

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«КАЗАНСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. А.Н. ТУПОЛЕВА-КАИ»

С.А. ЛЯШЕВА, М.П. ШЛЕЙМОВИЧ,  
З.Т. ЯХИНА

# ТЕОРИЯ ИНФОРМАЦИИ И КОДИРОВАНИЯ

*Учебно-методическое пособие*

*Рекомендовано к изданию Учебно-методическим управлением  
КНИТУ-КАИ*

Казань 2020

УДК 519.72

ББК 32.811

Ляш 99

*Рецензенты:*

доктор технических наук И.И. Исмагилов  
(Казанский (Приволжский) федеральный университет);  
кафедра интеллектуальных систем и управления  
информационными ресурсами (КНИТУ-КХТИ)

**Ляшева, С.А.**

Ляш 99 Теория информации и кодирования: учебно-методическое пособие / С.А. Ляшева, М.П. Шлеймович, З.Т. Яхина. – Казань: Изд-во КНИТУ-КАИ, 2020. – 120 с.

ISBN 978-5-7579-2493-1

Рассматриваются информационные характеристики случайных систем, информационные характеристики каналов связи, кодирование информации, сжатие информации, помехоустойчивое кодирование информации и шифрование информации. Предназначено для студентов направления 10.03.01 «Информационная безопасность», обучающихся по дисциплине «Теория информации и кодирования».

Ил. 7. Табл. 20. Библиогр.: 11 назв.

УДК 519.72

ББК 32.811

**ISBN 978-5-7579-2493-1**

© С.А. Ляшева, М.П. Шлеймович,  
З.Т. Яхина, 2020

© Изд-во КНИТУ-КАИ, 2020

# **1. ОСНОВЫ ТЕОРИИ ИНФОРМАЦИИ И КОДИРОВАНИЯ**

## **1.1. Информационные характеристики случайных систем**

В настоящее время в научной и технической деятельности применяется подход к объектам исследования и проектирования как к системам. В зависимости от характера деятельности в понятие «система» вкладывается различный смысл, но во всех случаях система есть подмножество взаимосвязанных элементов, выделенных из множества элементов любой природы в соответствии с требованиями решаемой задачи.

Для описания поведения системы используется понятие «состояние». Состояние системы – это совокупность значений, существенных с точки зрения решаемой задачи ее параметров. Если поведение системы описывается параметрами, значения которых представляют собой случайные величины, то система является случайной или вероятностной.

С точки зрения теории вероятностей, состояния случайной системы можно представлять как исходы некоторого опыта. При этом если система имеет счетное множество состояний, то она является дискретной, иначе – непрерывной.

Дискретную случайную систему  $X$  можно описать в следующем виде:

$$X = \{x_1, x_2, \dots, x_n\};$$

$$P(X) = \{p(x_1), p(x_2), \dots, p(x_n)\};$$

$$\sum_{i=1}^n p(x_i) = 1,$$

где  $x_1, x_2, \dots, x_n$  – состояния системы (значения случайной величины);  $p(x_1), p(x_2), \dots, p(x_n)$  – вероятности состояний системы (значений случайной величины).

Очевидно, что для определения состояния системы необходимо учитывать число возможных ее состояний и их вероятности. Поэтому в качестве меры неопределенности дискретной случайной системы была предложена величина:

$$H(X) = \sum_{i=1}^n p(x_i) \log_a \frac{1}{p(x_i)} = - \sum_{i=1}^n p(x_i) \log_a p(x_i),$$

где  $a$  – некоторое число;  $H(X)$  – есть энтропия системы  $X$ .

Основание логарифма  $a$  определяет единицу измерения энтропии. Например, энтропия измеряется в битах при  $a = 2$ , нитах при  $a = e$  и дитах при  $a = 10$ . При этом выбор единицы измерения непринципиален, поскольку величины будут отличаться только постоянным множителем:

$$\log_a b = \log_a c \times \log_c b = k \log_c b.$$

Для перехода от одной единицы измерения к другой можно воспользоваться следующими коэффициентами:

$$\begin{aligned} \log_e 2 &\approx 0,693, & \log_{10} 2 &\approx 0,301; \\ \log_e 10 &\approx 2,302, & \log_{10} e &\approx 0,434; \\ \log_2 e &\approx 1,441, & \log_2 10 &\approx 3,321. \end{aligned}$$

Таким образом,  $H_1(X) = kH_2(X)$ , где  $H_1(X)$ ,  $H_2(X)$  – значения энтропии, вычисленные с помощью различных единиц измерения. Очевидно, что на теоретические выводы выбор единицы измерения энтропии не влияет. Поэтому при дальнейшем изложении основание логарифма в некоторых случаях будет опускаться (будем предполагать, что основание равно 2).

Согласно формуле энтропии она представляет собой математическое ожидание случайной величины  $\log \frac{1}{p(x_i)} = -\log p(x_i)$ ,  $i = 1, 2, \dots, n$ :

$$H(X) = M \left[ \log \frac{1}{p(x_i)} \right] = M [-\log p(x_i)]$$

и энтропия имеет следующие свойства:

1) энтропия есть величина вещественная, ограниченная и неотрицательная:

$$H(X) \geq 0;$$

2) энтропия минимальна в случае, если одно из состояний системы достоверно известно (вероятность одного из состояний равно 1):

$$\min\{H(X)\} = 0;$$

3) энтропия максимальна в случае, если состояния системы равновероятны:

$$\max\{H(X)\} = \log n,$$

где  $n$  – число состояний системы.

Справедливость свойства 1) можно проиллюстрировать следующим образом. Так как вероятности  $p(x_1), p(x_2), \dots, p(x_n)$  есть вещественные величины, принимающие значения от 0 до 1, то значение функции

$$H(X) = \sum_{i=1}^n p(x_i) \log \frac{1}{p(x_i)}$$

также является вещественным. При этом  $p(x_i) \log \frac{1}{p(x_i)} \geq 0$ ,

$i = 1, 2, \dots, n$ , так как известно, что  $\lim_{x \rightarrow 0} \left( x \log \frac{1}{x} \right) = \lim_{x \rightarrow 1} \left( x \log \frac{1}{x} \right) = 0$ .

Это означает, что  $H(X) \geq 0$ , т.е. является неотрицательной величиной. Ограниченность энтропии следует из неравенства  $\ln x \leq x - 1$ :

$$\begin{aligned} H(X) &= \sum_{i=1}^n p(x_i) \log \frac{1}{p(x_i)} = k \sum_{i=1}^n p(x_i) \ln \frac{1}{p(x_i)} \leq \\ &\leq k \sum_{i=1}^n p(x_i) \left[ \frac{1}{p(x_i)} - 1 \right] = k \sum_{i=1}^n p(x_i) \frac{1}{p(x_i)} - k \sum_{i=1}^n p(x_i) = k(n-1), \end{aligned}$$

где  $k$  – константа.

Свойство 2) можно прокомментировать так. Если одно из состояний, например  $x_i$ , достоверно, т.е.  $p(x_i) = 1$ , то все остальные состояния будут иметь вероятности, равные 0, т.е.  $p(x_1) = \dots = p(x_{i-1}) = p(x_{i+1}) = \dots = p(x_n) = 0$ . Тогда в силу того, что  $\lim_{x \rightarrow 0} \left( x \log \frac{1}{x} \right) = \lim_{x \rightarrow 1} \left( x \log \frac{1}{x} \right) = 0$  и  $\lim_{x \rightarrow 0} \left( x \log \frac{1}{x} \right) = \lim_{x \rightarrow 1} \left( x \log \frac{1}{x} \right) = 0$  и энтропия есть величина неотрицательная, получаем минимальное значение  $H(X)$ , равное 0.

Доказательство свойства 3) также основано на неравенстве  $\ln x \leq x - 1$ , что эквивалентно  $k \log x \leq x - 1$ . С учетом данного неравенства имеем:

$$\begin{aligned} H(X) - \log n &= \sum_{i=1}^n p(x_i) \log \frac{1}{p(x_i)} - \sum_{i=1}^n p(x_i) \log n = \sum_{i=1}^n p(x_i) \log \frac{1}{p(x_i)n} \leq \\ &\leq k \left[ \sum_{i=1}^n p(x_i) \left( \frac{1}{p(x_i)n} - 1 \right) \right] = k \left[ \sum_{i=1}^n \frac{1}{n} - \sum_{i=1}^n p(x_i) \right] = k \left[ \frac{n}{n} - 1 \right] = 0. \end{aligned}$$

Тогда  $H(X) - \log n \leq 0$  и  $H(X) \leq \log n$ , т.е.  $\max \{H(X)\} = \log n$ . Очевидно, что указанное максимальное значение достигается при  $p(x_1) = \dots = p(x_n) = 1/n$ , т.е. в случае, когда состояния системы равновероятны.

Иногда к свойствам энтропии 1)–3) добавляют еще одно:

4) энтропия бинарной системы  $X = \{x_1, x_2\}$  изменяется от 0 до 1. Она равна 0, если вероятность одного из состояний равна 0, затем возрастает и достигает максимума при  $p(x_1) = p(x_2) = 0,5$ .

Это свойство является следствием свойств 1)–3). Его обычно рассматривают для того, чтобы графически проиллюстрировать свойства энтропии. В этом случае строят график функции:

$$H(p) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p},$$

где  $p$  – вероятность состояния  $x_1$ ;  $(1-p)$  – вероятность состояния  $x_2$ . Тогда  $H(p)$  есть энтропия бинарной случайной системы. На рис. 1.1 показаны значения  $H(p)$  в битах при изменении  $p$  от 0 до 1.

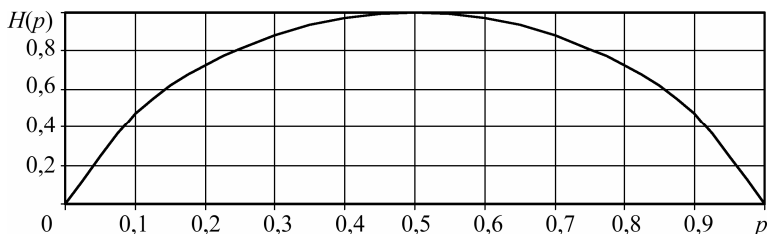


Рис. 1.1. Энтропия бинарной системы

Пусть имеется сложная система, состоящая из двух систем  $X$  и  $Y$ ,  $X = (x_1, \dots, x_i, \dots, x_n)$  и  $Y = (y_1, \dots, y_j, \dots, y_m)$ . Ее поведение определяется матрицей вероятностей совместных событий  $P(X, Y) = [p(x_i, y_j)]_{n \times m} = [p_{ij}]_{n \times m}$ :

$$P(X, Y) = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1m} \\ p_{21} & p_{22} & \dots & p_{2m} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{nm} \end{bmatrix}.$$

Энтропия сложной системы вычисляется по формуле:

$$H(X, Y) = M[-\log P(x, y)] = - \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(x_i, y_j).$$

В случае независимых систем  $X$  и  $Y$  для вероятностей совместных событий справедливы следующие выражения:

$$p(x_i, y_j) = p(x_i) p(y_j) \text{ и } \log p(x_i, y_j) = \log p(x_i) + \log p(y_j).$$

Тогда  $M[-\log p(x_i, y_j)] = M[-\log p(x_i) - \log p(y_j)] = M[-\log p(x_i)] + M[-\log p(y_j)]$  и  $H(X, Y) = H(X) + H(Y)$ , т.е. при объединении независимых систем их энтропии складываются.

В случае зависимых систем  $X$  и  $Y$  для вероятности совместных событий справедливо следующее равенство:

$$p(x_i, y_j) = p(x_i) p(y_j/x_i) = p(y_j) p(x_i/y_j),$$

где  $p(y_j/x_i)$  и  $p(x_i/y_j)$  – условные вероятности событий  $y_j$  и  $x_i$ , задающиеся матрицей  $P(X/Y)$ :

$$P(X/Y) = \begin{bmatrix} p(x_1/y_1) & p(x_1/y_2) & \dots & p(x_1/y_m) \\ p(x_2/y_1) & p(x_2/y_2) & \dots & p(x_2/y_m) \\ \dots & \dots & \dots & \dots \\ p(x_n/y_m) & p(x_n/y_m) & \dots & p(x_n/y_m) \end{bmatrix}.$$

Тогда энтропия системы  $Y$  относительно события  $x_i$  будет равна

$$H(Y/x_i) = M[-\log p(Y/x_i)] = -\sum_{j=1}^m p(y_j/x_i) \log p(y_j/x_i).$$

Энтропию  $H(Y/x_i)$  можно назвать  $i$ -й частной условной энтропией системы  $Y$  относительно системы  $X$ . Полную условную энтропию системы  $Y$  относительно системы  $X$  можно получить по формуле:

$$\begin{aligned} H(Y/X) &= \sum_{i=1}^n p(x_i) H(Y/x_i) = \sum_{i=1}^n p(x_i) \left[ -\sum_{j=1}^m p(y_j/x_i) \log p(y_j/x_i) \right] = \\ &= -\sum_{i=1}^n \sum_{j=1}^m p(x_i) p(y_j/x_i) \log p(y_j/x_i) = -\sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(y_j/x_i). \end{aligned}$$

Энтропия сложной системы в этом случае вычисляется следующим образом:



$$\begin{aligned}
H(X, Y) &= -\sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(x_i, y_j) = -\sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(x_i) - \\
&- \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(y_j / x_i) = -\sum_{i=1}^n p(x_i) \log p(x_i) \sum_{j=1}^m p(y_j / x_i) + H(Y / X) = \\
&= -\sum_{i=1}^n p(x_i) \log p(x_i) + H(Y / X) = H(X) + H(Y / X).
\end{aligned}$$

Очевидно, что справедливо также равенство:

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

Если системы  $X$  и  $Y$  независимы, то  $H(Y/X) = H(Y)$  и  $H(X/Y) = H(X)$ . Тогда  $H(X, Y) = H(X) + H(Y)$ . В общем случае  $H(X, Y) \leq H(X) + H(Y)$ . В случае полной зависимости систем  $H(X, Y) = H(X) = H(Y)$ , т.е.  $H(X/Y) = H(Y/X) = 0$ .

Очевидно, что если о некоторой системе все известно, то любое сообщение о ней не несет никакой информации. Например, сообщение о том, что Париж является столицей Франции, не несет никакой информации для грамотного человека. Некоторое количество информации можно получить только относительно системы, состояние которой неопределенно. И чем более неопределенным было состояние системы, тем большее количество информации будет получено. Поэтому количество информации о системе можно рассматривать как меру снятой неопределенности:

$$I(X) = H_1(X) - H_2(X),$$

где  $H_1(X)$  – энтропия системы  $X$  до получения информации;  $H_2(X)$  – энтропия системы  $X$  после получения информации.

Пусть энтропия некоторой системы  $X$  равна  $H(X)$ . Допустим, что в результате наблюдения этой системы о ней была получена полная информация, т.е. после окончания наблюдения энтропия системы равна 0, тогда полученное количество информации будет равно:

$$I(X) = H(X) - 0 = H(X).$$

Таким образом, количество информации, приобретаемое при полном выяснении состояния некоторой системы, равно энтропии этой системы. В случае если состояния системы обладают различными вероятностями, информация от разных сообщений неодинакова: наибольшую информацию несут сообщения о тех событиях, которые априори были наименее вероятны (например, сообщение о том, что 31 декабря в г. Казани выпал снег, несет гораздо меньше информации, чем аналогичное сообщение о том, что снег в г. Казани выпал 31 июля).

Поясним сказанное на следующем численном примере. Допустим, что нам интересно знать, сдал или не сдал экзамен некий студент Петров. Примем следующие вероятности этих двух событий:

$$p_1 = 7/8; \quad p_2 = 1/8.$$

Так как из распределения вероятностей видно, что этот студент является довольно сильным, то сообщение, что он сдал экзамен, несет мало информации:

$$I_1 = -\log_2 p_1 = 0,193 \text{ бит},$$

а вот сообщение о том, что Петров не сдал экзамен, несет гораздо больше информации:

$$I_2 = -\log_2 p_2 = 3 \text{ бита}.$$

Теперь более подробно рассмотрим понятие «количество информации». Будем полагать, что имеется источник информации о некоторой системе, при этом информация представляется в виде сообщения, состоящего из элементарных единиц – символов.

Для оценки количества переданной информации необходимо установить соответствующую меру, т.е. способ измерения информации. Очевидно, что количественная мера информации должна согласовываться с интуитивным представлением о содержании информации в сообщении. Так, чем длиннее телеграмма, тем больше информации она обычно содержит. Следовательно, мера информации должна монотонно возрастать с увеличением дли-

тельности сообщения, которую естественно измерять числом символов, содержащихся в нем. С учетом указанных соображений Р. Хартли в 1928 г. ввел меру количества информации, определяемую как

$$I = \log_2 N,$$

где  $N$  – количество различных сообщений. При этом Р. Хартли наложил ряд ограничений:

- 1) рассматриваются только дискретные сообщения;
- 2) множество различных сообщений конечно;
- 3) символы, составляющие сообщения, равновероятны и независимы.

Дальнейшее развитие данный подход получил в работах К. Шеннона. Он рассуждал следующим образом. В случае равной вероятности сообщений и независимости символов при любом числе символов в сообщении  $k$  все возможные сообщения оказываются также равновероятными, вероятность каждого из таких сообщений равна  $P = 1/N$ . Тогда количество информации можно выразить через вероятности появления сообщений  $I = -\log P$ . Пусть вероятность  $i$ -го символа равна  $p_i$  ( $i = 1, 2, \dots, n$ ). Символы образуют полную группу событий, т.е.  $\sum_{i=1}^n p_i = 1$ . Чтобы сообщения были

равновероятными, необходимо, чтобы относительные частоты появления отдельных символов во всех возможных сообщениях были равны. Это условие приближенно выполняется при достаточно длинных сообщениях. В силу статистической независимости символов вероятность сообщения длиной в  $k$  символов равна

$P = \prod_{i=1}^k p_i$ . Если  $i$ -й символ повторяется в данном сообщении  $k_i$  раз,

то  $P = \prod_{i=1}^n p_i^{k_i}$ . Так как при достаточно длинных сообщениях  $k_i \approx kp_i$ ,

то вероятность сообщений будет равна  $P = \prod_{i=1}^n p_i^{kp_i}$ . Тогда оконча-

тельно получим, что  $I = -\log P = -k \sum_{i=1}^n p_i \log p_i$ .

Таким образом, помимо длительности сообщений на содержание количества информации должны влиять и статистические характеристики, так как вероятности появления символов в сообщении могут быть различны.

Итак, количество информации по Шеннону выражается формулой

$$I = k \sum_{i=1}^n p_i \log \frac{1}{p_i},$$

где  $p_i$  – вероятность появления  $i$ -го символа (из множества  $n$  возможных символов) в сообщении;  $k$  – количество символов в сообщении.

Формула К. Шеннона для количества информации на отдельный символ сообщения совпадает с энтропией источника информации, т.е. системы, состояния которой соответствуют появлению символов в сообщении. Тогда количество информации сообщения, состоящего из  $k$  символов, определяется как:

$$I = kH,$$

т.е. при «статистическом» подходе К. Шеннона используется понятие энтропии как меры неопределенности, учитывающей вероятность появления и информативность того или иного сообщения.

Отметим, что количество информации по Хартли рассматривается как частный случай меры, введенной К. Шенноном. Действительно, в случае равновероятных символов  $p_i = 1/n$  ( $i = 1, \dots, n$ )

и  $I = k \sum_{i=1}^n \frac{1}{n} \log n = \log n^k$ . Величина  $n^k$  представляет собой число  $N$

возможных сообщений из  $k$  символов, т.е.  $I = \log_2 N$ .

Пусть имеются две системы  $X$  и  $Y$ . При этом система  $Y$  менее подробна по сравнению с системой  $X$  (некоторые состояния системы  $X$  не находят отображения в системе  $Y$ ) и имеются ошибки при передаче сообщений и неточности измерения параметров. Какое количество информации можно получить о системе  $X$ , наблюдая систему  $Y$ ?

Поясним вопрос. Понятно, что информацию о системе  $X$  можно получить, ведя наблюдение непосредственно за этой системой. Однако на практике часто бывает, что система  $X$  для наблюдения не доступна и тогда ведут наблюдение за другой системой  $Y$ , связанной с ней. Например, вместо непосредственного наблюдения за космическим кораблем (система  $X$ ) ведется наблюдение за системой сигналов, передаваемых его аппаратурой (система  $Y$ ). Или вместо наблюдения за футбольным матчем на стадионе (система  $X$ ), просматривается его запись по телевизору (система  $Y$ ). При этом очевидно между системами  $X$  и  $Y$  имеются различия.

Для определения того, какое количество информации о системе  $X$  дает наблюдение системы  $Y$ , используют следующее выражение:

$$I_{Y \rightarrow X} = H(X) - H(X/Y),$$

где  $H(X)$  – априорная энтропия (энтропия до наблюдения);  $H(X/Y)$  – апостериорная энтропия (энтропия после наблюдения). Величина  $I_{Y \rightarrow X}$  называется полной информацией о системе  $X$ , содержащейся в системе  $Y$ .

В общем случае, при наличии двух систем, каждая содержит относительно другой системы одну и ту же полную информацию:

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

тогда

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

т.е.  $I_{Y \rightarrow X} = I_{X \rightarrow Y} = I_{Y \leftrightarrow X}$ . Величина  $I_{Y \leftrightarrow X}$  называется полной взаимной информацией, содержащейся в системах  $X$  и  $Y$  относительно друг друга.

Если  $X$  и  $Y$  – независимые системы, то  $H(Y/X) = H(Y)$ ,  $H(X/Y) = H(X)$  и  $I_{Y \leftrightarrow X} = 0$ , т.е. нельзя получить сведения о системе, наблюдая вместо нее другую систему, никак не связанную с ней.

Если  $X$  и  $Y$  полностью определяют друг друга (информация о них совпадает), то  $H(X) = H(Y)$ ,  $H(X/Y) = H(Y/X) = 0$  и  $I_{Y \leftrightarrow X} = I_X = I_Y = H(X) = H(Y)$ .

Если между  $X$  и  $Y$  имеется жесткая односторонняя зависимость (состояние одной из систем полностью определяет состояние другой, но не наоборот), то по состоянию подчиненной системы вообще нельзя однозначно определить состояние связанной с ней системы. Пусть из двух систем  $X$  и  $Y$  подчиненной является  $Y$ . Тогда  $H(Y/X) = 0$  и  $I_{Y \leftrightarrow X} = H(Y)$ , т.е. полная взаимная информация, содержащаяся в системах, из которых одна является подчиненной, равна энтропии подчиненной системы.

