Министерство образования Российской Федерации
Новосибирский государственный университет
Высший колледж информатики

В. Н. Потапов

Теория информации. Кодирование дискретных вероятностных источников

Учебное пособие

Новосибирск
1999

Учебное пособие предназначено для студентов технического факультета НГУ, а также студентов-математиков, специализирующихся в области теории информации. В нем изложен курс лекций "Теория информации и кодирование", читаемый автором на техническом факультете НГУ.

Пособие содержит описание основных методов сжатия дискретных данных и оценки их эффективности. Теоретический материал дополнен примерами и задачами.

Рецензенты: д-р техн. наук, проф. Б. Я. Рябко,
канд. техн. наук А. Н. Фионов.

Предисловие

Событием, положившим начало современной теории информации, считается появление статьи Клода Шеннона "Математическая теория связи" в 1948 г. В настоящее время теория информации является динамично растущей областью математики, что обусловлено быстрым развитием технических средств передачи и обработки информации. К теории информации относят следующие связанные общностью происхождения и методов области науки: обработку непрерывных сигналов, сжатие дискретных данных, поиск и сортировку информации, теорию исправляющих ошибки кодов, криптографию, а также другие вопросы, расположенные на стыке теории информации и классических математических дисциплин. В настоящем издании рассмотрены некоторые проблемы кодирования (сжатия) дискретных вероятностных источников (главы 2–4) и простейшие понятия, связанные с передачей информации по каналу связи и кодами, исправляющими ошибки (глава 5).

Описанные в пособии методы кодирования широко используются на практике. В частности, большинство архиваторов основано на схеме кодирования Лемпела–Зива, арифметическом или интервальном кодировании, которые изложены в главах 3 и 4. С практической точки зрения наиболее важными характеристиками метода сжатия данных являются стоимость кодирования (объем сжатых данных), сложность кодирования (время обработки данных) и объем памяти, который используется в процессе работы алгоритма. В настоящем пособии доказаны теоретические оценки только для первой характеристики — стоимости кодирования. Оценки сложности вычислений для большинства рассмотренных алгоритмов читатели могут получить самостоятельно.

Основным источником для первых четырех глав пособия послужили книги [6] и [7]. Некоторые вопросы, рассмотренные в главах 1 и 3, подробно изложены с точки зрения теории вероятности в учебнике [2]. Исследование некоторых вопросов из глав 1, 2 и 5 с точки зрения алгебры и дискретной математики можно найти в [4], [5] и [8]. Основным источником для главы 5 послужила книга [10], эти же вопросы с гораздо более общих позиций подробно рассмотрены в учебниках [3] и [9]. Книги [1] и [10] в популярной форме знакомят читателя с основными идеями и понятиями теории информации и кодирования.

Автор благодарит Б. Я. Рябко, А. Н. Фионова, М. П. Шарову, А. Ю. Васильеву за замечания и уточнения, которые способствовали повышению качества изложения материала.

Литература

1.Аршинов М. Н., Садовский Л. Е. Коды и математика. М.: Наука, 1983.
2.Боровков А. А. Теория вероятностей. М.: Наука, 1976.
3.Галлагер Р. Теория информации и надежная связь. М.: Советское радио, 1974.
4.Гоппа В. Д. Введение в алгебраическую теорию информации. М: Наука, 1995.
5.Колмогоров А. Н. Теория информации и теория алгоритмов. М.: Наука, 1987.
6.Кричевский Р. Е. Сжатие и поиск информации. М.: Радио и связь, 1989.
7.Кричевский Р. Е. Лекции по теории информации. Новосибирск: Изд-во НГУ, 1970.
8.Марков А. А. Введение в теорию кодирования. М.: Наука, 1982.
9.Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976.
10.Яглом А. М., Яглом И. М. Вероятность и информация. М.: Наука, 1973.

1. Основы теории информации

1.1. Необходимые сведения из теории вероятности

Вспомним основные определения теории вероятности, которые будут использоваться в дальнейшем. Пусть Ω — *пространство элементарных событий*, F — некоторая совокупность подмножеств Ω.

