимы* над **GF(2)** тогда и только тогда, когда соотношение $\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + ... + \alpha_k\mathbf{x}_k = 0$ выполняется только при равенстве нулю всех коэффициентов $a_1 = a_2 = ... = a_k = 0$.

В любом линейном пространстве $\mathbf{V}_n$ существуют линейно независимые векторы $\mathbf{x_1, x_2, \ldots, x_k}$, такие, что любой вектор $\mathbf{y} \in \mathbf{X}^n$ может быть

единственным образом представлен в виде линейной комбинации векторов $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$. Набор векторов $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ называется *базисом линейного пространства*, число компонентов k этого вектора – *размерностью линейного пространства*. Любое k-мерное линейное пространство над полем **GF(2)** содержит 2^k векторов, и наоборот, если линейное пространство над полем **GF(2)** содержит 2^k векторов, то его размерность равна k.

На множестве $\mathbf{X}^n$ векторов длиной n над полем **GF(2)** определено *скалярное произведение* двух векторов $\mathbf{x}$ и $\mathbf{y}$ следующим образом

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i y_i \in \mathbf{GF(2)}.$$

Если $(\mathbf{x}, \mathbf{y}) = 0$, векторы называются *ортогональными*.

Пусть $\mathbf{V}_n$ – линейное k - мерное пространство над **GF(2)** и $\mathbf{W}_n$ – множество всех векторов из $\mathbf{X}^n$, ортогональных к $\mathbf{V}_n$ то есть $(\mathbf{x}, \mathbf{y}) = 0$ для любых $\mathbf{x} \in \mathbf{V}_n$ и $\mathbf{y} \in \mathbf{W}_n$. Тогда $\mathbf{W}_n$ – линейное пространство размерности k над полем **GF(2)**. Пространство $\mathbf{W}_n$ называется *ортогональным дополнением* пространства $\mathbf{V}_n$.

Приложение 2

Экстремальные значения энтропии непрерывных информационных сигналов, случайных шумов и помех

Непрерывные информационные сигналы, возникающие в некоторых источниках информации, преобразуются затем в дискретные сигналы с помощью аналого-цифровых преобразователей. Собственные тепловые шумы и помехи, действующие в линии связи, являются непрерывными аддитивными по отношению к информационному сигналу случайными процессами и дискретизируются в силу дискретного действия декодера канала. Для описания информационной характеристики подобных аналоговых сигналов применяется *дифференциальная* или *относительная* энтропия. Энтропия непрерывных сигналов называется так потому, что функционал энтропии, применяемый к дискретным случайным величинам (к дискретным сигналам), не может применяться к непрерывным величинам (сигналам), а потому дифференциальная энтропия вычисляется по отношению к энтропии непрерывной случайной величины, распределенной равномерно в интервале [0, 1]:

$$H_{\text{dif}} = \int_{-\infty}^{+\infty} \varphi(x) \log_2 \varphi(x) dx, \qquad \text{(П-1)}$$

где $\varphi(x)$ – одномерная плотность распределения мгновенных значений процесса (шума и помех).

Несмотря на то что мы рассматриваем дискретные совокупности, нахождение экстремальных значений энтропии непрерывных шумов и помех имеет смысл, поскольку эта энтропия позволяет выполнить оценку наибольшего возмущающего влияния указанных факторов на качество передачи информации.

Задача состоит в нахождении такой одномерной плотности распределения случайных помех в линии связи или случайного процесса, которая доставляет максимум энтропии при двух наиболее типичных вариантах технических ограничений:

1. о случайном процессе (шумах и помехах) известно лишь то, что его мгновенные значения не выходят за пределы ($x_{\max}$,+ $x_{\max}$): $|x| \le x_{\max}$;

2. о случайном процессе (шумах и помехах) известно лишь, что его дисперсия (мощность) ограничена, то есть ;

П р и о г р а н и ч е н и я х 1 задача формулируется следующим образом.

Найти плотность распределения *φ(x)*, при которой

$$H_{\text{diff}} = \int_{-x_{\max}}^{x_{\max}} \varphi(x)\log_2\varphi(x)dx = \max_{\varphi(x)}$$

в условиях $|x| \le x_{\max}$ и $\int_{-x_{\max}}^{x_{\max}} \varphi(x)dx = 1$.