Выведем выражение для информации  $I_{Y \rightarrow X}$  через энтропию объединенной системы  $H(X, Y)$  и энтропии отдельных ее частей  $H(X)$ ,  $H(Y)$ :

$$H(X, Y) = H(Y) + H(X/Y);$$

$$I_{Y \rightarrow X} = H(X) - H(X/Y) = H(X) + H(Y) - H(X, Y).$$

Полную взаимную информацию можно выразить также через вероятности состояний систем  $X$  и  $Y$ . Для этого запишем значения энтропии отдельных систем через математическое ожидание:

$$H(X) = M[-\log P(X)];$$

$$H(Y) = M[-\log P(Y)]; \quad H(X, Y) = M[-\log P(X, Y)],$$

тогда

$$I_{Y \rightarrow X} = M[-\log P(X) - \log P(Y) + \log P(X, Y)] = M \left[ \log \frac{P(X, Y)}{P(X)P(Y)} \right];$$

$$I_{Y \leftrightarrow X} = \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i) p(y_j)}.$$

Вычислить взаимную информацию можно также следующим образом:

$$I_{Y \leftrightarrow X} = \sum_{j=1}^m p(y_j) \sum_{i=1}^n p(x_i/y_j) \log \frac{p(x_i/y_j)}{p(x_i)}.$$

Тогда внутренняя сумма будет представлять частную информацию о системе  $X$ , получаемую с помощью отдельного события системы  $Y$ :

$$I_{y_j \rightarrow X} = \sum_{i=1}^n p(x_i/y_j) \log \frac{p(x_i/y_j)}{p(x_i)}.$$

Данное выражение представляет частную информацию «от события к системе». Также можно определить частную информацию о событии  $x_i$ , содержащуюся в событии  $y_j$  (информация «от события к событию»):

$$I_{y_j \rightarrow x_i} = \log \frac{p(x_i/y_j)}{p(x_i)} = \log \frac{p(x_i, y_j)}{p(x_i) p(y_j)}.$$

Информация «от события к событию» симметрична:

$$I_{y_j \rightarrow x_i} = I_{x_i \rightarrow y_j} = I_{y_j \leftrightarrow x_i}.$$

Ранее были рассмотрены информационные системы дискретных случайных систем. Как уже было сказано, помимо дискретных случайных систем, рассматриваются также непрерывные случайные системы, состояния которых описываются непрерывными параметрами, т.е. параметрами, значения которых принадлежат континуальному множеству (в интервале между любыми двумя значениями можно указать бесконечное множество значений).

В случае непрерывной случайной системы рассматривается непрерывная случайная величина, распределение вероятностей которой описывается с помощью плотности вероятности  $p(x)$ , где  $x$  определяет состояние системы. Пример плотности вероятности непрерывной случайной величины приведен на рис. 1.2.

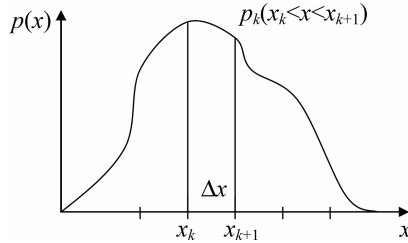


Рис. 1.2. Плотность вероятности случайной величины  $x$

Для перехода от дискретных систем к непрерывным произведем квантование значений случайной непрерывной величины  $x$  на счетное число  $n$  уровней с интервалом  $\Delta x$ . Полученная таким образом дискретная случайная величина  $x$  характеризуется распределением, в котором вероятность  $k$ -го состояния равна

$$p_k = \int_{x_k}^{x_k + \Delta x} p(x) dx, \text{ что приближенно можно представить как } p_k = p(x) \Delta x$$

(чем меньше  $\Delta x$ , тем более точной будет замена). Энтропию полученной дискретной системы можно определить следующим образом:

$$\begin{aligned} H(X) &= - \sum_{k=1}^n p(x_k) \Delta x \log \{ p(x_k) \Delta x \} = \\ &= - \sum_{k=1}^n p(x_k) \Delta x \log p(x_k) - \sum_{k=1}^n p(x_k) \Delta x \log \Delta x. \end{aligned}$$

С учетом того, что  $\lim_{\Delta x \rightarrow 0} \left\{ - \sum_{k=1}^n p(x_k) \Delta x \log p(x_k) \right\} =$

$$= - \int_{-\infty}^{\infty} p(x) \log p(x) dx \text{ и } \lim_{\Delta x \rightarrow 0} \left\{ - \sum_{k=1}^n p(x_k) \Delta x \log \Delta x \right\} = \lim_{\Delta x \rightarrow 0} \left\{ - \log \Delta x \sum_{k=1}^n p(x_k) \Delta x \right\} =$$



$$= \lim_{\Delta x \rightarrow 0} \left\{ - \sum_{k=1}^n p(x_k) \Delta x \log \Delta x \right\} = \lim_{\Delta x \rightarrow 0} \left\{ - \log \Delta x \sum_{k=1}^n p(x_k) \Delta x \right\} = - \log \Delta x \sum_{k=1}^n p_k = - \log \Delta x,$$

получим следующую формулу для энтропии непрерывной случайной системы:

$$H(X) = - \int_{-\infty}^{\infty} p(x) \log p(x) dx - \log \Delta x.$$

Если ввести обозначение  $H^*(X)$  для выражения  $-\int_{-\infty}^{\infty} p(x) \log p(x) dx$ , то энтропию можно представить в виде:

$$H(X) = H^*(X) - \log \Delta x.$$

Величину  $H^*(X)$  называют приведенной или дифференциальной энтропией.

Свойства информационных характеристик непрерывных случайных систем аналогичны свойствам соответствующих характеристик дискретных систем. Для энтропии объединения, например, также справедливы следующие выражения:

$$H(X, Y) = H(X) + H(Y/X) = H(Y) + H(X/Y);$$

$$H(X, Y) \leq H(X) + H(Y).$$

Однако здесь необходимо учесть особенности непрерывного распределения вероятностей (интегральные выражения):

$$H(X, Y) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \log p(x, y) dx dy - \log \Delta x - \log \Delta y;$$

$$H(Y / X) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \log p(y / x) dx dy - \log \Delta y;$$

$$H(X / Y) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \log p(x / y) dx dy - \log \Delta x.$$

В качестве еще одного примера рассмотрим взаимную информацию:

$$I_{X \leftrightarrow Y} = H(X) - H(X / Y) = H(Y) - H(Y / X) .$$

После подстановки соответствующих выражений получим:

$$I_{X \leftrightarrow Y} = - \int_{-\infty}^{\infty} p(x) \log p(x) dx - \log \Delta x + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \log p(x / y) dx dy + \log \Delta x ;$$

$$I_{X \leftrightarrow Y} = - \int_{-\infty}^{\infty} p(x) \log p(x) dx + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \log p(x / y) dx dy ,$$

теперь умножим первый интеграл на  $\int_{-\infty}^{\infty} p(y / x) dy = 1$  и учтем, что во втором интеграле  $p(x/y) = p(x, y)/p(y)$ . После этого окончательно получим следующее выражение для взаимной информации непрерывной случайной величины:

$$I_{X \leftrightarrow Y} = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(x, y) \log \frac{w(x, y)}{w(x)w(y)} dx dy .$$

## 1.2. Информационные характеристики каналов связи

Пусть имеется система передачи информации, состоящая из источника информации (ИИ), канала связи и приемника информации (ПИ) (рис. 1.3). Информация поступает в канал связи и затем в приемник информации в виде сообщений. При этом под сообщением будем понимать совокупность символов или первичных сигналов, содержащих информацию. Информация от источника информации, который представляет собой наблюдаемый объект, поступает в первичный преобразователь (датчик, человек-оператор и т.д.), воспринимающий информацию о протекающем в нем процессе или его состояниях. На выходе первичного преобразователя

как раз и формируются сообщения. Таким образом, источник информации и первичный преобразователь образуют источник сообщений.

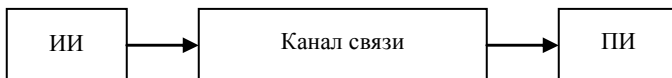


Рис. 1.3. Канал связи

С математической точки зрения под источником сообщений понимают множество возможных сообщений с заданной на этом множестве вероятностной мерой. Различают дискретные источники и непрерывные. Различие между ними в том, что символы в дискретном случае образуют счетное множество, а в непрерывном – множество континуума.

Дискретный источник определен, если перечислены все возможные символы, встречающиеся в сообщениях, и указаны их вероятности, т.е. задано множество:

$$X = \left\{ \langle x_i, p(x_i) \rangle; i = 1, 2, \dots, n; 0 \leq p(x_i) \leq 1; \sum_{i=1}^n p(x_i) = 1 \right\},$$

где  $x_1, x_2, \dots, x_n$  – символы источника сообщений;  $p(x_1), p(x_2), \dots, p(x_n)$  – их вероятности.

Энтропия дискретного источника сообщения (среднее количество информации), приходящаяся на один символ сообщения, задается формулой:

$$H(X) = - \sum_{i=1}^n p(x_i) \log p(x_i).$$

Если предположить, что вероятности символов равны, то данная формула принимает вид

$$H(X) = \log n.$$

Данное предположение является грубым, не учитывающим статистические взаимосвязи между символами. Модель, задавае-

мую таким образом, можно считать грубой моделью или моделью нулевого порядка. Тогда модель источника сообщений, определяемую формулой энтропии, можно считать моделью первого порядка. Если попытаться учесть статистические связи между двумя символами, т.е. условные вероятности  $p(x_j/x_i)$ , то можно построить модель второго порядка. Аналогично можно построить модели третьего, четвертого и других порядков.

Таким образом, получаем последовательность следующего вида:

$$H_0(X) = \log n;$$

$$H_1(X) = -\sum_{i=1}^n p(x_i) \log p(x_i);$$

$$H_2(X) = -\sum_{i=1}^n \sum_{j=1}^n p(x_i, x_j) \log p(x_j / x_i);$$

$$H_3(X) = -\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n p(x_i, x_j, x_k) \log p(x_k / x_i x_j), \dots$$

Так как модель более высокого порядка учитывает больше статистических связей между символами, то с ростом порядка модели значение энтропии уменьшается, т.е. справедливо неравенство:

$$H_0(X) \geq H_1(X) \geq H_2(X) \geq H_3(X) \geq \dots,$$

т.е. последовательность  $H_0(X)$ ,  $H_1(X)$ ,  $H_2(X)$ ,  $H_3(X)$  ... является монотонно убывающей. Поскольку энтропия ограничена снизу, то последовательность сходится к пределу:

$$H(X) = \lim_{m \rightarrow \infty} H_m(X) = H.$$

Например, если рассмотреть сообщения, представляющие собой слова на русском языке, то значение энтропии будет убывать в зависимости от порядка модели:  $H_0(X) = 5$  бит,  $H_1(X) = 4,42$  бит, ... Учитывая, что между буквами алфавита существуют взаимосвязи, например, в русском языке довольно часто встречаются сочетания *тся*, *ает*, *ций*, а сочетания *аь*, *иь* встретить невозможно, то моде-

ли более высоких порядков будут иметь все меньшее значение энтропии и в пределе стремиться к минимально возможному значению.

Так как энтропия характеризует среднее количество информации, приходящееся на один символ сообщения, то если источник сообщений выдает  $n$  символов в секунду, то скорость выдачи информации будет составлять:

$$R = nH.$$

Предположим, что необходимо передать сообщение с помощью наименьшего числа символов. Очевидно, что это возможно, если на каждый символ приходится максимальное количество информации, т.е. нужен источник сообщений, вырабатывающий символы, равномерно распределенные и статистически независимые. Назовем такой источник оптимальным, а его энтропию обозначим  $H_o$ . Реальные источники передают сообщения, состоящие из неравновероятных и статистически связанных символов. Поэтому энтропия реальных сообщений  $H_p$  оказывается меньше энтропии оптимальных сообщений  $H_o$ , а число символов для передачи одинакового количества информации – больше:

$$\begin{aligned} I &= n_o H_o = n_p H_p; \\ H_o &> H_p; \\ n_o &= \frac{I}{H_o} < n_p = \frac{I}{H_p}. \end{aligned}$$

Таким образом, часть символов  $n_p - n_o$  являются избыточными. Мера избыточности реальных сообщений по сравнению с оптимальными обозначается  $D$  (называется избыточность) и вычисляется по формуле

$$D = 1 - \frac{H_p}{H_o} = 1 - \frac{n_o}{n_p} = \frac{n_p - n_o}{n_p}.$$

Отметим, что наличие избыточности нельзя рассматривать как признак несовершенства источника сообщений. Избыточность

способствует повышению помехоустойчивости сообщений и точности их приема. Например, высокая избыточность естественных языков обеспечивает надежное общение между людьми.

Введение понятий энтропии, количества информации, скорости выдачи информации источником, избыточности позволяет характеризовать свойства систем передачи информации (при этом будем понимать в качестве систем передачи информации также системы ее обработки и хранения). Однако для их сравнения такого описания недостаточно, так как может интересовать не только передача определенного количества информации, но и передача его в возможно более короткий срок; не только хранение определенного количества информации, но также хранение его с помощью минимальной по объему аппаратуры и т.п.

Пусть количество информации, которое передается по каналу связи за время  $T$ , равно  $I_T = H_T(X) - H_T(X/Y)$ . Здесь под  $X$  понимается вход (сообщение на входе), а под  $Y$  – выход канала связи (сообщение на выходе). Если передача сообщения длится  $T$  единиц времени, то скорость передачи информации составит:

$$R = \frac{I_T}{T} = \frac{1}{T} [H_T(X) - H_T(X/Y)] = H(X) - H(X/Y).$$

Это количество информации, приходящееся в среднем на одно сообщение за единицу времени. Если в единицу времени передается  $n$  сообщений, то скорость передачи будет составлять  $R = n [H(X) - H(X/Y)]$ .

Пропускная способность канала связи есть максимально достижимая для него скорость передачи информации (или максимальное количество информации, передаваемое за единицу времени):

$$C = \max R = n[H(X) - H(X/Y)]_{\max} = n(I_{Y \rightarrow X})_{\max}.$$

Для упрощения записи далее вместо  $I_{Y \rightarrow X}$  будем писать  $I(X, Y)$ .

Пропускная способность является важнейшей характеристикой каналов связи, которая определяет, возможна ли передача без

задержек по каналу связи. Соответствующее условие формулируется в первой теореме Шеннона о кодировании (для каналов без помех).

**Теорема 1.1. Первая теорема Шеннона.** Пусть имеется источник информации  $X$  с энтропией  $H(X)$  и канал связи с пропускной способностью  $C$ . Если  $C \geq H(X)$ , то всегда можно закодировать достаточно длинное сообщение таким образом, что оно будет передано без задержек. Если же  $C < H(X)$ , то передача сообщений без задержек невозможна.

Ранее был рассмотрен канал связи без учета помех – идеальная модель. В отличие от нее в реальных каналах всегда присутствуют помехи. Однако если их уровень настолько мал, что вероятность искажения практически равна нулю, можно условно считать, что все сигналы передаются неискаженными. В этом случае все сказанное ранее остается справедливым. В противном случае необходимо использовать другие, более точные, модели.

Например, рассмотрим бинарный канал связи, пропускную способность которого нужно определить. В таком канале возможна передача только двух символов (двоичных сигналов). При этом с вероятностью  $p$  каждый из двоичных сигналов может перейти в противоположный сигнал (рис. 1.4). Такой канал связи называется симметричным бинарным каналом с помехами.

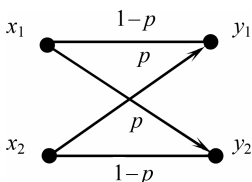


Рис. 1.4. Симметричный канал передачи сигналов в условиях помех:  
 $x_1, x_2$  – передаваемые двоичные сигналы;  $y_1, y_2$  – принимаемые двоичные сигналы;  
 $p$  – вероятность искажения сигнала;  $1 - p$  – вероятность неискаженной передачи

Матрица для нахождения условной вероятности имеет вид:

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

Найдем выражения для  $H(Y/X)$  и  $H(Y)$ , необходимые при определении пропускной способности канала связи:

$$\begin{aligned} H(Y / X) &= -\sum_{i=1}^2 p(x_i) \sum_{j=1}^2 p(y_j / x_i) \log p(y_j / x_i) = \\ &= -p(x_1)[(1-p)\log(1-p) + p\log p] - p(x_2)[p\log p + (1-p)\log(1-p)] = \\ &= -[p(x_1) + p(x_2)][p\log p + (1-p)\log(1-p)] = \\ &= -p\log p - (1-p)\log(1-p); \\ H(Y) &= \log 2 = 1. \end{aligned}$$

Тогда пропускная способность бинарного канала определяется по формуле:

$$C = n[1 + p\log p + (1-p)\log(1-p)].$$

Графически функция  $C = f(p)$  представлена на рис. 1.5. Наибольшее значение эта функция принимает при  $p = 0$  (при отсутствии помех) и при  $p = 1$  (при негативной передаче). При  $p = 1/2$  пропускная способность минимальна.

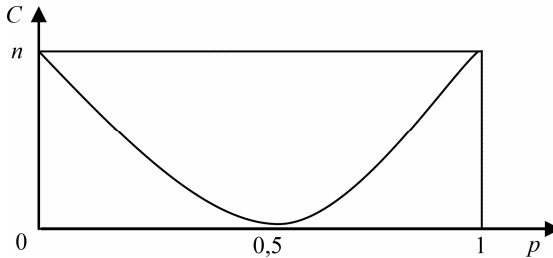


Рис. 1.5. График функции  $C = f(p)$

Подчеркнем, что при решении задачи использовалось равенство

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



т.е. вместо  $H(X)$  и  $H(X/Y)$  находились и применялись  $H(Y)$  и  $H(Y/X)$  (известно, что было послано по каналу связи и что при этом получено).

Теперь рассмотрим более общий случай. На рис. 1.6 представлена модель передачи информации по  $m$ -ичному каналу связи с помехами, где  $x_1, x_2, \dots, x_m$  – символы на входе;  $y_1, y_2, \dots, y_m$  – символы на выходе канала. Вероятность ошибки равна  $p$ , а вероятность безошибочной передачи сигналов равняется  $1 - p$ .

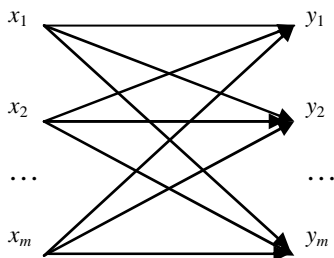


Рис. 1.6. Симметричный  $m$ -ичный канал связи с помехами

Переданный символ может с одинаковой вероятностью, равной  $\frac{p}{m-1}$ , быть воспринятым как любой из  $(m-1)$ -го отличных от него символов. Матрица для нахождения условной вероятности имеет вид:

$$P(Y / X) = \begin{bmatrix} 1-p & \frac{p}{m-1} & \dots & \frac{p}{m-1} \\ \frac{p}{m-1} & 1-p & \dots & \frac{p}{m-1} \\ \dots & \dots & \dots & \dots \\ \frac{p}{m-1} & \frac{p}{m-1} & \dots & 1-p \end{bmatrix}.$$

Получим выражения для энтропии  $H(Y/X)$  и  $H(Y)$ :

$$H(Y/X) = -(1-p)\log(1-p) - (m-1)\frac{p}{m-1}\log\frac{p}{m-1};$$

$$H(Y) = \log m.$$

Тогда пропускная способность канала связи определяется по формуле:  $C = n \left[ \log m + (1-p)\log(1-p) + (m-1)\frac{p}{m-1}\log\frac{p}{m-1} \right]$ .

График функции  $C = f(p)$  пропускной способности канала связи при  $m = 4$  представлен на рис. 1.7. Эта функция максимальна при  $p = 0$  и минимальна (равна 0) при  $p = \frac{m-1}{m} = 0.75$ . При  $p = 1$  пропускная способность равна  $C = n \log \frac{m}{m-1}$ .

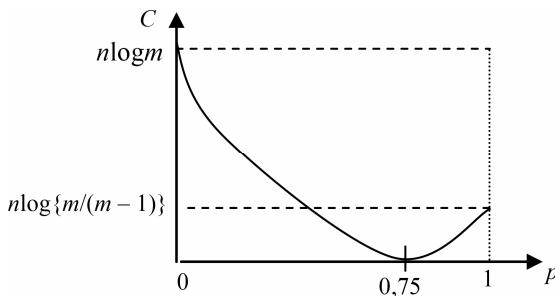


Рис. 1.7. График функции  $C=f(p)$

Условия передачи сообщений без искажений по каналу связи с помехами сформулированы К. Шенноном в его второй теореме о кодировании (для каналов с помехами).

**Теорема 1.2. Вторая теорема Шеннона.** Пусть имеется источник информации  $X$ , энтропия которого в единицу времени равна  $H(X)$ , и канал с пропускной способностью  $C$ . Если  $H(X) > C$ , то при любом кодировании передача сообщений без задержек и искажений невозможна. Если же  $H(X) \leq C$ , то любое достаточно длинное

сообщение можно всегда закодировать так, что оно будет передано без задержек и искажений с вероятностью, сколь угодно близкой к единице.

Каналы, используемые для передачи непрерывных сигналов, принято называть непрерывными. Реальные непрерывные каналы представляют собой сложные инерционные нелинейные объекты, характеристики которых случайным образом изменяются во времени. Для анализа таких каналов разработаны математические модели различных уровней сложности и степени адекватности реальным каналам. Наиболее широкое распространение получили модели, являющиеся разновидностями гауссова канала, под которым понимают математическую модель реального канала, построенную при следующих допущениях:

- 1) основные физические параметры канала являются известными детерминированными величинами;
- 2) полоса пропускания канала ограничена частотой  $F_k$ , Гц;
- 3) в канале действует аддитивный гауссовый белый шум – аддитивная флуктуационная помеха ограниченной мощности с равномерным частотным спектром и нормальным распределением амплитуд.

Предполагается также, что по каналу передаются сигналы с постоянной средней мощностью, статистические связи между сигналами и шумом отсутствуют, ширина спектра сигнала и помехи ограничены полосой пропускания канала.

Поясним ограничения, накладываемые на модель непрерывного канала связи. Первое из них достаточно очевидно – все параметры канала являются известными и неслучайными величинами.

Относительно второго ограничения можно сказать следующее. Для описания непрерывных сигналов используется математический аппарат, основанный на преобразовании Фурье, которое заключается в отображении сигнала, как функции времени, в функцию частоты и представляется следующей парой:

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt; \quad x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega)e^{j\omega t} d\omega,$$

где  $x(t)$  – функция, описывающая исходный сигнал;  $X(j\omega)$  – комплексная спектральная плотность или спектральная характеристика;  $j$  – мнимая единица;  $\omega$  – частота;  $t$  – время. Как комплексная величина спектральная характеристика может быть записана в виде

$$X(j\omega) = X(\omega)e^{-j\phi(\omega)},$$

где  $X(\omega) = |X(j\omega)|$  – спектральная плотность амплитуд или спектр сигнала.

С учетом того, что интеграл можно представить в виде суммы, а экспоненту с мнимой степенью – суммой гармонических функций, сигнал  $x(t)$  приближенно можно представить в виде суммы гармонических составляющих:

$$x(t) = \sum_{i=0}^{\infty} X(\omega_i) \cos(\omega_i t - \phi(\omega_i)).$$

Тогда второе ограничение на канал показывает, что гармонические составляющие с частотами, значения которых превышают  $2\pi F_k$ , будут искажены при прохождении через этот канал.

Отметим здесь же, что реальные сигналы являются ограниченными во времени. Это означает, что они имеют бесконечный спектр частот. Поэтому вводится некоторая частота  $F_c = \omega_c/2\pi$ , та-

кая, что  $|\hat{x}(t) - x(t)| \leq \varepsilon$ , где  $\hat{x}(t) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} X(j\omega)e^{j\omega t} d\omega$ ;  $\varepsilon$  – заданная

погрешность представления сигнала  $x(t)$ .

Третье ограничение говорит о том, что при прохождении через канал связи к сигналу  $x(t)$  добавляется (на него накладывается) помеха  $n(t)$ , представляющая сумму гармонических составляющих, амплитуды которых распределены по нормальному закону с нулевым средним. При этом все гармонические составляющие помехи имеют одинаковую мощность и любые две выборки помехи некор-

релированы между собой, как бы близко по времени они не располагались.

Непрерывные сигналы, имеющие спектр частот  $F_c$ , могут быть переданы в виде дискретных отсчетов через интервалы времени  $\Delta t = \frac{1}{2F_c}$  (по теореме Котельникова). Пусть в канале связи на передаваемое сообщение  $x(t)$  накладывается помеха  $n(t)$ , а длительность сообщения составляет  $T$ .