Множество F называется σ-*алгеброй* , если
1) $\Omega \in F$;
2) из $A_1, A_2, \dots A_i \dots \in F$ следует, что $\cap_{i=1}^{\infty} A_i \in F$;
3) из $A_1, A_2 \in F$ следует, что $A_1 \backslash A_2 \in F$.
Элементы F называют *событиями.* Если A_1 и A_2 — события, то $A_1 \cup A_2 = \Omega \backslash ((\Omega \backslash A_1) \cap (\Omega \backslash A_1))$ — тоже событие. Приняты обозначения $A_1 + A_2 = A_1 \cup A_2$ и $A_1 A_2 = A_1 \cap A_2$.

Вероятностью называется неотрицательная функция $p : F \to [0, 1]$, обладающая свойствами:
1) $p(\Omega) = 1$;
2) если $A_i \cap A_j = \emptyset$ для всех $i \neq j$, то $p(\sum_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} p(A_i)$.
Тройка (Ω, F, p) называется *вероятностным пространством*.

События $A_1, A_2 \in F$ называют *независимыми*, если $p(A_1A_2) = p(A_1)p(A_2)$. События $A_1, A_2, \ldots, A_n \in F$ называют *независимыми в совокупности*, если $p(A_1A_2 \ldots A_n) = p(A_1)p(A_2) \ldots p(A_n)$.

Разбиением события C будем называть множество
$A = \{A_1, A_2, \ldots, A_n\} \subset F$, если выполнены условия
1) $A_i \cap A_j = \emptyset$;
2) $\cup_i A_i = C$.
Если $C = \Omega$, то множество A будем называть просто разбиением.

Разбиения $A = \{A_1, A_2, \ldots, A_n\}$ и $B = \{B_1, B_2, \ldots, B_m\}$ называются *независимыми*, если для всех i, j события A_i и B_j независимы.

Вероятностью события A_1 *при условии* A_2 называется величина $p(A_1|A_2) = p(A_1A_2)/p(A_2)$.

Пусть $A = \{A_1, A_2, \ldots, A_n\}$ — разбиение события C, тогда справедлива *формула полной вероятности*

$$p(BC) = \sum_i p(B|A_i)p(A_i),$$

где $B \in F$ — произвольное событие.

Случайной величиной ξ называется функция $\xi : \Omega \to R$, если $\{a \in \Omega : \xi(a) < t\} \in F$ для всех $t \in R$.
В перечисленных ниже утверждениях подразумевается, что случайная величина имеет конечное число значений. В дальнейшем мы будем рассматривать только такие случайные величины.

Математическим ожиданием случайной величины ξ называется

$$M\xi = \sum_i x_i p(\{a \in \Omega : \xi(a) = x_i\}) = \sum_i \xi(A_i)p(A_i),$$

где $A_i = \{a \in \Omega : \xi(a) = x_i\}$.

Дисперсией случайной величины называется $D\xi = M\xi^2 - (M\xi)^2$.

Случайные величины $\xi_1, \xi_2, \ldots, \xi_n$ называются *независимыми*, если события $A_1 = \{a \in \Omega : \xi_1(a) = x_1\}, A_2 = \{a \in \Omega : \xi_2(a) = x_2\}, \ldots, A_n = \{a \in \Omega : \xi_n(a) = x_n\}$ независимы в совокупности для всевозможных $x_1, x_2, \ldots, x_n \in R$.

Справедливы равенства:
$M(\xi_1 + \xi_2) = M\xi_1 + M\xi_2$ — для произвольных случайных величин,
$D(\xi_1 + \xi_2) = D\xi_1 + D\xi_2$ — для независимых случайных величин.
Кроме того, если $\xi_1 \geq \xi_2$ для всех $a \in \Omega$, то $M\xi_1 \geq M\xi_2$.

Случайные величины ξ_1 и ξ_2 называются *одинаково распределенными* если $p(\{a \in \Omega : \xi_1(a) = x\}) = p(\{a \in \Omega : \xi_2(a) = x\})$ для всех

$x \in R$. Математические ожидания и дисперсии одинаково распределенных случайных величин совпадают.

1.2. Энтропия как мера неопределенности опыта

Задача этого пункта состоит в определении количества информации, которую приносит результат опыта, имеющего несколько случайных исходов. Другими словами, пусть $A = \{A_1, A_2, \ldots, A_k\}$ — некоторое разбиение (опыт, имеющий исходы $A_1, A_2, \ldots, A_k$). Нам нужно определить функцию $H(A)$, которая может служить мерой неопределенности разбиения A или количества информации, появляющейся в результате выполнения опыта A. Нас не интересует содержание опыта A, и мы считаем, что энтропия (неопределенность) опыта A зависит только от вероятностей его исходов, т. е. $H(A) = h(p(A_1), p(A_2), \ldots, p(A_k))$.