Покажем вначале, что

$$-\int_{-x_{\max}}^{x_{\max}} \varphi(x)\log_2\frac{1}{2x_{\max}}dx = -\log_2\frac{1}{2x_{\max}}\int_{-x_{\max}}^{x_{\max}}\varphi(x)dx = -\log_2\frac{1}{2x_{\max}}.$$

Теперь воспользуемся неравенством $\ln x \le x-1$ (см. рис. 3):

$$H_{\text{dif}}(x)+\log_2\frac{1}{2x_{\max}} = -\int_{-x_{\max}}^{x_{\max}}\varphi(x)\log_2\varphi(x)dx + \int_{-x_{\max}}^{x_{\max}}\varphi(x)\log_2\frac{1}{2x_{\max}}dx =$$

$$= \log_2 e\int_{-x_{\max}}^{x_{\max}}\varphi(x)\left[\ln\frac{1}{2x_{\max}} - \ln\varphi(x)\right]dx = \log_2 e\int_{-x_{max}}^{x_{max}}\varphi(x)\ln\frac{1}{2x_{\max}\varphi(x)}dx \le$$

$$\le \log_2 e\int_{-x_{max}}^{x_{\max}}\varphi(x)\left[\frac{1}{2x_{\max}\varphi(x)}-1\right]dx = \log_2 e\int_{-x_{\max}}^{x_{\max}}\left[\frac{1}{2x_{\max}}-\varphi(x)\right]dx = \log_2 e[1-1] = 0$$

. Отсюда следует, что $H_{\text{diff}}(x)+\log_2\frac{1}{2x_{\max}} \le 0$, то есть

$$H_{\text{diff}}(x) \le x_{\max}\log_2 2.$$

Но правая часть этого неравенства есть дифференциальная энтропия случайной величины, распределенной равномерно, что следует из (П-1):

$$H_{\text{dif}}(x) = -\frac{1}{2x_{\max}}\int_{-x_{\max}}^{x_{\max}}\log_2\frac{1}{2x_{\max}}dx = \log_2 2x_{\max}.$$

Это означает, что при указанных условиях максимальной энтропией обладает случайная величина, распределенная равномерно в пределах $[-x_{\max}, +x_{\max}]$.

П р и о г р а н и ч е н и я х 2 задача формулируется следующим образом.

Найти плотность распределения $\varphi(x)$, при которой

$$H_{\text{diff}}(x) = \int_{-x_{\max}}^{x_{\max}} \varphi(x)\log_2 \varphi(x)dx = \max_{\varphi(x)}$$

в условиях и $\int_{-x_{\max}}^{x_{\max}} \varphi(x)dx = 1$.

Поскольку дисперсия случайной величины и дифференциальная энтропия не зависят от сдвига, то есть от математического ожидания, примем математическое ожидание равным нулю. Тогда первое условие формулируется в виде:

$$\int_{-\infty}^{\infty} x^2 \varphi(x)dx \le \sigma^2.$$

Покажем вначале, что

$$-\int_{-\infty}^{\infty} \varphi(x)\log_2\left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}\right]dx = \frac{1}{2}\log_2\left(2\pi\sigma^2\right) + \frac{\log_2 e}{2\sigma^2}\int_{-\infty}^{\infty} x^2\varphi(x)dx =$$

$$= \frac{1}{2}\log_2\left(2\pi\sigma^2\right) + \frac{1}{2}\log_2 e = \frac{1}{2}\log_2\left(2\pi e\sigma^2\right).$$

Теперь воспользуемся неравенством $\ln x \le x - 1$ (см. рис. 3):

$$H_{\text{diff}}(x) - \frac{1}{2}\log_2\left(2\pi e\sigma^2\right) = -\int_{-\infty}^{\infty}\varphi(x)\log_2\varphi(x)dx + \int_{-\infty}^{\infty}\varphi(x)\log_2\left[\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}\right]dx =$$

$$= \log_2 e\int_{-\infty}^{\infty}\varphi(x)\ln\left[\frac{1}{\varphi(x)\sqrt{2\pi}\sigma}e^{-\frac{2x}{2\sigma^2}}\right]dx \le \log_2 e\int_{-\infty}^{\infty}\varphi(x)\left[\frac{1}{\varphi(x)\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}} - 1\right]dx =$$

$$= \log_2 e \int_{-\infty}^{\infty} \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} - \varphi(x) \right] dx = \log_2 e \cdot (1-1) = 0 .$$

Это означает, что $H_{\text{dif}f}(x) - \frac{1}{2}\log_2\left(2\pi e\sigma^2\right) \le 0$, то есть

$$H_{\text{diff}}(x) \le \frac{1}{2}\log_2\left(2\pi e\sigma^2\right).$$

Но правая часть этого неравенства есть дифференциальная энтропия случайной величины, распределенной в соответствии с нормальной плотностью распределения, что следует из (П-1):

$$H_{\text{diff}}(x) = -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \log_2 \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \right] dx =$$

$$= \frac{1}{2}\log_2\left(2\pi\sigma^2\right) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx + \frac{\log_2 e}{2\sigma^2} \int_{-\infty}^{\infty} \frac{x^2}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx =$$

$$= \frac{1}{2}\log_2\left(2\pi\sigma^2\right) + \frac{1}{2}\log_2 e = \frac{1}{2}\log_2\left(2\pi e\sigma^2\right).$$

Это означает, что при указанных условиях максимальной энтропией обладает случайная величина, распределенная нормально с дисперсией σ^2.