Количество информации, содержащееся в принятых сообщениях  $Y$  относительно переданных  $X$ , определяется равенством:

$$I(Y, X) = H(Y) - H(Y / X).$$

Значение  $H(Y/X)$  обусловлено только шумами и может быть заменено на энтропию шума  $H(N)$ . Тогда  $I(Y, X) = H(Y) - H(N)$ . При этом:

$$H(Y) = H(y_1, y_2, \dots, y_m); \quad H(N) = H(n_1, n_2, \dots, n_m),$$

где  $m = 2F_c T$ .

Скорость передачи информации будет равняться:

$$R = \lim_{T \rightarrow \infty} \frac{I(Y, X)}{T} = \lim_{T \rightarrow \infty} \frac{H(Y) - H(N)}{T}.$$

Максимальная скорость передачи информации называется пропускной способностью канала связи:

$$C = R_{\max} = \lim_{T \rightarrow \infty} \frac{I(Y, X)_{\max}}{T}.$$

Определим пропускную способность канала связи, когда помехи воздействуют на передаваемый сигнал по нормальному закону. Такие помехи обладают наибольшей эффективностью. Энтропия шума для одного отсчетного значения равна  $H(n) = -\log \sigma_n \sqrt{2\pi e} - \log \Delta x$ , где  $\sigma^2$  – дисперсия шума. Так как элементы независимы, то энтропия объединения для помехи равна сумме энтропии  $H(N) = 2F_c T H(n) = 2F_c T \left[ -\log \sigma_n \sqrt{2\pi e} - \log \Delta x \right]$ .

Если желательно передать наибольшее количество информации, то надо, чтобы энтропия объединения принятых сообщений была максимальной. Для этого необходимо, чтобы отсчеты принимаемого сигнала были статистически независимы и чтобы отсчетные значения были распределены по нормальному закону. В этом случае энтропия принимаемых сигналов будет равна:

$$H(Y)_{\max} = 2F_c T \left[ \log \sigma_y \sqrt{2\pi e} - \log \Delta y \right],$$

тогда

$$\begin{aligned} I(X, Y)_{\max} &= H(Y)_{\max} - H(N) = \\ 2F_c T \left[ \log \sigma_y \sqrt{2\pi e} - \log \Delta y - \log \sigma_n \sqrt{2\pi e} + \log \Delta x \right] &= \\ = 2F_c T \left[ \log \frac{\sigma_y}{\sigma_n} + \log \frac{\Delta x}{\Delta y} \right]. \end{aligned}$$

Если точности квантования  $\Delta x$  и  $\Delta y$  равны, то  $I(X, Y)_{\max} = 2F_c T \log \frac{\sigma_y}{\sigma_n}$ . Дисперсия принятых сообщений определяется как

сумма  $\sigma_y^2 = \sigma_x^2 + \sigma_n^2$ , тогда 
$$I(X, Y)_{\max} = 2F_c T \log \sqrt{\frac{\sigma_x^2 + \sigma_n^2}{\sigma_n^2}} =$$

$$= F_c T \log \frac{\sigma_x^2 + \sigma_n^2}{\sigma_n^2}.$$

Отношение дисперсии заменим отношением мощностей:

$$\frac{\sigma_x^2}{\sigma_n^2} = \frac{P}{N}.$$

Тогда получаем следующее выражение:

$$I(X, Y)_{\max} = F_c T \log \left( 1 + \frac{P}{N} \right),$$

где  $P$  и  $N$  – мощности сигнала и помехи соответственно. Таким образом, для увеличения  $I(X, Y)_{\max}$  необходимо увеличить  $F_c$ ,  $T$  и  $\frac{P}{N}$ .

Величину  $F_c T \log\left(\frac{P}{N}\right)$  называют «объемом сигнала». При сохранении объема сигнала можно передать одно и то же количество информации, используя различные  $F_c$ ,  $T$  и  $\frac{P}{N}$ .

С учетом сказанного определим пропускную способность непрерывного канала связи:

$$C = \lim_{T \rightarrow \infty} \frac{I(Y, X)_{\max}}{T} = F_c \log\left(1 + \frac{P}{N}\right).$$

Эта формула указывает, что наибольшая скорость передачи информации прямо пропорциональна полосе частот и соотношению между мощностью сигнала и мощностью помехи.

В заключение отметим, что для непрерывных каналов связи также справедливы теоремы Шеннона о кодировании (предполагается, что кодируются выборки непрерывного сигнала, взятые с интервалом дискретизации, величина которого не больше значения, определяемого теоремой Котельникова).

### 1.3. Кодирование информации

Ранее упоминались теоремы Шеннона о кодировании сообщений. Интуитивно понятно, что кодирование – это операция преобразования информации в форму, требуемую для последующей обработки (передачи по каналу связи, хранения в памяти вычислительной системы, использования для принятия решения и т.д.). Также понятно, что при построении любой информационной системы обойтись без кодирования невозможно: любое представление информации подразумевает использование каких-нибудь кодов.

Поэтому далее подробно разберем теоретические основы кодирования информации.

Пусть  $A$  – произвольный алфавит. Элементы алфавита  $A$  называют буквами (или символами), а конечные последовательности, составленные из букв, – словами в  $A$ . При этом считается, что в любом алфавите существует пустое слово, не содержащее букв.

Слово  $\alpha_1$  называют началом (префиксом) слова  $\alpha$ , если существует слово  $\alpha_2$ , такое, что  $\alpha = \alpha_1\alpha_2$ ; при этом слово  $\alpha_1$  называют собственным началом слова  $\alpha$ , если  $\alpha_2$  – не пустое слово. Длина слова – это число букв в слове (пустое слово имеет длину 0). Запись  $\alpha_1\alpha_2$  обозначает соединение (конкатенацию) слов  $\alpha_1$  и  $\alpha_2$ . Слово  $\alpha_2$  называют окончанием (суффиксом) слова  $\alpha$ , если существует слово  $\alpha_1$ , такое что  $\alpha = \alpha_1\alpha_2$ ; при этом слово  $\alpha_2$  называют собственным окончанием слова  $\alpha$ , если  $\alpha_1$  – не пустое слово. Пустое слово по определению считается началом и окончанием любого слова  $\alpha$ .

Рассмотрим алфавит  $B = \{0, 1, \dots, D - 1\}$ , где  $D \geq 2$ , и произвольное множество  $C$ . Произвольное отображение множества  $C$  во множество слов в алфавите  $B$  называют  $D$ -ичным кодированием множества  $C$  (при  $D = 2$  кодирование будет двоичным). Обратное отображение называют декодированием. Приведем примеры кодирований.

1. Кодирование множества натуральных чисел, при котором числу  $n = 0$  ставится в соответствие слово  $e(0) = 0$ , а числу  $n \geq 1$  двоичное слово

$$e(n) = b_1b_2 \dots b_{l(n)}$$

наименьшей длины, удовлетворяющее условию

$$\sum_{j=1}^{l(n)} b_j 2^{l(n)-j} = n.$$

Очевидно, что  $b_1 = 1$ ,  $2^{l(n)-1} \leq n < 2^{l(n)}$  и, следовательно

$$l(n) = [\log n] + 1 = \lceil \log(n + 1) \rceil,$$



где  $[x]$  и  $]x[$  обозначает соответственно наибольшее целое число, не превосходящее  $x$ , и наименьшее целое число, превосходящее  $x$ . Слово  $e(n)$  называют двоичной записью числа  $n$ , а данное кодирование – представление чисел в двоичной системе счисления. Данное кодирование является взаимно-однозначным, поскольку при  $n_1 \neq n_2$  слова  $e(n_1)$  и  $e(n_2)$  различны. В табл. 1.1 приведено представление первых 16 натуральных чисел в двоичной системе счисления.

Таблица 1.1

Кодирование $e(n)$							
$n$	$e(n)$	$n$	$e(n)$	$n$	$e(n)$	$n$	$e(n)$
0	0	4	100	8	1000	12	1100
1	1	5	101	9	1001	13	1101
2	10	6	110	10	1010	14	1110
3	11	7	111	11	1011	15	1111

2. Кодирование первых  $2^k$  натуральных чисел, при котором каждому числу  $n$  ( $0 \leq n < 2^k$ ) ставится в соответствие слово:

$$e_k(n) = 0^{k-l(n)} e(n),$$

где запись  $0^{k-l(n)}$  обозначает слово, состоящее из  $k - l(n)$  нулей;  $e(n)$  – рассмотренное ранее представление числа  $n$  в двоичной системе счисления. Данное кодирование для первых 16 натуральных чисел ( $k = 4$ ) приведено в табл. 1.2.

Таблица 1.2

Кодирование $e_k(n)$							
$n$	$e_k(n)$	$n$	$e_k(n)$	$n$	$e_k(n)$	$n$	$e_k(n)$
0	0000	4	0100	8	1000	12	1100
1	0001	5	0101	9	1001	13	1101
2	0010	6	0110	10	1010	14	1110
3	0011	7	0111	11	1011	15	1111

Пусть  $A = \{a_i, i = 1, 2, \dots\}$  – конечный или счетный алфавит, буквы которого занумерованы натуральными числами. В этом случае кодирование букв алфавита  $A$  можно задать последовательно-

стью  $D$ -ичных слов  $V = \{v_i, i = 1, 2, \dots\}$ , где  $v_i$  – образ буквы  $a_i$ . Такие последовательности слов (из множества  $V$ ) называют кодами (алфавита  $A$ ). Если задан код  $V$  алфавита  $A$ , то кодирование слов, при котором каждому слову  $a_{i1}a_{i2} \dots a_{ik}$  ставится в соответствие слово  $v_{i1}v_{i2} \dots v_{ik}$ , называют побуквенным кодированием.

При переходе от взаимно-однозначного кодирования букв алфавита к побуквенному кодированию слов в алфавите свойство взаимной однозначности может не сохраниться. Например, кодирование  $e(n)$  не сохраняет данного свойства, а кодирование  $e_k(n)$  его сохраняет. Свойство взаимной однозначности сохраняют разделимые коды.

Код  $V = \{v_i, i = 1, 2, \dots\}$  называют разделимым, если из каждого равенства вида:

$$v_{i1}v_{i2} \dots v_{ik} = v_{j1}v_{j2} \dots v_{jl}$$

следует, что  $l = k$  и  $v_{i1} = v_{j1}, v_{i2} = v_{j2}, \dots, v_{ik} = v_{jl}$ . Разделимые коды называют также однозначно декодируемыми кодами.

К классу разделимых кодов принадлежат префиксные коды. Код  $V = \{v_i, i = 1, 2, \dots\}$  называют префиксным, если никакое слово  $v_k$  не является началом (префиксом) никакого слова  $v_l, l \neq k$ . Если каждое слово префиксного кода заменить наименьшим его началом, которое не является началом других кодовых слов, то полученный код также будет префиксным. Такую операцию называют усечением префиксного кода.

Для произвольного кода  $V$ , состоящего из различных слов, можно построить кодовое дерево. Это ориентированный граф, не содержащий циклов, в котором вершина  $\beta_1$  соединена с вершиной  $\beta_2$  ребром, направленным от  $\beta_1$  к  $\beta_2$ , тогда и только тогда, когда  $\beta_2 = \beta_1 b$ , где  $b \in B = \{0, 1, \dots, D - 1\}, D \geq 2$ . Для префиксных кодов (и только для них) множество кодовых слов совпадает с множеством конечных вершин (вершин, из которых не исходят ребра) кодового дерева.

Свойства кодов, полезные для их практического применения, определяются основными теоремами кодирования.

**Теорема 1.3. Неравенство Крафта.** Для существования однозначно декодируемого (разделимого) кода, содержащего  $N$  кодовых слов в множестве  $\{0, 1, D - 1\}$  с длинами  $n_1, n_2, \dots, n_N$ , необходимо и достаточно, чтобы выполнялось неравенство

$$\sum_{i=1}^N D^{-n_i} \leq 1.$$

*Доказательство.* Представим, что имеется кодовое дерево для префиксного кода. Корень кодового дерева образует уровень 0, вершины, связанные с корнем, – уровень 1 и т.д. Возможное количество вершин на  $k$ -м уровне обозначим как  $D^k$ . Каждая вершина  $k$ -го уровня порождает точно  $D^{n-k}$  вершин  $n$ -го уровня.

Далее для простоты упорядочим длины кодовых слов:

$$n_1 \leq n_2 \leq \dots \leq n_N = n.$$

Очевидно, что кодовое слово длины  $k$  запрещает в точности  $D^{n-k}$  возможных конечных вершин (вершин последнего уровня).

Тогда все кодовые слова префиксного кода запрещают  $\sum_{i=1}^N D^{n-n_i}$  конечных вершин. Так как общее число конечных вершин равно  $D^n$ , то справедливо неравенство

$$\sum_{i=1}^N D^{n-n_i} \leq D^n,$$

из которого следует, что:

$$\sum_{i=1}^N D^n D^{-n_i} \leq D^n; \quad D^n \sum_{i=1}^N D^{-n_i} \leq D^n; \quad \sum_{i=1}^N D^{-n_i} \leq 1.$$

Таким образом, неравенство Крафта доказано.

В результате доказательства теоремы 1.3 делается вывод о том, что существуют хотя бы префиксные коды, которые являются однозначно декодируемыми кодами, с длинами кодовых слов

$n_1, n_2, \dots, n_N$ , удовлетворяющими неравенству Крафта. Следующая теорема, называемая утверждением Мак-Миллана, обобщает данный вывод на все однозначно декодируемые коды.

**Теорема 1.4. Неравенство Мак-Миллана.** Каждый однозначно декодируемый код удовлетворяет неравенству Крафта.

*Доказательство.* Возведем сумму  $\sum_{i=1}^N D^{-n_i}$  в степень  $L$ :

$$\left[ \sum_{i=1}^N D^{-n_i} \right]^L = \sum_{i_1=1}^N \sum_{i_2=1}^N \dots \sum_{i_L=1}^N D^{-[n_{i_1} + n_{i_2} + \dots + n_{i_L}]} . \quad (1.1)$$

Пусть  $A_k$  – число комбинаций, содержащих  $L$  кодовых слов с суммарной длиной  $k$ . Тогда выражение (1.1) можно представить в виде

$$\left[ \sum_{i=1}^N D^{-n_i} \right]^L = \sum_{k=1}^{L_{\max}} A_k D^{-k} ,$$

где  $L_{\max}$  – максимальная длина сообщения, содержащего  $L$  кодовых слов. Если код является однозначно декодируемым, то все последовательности из  $L$  кодовых слов суммарной длины  $k$  различны. Так как имеется всего  $D^k$  возможных последовательностей, то  $A_k \leq D^k$  и тогда:

$$\begin{aligned} \left[ \sum_{i=1}^N D^{-n_i} \right]^L &= \sum_{k=1}^{L_{\max}} A_k D^{-k} \leq \sum_{k=1}^{L_{\max}} D^k D^{-k} = \sum_{k=1}^{L_{\max}} 1 = L_{\max} ; \\ \sum_{i=1}^N D^{-n_i} &\leq [L_{\max}]^{1/L} . \end{aligned}$$

Так как  $L$  – это число независимых кодовых слов, которые используются для построения всех возможных последовательностей длины, не превышающей  $L_{\max}$ . Поэтому  $L \leq L_{\max}$  и  $\lim_{L \rightarrow \infty} [L_{\max}]^{1/L} = \lim_{L \rightarrow \infty} L^{1/L} = 1$ , из чего следует, что

$$\sum_{i=1}^N D^{-n_i} \leq 1 .$$

Поскольку приведенные рассуждения справедливы для каждого однозначно декодируемого кода, а не только для префиксных кодов, то утверждение Мак-Миллана доказано.

Следующие теоремы связывают энтропию источника сообщений и среднюю длину кодового слова.

**Теорема 1.5. Теорема кодирования источников I.** Для любого дискретного источника без памяти  $X$  с конечным алфавитом и энтропией  $H(X)$  существует  $D$ -ичный префиксный код, в котором средняя длина кодового слова  $\bar{n}$  удовлетворяет неравенству

$$\frac{H(X)}{\log D} \leq \bar{n} \leq \frac{H(X)}{\log D} + 1. \quad (1.2)$$

*Доказательство.* Прежде всего, поясним, что дискретный источник без памяти, описывается моделью, в которой не учитываются связи между символами сообщения. Теперь докажем левую часть неравенства (1.2):

$$H(X) - \bar{n} \log D \leq 0.$$

Для этого используем определение энтропии и неравенство Крафта:

$$\begin{aligned} H(X) - \bar{n} \log D &= \sum_{i=1}^N p_i \log \frac{1}{p_i} - \sum_{i=1}^N p_i n_i \log D = \\ &= \sum_{i=1}^N p_i \log \frac{D^{-n_i}}{p_i} \leq \log e \left( \sum_{i=1}^N p_i \left[ \frac{D^{-n_i}}{p_i} - 1 \right] \right) = \log e \left( \sum_{i=1}^N D^{-n_i} - \sum_{i=1}^N p_i \right) \leq 0. \end{aligned}$$

Для доказательства правой части неравенства (1.2) перепишем неравенство Крафта в следующем виде:

$$\sum_{i=1}^N D^{-n_i} \leq \sum_{i=1}^N p_i.$$

Затем выберем для каждого слагаемого такое наименьшее целое  $n_i$ , при котором

$$D^{-n_i} \leq p_i.$$

Итак, неравенство Крафта при таком выборе сохраняется, значит можно построить соответствующий префиксный код. Так как  $n_i$  – наименьшее целое, то для  $n_i - 1$  справедливо

$$p_i < D^{-(n_i-1)}.$$

Тогда

$$p_i \log p_i < p_i \log D^{-(n_i-1)} = p_i (-n_i + 1) \log D;$$

$$\sum_{i=1}^N p_i \log p_i < \log D \sum_{i=1}^N p_i (-n_i) + \log D \sum_{i=1}^N p_i;$$

$$-H(X) < -\log D\bar{n} + \log D; \quad \log D\bar{n} < H(X) + \log D; \quad \bar{n} < \frac{H(X)}{\log D} + 1.$$

Таким образом, теорема кодирования источников I доказана. Она определяет, что средняя длина кодового слова не может быть меньше энтропии источника сообщений. Отметим, что при доказательстве теоремы использовались те же обозначения, что и при рассмотрении неравенства Крафта.

**Теорема 1.6. Теорема кодирования источников II.** Для блока длины  $L$  существует  $D$ -ичный префиксный код, в котором средняя длина кодового слова на один символ  $\bar{n}$  удовлетворяет неравенству

$$\frac{H_L(X)}{\log D} \leq \bar{n} \leq \frac{H_L(X)}{\log D} + \frac{1}{L},$$

где  $H_L(X) = \frac{1}{L} H(X_1, X_2, \dots, X_L)$ .

*Доказательство.* Здесь в качестве единиц сообщений рассматриваются блоки символов и  $H(X_1, X_2, \dots, X_L)$  – энтропия источника сообщений, приходящаяся на блок из  $L$  символов. Для доказательства теоремы можно воспользоваться теоремой о кодировании источников I:

$$\frac{H(X_1, X_2, \dots, X_L)}{\log D} \leq \bar{n}_L \leq \frac{H(X_1, X_2, \dots, X_L)}{\log D} + 1;$$

$$\frac{LH_L(X)}{\log D} \leq L\bar{n} \leq \frac{LH_L(X)}{\log D} + 1;$$

$$\frac{H_L(X)}{\log D} \leq \bar{n} \leq \frac{H_L(X)}{\log D} + \frac{1}{L}.$$

Теорема о кодировании источников II позволяет утверждать, что существуют такие способы кодирования для достаточно длинного сообщения, что средняя длина кодового слова может быть сделана сколь угодно близкой к величине  $\frac{H(X)}{\log D}$ . Действительно, при  $L \rightarrow \infty$ ,  $H_L(X) \rightarrow H$ , где  $H$  – энтропия источника сообщений на один символ, справедливо неравенство

$$\frac{H}{\log D} \leq \bar{n} \leq \frac{H}{\log D} + \varepsilon,$$

где  $\varepsilon = \frac{1}{L} \rightarrow 0$ . Это можно интерпретировать также следующим образом: для любого сколь угодно малого числа  $\varepsilon$  существует метод кодирования блоков, содержащих  $L > \frac{1}{\varepsilon}$  символов, при котором для средней длины кодового слова на символ  $\bar{n}$  выполняется указанное неравенство.

Кроме того, так как минимально достижимой длиной кодового слова на символ является величина  $\frac{H}{\log D}$ , при  $D = 2$  избыточ-

ность кода можно определить по формуле  $\frac{\bar{n} - H}{\bar{n}} = 1 - \frac{H}{\bar{n}}$ .

Задача построения оптимального кода заключается в отыскании целых положительных чисел  $n_1, n_2, \dots, n_N$ , минимизирующих

среднюю длину кодового слова при условии выполнения неравенства Крафта:

$$\bar{n} = \sum_{i=1}^N p_i n_i \rightarrow \min; \quad \sum_{i=1}^N D^{-n_i} \leq 1.$$

При построении кодов в случае алфавита  $A = \{a_i, i = 1, 2, \dots, N\}$  с известным распределением вероятностей  $P = \{p_i, i = 1, 2, \dots, N\}$  без ограничения общности можно считать, что буквы алфавита  $A$  занумерованы в порядке убывания их вероятностей, т.е.  $p_1 \geq p_2 \geq \dots \geq p_N$ . Кроме того, будем рассматривать только двоичные коды.

Известны два метода (Фано и Шеннона) построения кодов, близких к оптимальным. Метод Фано заключается в следующем. Упорядоченный в порядке убывания вероятностей список букв делится на две последовательные части так, чтобы суммы вероятностей входящих в них букв как можно меньше отличались друг от друга. Буквам из первой части приписывается символ 0, а буквам из второй части – символ 1. Далее точно также поступают с каждой из полученных частей, если она содержит, по крайней мере, две буквы. Процесс продолжается до тех пор, пока весь список не разобьется на части, содержащие по одной букве. Каждой букве ставится в соответствие последовательность символов, приписанных в результате этого процесса данной букве. Легко видеть, что полученный код является префиксным.

Метод Шеннона применим лишь в том случае, когда все вероятности положительны. Он состоит в том, что букве  $a_i$ , имеющей вероятность  $p_i > 0$ , ставится в соответствие последовательность из  $n_i = \lceil \log(1/p_i) \rceil$  первых после дробной точки цифр разложения числа

$q_i = \sum_{j=1}^{i-1} p_j$  в бесконечную дробь (для  $a_1$  полагаем, что  $q_1 = 0$ ). По-

скольку при  $l > k$  (в силу того, что  $p_l \leq p_k$ )  $n_l \geq n_k$  и  $1 > q_l \geq q_k + p_k \geq q_k + 2^{-n_k}$ , то полученный таким образом код является префиксным. На основе полученного префиксного кода



строится усеченный префиксный код, который и является результатом кодирования по методу Шеннона.

Пусть, например, имеется множество букв  $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$  с распределением вероятностей  $P = \{0,2, 0,2, 0,19, 0,12, 0,11, 0,09, 0,09\}$ . Выполним кодирование букв по методу Фано.

1. Разобьем список на две части так, чтобы суммы вероятностей входящих в них букв как можно меньше отличались друг от друга:

$$A_1 = \{a_1, a_2, a_3\}, P_1 = \{0,2, 0,2, 0,19\};$$

$$A_2 = \{a_4, a_5, a_6, a_7\}, P_2 = \{0,12, 0,11, 0,09, 0,09\}.$$

2. Припишем буквам из первой части символ 0, а буквам второй части символ 1:

$$A_1 = \{a_1/0, a_2/0, a_3/0\};$$

$$A_2 = \{a_4/1, a_5/1, a_6/1, a_7/1\}.$$

3. Повторим последовательно указанные действия для каждой из частей по отдельности. В результате получим:

$$A_{11} = \{a_1/00\};$$

$$A_{121} = \{a_2/010\};$$

$$A_{122} = \{a_3/011\};$$

$$A_{211} = \{a_4/100\};$$

$$A_{212} = \{a_5/101\};$$

$$A_{221} = \{a_6/110\};$$

$$A_{222} = \{a_7/111\}.$$

Кодовые слова, полученные в результате кодирования, приведены для каждой буквы справа от наклонной черты. При этом порядок индексов полученных однобуквенных списков показывает последовательность разбиения исходного списка групп на части.

Процесс кодирования по методу Фано удобно оформлять в виде таблицы. Для рассматриваемого примера он приведен в табл. 1.3.

Таблица 1.3

**Кодирование по методу Фано**

$a_1$	0,20	0	0		00
$a_2$	0,20		1	0	010
$a_3$	0,19			1	011
$a_4$	0,12	1	0	0	100
$a_5$	0,11			1	101
$a_6$	0,09		1	0	110
$a_7$	0,09			1	111

Определим среднюю длину кодового слова:

$$\bar{n} = \sum_{i=1}^N p_i n_i \approx 2,8.$$

Теперь выполним кодирование по методу Шеннона. Процесс кодирования приведен в табл. 1.4.

Таблица 1.4

**Кодирование по методу Шеннона**

$a_i$	$n_i$	$q_i$	Код $a_i$	Усеченный код $a_i$
$a_1$	$]2,321 \dots[ = 3$	0	000	000
$a_2$	$]2,321 \dots[ = 3$	0,2	001	001
$a_3$	$]2,395 \dots[ = 3$	0,4	011	01
$a_4$	$]3,058 \dots[ = 4$	0,59	1001	100
$a_5$	$]3,183 \dots[ = 4$	0,71	1011	101
$a_6$	$]3,472 \dots[ = 4$	0,82	1101	110
$a_7$	$]3,472 \dots[ = 4$	0,91	1110	111

Как и для предыдущего случая, найдем среднюю длину кодового слова:

$$\bar{n} = \sum_{i=1}^N p_i n_i \approx 2,81.$$

Как можно видеть, результаты кодирования по методам Фано и Шеннона, с точки зрения минимизации средней длины кода, практически совпали. Поэтому часто эти методы рассматривают как один (в формулировке Фано) и называют методом Шеннона – Фано.

В 1952 г. Давид Хаффмен предложил метод оптимального префиксного кодирования для дискретных источников, который в отличие от методов Шеннона и Фано до сих пор применяется на практике. Д. Хаффмен доказал, что средняя длина кодового слова, полученная с помощью его метода, будет минимальна. Кодирование Хаффмена производится за три шага:

1) упорядочение: буквы располагаются в порядке убывания их вероятностей;

2) редукция: две буквы с наименьшими вероятностями объединяются в одну с суммарной вероятностью; список букв переупорядочивается в соответствии с шагом 1; процесс продолжается до тех пор, пока все буквы не будут объединены в одну. При этом можно добиться выравнивания длин кодовых слов с помощью следующей стратегии: если несколько букв имеют одинаковые вероятности, то объединяют те две из них, которые до этого имели наименьшее число объединений (правда, на среднюю длину кода это не повлияет);

3) кодирование: начиная с последнего объединения, последовательно приписываются одной компоненте составной буквы символ 0, а второй – символ 1; процесс продолжается до тех пор, пока все исходные буквы не будут закодированы.

Выполним кодирование по методу Хаффмена для множества, рассматривавшегося в примерах применения методов Фано и Шеннона.

1. Исходный список букв  $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$  уже упорядочен, так как  $P = \{0,2; 0,2; 0,19; 0,12; 0,11; 0,09; 0,09\}$ .

2. Объединим буквы  $a_6$  и  $a_7$  в одну букву  $a^1$  с вероятностью 0.18 и переупорядочим список:

$$P^1 = \{0,2; 0,2; 0,19; 0,18; 0,12; 0,11\}; \quad A^1 = \{a_1, a_2, a_3, a^1, a_4, a_5\}.$$

3. Повторим шаг 2 до тех пор, пока не останется одна буква в списке:

$$P^2 = \{0,23; 0,2; 0,2; 0,19; 0,18\}, \quad A^2 = \{a^2, a_1, a_2, a_3, a^1\};$$

$$P^3 = \{0,37, 0,23, 0,2, 0,2\}, \quad A^3 = \{a^3, a^2, a_1, a_2\};$$

$$P^4 = \{0,4; 0,37; 0,23\}, \quad A^4 = \{a^4, a^3, a^2\};$$

$$P^5 = \{0,6; 0,4\}, \quad A^5 = \{a^5, a^4\};$$

$$P^6 = \{1\}, \quad A^6 = \{a^6\}.$$

4. Присвоим двоичные коды символам:

$$a^6: a^5 = 0, \quad a^4 = 1;$$

$$a^5: a^3 = 00, \quad a^2 = 01;$$

$$a^4: a_1 = 10, \quad a_2 = 11;$$

$$a^3: a_3 = 000, \quad a^1 = 001;$$

$$a^2: a_4 = 010, \quad a_5 = 011;$$

$$a^1: a_6 = 0010, \quad a_7 = 0011.$$

Таким образом, исходным буквам присвоены следующие двоичные коды:  $a_1 = 10$ ;  $a_2 = 11$ ;  $a_3 = 000$ ;  $a_4 = 010$ ;  $a_5 = 011$ ;  $a_6 = 0010$ ;

$a_7 = 0011$ , что дает среднюю длину кода  $\bar{n} = \sum_{i=1}^N p_i n_i \approx 2,78$ , мень-

шую, чем в случае кодирования Фано и Шеннона.

Определим избыточность полученных кодов. Для этого найдем энтропию источника сообщений:

$$H(X) = \sum_{i=1}^N p_i \log \frac{1}{p_i} \approx 2,73.$$

Тогда коды имеют следующую избыточность:

$$\text{код Фано: } 1 - \frac{H}{\bar{n}} \approx 1 - \frac{2,73}{2,8} \approx 0,025;$$

$$\text{код Шеннона: } 1 - \frac{H}{\bar{n}} \approx 1 - \frac{2,73}{2,81} \approx 0,028;$$

$$\text{код Хаффмена: } 1 - \frac{H}{\bar{n}} \approx 1 - \frac{2,73}{2,78} \approx 0,018.$$

Таким образом, избыточность кода Хаффмена минимальна.

Для снижения избыточности, т.е. снижения средней длины кодового слова на один символ, можно воспользоваться блочным кодированием, обоснование которого дано в теореме кодирования источников II. В этом случае необходимо получить всевозможные группы букв заданной длины, найти вероятности групп, как вероятности совместного появления букв группы одновременно, и выполнить кодирование, рассматривая группы как символы нового алфавита.

### **1.4. Контрольные вопросы**

1. Дискретная случайная система.
2. Непрерывная случайная система.
3. Энтропия.
4. Источник информации.
5. Количество информации.
6. Взаимная информация.
7. Модель канала связи.
8. Пропускная способность канала связи.
9. Теоремы Шеннона о пропускной способности каналов связи.
10. Пропускная способность симметричного канала связи с помехами.
11. Кодирование.
12. Префиксный код.
13. Неравенство Крафта.
14. Задача оптимального кодирования.
15. Теоремы о кодировании источника без памяти.
16. Групповое кодирование.
17. Методы Хаффмена.
18. Метод Шеннона – Фано.

## 2. МЕТОДЫ РЕШЕНИЯ ПРАКТИЧЕСКИХ ЗАДАЧ КОДИРОВАНИЯ ИНФОРМАЦИИ

### 2.1. Сжатие информации

#### 2.1.1. Основные понятия сжатия информации

При проектировании информационных систем часто необходимо минимизировать затраты на передачу и хранение информации. Добиться этого можно с помощью кодирования, устраняющего избыточность информации. В решении практических задач такое кодирование называют сжатием информации. При сжатии информации, изначально представленной в дискретном виде, говорят о сжатии данных, о том, что сжимается файл или поток данных. Для описания методов сжатия данных используются следующие основные понятия и характеристики.

*Компрессор* (или кодер) – программа (устройство), которая сжимает исходные данные, т.е. преобразует входной несжатый файл в выходной сжатый файл. Программа (устройство), выполняющая обратное преобразование (восстановление исходных данных из сжатого файла), называется *декомпрессором* (или декодером). Компрессор и декомпрессор вместе образуют *кодек*.

Методы сжатия данных делятся на неадаптивные, адаптивные, полуадаптивные и локально адаптивные методы. Неадаптивный метод сжатия данных (метод неадаптивного сжатия данных) – метод, в котором не предусмотрена возможность изменения опера-

ций, параметров и настроек в зависимости от характера сжимаемой информации.

Метод, в котором предусмотрена возможность изменения операций, параметров и настроек в зависимости от характера сжимаемых данных, называется адаптивным методом. Если для изменения операций, параметров и настроек предварительно собирается некоторая статистика сжимаемой информации, то метод сжатия называется полуадаптивным (говорят, что реализуется двухпроходное сжатие: на первом проходе выполняется анализ информации, а на втором – собственно сжатие). Метод сжатия называется локально адаптивным, если в нем предусмотрена возможность изменения параметров в зависимости от локальных особенностей входного файла.

Методы сжатия данных делятся также на методы сжатия без потерь и с потерями информации. Метод сжатия без потерь (метод неискажающего сжатия) позволяет восстановить сжатую информацию без искажений. Метод сжатия с потерями (метод искажающего сжатия) предусматривает искажение сжимаемой информации для получения требуемых характеристик сжатия (скорости, качества, простоты и т.д.).

Симметричный метод сжатия – это метод, при использовании которого кодер и декодер выполняют одни и те же действия, но в противоположных направлениях. Если либо кодер, либо декодер выполняет значительно большую работу, то соответствующий метод сжатия называется асимметричным.

Для определения производительности метода сжатия данных часто используют коэффициент сжатия, фактор сжатия, качество сжатия, время сжатия и время восстановления. Коэффициент сжатия – величина, получающаяся в результате деления размера выходного (сжатого) файла на размер входного (несжатого) файла. Фактор сжатия – величина, обратная коэффициенту сжатия. Качество сжатия определяется по формуле  $100 \times (1 - K)$ , выражается в процентах и показывает, на сколько процентов уменьшается раз-

мер исходного файла после сжатия. Время сжатия – это время, затрачиваемое на сжатие данных, время восстановления – это время, необходимое для восстановления данных из сжатого файла.

### ***2.1.2. Энтропийные методы сжатия***

К энтропийным методам сжатия относятся методы, в которых средняя длина кода приближается к значению энтропии источника. Из подобных методов на практике часто применяются сжатие по Хаффмену и арифметическое кодирование.

Методы сжатия по Хаффмену основаны на методе оптимального кодирования Хаффмена. Непосредственная реализация метода Хаффмена приводит к неадаптивному сжатию, так как вероятности символов должны быть известны заранее (неадаптивный метод далее рассматриваться не будет). Полуадаптивный метод сжатия по Хаффмену заключается в том, что сначала весь исходный файл просматривается, определяются частоты символов в файле, а затем производится кодирование на основании полученных частот, которые используются как оценки вероятностей символов. При этом в выходной файл со сжатыми данными заносятся значения частот символов и/или коды символов, например в виде кодовой таблицы. Адаптивный метод сжатия по Хаффмену основан на использовании некоторой вероятностной модели источника исходных данных, корректируемой после кодирования каждого символа. В этом случае хранить информацию о частотах символов не нужно, поскольку декодер будет использовать точно такую же модель и точно также ее корректировать после восстановления каждого символа.

Рассмотрим два адаптивных метода сжатия по Хаффмену. Первый из них заключается в использовании модели, которая изначально является равномерной, содержащей все возможные символы (в том числе и символ конца файла EOF) с частотами, равными 1. После кодирования очередного символа его частота увеличи-



вается на единицу. При этом подсчитывается суммарная частота и если она достигает заранее заданного, максимального, значения, то производится нормализация – уменьшение частот всех символов. Как правило, при нормализации все частоты делят на 2. При этом частотам, получившимся равными 0, присваивают значение 1. Процесс продолжается до тех пор, пока не будет достигнут конец файла (считан символ EOF).

Второй метод адаптивного сжатия по Хаффмену основан на использовании дополнительного символа ESC, частота которого всегда равна 1. Исходная модель в этом случае содержит только два символа – ESC и EOF с единичными частотами. При появлении очередного символа проверяется его наличие в текущей модели. Если символ имеется в модели (его частота не равна 0), то ему присваивается код по методу Хаффмена. Если же символ в модели отсутствует (его частота равна 0), то ему присваивается код, состоящий из кода Хаффмена символа ESC, к которому приписывается справа исходный код символа (код, считанный из файла). Если символ при кодировании уже присутствовал в модели, то после кодирования его частота увеличивается на 1, иначе – он добавляется в модель с частотой 1. Как и в предыдущем случае, подсчитывается суммарная частота и производится нормализация, когда суммарная частота достигает заданного, максимального, значения. При этом можно модифицировать нормализацию: после уменьшения значений частот все символы (кроме ESC и EOF), у которых частоты стали меньше 1, удаляются из модели. Отметим, что кодирование выполняется таким образом, чтобы коды символов ESC и EOF имели не меньшую длину, чем другие символы.

Восстановление данных, сжатых с помощью первого адаптивного метода, выполняется следующим образом: побитно считывается код из сжатого файла, производится попытка его декодировать и, если это удастся, соответствующий символ записывается в выходной файл, его частота увеличивается на 1, определяется

суммарная частота и при необходимости выполняется нормализация. Процесс продолжается до тех пор, пока не будет декодирован символ EOF. Восстановление по второму методу отличается только тем, что если декодирован символ ESC, то далее считывается известное количество бит несжатого символа, который затем добавляется в модель с частотой 1. При использовании обоих методов исходная модель должна быть точно такой же, что и при сжатии данных.

Проиллюстрируем описание методов сжатия по Хаффмену на примерах. Пусть файл данных, который нужно сжать, имеет вид

$$s_1s_2s_2s_2s_1s_3s_3\text{EOF}.$$

При сжатии файла полуадаптивным методом частоты символов и их коды будут найдены после первого прохода (табл. 2.1), а затем использованы на втором проходе. В результате будет получена двоичная последовательность:

$$01111000000001.$$

При этом в результирующий файл необходимо поместить также частоты символов и/или их коды.

Таблица 2.1

Полуадаптивные коды Хаффмена				
Символ	$s_1$	$s_2$	$s_3$	EOF
Частота	2	3	2	1
Код	01	1	000	001

Теперь выполним сжатие того же файла, но с помощью адаптивных методов. При этом будем полагать, что максимальное значение суммарной частоты, при которой необходимо проводить нормализацию, равно 8. Коды, получаемые при сжатии первым адаптивным методом, приведены в табл. 2.2. На выходе в результате будет получена последовательность

$$1001001010101001.$$

Отметим, что после выполнения четвертого и седьмого (перед выполнением пятого и восьмого) шагов была проведена нормализация, заключающаяся в делении значений всех частот на два и замене нулей на единицы.

Для сжатия вторым адаптивным методом необходимо определить исходные коды символов в несжатом виде. Пусть они будут следующими:

$$s_1 - 00; \quad s_2 - 01; \quad s_3 - 10; \quad \text{EOF} - 11,$$

тогда в результате на выходе получим

0000001110001101001011.

Процесс получения соответствующих кодов показан в табл. 2.3. Здесь также после шестого шага (перед седьмым) выполнялась нормализация модели.

Таблица 2.2

**Кодирование первым адаптивным методом Хаффмена**

№ п/п	Символ	Модель			№ п/п	Символ	Модель		
		Символ	Частота	Код			Символ	Частота	Код
1	$s_1$	$s_1$	1	10	5	$s_1$	$s_1$	1	01
		$s_2$	1	11			$s_2$	2	1
		$s_3$	1	00			$s_3$	1	000
		EOF	1	01			EOF	1	001
2	$s_2$	$s_1$	2	1	6	$s_3$	$s_1$	2	1
		$s_2$	1	01			$s_2$	2	00
		$s_3$	1	000			$s_3$	1	010
		EOF	1	001			EOF	1	011
3	$s_2$	$s_1$	2	1	7	$s_3$	$s_1$	2	00
		$s_2$	2	00			$s_2$	2	01
		$s_3$	1	010			$s_3$	2	10
		EOF	1	011			EOF	1	11
4	$s_2$	$s_1$	2	00	8	EOF	$s_1$	1	10
		$s_2$	3	1			$s_2$	1	11
		$s_3$	1	010			$s_3$	1	00
		EOF	1	011			EOF	1	01

Таблица 2.3

## Кодирование вторым адаптивным методом Хаффмена

№ п/п	Символ	Модель			№ п/п	Символ	Модель		
1	$s_1$	Символ	Частота	Код	5	$s_1$	Символ	Частота	Код
		$s_1$	0	–			$s_1$	1	01
		$s_2$	0	–			$s_2$	3	00
		$s_3$	0	–			$s_3$	0	–
		ESC	1	0			ESC	1	10
2	$s_2$	EOF	1	1	6	$s_3$	EOF	1	11
		Символ	Частота	Код			Символ	Частота	Код
		$s_1$	1	1			$s_1$	2	01
		$s_2$	0	–			$s_2$	3	00
		$s_3$	0	–			$s_3$	0	–
3	$s_2$	ESC	1	00	7	$s_3$	ESC	1	10
		EOF	1	01			EOF	1	11
		Символ	Частота	Код			Символ	Частота	Код
		$s_1$	1	10			$s_1$	1	11
		$s_2$	1	11			$s_2$	1	10
4	$s_2$	$s_3$	0	–	8	EOF	$s_3$	1	01
		ESC	1	00			ESC	1	000
		EOF	1	01			EOF	1	001
		Символ	Частота	Код			Символ	Частота	Код
		$s_1$	1	01			$s_1$	1	11
		$s_2$	2	00			$s_2$	1	10
		$s_3$	0	–			$s_3$	2	00
		ESC	1	10			ESC	1	010
		EOF	1	11			EOF	1	011

Альтернативой методу сжатия по Хаффмену является арифметическое кодирование. При его применении код присваивается всему передаваемому файлу вместо кодирования отдельных символов. Кодер читает входной файл символ за символом и добавляет биты к сжатому файлу. Получающийся код представляет собой дробную часть числа из полуинтервала  $[0, 1)$ . В общем виде алгоритм арифметического кодирования имеет следующие шаги:

1. Задать текущий интервал  $[0, 1)$ .
2. Повторить следующие действия для каждого символа  $s$  входного файла.

2.1. Разделить текущий интервал на части пропорционально частотам каждого символа.

2.2. Выбрать подынтервал, соответствующий символу  $s$ , и назначить его новым текущим интервалом.

3. Когда весь входной файл будет обработан, выходом алгоритма объявляется любая внутренняя точка текущего интервала.

Для восстановления сжатых данных последовательно определяются интервалы, в которые попадает анализируемое значение (входной код из сжатого файла), и соответствующие им символы, которые и будут выдаваться на выход декодера.

В качестве примера выполним сжатие той же последовательности, что рассматривалась при изучении методов сжатия по Хаффмену. Процесс образования интервалов показан в табл. 2.4. При этом используются частоты из табл. 2.1. На последнем (восьмом) шаге выбирается подынтервал

$[0,097481728; 0,097507477)$ ,

соответствующий символу EOF. В качестве результата можно взять любую точку из этого подынтервала. Выберем такое значение, которое имеет наименьшее число цифр в двоичной форме:

$0.097481728_{10} = 0,00011000111101001001_2$ .

Таким образом, арифметическим кодом исходной последовательности будет

00011000111101001001.

Таблица 2.4

Образование интервалов при арифметическом кодировании

№ п/п	Интервалы	Символ
1	$s_1: [0, 0,25)$ $s_2: [0,25, 0,625)$ $s_3: [0,625, 0,875)$ EOF: $[0,875, 1)$	$s_1$
2	$s_1: [0, 0,0625)$ $s_2: [0,0625, 0,15625)$ $s_3: [0,15625, 0,21875)$ EOF: $[0,21875, 0,25)$	$s_2$

№ п/п	Интервалы	Символ
3	$s_1$ : [0,0625, 0,0859375) $s_2$ : [0,0859375, 0,12109375) $s_3$ : [0,12109375, 0,14453125) EOF: [0,14453125, 0,15625)	$s_2$
4	$s_1$ : [0,0859375, 0,094726563) $s_2$ : [0,094726563, 0,107910156) $s_3$ : [0,107910156, 0,116699219) EOF: [0,116699219, 0,12109375)	$s_2$
5	$s_1$ : [0,094726563, 0,098022461) $s_2$ : [0,098022461, 0,102966309) $s_3$ : [0,102966309, 0,106262207) EOF: [0,106262207, 0,107910156)	$s_1$
6	$s_1$ : [0,094726563, 0,095550538) $s_2$ : [0,095550538, 0,096786499) $s_3$ : [0,096786499, 0,097610474) EOF: [0,097610474, 0,098022461)	$s_3$
7	$s_1$ : [0,096786499, 0,096992493) $s_2$ : [0,096992493, 0,097301483) $s_3$ : [0,097301483, 0,097507477) EOF: [0,097507477, 0,097610474)	$s_3$
8	$s_1$ : [0,097301483, 0,097352982) $s_2$ : [0,097352982, 0,097430229) $s_3$ : [0,097430229, 0,097481728) EOF: [0,097481728, 0,097507477)	EOF

Приведенный пример соответствует полуадаптивному методу. Существует также адаптивное арифметическое кодирование, основанное на модели, которая была рассмотрена при описании первого адаптивного метода сжатия по Хаффмену.

### 2.1.3. Словарные методы

Словарные методы основаны на применении динамических словарей. Эти методы базируются на алгоритмах LZ77 и LZ78, предложенных Лемпелем и Зивом. В основе словарных методов лежат следующие идеи:

1) каждая очередная закодированная последовательность символов добавляется к ранее закодированным символам таким образом, что вместе с ними она образует разложение всей переданной и принятой информации на несовпадающие между собой слова;

2) слова хранятся в памяти и используются в дальнейшем в качестве словаря;

3) кодирование осуществляется при помощи указателей на слова из сформированного словаря;

4) кодирование является динамической процедурой, ориентированной на блоки символов.

При реализации методов Лемпеля – Зива, как правило, используется скользящее окно, состоящее из словаря и упреждающего буфера.

В результате сжатия файла с помощью метода LZ77 формируется последовательность вида:

$$\langle A, L, S \rangle,$$

где  $A$  – относительный адрес в словаре;  $L$  – количество совпадающих символов;  $S$  – символ из буфера, который отличается от продолжения слова в словаре. При этом код  $\langle -1, 0, S \rangle$  или  $\langle 0, 0, S \rangle$  означает, что в словаре ничего не найдено.

Алгоритм LZ77 работает следующим образом. Сначала упреждающий буфер заполняется символами из начала сжимаемого файла. Затем осуществляется поиск в словаре слова, максимально совпадающего со словом в начале упреждающего буфера. После этого формируется и выдается код. После выдачи кода  $L + 1$  символов из начала упреждающего буфера переписываются в словарь, оставшиеся символы смещаются в начало, а на их место записываются символы из файла. Так как размер словаря ограничен, то из его начала часть символов может удаляться со смещением остальных на соответствующее число позиций.

Пусть, например, необходимо закодировать последовательность символов:

*aaaabaabbbEOF*

методом LZ77 с использованием словаря и упреждающего буфера, рассчитанных на 8 и 4 символов соответственно. Процесс кодирования показан в табл. 2.5.

Таблица 2.5

Пример кодирования по методу LZ77

№ п/п	Словарь								Упреждающий буфер				Код
1	0	1	2	3	4	5	6	7	0	1	2	3	<-1, 0, a>
	-	-	-	-	-	-	-	-	a	a	a	a	
2	0	1	2	3	4	5	6	7	0	1	2	3	<0, 1, a>
	a	-	-	-	-	-	-	-	a	a	a	b	
3	0	1	2	3	4	5	6	7	0	1	2	3	<0, 1, b>
	a	a	a	-	-	-	-	-	a	b	a	a	
4	0	1	2	3	4	5	6	7	0	1	2	3	<0, 2, b>
	a	a	a	a	b	-	-	-	a	a	b	b	
5	0	1	2	3	4	5	6	7	0	1	2	3	<4, 1, b>
	a	a	a	a	b	a	a	b	b	b	EOF	-	
6	0	1	2	3	4	5	6	7	0	1	2	3	<-1, 0, EOF>
	a	a	b	a	a	b	b	b	EOF	-	-	-	
7	0	1	2	3	4	5	6	7	0	1	2	3	-
	a	b	a	a	b	b	b	EOF	-	-	-	-	

Метод LZ78 отличается от LZ77 тем, что в каждой позиции словаря хранятся слова. Поэтому в выходном коде количество совпадающих символов *L* отсутствует, т.е. код имеет вид

$$<A, S>,$$

где *A* – адрес слова в словаре, совпадающего со словом в начале упреждающего буфера; *S* – символ из буфера, который отличается от продолжения слова в словаре. При этом код < -1, *S* > означает, что в словаре ничего не найдено. После выдачи кода в словарь добавляется слово, начало которого представляет собой слово из словаря по адресу *A* (*A* ≠ -1), а окончание – символ *S*.



Процесс кодирования по методу LZ78 последовательности символов, рассмотренной в примере применения метода LZ77, показан в табл. 2.6.

Таблица 2.6

**Пример кодирования по методу LZ78**

№ п/п	Словарь (позиция: слово)	Упреждающий буфер	Код
1	–	<i>aaaa</i>	$\langle -1, a \rangle$
2	0: <i>a</i>	<i>aaab</i>	$\langle 0, a \rangle$
3	0: <i>a</i> 1: <i>aa</i>	<i>abaa</i>	$\langle 0, b \rangle$
4	0: <i>a</i> 1: <i>aa</i> 2: <i>ab</i>	<i>aabb</i>	$\langle 1, b \rangle$
5	0: <i>a</i> 1: <i>aa</i> 2: <i>ab</i> 3: <i>aab</i>	<i>bbEOF</i>	$\langle -1, b \rangle$
6	0: <i>a</i> 1: <i>aa</i> 2: <i>ab</i> 3: <i>aab</i> 4: <i>b</i>	<i>bEOF</i>	$\langle 4, EOF \rangle$
7	0: <i>a</i> 1: <i>aa</i> 2: <i>ab</i> 3: <i>aab</i> 4: <i>b</i> 5: <i>bEOF</i>	–	–

Одной из практических модификаций LZ77 является алгоритм LZSS, который отличается от LZ77 тем, как поддерживается скользящее окно и какие коды выдает кодер. LZSS, помимо собственно окна с содержимым сообщения, поддерживает двоичное дерево для ускорения поиска совпадения. Каждая подстрока, покидающая буфер, добавляется в дерево поиска, а подстрока, покидающая словарь, удаляется из него. Такая организация модели данных позволяет добиться существенного увеличения скорости поиска совпадения, которая, в отличие от LZ77, становится про-

порциональна не произведению размеров окна и подстроки, а его двоичному логарифму. На выходе кодера формируются либо пары  $\langle A, L \rangle$ , либо незакодированные символы. Для их различения кодер LZSS использует однобитовый префикс.

Как и в LZ77, в этом алгоритме используется обычный символьный буфер для хранения содержимого окна. Доступ к нему организован как к кольцевому буферу, а размер кратен степени 2. Дерево поиска представляет собой двоичное лексикографически упорядоченное дерево. Каждый узел в дереве соответствует одной подстроке словаря и содержит ссылки на родителя и двух детей: «большого» и «меньшего» в смысле лексикографического сравнения символьных строк.

Для того чтобы кодер LZSS мог начать работать, необходимо загрузить буфер очередными символами сообщения и проинициализировать дерево. Для этого в дерево вставляется содержимое буфера. После этого алгоритм последовательно выполняет следующие действия:

- 1) кодирует содержимое буфера;
- 2) считывает очередные символы в буфер, удаляя при необходимости наиболее «старые» строки из словаря;
- 3) вставляет в дерево новые строки, соответствующие считанным символам.

Для того чтобы декодер смог вовремя остановиться, декодируя сжатое сообщение, кодер помещает в сжатый файл специальный символ конца файла после того, как он обработал все символы сообщения.

Вся работа по поиску расположения и установлению длины максимального совпадения содержимого словаря с буфером происходит в процессе добавления в дерево новой строки. При добавлении строки в дерево алгоритм последовательно перемещается от корня дерева к его листу, каждый раз осуществляя лексикографическое сравнение новой строки и узла дерева. В зависимости от

результата сравнения выбирается больший или меньший ребенок этого узла и запоминаются длина совпадения строк и положение текущего узла. Если в результате сравнения оказывается, что содержимое буфера и строка, на которую ссылается текущий узел, в точности совпадают, то ссылки в текущем узле обновляются так, чтобы они указывали на буфер, и процедура добавления строки в дерево завершается. Если ребенок текущего узла, выбранный для очередного шага, отмечен как несуществующий, остается заполнить его соответствующей ссылкой и завершить работу.

При удалении узла из дерева возможны два варианта. Если узел имеет не более одного ребенка, то удаление узла осуществляется простым исправлением ссылок «родитель – ребенок». Если узел имеет двоих детей, то необходимы другие действия. Для этого найдем следующий меньший узел в дереве, удалим его из дерева и заменим им текущий удаляемый узел. Следующий меньший узел находится выбором меньшего ребенка и последующим перемещением от него по дереву до листа по большим ветвям.

При восстановлении данных декодер читает один бит сжатой информации и определяет, что следует далее. Если далее следует символ, то следующие 8 битов выдаются как раскодированный символ и помещаются в скользящее окно. Если же далее следует пара  $\langle A, L \rangle$ , то соответствующее количество символов словаря помещается в окно и выдается в раскодированном виде. Декодирование заканчивается при обнаружении символа конца файла.

Примером практической модификации алгоритма LZ78 является алгоритм LZW. Кодер LZW никогда не выдает сами символы сжимаемого сообщения, только коды фраз. Он построен вокруг таблицы фраз (словаря), которая отображает строки символов сжимаемого сообщения в коды фиксированной длины вида  $\langle A \rangle$ . Таблица обладает так называемым свойством предшествования, т.е. для каждой фразы словаря, состоящей из некоторой фразы  $w$  и символа  $S$ , фраза  $w$  тоже содержится в словаре.

Очевидно, что декодер LZW использует тот же словарь, что и кодер, строя его по аналогичным правилам при восстановлении сжатого сообщения. Каждый считываемый код разбивается с помощью словаря на предшествующую фразу  $w$  и символ  $S$ . Затем рекурсия продолжается для предшествующей фразы  $w$  до тех пор, пока она не окажется кодом одного символа, что и завершает декомпрессию этого кода. Обновление словаря происходит для каждого декодируемого кода, кроме первого. После завершения декодирования кода его последний символ, соединенный с предыдущей фразой, добавляется в словарь. Новая фраза получает то же значение кода (позицию в словаре), что присвоил ей кодер. Так, шаг за шагом декодер восстанавливает тот словарь, который построил кодер.

Отметим, что словарные методы являются асимметричными методами, поскольку при восстановлении данных не нужно выполнять поиск: ссылка на словарь содержится непосредственно в коде. Например, для LZ77 декодер по адресу  $A$  и количеству совпадающих символов  $L$  находит слово в словаре, добавляет к нему символ  $S$  и выдает на выход.

### ***2.1.4. Методы кодирования длин серий***

Методы кодирования длин серий, которые еще называют методами RLE (Run Length Encoding), являются одними из наиболее старых методов сжатия. Однако благодаря своей простоте и эффективности они до сих пор используются либо непосредственно, либо в составе других методов. При использовании простейшего метода RLE последовательность (серия) одинаковых символов заменяется парой  $\langle C, S \rangle$ , где  $C$  – длина серии;  $S$  – символ, из которого состоит серия. Например, для файла:

*aaaaaaaaabbbbbaaaaaabbbbaaaaaabbaaaacccccdddEOF*

после применения данного метода будет получена следующая последовательность кодов:

$$\langle a, 9 \rangle \langle b, 4 \rangle \langle a, 5 \rangle \langle b, 2 \rangle \langle a, 6 \rangle \langle b, 4 \rangle \langle a, 6 \rangle \langle b, 2 \rangle \\ \langle a, 4 \rangle \langle c, 5 \rangle \langle d, 3 \rangle \langle \text{EOF}, 1 \rangle.$$

Недостатком простейшего метода RLE является увеличение размеров информации (вместо сокращения) в том случае, если сжимаемая последовательность содержит большое число одиночных символов. Поэтому для устранения указанного недостатка на практике используют различные модификации RLE-кодирования, например, можно использовать  $C$  переменного размера:

- при  $C = 0$  – размер равен 1 биту;
- при  $C > 0$  – размер равен  $n$  бит, старший из которых содержит 1.

Процесс восстановления данных при помощи декодера RLE вполне очевиден.

### ***2.1.5. Методы контекстного моделирования***

Для осуществления эффективного сжатия сообщений, поступающих с выхода источника информации, требуется знание его характеристик. При рассмотрении источников с памятью в качестве таких характеристик принято брать условные вероятности появления символов в сообщении вслед за различными символьными последовательностями. В случае, когда нет никакой дополнительной информации об источнике, определить эти вероятности можно только путем статистического анализа его информационной выборки. Данный принцип лежит в основе методов контекстного моделирования.

Последовательность символов, непосредственно предшествующую некоторому символу в информационном сообщении, принято называть контекстом этого символа, а длину этой последовательности – порядком контекста. На основе статистики, собираемой во время обработки информации, метод контекстного моделирования позволяет оценить вероятность появления в текущем контексте произвольного символа или последовательности символов. Оценка

вероятности определяет длину кода, который генерируется с использованием алгоритма Хаффмена, арифметического кодирования или другого метода, основанного на вероятностной модели (такие методы называются методами энтропийного кодирования). Методы контекстного моделирования в большинстве своем являются адаптивными. Вычисление вероятностных оценок осуществляется одним и тем же способом на этапах кодирования и декодирования, что позволяет не хранить описания информационной модели.

Методы контекстного моделирования отличаются способом получения оценок условных вероятностей появления символов в различных контекстах. Наибольшее распространение среди методов контекстного моделирования получил метод PPM. Оценку вероятности появления символов на выходе источника информации можно получать на основе контекстов только некоторого строго определенного порядка. В этом случае говорят, что используется контекстная модель соответствующего порядка. Такой метод малоэффективен по причине того, что реальные информационные источники, как правило, не обладают постоянной памятью. Более разумно при определении вероятностей появления символов учитывать модели разных порядков, т.е. производить так называемое смешивание моделей.

Очевидно, что конечный вклад различных моделей в вероятностную оценку должен как-то различаться. Одним из возможных решений этой проблемы является введение весов, определяющих влияние той или иной модели на результирующую оценку. В методе PPM используется несколько более упрощенный подход. Вероятность появления символа всегда оценивается в текущем контексте какого-то одного порядка, т.е. в некоторой конкретной контекстной модели. Метод PPM работает по следующей схеме. Первоначально для оценки вероятности появления символа берется контекст некоторого заранее определенного достаточно большого

порядка (выбирается некая конкретная контекстная модель). Если кодируемый символ ранее в таком контексте не встречался, то вероятность его появления не может быть оценена с использованием выбранной модели. В этом случае кодер генерирует код служебного символа перехода (ESCAPE-символ), сигнализирующего о невозможности использования рассматриваемой модели и необходимости перехода к другой модели. После этого осуществляется попытка оценить вероятность появления кодируемого символа с помощью модели следующего по убыванию порядка (возможен переход и на модель более низкого порядка). Если использование новой модели также не позволяет получить вероятностную оценку, кодер снова генерирует код служебного символа перехода и опять рассматривается модель меньшего порядка и т.д. Для гарантии завершения описанного процесса вводится дополнительная модель 1-го порядка, с одинаковой вероятностью оценивающая появление всех символов информационного алфавита (т.е. символ будет оценен, даже если он ранее вообще не встречался). Такая схема называется схемой контекстного спуска.

При оценке вероятности появления символа в контексте можно использовать два различных подхода. Первый сопряжен с вычислением вероятностной оценки в модели данного порядка без учета моделей более высоких порядков. Такой подход недостаточно эффективен, так как если символ может быть оценен моделью некоторого порядка, то, согласно описанной схеме, он никогда не будет оценен моделью более порядка, что, однако, никак не учитывается при получении оценки с применением моделей низших порядков.

Для примера рассмотрим ситуацию, когда требуется оценить вероятность появления символа *c* в контексте *bcabdad*, при условии, что оценка производится, начиная с модели второго порядка. Так как символ *c* в контексте *ab* ранее не встречается, то в кодовую последовательность добавляется код служебного символа перехода, сигнализирующий о необходимости оценки символа *c* исполь-

зованием модели первого порядка. Теперь оценка производится с учетом текущего контекста первого порядка –  $b$ . В данном контексте встречаются два символа:  $c$  и  $d$ . Однако символ  $d$  встречается также и в контексте второго порядка  $ab$ . Значит, в случае его появления в текущем контексте вместо символа  $c$  он был бы закодирован с применением модели второго, а не первого порядка. Следовательно, символ  $d$  не имеет смысла учитывать при оценке вероятности появления символа  $c$  в контексте  $b$ .

Итак, второй подход заключается в учете при вычислении вероятностной оценки контекстных моделей более высокого порядка по сравнению с порядком модели, используемой для оценки. Про модификации метода PPM, основанные на последнем подходе, говорят, что они используют механизм исключений, так как из вероятностной оценки исключается вклад моделей более высоких порядков. Применение механизма исключений приводит к повышению эффективности кодирования, однако из-за увеличения сложности оценки значительно снижается его производительность.

### ***2.1.6. Методы сжатия с преобразованием блоков***

Методы сжатия с преобразованием блоков выполняются в два этапа – на первом происходит преобразование данных в новую форму, а на втором – сжатие результатов преобразования. Преобразование данных позволяет выделить зависимости между символами сжимаемого сообщения и тем самым более полно учесть их при кодировании с помощью статистических методов.

Один из подходов здесь базируется на преобразовании Барроуза – Уилера BWT (Burrows – Wheeler Transformation), которое выполняется для блоков данных и заключается в следующем. Пусть имеется блок символов (строка)  $S_1, S_2, \dots, S_{N-1}, S_N$  длины  $N$ . Поместим данный блок в буфер такого же размера. Теперь последовательно выполним циклический сдвиг буфера на один символ влево:

$$S_1, S_2, \dots, S_{N-1}, S_N \rightarrow S_2, S_3, \dots, S_N, S_1.$$



Повторяем эту операцию до тех пор, пока не получится  $N$  строк – одна исходная и  $N - 1$  результатов циклического сдвига. При этом будем условно считать, что все  $N$  строк размещаются в некотором массиве. Отсортируем полученный массив строк в лексикографическом порядке и сформируем из последних символов каждой строки новую строку, которая вместе с индексом исходной строки и есть результат преобразования.

Преобразование BWT обратимо, т.е. по его результату можно восстановить исходный блок символов. Обратимость преобразования обуславливается тем, что все обрабатываемые строки получены из единственной исходной строки. Если отсортировать символы результата, то получим первые символы всех строк отсортированного массива. Если теперь выполнить циклический сдвиг влево и сортировку по двум символам, то получим по два первых символа в каждой строке. Процесс можно продолжить до восстановления всего массива строк. Исходная строка определяется ее индексом в этом массиве.

Применение преобразования BWT предполагает, что обрабатываемая информация порождается источником с памятью. Символы в такой информации зависят от контекстов, в которых они появляются. Отсюда, в частности, следует, что вероятность совпадения символов, имеющих частично совпадающие контексты, достаточно велика. Преобразование BWT позволяет группировать символы исходного информационного блока по схожести их контекстов. Результат преобразования обладает свойством локальной идентичности, которое заключается в том, что с уменьшением расстояния между позициями повышается вероятность совпадения символов, находящихся на этих позициях. Это, в свою очередь, позволяет осуществить эффективное кодирование результата преобразования с помощью какого-нибудь локально адаптивного статистического метода.

Для повышения эффективности сжатия на основе преобразования Барроуза – Уилера часто его результат подвергают дополнительному преобразованию, например преобразованию стопки книг или интервальному кодированию. Указанные преобразования, имеющие самостоятельное значение, будут рассмотрены далее.

Преобразование стопки книг или MTF (Move To Front) заключается в замещении символа номером его позиции в таблице, содержащей символы информационного алфавита. Символы замещаются последовательно, начиная с начала или с конца строки сообщения. Замещение сопровождается перестройкой таблицы, производимой следующим образом: замещенный символ перемещается на первую позицию в таблице, а все символы, находившиеся в момент замещения на позициях с номерами, меньшими номера позиции замещаемого символа, сдвигаются на одну позицию в направлении от начала таблицы. Таким образом, если какой-то символ недавно выступал в качестве замещаемого символа, то он будет находиться близко к началу таблицы.

Результатом преобразования является последовательность номеров символов в таблице. При этом маленькие номера будут встречаться в данной последовательности значительно чаще, чем большие. Такое перераспределение позволяет получить более эффективное сжатие.

При интервальном кодировании вместо символа  $S$  исходного сообщения кодируется длина интервала (количество символов) между предыдущим и текущим вхождением данного символа. При этом считается, что первому символу сообщения предшествуют все символы алфавита в заранее определенном порядке.

### ***2.1.7. Методы сжатия с потерями***

Все методы сжатия, описанные до этого, являются методами без потерь. Методы сжатия с потерями применяются, как правило, для хранения измерительной, звуковой или графической информа-

ции. В качестве примера рассмотрим подход к сжатию графической информации на основе дискретного косинусного преобразования.

Пусть имеется непрерывный периодический сигнал  $a(t)$  с периодом  $T$ :

$$a(t) = \alpha_0 \varphi_0(t) + \alpha_1 \varphi_1(t) + \dots + \alpha_{N-1} \varphi_{N-1}(t) = \sum_{s=0}^{N-1} \alpha_s \varphi_s(t).$$

Можно подобрать систему функций  $\{\varphi_s(t)\}$ ,  $s = 0, 1, \dots, N-1$ , так, чтобы:

- ни одна из них не могла быть получена линейной комбинацией других:

$$\varphi_l(t) \neq \sum_{s=0}^{N-1} \alpha_s \varphi_s(t), \quad s \neq l;$$

- за скалярное произведение принимается

$$(\varphi_s, \varphi_l) = \int_T \varphi_s(t) \varphi_l(t) dt;$$

- они попарно ортогональны и нормированы:

$$(\varphi_s, \varphi_l) = \begin{cases} 1, & \text{при } s = l, \\ 0, & \text{при } s \neq l. \end{cases}$$

Такой выбор системы функций  $\{\varphi_s(t)\}$ ,  $s = 0, 1, \dots, N-1$ , позволяет представить сигнал  $a(t)$  в виде точки  $N$ -мерного функционального пространства с координатами  $\alpha_s$ . Систему функций  $\{\varphi_s(t)\}$ ,  $s = 0, 1, \dots, N-1$ , называют базисом пространства сигналов, каждую из функций – базисной, представление сигнала в форме композиции базисных функций – разложением по заданному базису. Координаты точки, соответствующие сигналу  $a(t)$ , можно найти по формуле

$$\alpha_s = (a(t), \varphi_s(t)) = \int_T a(t) \varphi_s(t) dt.$$

Существует множество систем функций, обладающих свойствами базиса. В частности, базис образует система функций вида:

$$\varphi_s(t) = \frac{1}{\sqrt{T}} \exp\left(\frac{i2\pi st}{T}\right),$$

где  $i$  – мнимая единица. Разложение сигнала по этому базису называют разложением Фурье, а нахождение коэффициентов разложения – преобразованием Фурье. Преобразование Фурье выполняется по формуле:

$$\alpha_s = (a(t), \varphi_s(t)) = \int_T a(t) \varphi_s(t) dt = \frac{1}{\sqrt{T}} \int_0^T a(t) \exp\left(i2\pi \frac{st}{T}\right) dt.$$

В дальнейшем будем рассматривать периодический сигнал, который принимает значения в дискретные моменты времени:

$$t = k \Delta t,$$

где  $\Delta t = T / N$ ;  $N$  – количество разбиений интервала  $T$ . Тогда непрерывный сигнал  $a(t)$  заменяется на последовательность отсчетов:

$$a((k + 0,5) \Delta t) = a_k, \quad k = 0, 1, \dots, N - 1,$$

взятых посередине отрезков разбиения  $\Delta t$ , а формулы преобразования Фурье приобретают вид

$$\alpha_s = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} a_k c_s \cos\left(\pi \frac{s(k + 0,5)}{N}\right), \quad s = 1, 2, \dots, N - 1.$$

Данная формула задает прямое дискретное косинусное преобразование, которое выполняется по базису:

$$\{\varphi_{sk}\} = \left\{ \sqrt{\frac{2}{N}} c_s \cos\left(\pi \frac{s(k + 0,5)}{N}\right), \quad s = 1, 2, \dots, N - 1, \quad k = 1, 2, \dots, N - 1 \right\},$$

где  $c_s$  – нормировочный множитель, определяемый как

$$c_s = \begin{cases} \frac{1}{\sqrt{2}} & \text{при } k = 0; \\ 1 & \text{при } k \neq 0. \end{cases}$$

Данное преобразование обратимо – существует обратное дискретное косинусное преобразование:

$$a_k = \sqrt{\frac{2}{N}} \sum_{s=0}^{N-1} \alpha_s c_s \cos\left(\pi \frac{s(k+0,5)}{N}\right), \quad k=0, 1, \dots, N-1.$$

В этом выражении элементы  $\{\varphi_{sk}\}$  есть дискретные отсчеты базисных функций, где  $s$  – номер базисной функции;  $k$  – номер отсчета дискретизации базисной функции. Их значения образуют матрицу  $N \times N$ , которую обозначим  $F_N = \{\varphi_{sk}\}$ . Тогда прямое и обратное дискретное косинусное преобразование можно записать в матричной форме:

$$\alpha = F_N a; \quad a = F_N^T \alpha,$$

где  $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_{N-1})^T$ ;  $a = (a_0, a_1, \dots, a_{N-1})^T$ .

Описанные преобразования являются одномерными преобразованиями. Однако их можно обобщить для двумерного сигнала, представленного в виде матрицы размера  $N \times N$ . В результате дискретного косинусного преобразования над двумерным сигналом получается также матрица размера  $N \times N$  коэффициентов разложения по базисным функциям:

$$\alpha = \{\alpha_{sr}\}; \quad s = 0, 1, \dots, N-1; \quad r = 0, 1, \dots, N-1.$$

Формулы в матричной форме для выполнения прямого и обратного двумерного дискретного косинусного преобразования имеют вид:

$$\alpha = F_N a F_N^T; \quad a = F_N^T \alpha F_N.$$

Матрица  $F_N$  зависит только от  $N$  и может быть просчитана заранее. Например, при  $N = 8$  она будет иметь вид:

$$F_N = \begin{bmatrix} 0,35 & 0,35 & 0,35 & 0,35 & 0,35 & 0,35 & 0,35 & 0,35 \\ 0,49 & 0,42 & 0,28 & 0,09 & -0,09 & -0,28 & -0,42 & -0,49 \\ 0,46 & 0,19 & -0,19 & -0,46 & -0,46 & -0,19 & 0,19 & 0,46 \\ 0,42 & -0,09 & -0,49 & -0,28 & 0,28 & 0,49 & 0,09 & -0,42 \\ 0,35 & -0,35 & -0,35 & 0,35 & 0,35 & -0,35 & -0,35 & 0,35 \\ 0,28 & -0,49 & 0,09 & 0,42 & -0,42 & -0,09 & 0,49 & -0,28 \\ 0,19 & -0,46 & 0,46 & -0,19 & -0,19 & 0,46 & -0,46 & 0,19 \\ 0,09 & -0,28 & 0,42 & -0,49 & 0,49 & -0,42 & 0,28 & -0,09 \end{bmatrix}.$$

При сжатии графической информации исходный файл представляет собой последовательность пикселей – значений цвета точек изображения. В случае полутоновых изображений цвета задаются в виде чисел, характеризующих яркости соответствующих точек. При использовании дискретного косинусного преобразования последовательность пикселей разбивают на блоки размером  $8 \times 8$  и обрабатывают эти блоки по отдельности.

Предположим, что матрица яркостей какого-либо блока имеет вид:

$$a = \begin{bmatrix} 95 & 88 & 88 & 87 & 95 & 88 & 95 & 95 \\ 143 & 144 & 151 & 151 & 153 & 170 & 183 & 181 \\ 153 & 151 & 162 & 166 & 162 & 151 & 126 & 117 \\ 143 & 144 & 133 & 130 & 143 & 153 & 159 & 175 \\ 123 & 112 & 116 & 130 & 143 & 147 & 162 & 189 \\ 133 & 151 & 162 & 166 & 170 & 188 & 166 & 128 \\ 160 & 168 & 166 & 159 & 135 & 101 & 93 & 98 \\ 154 & 155 & 153 & 144 & 126 & 106 & 118 & 133 \end{bmatrix}.$$

Теперь последовательно выполним следующие шаги:

1) из элементов матрицы вычитается величина 128:

$$a^{cm} = \begin{bmatrix} -33 & -40 & -40 & -41 & -33 & -40 & -33 & -33 \\ 15 & 16 & 23 & 23 & 25 & 42 & 55 & 53 \\ 25 & 23 & 34 & 38 & 34 & 23 & -2 & -11 \\ 15 & 16 & 5 & 2 & 15 & 25 & 31 & 47 \\ -5 & -16 & -12 & 2 & 15 & 19 & 34 & 61 \\ 5 & 23 & 34 & 38 & 42 & 60 & 38 & 0 \\ 32 & 40 & 38 & 31 & 7 & -27 & -35 & -30 \\ 26 & 27 & 25 & 16 & -2 & -22 & -10 & 5 \end{bmatrix};$$

2) выполняется дискретное косинусное преобразование над смещенными данными с округлением результатов до целых значений:

$$\alpha = \begin{bmatrix} 92 & 2 & -8 & -7 & 4 & -1 & 1 & 1 \\ -39 & -57 & 11 & 17 & -2 & 2 & 4 & 2 \\ -85 & 64 & 0 & 17 & 3 & 5 & -6 & 5 \\ -52 & -37 & -10 & 13 & -9 & 3 & -2 & 0 \\ -86 & -41 & 50 & -8 & 18 & -7 & -2 & 5 \\ -63 & 66 & -13 & -1 & 2 & -8 & -2 & 1 \\ -17 & 15 & -37 & 18 & -12 & 3 & 3 & -1 \\ -53 & 31 & -7 & -10 & 23 & -1 & 1 & 3 \end{bmatrix};$$

3) квантуются коэффициенты разложения:

$$\alpha_{sr}^1 = \frac{\alpha_{sr}}{q_{sr}},$$

где  $q_{sr}$  – коэффициент квантования, например,  $q_{sr} = 1 + (s + r)Q$ .  
При  $Q = 4$  получим:

$$\alpha^1 = \begin{bmatrix} 18 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -4 & -4 & 1 & 1 & 0 & 0 & 0 & 0 \\ -7 & 4 & 0 & -1 & 0 & 0 & 0 & 0 \\ -3 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4 & -2 & 2 & 0 & 0 & 0 & 0 & 0 \\ -3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$





тия, применяемого при представлении видеоданных в стандарте MPEG. Аналогичные методы применяются также и для сжатия звуковых данных.

### ***2.1.8. Сжатие непрерывной информации***

Очевидным способом сжатия непрерывной информации является ее преобразование в цифровую форму, так как при этом часть информации удаляется. В этом случае получают последовательность чисел – отсчетов (выборок) исходного сигнала, которую затем можно дополнительно сжать любым из рассмотренных методов (в этом случае будет удалена избыточность в получившихся дискретных данных).

В основе преобразования непрерывных сигналов в цифровую форму лежат две операции – дискретизация и квантование. Дискретизация заключается в периодическом измерении непрерывного сигнала (с некоторым интервалом дискретизации) и использовании моментальных значений (отсчетов) вместо исходной «волны». Квантование представляет собой приведение значения очередного отсчета к допустимой величине из конечного множества значений, соответствующих уровням квантования, отличающимся на некоторый интервал квантования.

Для представления результатов дискретизации используют амплитудно-импульсную модуляцию: сигнал преобразуется в набор импульсов определенной длительности (меньшей интервала дискретизации) с амплитудами, равными значениям, зафиксированным в момент опроса. Результат квантования можно воспринимать как представление каждого отсчета в виде последовательности импульсов, соответствующей двоичному коду ближайшего к его значению уровня квантования (импульсно-кодовая модуляция). Данные представления результатов дискретизации и квантования играют роль при передаче информации по каналу связи. С точки

зрения обработки и хранения информации, последовательность отсчетов хранится в памяти вычислительной системы в виде чисел.

При равномерной дискретизации (равномерном квантовании) интервал дискретизации (интервал квантования) является постоянным независимо от характера сигнала в тот или иной момент времени. Однако можно значительно сократить избыточность информации за счет применения адаптивных методов преобразования сигнала в цифровую форму.

Рассмотрим метод дифференциальной импульсно-кодовой модуляции, основанный на том, что фиксируются не сами моментальные значения, а разница между последовательными значениями. Так как при малом интервале дискретизации последовательно идущие выборки отличаются друг от друга незначительно. В результате понадобится меньшее количество бит для хранения информации по сравнению с равномерным квантованием. Пусть, например, три последовательные выборки имеют величины 17, 22 и 23. Тогда после дифференциальной импульсно-кодовой модуляции получим последовательность чисел 17, 5 и 1, для которых можно получить коды с меньшей средней длиной кодовых слов. При практическом использовании этого метода набор допустимых приращений задается заранее.

Для сжатия звуковой информации получили широкое распространение методы на основе адаптивной дифференциальной импульсно-кодовой модуляции. Здесь вместо того, чтобы использовать заранее заданные приращения, набор допустимых приращений определяется, исходя из предварительно считанных данных. Часто данный набор допустимых приращений принимает форму изменяющегося масштабного коэффициента. Подгоняя значение масштабного коэффициента к параметрам конкретного фрагмента, например, звуковой записи, можно добиться более высокого качества, чем при использовании обычной дифференциальной импульсно-кодовой модуляции.

Широкое распространение получили также методы сжатия с прогнозированием. В основе таких методов лежит задача поиска способа предсказания последующего элемента данных на основе уже имеющихся данных. Если удастся угадать, каким будет следующий элемент, хранить его не нужно, так как декодер, используя тот же способ прогнозирования, просто вычислит пропущенное значение. Такие методы часто используются для сжатия измерительной информации, поступающей в систему с датчиков (интерполяционные и экстраполяционные методы).

## **2.2. Помехоустойчивое кодирование информации**

### ***2.2.1. Помехи***

При передаче информации по каналу связи она преобразуется в сигналы, удобные для прохождения по конкретной линии связи. Линия связи – физическая среда, обеспечивающая поступление сигнала от передающего устройства к приемному. Сигналы на выходе линии связи могут отличаться от переданных из-за действия помех. Помехами называют любые мешающие возмущения как внешние, так и внутренние (источником которых являются технические средства канала связи), вызывающие отклонения принятых сигналов от переданных и затрудняющие их прием. К внешним помехам относятся, например, атмосферные, промышленные и преднамеренные помехи, а к внутренним – помехи, вызываемые тепловым и дробовым эффектами.

Для уменьшения влияния помех применяют помехоустойчивые коды, к которым относятся коды, обнаруживающие ошибки, и корректирующие коды.

Построение помехоустойчивых кодов в основном связано с добавлением к исходной комбинации из  $k$  информационных символов  $r$  контрольных символов. Так как закодированная комбинация

ция будет составлять  $n = k + r$  символов, то такие коды часто называют  $(n, k)$ -кодами.

Отметим, что при помехоустойчивом кодировании, в отличие от сжатия, в передаваемую информацию вносится некоторая избыточность, т.е. наличие избыточности, как уже говорилось, не всегда является отрицательным показателем.

### ***2.2.2. Расстояние Хемминга***

Основные свойства помехоустойчивых кодов определяются с помощью расстояния Хемминга (хемминговоe расстояние или кодовое расстояние). Представим  $N$ -мерный куб, длина ребер в котором равна одной единице. Вершины такого куба отображают двоичные коды. Минимальное расстояние между двумя вершинами определяется минимальным количеством ребер, находящихся между этими вершинами. Расстояние, определяемое таким образом, и есть расстояние Хемминга. Другими словами, кодовое расстояние – это минимальное число элементов, в которых одна кодовая комбинация отличается от другой. Для определения кодового расстояния между двумя кодовыми комбинациями можно сложить их по модулю 2 и найти число единиц в получившейся сумме. Например, сложив две комбинации 10110101101 и 11001010101, получим 0111111000 и определим, что расстояние между ними  $d = 7$ .

Если код имеет кодовое расстояние  $d = 1$ , то все кодовые комбинации размещаются в вершинах куба. Такой код не является помехоустойчивым – он не в состоянии обнаружить ошибку. Если же выбрать комбинации с кодовым расстоянием  $d = 2$ , например, 000, 110, 101, 011, то соответствующий код будет уже более помехоустойчивым – позволит обнаруживать одиночные ошибки. Назовем выбранные комбинации разрешенными, предназначенными для передачи информации, а все остальные (001, 010, 100, 111) – запрещенными. Очевидно, что любая одиночная ошибка приводит к тому, что разрешенная комбинация переходит в ближайшую за-

прещенную комбинацию, что можно обнаружить после ее получения. При выборе комбинаций с кодовым расстоянием  $d = 3$  получим код, который позволит исправить одну одиночную ошибку или обнаружить две ошибки. Таким образом, увеличивая кодовое расстояние, можно увеличить помехоустойчивость кода. В общем случае кодовое расстояние определяется по формуле

$$d = t + l + 1,$$

где  $t$  – число исправляемых ошибок;  $l$  – число обнаруживаемых ошибок, при этом обычно  $l > t$ .

### **2.2.3. Коды для обнаружения ошибок**

Один из самых старых, простых и используемых кодов для обнаружения ошибок образуется путем добавления к передаваемой двоичной комбинации, состоящей из  $k$  информационных символов (битов), одного контрольного символа (0 или 1), так, чтобы общее число единиц в передаваемой комбинации было четным. Такой код называется кодом с проверкой на четность и позволяет обнаруживать одиночные ошибки. Например, пусть даны две комбинации 11011 и 11100, для которых нужно построить код с проверкой на четность. Первая комбинация имеет четное, а вторая – нечетное число единиц. Поэтому к первой комбинации добавляется 0, а ко второй – 1. В результате получаем коды 110110 и 111001. Если предположить, что в результате ошибок вместо указанных комбинаций были получены 111110 и 110001, то проверка на четность их обнаружит, так как контрольные символы у принятых комбинаций (если они правильные) должны быть 1 и 0 соответственно.

Другим примером кодов с обнаружением ошибок является код с постоянным весом, который содержит постоянное число единиц и нулей. В этом случае число возможных кодовых комбинаций составит

$$N = C_n^m = \frac{n!}{m!(n-m)!},$$

где  $n$  – общее число символов;  $m$  – число единиц в кодовой комбинации. Например, в табл. 2.7 приведены коды с двумя единицами из пяти  $\left( \text{число кодируемых комбинаций равно } C_5^2 = \frac{5!}{2!(5-2)!} = 10 \right)$ .

Этот код позволяет обнаруживать любые одиночные ошибки и часть многократных ошибок. Однако с помощью этого кода не обнаруживаются ошибки смещения, когда одна единица переходит в ноль и один ноль переходит в единицу или два нуля и две единицы меняются на обратные символы и т.д.

Таблица 2.7

**Код с двумя единицами из пяти**

Значение		Код
десятичный вид	двоичный вид	
0	0000	00011
1	0001	00101
2	0010	00110
3	0011	01001
4	0100	01010
5	0101	01100
6	0110	10001
7	0111	10010
8	1000	10100
9	1001	11000

Часто также используется корреляционный код (код с удвоением). При его построении 1 преобразуется в 10, а 0 – в 01. Тогда, например, вместо комбинации 1010011 будет передаваться 10011001011010. Ошибка обнаруживается в том случае, если в парных элементах будут одинаковые символы 00 или 11 (вместо 01 и 10).

Еще одним простым кодом является инверсный код. В этом случае к исходной комбинации добавляется такая же комбинация по длине. В линию посылается удвоенное число символов. Если в исходной комбинации четное число единиц, то добавляемая комбинация повторяет исходную комбинацию, если нечетное – то до-

бавляемая комбинация является инверсной относительно исходной. Например, инверсный код для комбинации 11011 будет иметь вид 1101111011 (контрольная комбинация 11011), а для комбинации 11100 – 1110000011 (контрольная комбинация 00011). Прием инверсного кода осуществляется в два этапа. На первом этапе суммируются единицы в первой основной группе символов. Если число единиц четное, то контрольные символы принимаются без изменения, если нечетное, то контрольные символы инвертируются. На втором этапе контрольные символы суммируются с информационными символами по модулю два. Нулевая сумма говорит об отсутствии ошибок. При ненулевой сумме принятая комбинация считается ошибочной. Обнаруживающие способности данного кода достаточно велики: определяет практически любые ошибки, кроме редких ошибок смещения, которые одновременно происходят как среди информационных символов, так и среди соответствующих контрольных.

#### ***2.2.4. Коды для исправления ошибок***

Коды, которые позволяют обнаруживать и исправлять ошибки, называют корректирующими. Большинство корректирующих кодов являются линейными кодами, у которых контрольные символы образуются путем линейной комбинации информационных символов. Кроме того, корректирующие коды являются групповыми кодами, т.е. образуют замкнутую группу с нулевым элементом и операцией сложения. Замкнутость группы означает, что при сложении двух элементов группы получается элемент, принадлежащий этой же группе. В качестве операции сложения используется поразрядное сложение по модулю 2 (так как число разрядов у кодов не должно увеличиваться при выполнении операции сложения).

Для построения кода, способного обнаруживать и исправлять одиночную ошибку, необходимое число контрольных разрядов будет составлять

$$n - k \geq \log(n + 1),$$

где  $k$  – число разрядов исходной кодовой комбинации;  $n$  – число разрядов после добавления контрольных символов;  $r$  – число контрольных символов. Если необходимо иметь возможность исправлять однократные и двукратные ошибки, то число контрольных разрядов определяется неравенством

$$n - k \geq \log(1 + C_n^1 + C_n^2).$$

В общем случае число контрольных символов определяется неравенством Хэмминга:

$$n - k \geq \log(1 + C_n^1 + C_n^2 + \dots + C_n^t) = \log \sum_{i=0}^t C_n^i,$$

где  $t$  – максимальное число одновременно исправляемых ошибок.

В качестве примеров корректирующих кодов рассмотрим коды Хемминга и циклические коды для исправления одиночной ошибки.

Код Хэмминга является групповым  $(n, k)$ -кодом с минимальным расстоянием  $d = 3$ , который позволяет обнаруживать и исправлять однократные ошибки.

При декодировании кода Хемминга формируется синдром, значение которого при отсутствии ошибки равно 0, а при обнаружении ошибки равно номеру разряда с ошибкой.

При построении кода Хемминга требуется:

- 1) задать количество информационных разрядов  $k$ ;
- 2) определить количество проверочных разрядов  $r$ ;
- 3) задать проверочные равенства;
- 4) определить позиции проверочных разрядов;
- 5) определить значения проверочных разрядов;
- 6) построить кодовые комбинации.

Пример построения кода Хемминга:

- 1) задать количество информационных разрядов:  
 $k = 4$ ;



2) определить количество проверочных разрядов:

$$2^r \geq r + k + 1 \rightarrow r = 3;$$

3) задать проверочные равенства:

$$e_1 = a_1 \oplus a_3 \oplus a_5 \oplus a_7,$$

$$e_2 = a_2 \oplus a_3 \oplus a_6 \oplus a_7,$$

$$e_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7;$$

4) определить позиции проверочных разрядов:

$$a_1, a_2, a_4;$$

5) определить значения проверочных разрядов:

$$a_1 = a_3 \oplus a_5 \oplus a_7,$$

$$a_2 = a_3 \oplus a_6 \oplus a_7,$$

$$a_4 = a_5 \oplus a_6 \oplus a_7;$$

6) построить кодовые комбинации: кодовые комбинации для данного примера приведены в табл. 2.8.

Таблица 2.8

Код Хемминга

Исходная кодовая комбинация	Кодовая комбинация Хемминга	Исходная кодовая комбинация	Кодовая комбинация Хемминга
0000	0000000	1000	1110000
0001	1101001	1001	0011001
0010	0101010	1010	1011010
0011	1000011	1011	0110011
0100	1001100	1100	0111100
0101	0100101	1101	1010101
0110	1100110	1110	0010110
0111	0001111	1111	1111111

Действия при декодировании кода Хемминга аналогичны:

1) получить кодовую комбинацию Хемминга;

2) сформировать синдром согласно проверочным равенствам;

3) если синдром не равен 0, то исправить ошибку в разряде, номер которого равен значению синдрома;

4) определить по таблице исходную комбинацию.

Пример декодирования кода Хемминга:

1) получить кодовую комбинацию Хемминга:

1101011;

2) сформировать синдром согласно проверочным равенствам:

$$e_1 = 1 \oplus 0 \oplus 0 \oplus 1 = 0,$$

$$e_2 = 1 \oplus 0 \oplus 1 \oplus 1 = 1,$$

$$e_3 = 1 \oplus 0 \oplus 1 \oplus 1 = 1;$$

3) если синдром не равен 0, то исправить ошибку в разряде, номер которого равен значению синдрома:

$$e_3 e_2 e_1 = 110_2 = 6 - \text{ошибка в 6-м разряде,}$$

правильная кодовая комбинация:

1101001;

4) определить по таблице исходную комбинацию:

0001.

Для построения кода Хэмминга можно использовать порождающую матрицу:

$$G_{k \times n} = (A_{k \times (n-k)} I_k),$$

где  $A_{k \times (n-k)}$  – матрица двоичных элементов;  $I_k$  – единичная матрица;  $k$  – число информационных разрядов кода;  $n$  – общее число разрядов кода. Порождающей матрице соответствует проверочная матрица:

$$H_{(n-k) \times n} = (I_{n-k} A_{k \times (n-k)}^T).$$

Матрица  $A_{k \times (n-k)}$  содержит  $k$  строк, представляющих все возможные двоичные комбинации длины  $n - k$  с не менее чем двумя единицами. Например, (7, 4) – код Хэмминга из табл. 2.9 имеет следующие порождающую и проверочную матрицы:

$$G_{4 \times 7} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}; \quad H_{3 \times 7} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Таблица 2.9

(7, 4) – код Хэмминга

Исходный код	Код Хэмминга	Исходный код	Код Хэмминга	Исходный код	Код Хэмминга	Исходный код	Код Хэмминга
0000	0000000	0100	0110100	1000	1101000	1100	1011100
0001	1010001	0101	1100101	1001	0111001	1101	0001101
0010	1110010	0110	1000110	1010	0011010	1110	0101110
0011	0100011	0111	0010111	1011	1001011	1111	1111111

Данный код является систематическим, т.е. каждую кодовую комбинацию можно представить в виде:

$$e_1 e_2 \dots e_{n-k} x_1 x_2 \dots x_k,$$

где  $e_1, e_2, \dots, e_{n-k}$  – проверочные символы;  $x_1, x_2, \dots, x_k$  – информационные символы. С помощью порождающей матрицы  $G_{k \times n}$  из исходной кодовой  $k$ -символьной комбинации  $C_k$  получается  $n$ -символьный код Хэмминга  $X_n$  следующим образом:

$$X_n = C_k G_{k \times n}.$$

Обнаружение ошибок основано на том, что для разрешенных кодовых комбинаций справедливо равенство:

$$X_n H_{(n-k) \times n}^T = 0,$$

поэтому если результат операции (синдром) не будет нулевым, то можно сделать вывод о том, что кодовая комбинация содержит ошибку. В этом случае определяется номер строки транспонированной проверочной матрицы  $H_{(n-k) \times n}^T$ , равной синдрому, который и будет номером разряда кодовой комбинации, содержащим ошибку. Например, пусть вместо комбинации 0011010 была получена комбинация 0111010, тогда синдром равен:

$$(0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0) \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = (0 \ 1 \ 0),$$

что соответствует второй строке проверочной матрицы и, значит, ошибка обнаружена во втором разряде кодовой комбинации. После исправления символа во втором разряде с 1 на 0 получим правильную комбинацию:

0011010.

При построении кода Хэмминга необходимо учитывать, что проверочная матрица  $H_{(n-k) \times n}$  не должна содержать одинаковых столбцов, а строки матрицы  $A_{k \times (n-k)}$  должны содержать, по крайней мере, две единицы.

Циклическими кодами называют специальную группу линейных кодов, которые имеют полиномиальное представление. Например, кодовая комбинация 101101 имеет следующее полиномиальное представление:

$$P(x) = 1 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 = x^5 + x^3 + x^2 + 1.$$

Циклические коды относятся к систематическим  $(n, k)$ -кодам, в которых контрольные  $n - k$  и информационные  $k$  разряды расположены на строго определенных местах.

Над полиномами, представляющими циклические коды, определены операции умножения, деления, сложения и вычитания. При этом операции сложения и вычитания выполняются по модулю 2.

Циклический код  $F(x)$  получают следующим образом:

$$\frac{h(x)x^{n-k}}{g(x)} = Q(x) + \frac{R(x)}{g(x)};$$

$$F(x) = Q(x)g(x) + R(x),$$

где  $h(x)$  – заданный многочлен, соответствующий исходной кодовой комбинации;  $g(x)$  – образующий многочлен, который является неприводимым сомножителем при разложении двучлена  $x^n + 1$ . Некоторые образующие многочлены приведены в табл. 2.10.

Таблица 2.10

**Примеры образующих многочленов**

$r$	$g(x)$	$r$	$g(x)$
1	$x + 1$	5	$x^5 + x^2 + 1$
2	$x^2 + x + 1$	6	$x^6 + x + 1$
3	$x^3 + x + 1$	7	$x^7 + x + 1$
4	$x^4 + x + 1$	8	$x^8 + x^4 + x^3 + x + 1$

При декодировании принятую кодовую комбинацию необходимо разделить на  $g(x)$ . Если в результате деления остаток не равен нулю, то означает, что в кодовой комбинации имеется ошибка.

Для выбора образующего полинома по заданной кодовой комбинации сначала определяют число контрольных символов из соотношения:

$$n - k = \log(n + 1),$$

которое в численном виде показано в табл. 2.11, а затем из таблицы неприводимых многочленов выбирают самый короткий многочлен со степенью, равной числу контрольных символов.

Таблица 2.11

**Соотношения между числом информационных и контрольных символов**

$N$	3	5	6	7	9 ... 15	17 ... 31	33 ... 63	65 ... 127
$K$	1	2	3	4	5 ... 11	12 ... 26	27 ... 57	28 ... 120
$n - k$	2	3	3	3	4	5	6	7

Пусть, например, требуется закодировать комбинацию вида 1101, что соответствует  $h(x) = x^3 + x^2 + 1$ . Процесс кодирования последовательно выполняется в виде следующих шагов:

- 1) определяем число контрольных символов  $n - k = 3$ ;
- 2) из табл. 2.12 выбираем многочлен  $g(x) = x^3 + x + 1 = 1011$ ;
- 3) умножаем  $h(x)$  на  $x^{n-k}$ :

$$h(x) x^{n-k} = (x^3 + x^2 + 1) x^3 = x^6 + x^5 + x^3 = 1101000;$$

4) делим полученное произведение на образующий полином  $g(x)$ :

$$\frac{h(x)x^{n-k}}{g(x)} = \frac{x^6 + x^5 + x^3}{x^3 + x + 1} = x^3 + x^2 + x + 1 + \frac{1}{x^3 + x + 1} = 1111 + \frac{001}{1011};$$

- 5) суммируем остаток с  $h(x) x^{n-k}$ :

$$F(x) = x^6 + x^5 + x^3 + 1 = 1101001.$$

В полученной кодовой комбинации циклического кода информационные символы  $h(x) = 1101$ , а контрольные  $R(x) = 001$ . Закодированное сообщение делится на образующий полином без остатка:

$$1101001 / 1011 = 1111.$$

Образующей (порождающей) матрицей можно воспользоваться для образования циклических кодов. Образующая матрица  $G_{k \times n}$  составляется на основе единичной матрицы  $I_k$ , к которой справа дописывается матрица остатков  $R_{k \times (n-k)}$ :

$$G_{k \times n} = (I_k R_{k \times (n-k)}).$$

Матрица  $R_{k \times (n-k)}$  получается из остатков  $r_i$  от деления полинома  $x^{2(n-k)-i-1}$  на образующий многочлен  $g(x)$  для всех строк с номерами  $i = 1, 2, \dots, k$ . Например, для (7, 4) – кода и  $g(x) = x^3 + x + 1$  будут получены остатки, представленные в табл. 2.12.

Образование элементов матрицы остатков

$i$	$x^{2(n-k)-1}$	$x^{2(n-k)-1}/g(x)$	$r_i$
1	$x^6$	$x^3 + x + 1$	$x^2 + 1 = 101$
2	$x^5$	$x^2 + 1$	$x^2 + x + 1 = 111$
3	$x^4$	$x$	$x^2 + x = 110$
4	$x^3$	1	$x + 1 = 011$

Порождающая матрица для (7, 4) – кода с порождающим многочленом  $g(x) = x^3 + x + 1$  имеет вид:

$$G_{4 \times 7} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Для обнаружения ошибок можно использовать проверочную матрицу:

$$H_{(n-k) \times n} = (R_{k \times (n-k)}^T I_{n-k}).$$

Например, для (7, 4) – кода проверочная матрица будет следующей

$$H_{3 \times 7} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Применение порождающей и проверочной матриц точно такое же, как и в случае кодов Хэмминга. Пусть, например, необходимо получить линейный код для 1101. Тогда, умножив 1101 на порождающую матрицу, получим искомую кодовую комбинацию 1101001. Теперь, пусть вместо 1101001 была принята комбинация 1101011. Умножив ее на транспонированную проверочную матрицу, получим синдром 010, соответствующий 6-й строке этой матрицы. Это означает, что необходимо исправить 6-й разряд проверяемой кодовой комбинации. Выполнив исправление, получаем 1101001.

## 2.3. Шифрование информации

Одной из важнейших задач, которые необходимо решать в информационных системах, является обеспечение информационной безопасности, под которой понимается защищенность информации и поддерживающей инфраструктуры от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры.

В качестве одного из основных механизмов информационной безопасности используются средства шифрования информации, которые базируются на применении криптографических методов. Разработка и исследование таких методов составляет предмет специальной науки – криптографии.

*Криптография* (тайнопись) – это раздел математики, в котором изучаются и разрабатываются системы изменения письма с целью сделать его непонятным для непосвященных лиц. Теоретические основы классической криптографии впервые были изложены Клодом Шенноном в конце 1940-х годов.

Основной криптографический метод защиты информации для обеспечения ее конфиденциальности – шифрование информации. При шифровании и расшифровке (дешифрировании) информации выполняется преобразование исходных (открытых) данных в зашифрованные и наоборот. Шифрование данных можно представить в виде следующих формул:

$$C = E_{k1}(M); \quad M' = D_{k2}(C),$$

где  $M$  – открытая информация (открытый текст);  $C$  – полученный в результате зашифровывания шифротекст (криптограмма);  $E$  – функция зашифровывания, выполняющая криптографические преобразования над исходным текстом;  $k1$  – параметр функции зашифрования, называемый ключом зашифровывания;  $M'$  – информация,



полученная в результате расшифровывания;  $k_2$  – параметр для расшифровывания информации;  $D$  – функция расшифровывания, выполняющая обратные относительно зашифровывания криптографические преобразования над шифротекстом.

В стандарте ГОСТ 28147-89 понятие «ключ» определено следующим образом: «Конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования, обеспечивающее выбор одного преобразования из совокупности всевозможных для данного алгоритма преобразований». Другими словами, ключ является уникальным элементом, с помощью которого можно варьировать результат работы алгоритма шифрования: один и тот же открытый текст при использовании различных ключей будет зашифрован по-разному.

Для того чтобы результат последовательного выполнения операций зашифровывания и расшифровывания совпал с исходным сообщением, необходимо выполнение двух условий:

- 1) функция  $D$  должна соответствовать функции  $E$ ;
- 2) ключ  $k_2$  должен соответствовать ключу  $k_1$ .

При отсутствии верного ключа  $k_2$  получить исходное сообщение  $M' = M$  невозможно, если для зашифровывания использовался криптографически стойкий алгоритм шифрования. Криптостойкость является количественной характеристикой алгоритма шифрования, определяемой требуемыми ресурсами для его вскрытия. Ресурсами являются:

- количество информации;
- время;
- память.

Совокупность этих трех величин характеризует конкретную атаку на конкретный алгоритм шифрования, а лучшая (с минимальным набором ресурсов) из возможных атак на алгоритм характеризует его криптостойкость. Кроме того, понятие криптостойкого алгоритма может быть определено следующим образом:

– алгоритм является криптографически стойким, если не существует каких-либо методов его вскрытия, кроме перебора всех возможных вариантов;

– размер ключа алгоритма при этом является достаточно большим для того, чтобы перебор вариантов стал невозможным при текущем уровне вычислительной техники.

Методы криптографии начали развиваться еще в глубокой древности. Известно, что еще в V веке до нашей эры тайнопись использовалась в Древней Греции. Очень распространена была простейшая (для настоящего времени) система шифрования – это замена каждого знака письма на другой знак по выбранному правилу. Юлий Цезарь, например, заменял в своих секретных письмах первую букву алфавита на четвертую, вторую – на пятую, последнюю – на третью и т.п., т.е. *A* на *D*, *B* на *E*, *Z* на *C* и т.п. Его наследник Октавиан Август заменял каждую непоследнюю букву алфавита на следующую, а последнюю на первую. Подобные шифры, называемые простой заменой или подстановкой, описаны, например, в рассказах «Пляшущие человечки» А.К. Дойла и «Золотой жук» Э. По.

Шифры простой замены легко поддаются расшифровке при знании исходного языка сообщения, так как каждый письменный язык характеризуется частотой встречаемости своих знаков. Например, в английском языке чаще всего встречается буква *E*, а в русском – *О*. Таким образом, в шифрованном подстановкой сообщения на русском языке самому частому знаку будет с большой вероятностью соответствовать буква *О*. При этом вероятность будет расти с ростом длины сообщения.

Усовершенствованные шифры-подстановки используют возможность заменять символ исходного сообщения на любой символ из заданного для него множества символов, что позволяет выравнивать частоты встречаемости различных знаков шифра, но подобные шифры удлинляют сообщение и замедляют скорость обмена информацией.

В шифрах-перестановках знаки сообщения специальным образом переставляются между собой, например, записывая сообщение в строки заданной длины и беря затем последовательность слов в столбцах в качестве шифра. Сообщение «ТЕОРИЯИНФОРМАЦИИ», используя строки длины 4, будет в шифрованном таким методом виде выглядеть как «ТИФАЕЯОЦОИРИРНИ», потому что при шифровании использовался следующий прямоугольник:

Т	Е	О	Р
И	Я	И	Н
Ф	О	Р	М
А	Ц	И	И

Шифры-перестановки в общем случае практически не поддаются дешифрированию – для их дешифрирования необходимо знать дополнительную информацию. Недостаток подобных шифров заключается в том, что если удастся каким-то образом расшифровать хотя бы одно сообщение, то в дальнейшем можно расшифровать и любое другое. Модификацией шифров-перестановок являются шифры-перестановки со словом-ключом, которое определяет порядок взятия слов-столбцов.

Системы с ключевым словом или просто ключом, известные с XVI века, широко применяются до сих пор. Их особенностью являются два уровня секретности. Первый уровень – это собственно способ составления кода, который постоянно известен лицам, использующим данный шифр. Второй уровень – это ключ, который посылается отдельно от основного сообщения по особо защищенным каналам и без которого расшифровка основного сообщения невозможна. Наиболее простой способ использования ключа хорошего шифра следующий: под символами сообщения записывается раз за разом ключ, затем номера соответствующих знаков сообщения и ключа складываются. Если полученная сумма больше общего числа знаков, то от нее отнимается это общее число знаков. Полученные числа будут номерами символов кода. С ростом дли-

ны ключа трудоемкость дешифрирования подобного шифра стремительно растет. Например, рассмотренное ранее сообщение с ключом «КИБЕРНЕТИКА» в зашифрованном виде будет выглядеть как «ЮОРЦЬНОБЮЪСШЙШОЪ»:

Т	Е	О	Р	И	Я	И	Н	Ф	О	Р	М	А	Ц	И	И
20	6	16	18	10	33	10	15	22	16	18	14	1	24	10	10
К	И	Б	Е	Р	Н	Е	Т	И	К	А	К	И	Б	Е	Р
12	10	2	6	18	15	6	20	10	12	1	12	10	2	6	18
32	16	18	24	28	15	16	2	32	28	19	26	11	26	16	28
Ю	О	Р	Ц	Ъ	Н	О	Б	Ю	Ъ	С	Ш	Й	Ш	О	Ъ

Если в качестве ключа использовать случайную последовательность, то получится нераскрываемый шифр. Проблема такого шифра – это способ передачи ключа.

В информационных сетях использование традиционных систем шифрования с ключом затруднено необходимостью иметь специальный особо защищенный способ для передачи ключа. В 1976 году У. Диффи и М. Хеллман – инженеры-электрики из Станфордского университета, а также студент Калифорнийского университета Р. Меркль предложили новый принцип построения криптосистем, не требующий передачи ключа принимающему сообщению и сохранения в тайне метода шифрования.

Алгоритмы шифрования можно разделить на две категории:

1) алгоритмы симметричного шифрования, в которых  $k_2 = k_1 = k$ ;

2) алгоритмы асимметричного шифрования, в которых ключ зашифровывания  $k_1$  вычисляется из ключа  $k_2$  таким образом, что обратное вычисление невозможно, например, по формуле  $k_1 = a^{k_2} \bmod p$ , где  $a$  и  $p$  – параметры алгоритма.

Симметричное шифрование делится на два вида: блочное и потоковое. При блочном шифровании информация разбивается на блоки фиксированной длины, после чего они поочередно шифруются. Алгоритмы потокового шифрования обрабатывают данные

побитно или посимвольно (можно считать, что данные разбиваются на блоки единичной длины).

Большинство симметричных алгоритмов работают следующим образом: над шифруемым текстом выполняется некоторое преобразование с участием ключа шифрования, которое повторяется определенное число раз (раундов).

При использовании алгоритмов симметричного шифрования каждая из обменивающихся информацией сторон должна иметь копию общего секретного ключа, что создает сложнейшую проблему управления ключами. Этого недостатка лишены алгоритмы ассиметричного шифрования (однако у них есть свои собственные недостатки, например, они более медленные).

В ассиметричных алгоритмах (алгоритмах с открытым ключом) используются два ключа: открытый и секретный. Открытый ключ может быть опубликован в справочнике наряду с именем пользователя. В результате любой желающий может зашифровать с его помощью свое сообщение и послать закрытую информацию владельцу соответствующего секретного ключа. Расшифровать посланное сообщение сможет только тот, у кого есть секретный ключ.

В основе алгоритма шифрования с открытым ключом лежит идея использования легко осуществимого на стадии шифрования математического преобразования, которое сложно было бы обратить (без знания специальной секретной информации) для реализации второй стадии алгоритма, т.е. расшифрования. Преобразование, обладающее указанным свойством, называется односторонней функцией или функцией-ловушкой.

Имеется набор широко известных и всесторонне изученных односторонних функций. К ним относятся функции, изученные при решении задачи разложения целых чисел на множители, проблемы вычисления дискретных логарифмов или вычисления квадратных корней по модулю составного числа. Отметим, что применяемые

функции являются односторонними только в вычислительном отношении, т.е. имея достаточно большие компьютерные мощности, их вполне можно обратить, причем быстрее, чем найти секретный ключ в результате полного перебора.

Рассмотрим построение ассиметричных алгоритмов на примере системы, разработанной в 1978 году американцами Р. Ривестом, Э. Шамиром и Л. Адлеманом. По их именам эта система получила название RSA. Вообще она является первой и наиболее известной системой с открытым ключом. Алгоритм RSA заключается в выполнении следующих действий:

- 1) выбрать два больших простых числа  $p_1, p_2$ ;
- 2) вычислить  $r = p_1 p_2$ ;
- 3) вычислить функцию Эйлера  $\phi(r) = (p_1 - 1)(p_2 - 1)$ ;
- 4) определить случайное число  $a < \phi(r)$ , взаимнопростое с  $\phi(r)$ ;
- 5) найти число  $\alpha$ , удовлетворяющее уравнению:  
 $a\alpha \equiv 1 \pmod{\phi(r)}, 0 < \alpha < \phi(r)$ ;
- 6) зашифровать сообщение  $m$  ( $m < r$ ):

$$m_1 \equiv m^a \pmod{r}.$$

Дешифровать сообщение  $m_1$ :

$$m \equiv m_1^\alpha \pmod{r}.$$

Задача нахождения секретного ключа в алгоритме RSA будет иметь такую же сложность, что и задача разложения числа на простые множители, поскольку здесь использовалась соответствующая односторонняя функция. Рассмотрим алгоритм RSA на примере. Пусть  $p_1 = 7$  и  $p_2 = 23$ . Тогда  $r = p_1 p_2 = 7 \times 23 = 161$ ;  $\phi(r) = \phi(161) = (7 - 1) \times (23 - 1) = 6 \times 22 = 132$ ;  $a = 7$ ;  $\alpha = 19$ . Если кто-то захочет отправить секретное сообщение  $m = 3$ , то он должен будет преобразовать его в  $m_1 \equiv 3^7 \equiv 94 \pmod{161}$ . При получении  $m_1 = 94$  оно будет дешифровано по формуле  $m = 94^{19} \equiv 3 \pmod{161}$ .

Для генерирования цифровой подписи могут использоваться некоторые из ассиметричных алгоритмов. Цифровой подписью называют блок данных, сгенерированный с использованием неко-

торого секретного ключа. При этом с помощью открытого ключа можно проверить, что данные были действительно сгенерированы с помощью этого секретного ключа.

Цифровые подписи используются для подтверждения, что сообщение пришло действительно от заявленного отправителя (в предположении, что лишь отправитель обладает секретным ключом, соответствующим его открытому ключу). Также подписи используются для проставления штампа времени на документах: сторона, которой доверяют, подписывает документ со штампом времени с помощью своего секретного ключа и, таким образом, подтверждает, что документ уже существовал в момент, объявленный в штампе времени.

Цифровые подписи также можно использовать для удостоверения (сертификации) принадлежности документа определенному лицу: открытый ключ и информация относительно его принадлежности подписываются стороной, которой доверяют. При этом доверять подписывающей стороне можно на основании того, что ее ключ был подписан третьей стороной.

Таким образом, возникает иерархия доверия. Очевидно, что некоторый ключ должен быть корнем иерархии (т.е. ему доверяют не потому, что он кем-то подписан, а потому что априорно известно, что ему можно доверять). В централизованной инфраструктуре ключей имеется очень небольшое количество корневых ключей сети (например, облеченные полномочиями государственные агентства, которые также называют сертификационными агентствами). В распределенной инфраструктуре нет необходимости иметь универсальные для всех корневые ключи, и каждая из сторон может доверять своему набору корневых ключей (скажем, своему собственному ключу и ключам, ею подписанным). Эта концепция носит название сети доверия.

Цифровая подпись документа обычно создается следующим образом: из документа генерируется так называемый дайджест

и к нему добавляется информация о том, кто подписывает документ, штамп времени и прочее. Получившаяся строка далее зашифровывается секретным ключом подписывающего с использованием того или иного алгоритма. Получившийся зашифрованный набор бит и представляет собой подпись, к которой обычно прикладывается открытый ключ подписавшей стороны. Получатель сначала решает для себя, доверяет ли он тому, что открытый ключ принадлежит именно тому, кому должен принадлежать (с помощью сети доверия или априорного знания), и затем дешифрует подпись с помощью открытого ключа. Если подпись нормально дешифрировалась и ее содержимое соответствует документу (дайджест и др.), то сообщение считается подтвержденным.

Свободно доступны несколько методов создания и проверки цифровых подписей. Наиболее известным является метод, основанный на применении RSA.

Также применяются криптографические хэш-алгоритмы для генерации дайджеста сообщения при создании цифровой подписи. Они основаны на применении хэш-функций, отображающих сообщение в имеющее фиксированный размер хэш-значение таким образом, что все множество возможных сообщений распределяется равномерно по множеству хэш-значений. При этом криптографическая хэш-функция делает это таким образом, что практически невозможно подогнать документ к заданному хэш-значению. Криптографические хэш-функции обычно производят значения длиной в 128 и более бит (это число значительно больше, чем количество сообщений, которые когда-либо будут существовать в мире). Широко известны такие хэш-алгоритмы, как MD5 и SHA.

## **2.4. Контрольные вопросы**

1. Основные понятия сжатия информации.
2. Энтропийные методы сжатия.



3. Методы контекстного моделирования.
4. Словарные методы сжатия.
5. Методы сжатия с преобразованием блоков.
6. Методы сжатия с потерями.
7. Помехи.
8. Расстояние Хемминга.
9. Коды для обнаружения ошибок.
10. Коды для исправления ошибок.
11. Введение в информационную безопасность.
12. Криптография.
13. Классические шифры.
14. Стойкость шифров.
15. Алгоритмы симметричного шифрования.
16. Алгоритмы асимметричного шифрования.

### **3. ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

#### **3.1. Информационные характеристики дискретного источника информации**

##### ***3.1.1. Порядок выполнения лабораторной работы***

1. Ознакомиться с основными сведениями об информационных характеристиках дискретных случайных систем.
2. Получить задание на выполнение лабораторной работы.
3. Выполнить расчеты информационных характеристик дискретных случайных систем.
4. Сделать выводы о свойствах информационных характеристик дискретных случайных систем.
5. Оформить отчет о выполнении лабораторной работы.
6. Ответить на контрольные вопросы.

##### ***3.1.2. Контрольные вопросы***

1. Энтропия дискретной случайной системы.
2. Основные свойства энтропии.
3. Условия минимальной энтропии.
4. Условия максимальной энтропии.
5. Энтропия объединения независимых систем.
6. Энтропия объединения зависимых систем.
7. Количество информации.
8. Объем информации.

9. Связь количества информации в сообщении о событии с вероятностью события.

10. Взаимная информация.

11. Взаимная информация независимых и зависимых систем.

12. Связь взаимной информации с энтропией объединения.

### ***3.1.3. Задания на лабораторную работу***

1. Рассчитать значения функции  $H(p) = -p \log_2 p$ . Значения аргумента функции изменяются от 0 до 1 с шагом 0,05. Построить график функции  $H(p)$ .

2. Задать систему с достоверным состоянием и определить ее энтропию.

3. Задать систему с равномерным распределением вероятностей состояний (число состояний должно быть не менее четырех) и определить ее энтропию. Для проверки: задать систему с таким же количеством событий, но не с одинаковыми вероятностями и определить ее энтропию.

4. Для заданных из табл. 3.1 систем  $X$  и  $Y$  с состояниями, определяемыми символами алфавита  $A = \{a, b, c, d\}$ , определить:

4.1) вероятности состояний систем  $X$  и  $Y$ ;

4.2) энтропии независимых систем  $X$  и  $Y$ ;

4.3) условные энтропии систем  $X$  и  $Y$ , считая, что каждому символу одной системы соответствует соответствующий по индексу символ второй системы;

4.4) энтропию объединения независимых систем  $X$  и  $Y$ ;

4.5) энтропию объединения зависимых систем  $X$  и  $Y$ ;

4.6) взаимную информацию систем  $X$  и  $Y$ ;

4.7) объем информации для систем  $X$  и  $Y$ , считая, что каждый символ алфавита  $A$  кодируется двумя символами вторичного алфавита.

Таблица 3.1

## Состояния систем X и Y

№ п/п	X	Y
1	<i>abcababacabbacbbacbbddadadaa</i>	<i>acabacabbacbbaccabcbacbbddadcdaa</i>
2	<i>bcabbcdabacbcacbbddcbbacbbdbdadaac</i>	<i>cacaddabbbaaccaabacadaabacabbacbbacbb</i>
3	<i>acabacabbacbbaccabcbacbbddadadaa</i>	<i>acdabbabacabbacbbacbbacbbddadada</i>
4	<i>abcaacaabbacbaacccabacbbacbbddadd</i>	<i>bcabbcdabbacbbddcbbacbbdbdadabcd</i>
5	<i>acdaddabacabbacbbacbbacbbddadada</i>	<i>dabcadabacabbacbbacbbacbbddadadac</i>
6	<i>cacaddabbbaaccaabcaaaabacabbacbbacbb</i>	<i>bcacbbbaabacabbacbbacbbacdbbdddadada</i>
7	<i>dbdacdabacabbacbbacbbacbbddadadac</i>	<i>abcaadaabbacbaacccabadddbacbbddadd</i>
8	<i>babcbabacabbacbbacbbacdbbdddadadac</i>	<i>dddadabbabbacbbacbbacbbadcaaddadddaa</i>
9	<i>aacababacacdbbacbbddccbbacbbddadadbb</i>	<i>bccbcbbacabbacbbadaaaaaaabddadadcc</i>
10	<i>ddcaadabbabbacbbacbbacbbddadddaa</i>	<i>bbbcbdbbbabacddacdbacbbadadadddaa</i>
11	<i>abcccdbacabbacbbadbbdacbbddadadacc</i>	<i>abcadaabacabbacbbacbbacbbddadadaa</i>
12	<i>abcbbdbbacabbacddacdbbacbbdadadddd</i>	<i>aacabaacacdbbacbbddcbbacbbddadadbb</i>

## 3.2. Оптимальное кодирование информации

### 3.2.1. Порядок выполнения лабораторной работы

1. Ознакомиться с основными сведениями об оптимальном кодировании.
2. Получить задание на выполнение лабораторной работы.
3. Выполнить необходимые расчеты.
4. Сделать выводы о свойствах кодов.
5. Оформить отчет о выполнении лабораторной работы.
6. Ответить на контрольные вопросы.

### 3.2.2. Контрольные вопросы

1. Кодирование сообщений.
2. Декодирование сообщений.
3. Префиксные коды.
4. Неравенство Крафта.
5. Нижний предел средней длины кода.
6. Принципы построения оптимальных кодов.

7. Условие оптимальности равномерного кода.
8. Кодирование Шеннона – Фано.
9. Кодирование Хаффмана.
10. Блочное кодирование.

### 3.2.3. Задания на лабораторную работу

1. Определить вероятности появления символов заданного источника с алфавитом  $A = \{a, b, c, d\}$  из табл. 3.2 (использовать частоты символов).
2. Определить энтропию сообщения.
3. Построить коды Шеннона – Фано и Хаффмана для отдельных символов.
4. Построить коды Шеннона – Фано и Хаффмана для двухбуквенных блоков символов.
5. Закодировать сообщение.
6. Декодировать сообщение.
7. Определить среднюю длину и избыточность для всех кодов.
8. Определить наиболее оптимальное кодирование для источника сообщений.

Таблица 3.2

Сообщения дискретного источника

№ п/п	Сообщение
1	<i>abcacaabacabbacbbacbbacbbddadadaa</i>
2	<i>bcabdc dabacbbacbbddcbbacbbdbdadaac</i>
3	<i>acabacabbacbbaccabcabbacbbddadadaa</i>
4	<i>abcaadaabbacbaaccacacbbacbbddadd</i>
5	<i>aadcaddabacabbacbbacbbacbbddadada</i>
6	<i>cbcaddabbbaccaabcaaaabacabbacbbacbb</i>
7	<i>dbdabdabacabbacbbacbbacbbddadadac</i>
8	<i>bbcbbaabacabbacbbacbbacbbddadadac</i>
9	<i>aacabadacdbbacbbddcbbacbbddadadbb</i>
10	<i>dcdaadabbabbacbbacbbacbbddadddaa</i>
11	<i>abcchcbacabbacbbaddbdacbbddadadcc</i>
12	<i>abcbdbbbacabbacddacdbbaccbbadadadd</i>

### **3.3. Коды для обнаружения и исправления ошибок**

#### ***3.3.1. Порядок выполнения лабораторной работы***

1. Ознакомиться с основными сведениями по помехоустойчивому кодированию.
2. Получить задание на выполнение лабораторной работы.
3. Выполнить необходимые расчеты.
4. Сделать выводы по результатам выполнения лабораторной работы.
5. Оформить отчет о выполнении лабораторной работы.
6. Ответить на контрольные вопросы.

#### ***3.3.2. Контрольные вопросы***

1. Помехоустойчивые коды.
2. Коды с проверкой на четность, с удвоением, с постоянным весом, инверсные.
3. Корректирующие коды.
4. Кодовое расстояние.
5. Линейные коды.
6. Циклические коды.
7. Образующий многочлен.
8. Образующая и проверочная матрицы циклического кода.
9. Кодирование и декодирование циклического кода.
10. Выявление и исправление ошибки в циклическом коде.

#### ***3.3.3. Задания на лабораторную работу***

Дан алфавит латинских символов  $A = \{a, \dots, z\}$ .

1. Определить кодовые комбинации символов алфавита.

2. Выбрать неприводимый полином из табл. 3.2.
3. Выполнить кодирование сообщения из табл. 3.3 с использованием циклического кода.
4. Выполнить декодирование закодированного сообщения.
5. Внести ошибку в закодированное сообщение.
6. Выполнить декодирование закодированного сообщения с ошибкой.
7. Выполнить кодирование сообщения с использованием кодов с проверкой на четность, с удвоением, с постоянным весом, инверсные.

Таблица 3.3

**Сообщения дискретного источника**

№ п/п	Сообщение
1	<i>Abcakaabacabbacbbacccbbacccbddadadaa</i>
2	<i>Ecaelcdabacbbacbbddcbacccbbdbdadadae</i>
3	<i>Dddbhjgffsdkkkdfffghgdhgfjhjfhghkfgjk</i>
4	<i>Abcazzgabbacbaacljabacccbbacccssddadd</i>
5	<i>Aadhaddabacabbacbmnnccbbaccjjidadada</i>
6	<i>Cncaddabtdeccaabcaassxacabbacbbacbb</i>
7	<i>Dbdandaxghabbacbbacnvvbacccbddadadac</i>
8	<i>Bsbiiabacabbacbbaccllbacdbffdadadac</i>
9	<i>Alcabaarrcdbbacbdewcbbacqobddadadbo</i>
10	<i>Dppuadabbttirbacbbacyubacccbuuwdadada</i>
11	<i>Abmcppjbacabnscbbadsmdacccbmndadadacj</i>
12	<i>Abcbbgqqacabqqwddacerbacvbuydadadydd</i>

### 3.4. Симметричное и асимметричное шифрование информации

#### 3.4.1. Порядок выполнения лабораторной работы

1. Ознакомиться с основными сведениями по шифрованию данных.
2. Получить задание на выполнение лабораторной работы.

3. Выполнить необходимые расчеты для выполнения задания с помощью программы Microsoft Excel.
4. Сделать выводы по результатам выполнения лабораторной работы.
5. Оформить отчет о выполнении лабораторной работы.
6. Ответить на контрольные вопросы.

### ***3.4.2. Контрольные вопросы***

1. Модель шифрования.
2. Ключ согласно ГОСТ.
3. Криптографическая стойкость алгоритма шифрования.
4. Принцип работы симметричного алгоритма шифрования.
5. Шифры простой замены.
6. Шифры замены с множеством подстановочных символов.
7. Шифры-перестановки.
8. Шифры-перестановки с ключом.
9. Асимметричные алгоритмы шифрования.
10. Алгоритм RSA.

### ***3.4.3. Задания на лабораторную работу***

1. Выбрать сообщение из табл. 3.4.
2. Выполнить шифрование и дешифрирование заданного сообщения с помощью метода симметричного шифрования с ключом (использовать латинский алфавит).
3. Выполнить шифрование и дешифрирование заданного сообщения по методу RSA. Исходным кодом символа считать его порядковый номер в латинском алфавите.



Таблица 3.4

## Сообщения для шифрования

№ п/п	Сообщение
1	<i>abcaraabacabbacbbacccbaccbbddadadaa</i>
2	<i>bkabbcadabacbbacbbddcbbaccbbdbdadaac</i>
3	<i>ajabacabbacbbaccabcabbaccbbddadadaa</i>
4	<i>abcaahaabbacbaacccabaccbbacbbddadd</i>
5	<i>agddaddabacabbacbbacccbaccbbddadada</i>
6	<i>cncaddabbaccaabcaaaabacabbacbbacbb</i>
7	<i>abcamaabacabbacbbacccbaccbbddadadaa</i>
8	<i>bfabbcgabacbbacbeddcbbaccbbdbdadaac</i>
9	<i>atabacabbacbbaccabcabbaccbbddadadaa</i>
10	<i>abcaasaabbacbaacccabaccbbaccbbddadd</i>
11	<i>afddaddabacabbacbbacccbaccbbddadada</i>
12	<i>ckcaddabbaccaabcaaaabacabbacbbacbb</i>

## ПРИЛОЖЕНИЕ

### ЭЛЕМЕНТЫ ТЕОРИИ ВЕРОЯТНОСТЕЙ

#### *Приложение 1*

#### **Основные понятия и определения теории вероятностей**

Многие положения теории информации базируются на математическом аппарате теории вероятностей, которая изучает закономерности случайных явлений, т.е. таких явлений, которые при многократном повторении в одних и тех же условиях протекают каждый раз несколько по-иному.

Основными понятиями теории вероятностей являются понятия события и вероятности события. Событие – это всякий факт, который может произойти или не произойти в результате некоторого эксперимента. Различают достоверные, невозможные и случайные события. *Достоверным* событием называют событие, которое обязательно произойдет в результате эксперимента. *Невозможным* событием называют событие, которое заведомо не произойдет в результате эксперимента. *Случайным* событием называют событие, которое может либо произойти, либо не произойти в результате эксперимента.

Вероятность  $P(A)$  события  $A$  есть численная мера объективной возможности его наступления при осуществлении определенного комплекса условий эксперимента  $Q$ . Если вероятность наступления события  $A$  равна  $p$ , то это записывают в виде  $P(A) = p$ . Вероятность имеет следующие свойства:

- 1) вероятность достоверного события равна единице;
- 2) вероятность невозможного события равна нулю;
- 3) вероятность случайного события  $A$  есть положительное число из интервала от 0 до 1.

Оценить значение вероятности события  $A$  можно с помощью относительной частоты

$$v(A) = m/n,$$

где  $n$  – число независимых испытаний;  $m$  – число испытаний, в которых появилось событие  $A$ .

## Приложение 2

### Основные теоремы теории вероятностей

#### Теорема П1. Сложение вероятностей несовместных событий.

Вероятность суммы несовместных событий, т.е. событий, из которых в результате эксперимента может произойти только одно, равна сумме их вероятностей:

$$P\left(\sum_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i).$$

Если события  $A_1, A_2, \dots, A_n$  образуют полную группу несовместных событий, т.е. в результате эксперимента одно из событий должно произойти, то сумма их вероятностей равна 1:

$$\sum_{i=1}^n P(A_i) = 1.$$

В частности, сумма вероятностей двух противоположных событий равна 1:

$$P(A) + P(\bar{A}) = 1.$$

#### Теорема П2. Сложение вероятностей совместных событий.

Вероятность суммы совместных (произвольных) событий определяется через вероятность произведений этих событий, взятых по одному, по два, по три и т.д. по формуле:

$$P\left(\sum_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i) - \sum_{i=1}^{n-1} \sum_{j=i+1}^n P(A_i A_j) + \dots + (-1)^n P\left(\prod_{i=1}^n A_i\right).$$

При этом имеет место неравенство:

$$P(A_1 + A_2 + \dots + A_n) \leq P(A_1) + P(A_2) + \dots + P(A_n).$$

**Теорема П3. Произведение произвольного числа событий.**

Вероятность произведения произвольного числа событий определяется через вероятности суммы этих событий, взятых по одному, по два, по три и т.д. по формуле

$$P\left(\prod_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i) - \sum_{i=1}^{n-1} \sum_{j=i+1}^n P(A_i + A_j) + \dots + (-1)^n P\left(\sum_{i=1}^n A_i\right).$$

Вероятность события  $A$ , вычисленная при условии, что имело место событие  $B$ , называется условной вероятностью события  $A$  и обозначается  $P(A | B)$ . Событие  $A$  называется независимым от события  $B$ , если вероятность события  $A$  не зависит от того, произошло событие  $B$  или нет:

$$P(A | B) = P(A).$$

И обратно, событие  $A$  называется зависимым от события  $B$ , если вероятность события  $A$  меняется в зависимости от того, произошло событие  $B$  или нет:

$$P(A | B) \neq P(A).$$

**Теорема П4.** Вероятность произведения двух событий равна произведению вероятности одного из них на условную вероятность другого, вычисленную при условии, что первое имело место:

$$P(AB) = P(A) P(B | A).$$

Если событие  $A$  не зависит от события  $B$ , то и событие  $B$  не зависит от события  $A$ . Вероятность произведения двух независимых событий  $A$  и  $B$  равна произведению вероятностей этих событий:

$$P(AB) = P(A) P(B).$$

**Теорема П5.** Вероятность произведения независимых событий равна произведению вероятностей этих событий:

$$P(A_1, A_2, \dots, A_n) = P(A_1) P(A_2), \dots, P(A_n).$$

Следствием теорем сложения и умножения вероятностей является формула полной вероятности:

$$P(A) = \sum_{i=1}^n P(H_i)P(A | H_i),$$

где  $A$  – произвольное случайное событие;  $H_1, H_2, \dots, H_n$  – несовместные события, образующие полную группу (гипотезы).

**Теорема Пб. Теорема гипотез (формула Байеса).** Пусть  $H_1, H_2, \dots, H_n$  – полная группа несовместных событий. Тогда, если произошло событие  $A$ , то имеет место равенство

$$P(H_i | A) = \frac{P(H_i)P(A | H_i)}{\sum_{i=1}^n P(H_i)P(A | H_i)}.$$

### **Случайные величины и их распределения**

Случайной величиной называется величина  $X$ , которая в результате опыта может принимать одно из значений  $x_1, x_2, \dots, x_i, \dots, x_n$ , образующих полную группу несовместных событий, причем неизвестно заранее, какое именно.

По характеру реализаций случайные величины делят на дискретные и непрерывные. Дискретной случайной величиной называют случайную величину  $X$ , которая принимает отдельные, изолированные возможные значения  $x_i$  с определенными вероятностями  $p_i$ .

Законом распределения случайной величины  $X$  называется совокупность пар чисел  $(x_i, p_i)$ , где  $x_i$  – возможные значения случайной величины;  $p_i$  – вероятности, с которыми она принимает эти значения. При этом

$$\sum_{i=1}^n P(X = x_i) = \sum_{i=1}^n p_i = 1.$$

Простой формой задания закона распределения дискретной случайной величины является ряд распределения. Он описывается в виде таблицы, в которой перечислены возможные значения случайной величины и соответствующие им вероятности.

Непрерывной случайной величиной называют случайную величину, которая может принимать все значения из некоторого конечного или бесконечного промежутка. Число возможных значений случайной непрерывной величины бесконечно.

Функцией распределения непрерывной случайной величины называют функцию  $F(x)$ , определяющую вероятность того, что случайная величина  $X$  в результате испытания примет значение, не больше  $x$ :

$$F(x) = P(X \leq x).$$

Очевидно, что функция распределения является неубывающей и ее значения принадлежат отрезку  $[0, 1]$ . Из определения функции  $F(x)$  следует, что вероятность попадания значения случайной величины в интервал от  $x_1$  до  $x_2$  равна приращению функции распределения на этом интервале:

$$P(x_1 \leq X \leq x_2) = F(x_2) - F(x_1).$$

Отсюда можно сделать следующие заключения:

1) вероятность того, что непрерывная случайная величина примет одно определенное значение, равна 0;

$$2) F(-\infty) = 0;$$

$$3) F(\infty) = 1.$$

Так как функция распределения  $F(x)$  в силу непрерывности случайной величины  $X$  предполагается непрерывной и дифференцируемой, то можно найти ее производную:

$$p(x) = \lim_{\Delta x \rightarrow 0} \frac{P(x \leq X < x + \Delta x)}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{F(x + \Delta x) - F(x)}{\Delta x} = \frac{dF(x)}{dx}.$$

Функцию  $p(x)$  называют плотностью распределения или плотностью вероятности непрерывной случайной величины. Она имеет следующие свойства:

1) плотность распределения есть неотрицательная функция;

2) имеет место равенство:

$$\int_{-\infty}^{\infty} p(x) dx = 1;$$

3) плотность распределения определяет функцию распределения случайной величины по формуле:

$$F(x) = \int_{-\infty}^x p(y) dy = 1;$$

4) вероятность попадания случайной величины в интервал  $[a, b]$  определяется по формуле:

$$P(a \leq X < b) = \int_a^b p(x) dx = 1.$$



### **Числовые характеристики случайной величины**

Несмотря на то, что наиболее полными характеристиками случайной величины являются законы их распределения, в ряде случаев не требуется столь исчерпывающей информации, достаточно иметь лишь некоторые ее числовые характеристики. Различают характеристики положения и характеристики рассеивания и вероятностных взаимодействий.

Основными характеристиками положения случайной величины являются математическое ожидание, мода и медиана. Математическое ожидание дискретной случайной величины  $M[X]$  есть сумма произведений всех возможных значений случайной величины  $x_i$  на их вероятности  $p_i$ :

$$M[X] = \sum_{i=1}^n x_i p_i .$$

Для непрерывной случайной величины математическое ожидание вычисляется по формуле

$$M[X] = \int_{-\infty}^{\infty} x_i p(x) dx = \int_{-\infty}^{\infty} x_i dF(x) .$$

Модой дискретной случайной величины называется ее наиболее вероятное значение. Модой непрерывной случайной величины называется ее значение, при котором плотность вероятности принимает максимальное значение. Медианой случайной величины  $X$  называется такое ее значение  $Y$ , для которого выполняется равенство

$$P(X < Y) = P(X > Y) = 0,5.$$

Основными характеристиками рассеивания случайной величины являются ее начальные и центральные моменты. Начальным

моментом  $k$ -го порядка  $\alpha_k[X]$  случайной величины  $X$  называется математическое ожидание  $k$ -й степени от этой случайной величины:

$$\alpha_k [X] = M [X^k].$$

Для определения центрального момента введем понятие центрированной случайной величины  $X^\circ$ . Центрированной случайной величиной  $X^\circ$ , соответствующей случайной величине  $X$ , называется отклонение случайной величины от ее математического ожидания  $M[X] = m$ , т.е.  $X^\circ = X - m$ . Центральным моментом  $k$ -го порядка  $\mu_k [X]$  случайной величины  $X$  называется математическое ожидание  $k$ -й степени центрированной случайной величины  $X^\circ$ .

Второй центральный момент называется дисперсией случайной величины. Дисперсия характеризует рассеивание значений случайной величины. Она обозначается следующим образом:

$$\mu_2 [X] = D [X] = D_x = \sigma^2.$$

Для дискретной случайной величины дисперсия вычисляется по формуле:

$$D[X] = \sum_{i=1}^n (x_i - m)^2 p_i ,$$

для непрерывной случайной величины:

$$D[X] = \int_{-\infty}^{\infty} (x_i - m)^2 p(x) dx .$$

Величина  $\sigma = \sqrt{D[X]}$  называется среднее квадратическое, или стандартное, отклонение случайной величины.

Для оценки степени независимости случайных величин  $X$  и  $Y$  вводится числовая характеристика, называемая корреляционным (ковариационным) моментом случайных величин  $X$  и  $Y$ . Это есть число

$$K(x, y) = M \{ (X - M [X]) (Y - M [Y]) \} = M [XY] - M [X] M [Y].$$

Корреляционный момент характеризует вероятностную зависимость между случайными величинами.

## **Законы распределения непрерывных случайных величин**

Рассмотрим несколько распространенных законов распределения случайных величин.

1. *Равномерное распределение.* Непрерывная случайная величина называется равномерно распределенной на интервале  $[a, b]$ , если плотность ее распределения имеет постоянное значение. Математическое ожидание и дисперсия равномерно распределенной случайной величины равны:

$$M[X] = \frac{b+a}{2}; \quad D[X] = \frac{(b-a)^2}{12}.$$

2. *Показательное распределение.* Показательным (экспоненциальным) распределением случайной величины называют распределение случайной величины, которое описывается плотностью распределения:

$$p(x) = \begin{cases} 0, & x < 0; \\ \lambda \exp(-\lambda x), & x \geq 0, \end{cases}$$

где  $\lambda$  – положительная постоянная величина. Математическое ожидание и дисперсия в этом случае определяются по формулам

$$M[X] = \frac{1}{\lambda}; \quad D[X] = \frac{1}{\lambda^2}.$$

3. *Нормальное распределение.* Нормальный закон распределения (закон Гаусса), наиболее часто встречающийся на практике закон распределения. На его основе описывают случайные возмущения и отклонения основных характеристик процессов и систем,

ошибки измерений и т.д. Этот закон является предельным законом, к которому приближаются другие законы распределения при весьма часто встречающихся типичных условиях. Непрерывная случайная величина называется распределенной по нормальному закону, если ее плотность вероятности определяется выражением:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-(x - m_x)^2 / (x - m_x)^2\right],$$

где  $m_x$  и  $\sigma$  – параметры распределения, представляющие собой математическое ожидание и среднеквадратическое отклонение соответственно.

## СПИСОК ЛИТЕРАТУРЫ

1. *Белов, В.М.* Теория информации. Курс лекций: учебное пособие для студентов вузов / В.М. Белов, С.Н. Новиков, О.И. Солонская. – М.: Горячая линия-Телеком, 2014. – 143 с.
2. *Вернер, М.* Основы кодирования: учебник для вузов / М. Вернер. – М.: Техносфера, 2004. – 288 с.
3. *Гашков, С.Б.* Криптографические методы защиты информации: учебное пособие / С.Б. Гашков, Э.А. Применко, М.А. Черепнев. – М.: Академия, 2010. – 304 с.
4. *Дмитриев, В.И.* Прикладная теория информации: учебник для студентов вузов по спец. «Автоматизированные системы обработки информации и управления» / В.И. Дмитриев. – М.: Высшая школа, 1989. – 320 с.
5. *Кудряшов, Б.Д.* Основы теории кодирования: учебное пособие / Б.Д. Кудряшов. – СПб.: БХВ-Петербург, 2016. – 400 с.
6. *Кудряшов, Б.Д.* Теория информации: учебник для вузов / Б.Д. Кудряшов. – СПб.: Питер, 2009. – 320 с.
7. *Морелос-Сарагоса, Р.* Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение: пер. с англ. В.Б. Афанасьева / Р. Морелос-Сарагоса. – М.: Техносфера, 2006. – 319 с.
8. *Лидовский, В.В.* Теория информации: учебное пособие / В.В. Лидовский. – М.: Компания Спутник+, 2004. – 111 с.
9. *Ляшева, С.А.* Теория вероятностей и математическая статистика: учебное пособие / С.А. Ляшева, М.П. Шлеймович. – Казань: Изд-во КНИТУ-КАИ, 2015. – 109 с.

10. *Ляшева, С.А.* Теория информации: учебное пособие / С.А. Ляшева, И.С. Ризаев, М.П. Шлеймович. – Казань: Изд-во Казан. гос. техн. ун-та, 2013. – 104 с.

11. *Сэломон, Д.* Сжатие данных, изображений и звука: пер. с англ. В.В. Чепыжова / Д. Сэломон. – М.: Техносфера, 2006. – 368 с.

## ОГЛАВЛЕНИЕ

1. ОСНОВЫ ТЕОРИИ ИНФОРМАЦИИ И КОДИРОВАНИЯ .....	3
1.1. Информационные характеристики случайных систем .....	3
1.2. Информационные характеристики каналов связи .....	18
1.3. Кодирование информации .....	31
1.4. Контрольные вопросы .....	45
2. МЕТОДЫ РЕШЕНИЯ ПРАКТИЧЕСКИХ ЗАДАЧ КОДИРОВАНИЯ ИНФОРМАЦИИ .....	46
2.1. Сжатие информации .....	46
2.2. Помехоустойчивое кодирование информации .....	75
2.3. Шифрование информации .....	88
2.4. Контрольные вопросы .....	96
3. ЛАБОРАТОРНЫЙ ПРАКТИКУМ .....	98
3.1. Информационные характеристики дискретного источника информации .....	98
3.2. Оптимальное кодирование информации .....	100
3.3. Коды для обнаружения и исправления ошибок .....	102
3.4. Симметричное и асимметричное шифрование информации .....	103
ПРИЛОЖЕНИЕ. Элементы теории вероятностей .....	106
Список литературы .....	117

ЛЯШЕВА Стелла Альбертовна  
ШЛЕЙМОВИЧ Михаил Петрович  
ЯХИНА Зухра Талгатовна

## ТЕОРИЯ ИНФОРМАЦИИ И КОДИРОВАНИЯ

*Учебно-методическое пособие*

Редактор Н.И. Данич  
Компьютерная верстка и дизайн обложки – С.В. Филаретов

---

Подписано к печати 22.12.20.  
Формат 60×84 1/16. Бумага офсетная. Печать цифровая.  
Усл. печ. л. 6,98. Тираж 140. Заказ Д 95.

---

Издательство КНИТУ-КАИ  
420111, Казань, К.Маркса, 10

ISBN 978-5-7579-2493-1