Сначала предположим, что все исходы опыта A равновероятны, тогда $H(A) = h(1/k, 1/k, \ldots, 1/k) = \phi(k)$. Естественно предположить, что опыт, имеющий большее число равновероятных исходов, имеет большую неопределенность, поэтому $\phi(k)$ — возрастающая функция. Пусть опыт B состоит в последовательном выполнении двух независимых экспериментов A' и A'', имеющих по k равновероятных исходов. Тогда опыт B имеет k^2 равновероятных исходов, и его неопределенность равна сумме неопределенностей опытов A' и A''. То есть $H(B) = H(A') + H(A'')$ и $\phi(k^2) = 2\phi(k)$. Аналогичным образом заключаем, что $\phi(k^i) = i\phi(k)$ для всех целых $i > 0$. Покажем, что единственной с точностью до умножения на постоянную функцией, удовлетворяющей этим условиям, является $\log k$[1], т. е. $\phi(k) = C \log k$, где $C > 0$ — константа.

Пусть $n > 1$ и $a > 2$ — целые числа. Найдется такое натуральное m, что справедливы неравенства

$$2^m \le a^n < 2^{m+1}.$$

Тогда из монотонности функции ϕ имеем $m\phi(2) \le n\phi(a) < (m+1)\phi(2)$. Кроме того, $m \le n \log a < m + 1$. Из двух последних неравенств получаем

$$\left| \frac{\phi(a)}{\log a} - \phi(2) \right| \le \frac{1}{m}.$$

Увеличивая n, число m можно сделать сколь угодно большим. Тогда из предыдущего неравенства следует, что $\phi(a) = \phi(2) \log a$. Поскольку

[1]Здесь и далее логарифм берется по основанию 2.

для записи результата опыта с двумя равновероятными исходами достаточно одного бита, неопределенность такого опыта удобно считать равной 1, т. е. $\phi(2) = 1$.

Пусть опыт A имеет k исходов, вероятности которых равны некоторым рациональным числам p_i. Тогда $H(A) = h\left(\frac{m_1}{n}, \frac{m_2}{n}, \ldots, \frac{m_k}{n}\right)$, где m_i — целые, а n — общий знаменатель чисел p_i. Рассмотрим опыт B, имеющий n равновероятных исходов. Сгруппируем исходы опыта B в k групп так, чтобы i-я группа имела вероятность $\frac{m_i}{n}$. Тогда исход опыта B можно указать, определив сначала группу, в которую попал данный исход, а затем найдя место нужного исхода в этой группе. Поскольку исходы опытов, содержащихся в каждой группе, равновероятны и опыт по определению места нужного исхода в i-й группе нужно проводить с вероятностью $\frac{m_i}{n}$, справедлива формула

$$h\left(\frac{1}{n}, \ldots, \frac{1}{n}\right) = h\left(\frac{m_1}{n}, \frac{m_2}{n}, \ldots, \frac{m_k}{n}\right) + \frac{m_1}{n} h\left(\frac{1}{m_1}, \ldots, \frac{1}{m_1}\right) + \ldots$$

$$\ldots + \frac{m_k}{n} h\left(\frac{1}{m_k}, \ldots, \frac{1}{m_k}\right).$$

Тогда

$$h\left(\frac{m_1}{n}, \frac{m_2}{n}, \ldots, \frac{m_k}{n}\right) = \log n - \frac{m_1}{n}\log m_1 - \ldots - \frac{m_k}{n}\log m_k = \sum_{i=1}^{k} \frac{m_i}{n} \log \frac{n}{m_i}.$$

Естественно предположить, что функция h непрерывна по всем своим аргументам. Из этого следует, что предыдущая формула верна и для опытов с иррациональными вероятностями исходов.

На основании проведенных выше рассуждений определим *энтропию* (неопределенность) разбиения $A = \{A_1, A_2, \ldots, A_k\}$ следующим образом[2]:

$$H(A) = -\sum_{i=1}^{k} p(A_i) \log p(A_i).$$

Заметим, что $H(A) \geq 0$ и $H(A) = 0$ только тогда, когда найдется элемент разбиения A_i с вероятностью $p(A_i) = 1$.

[2]Это определение энтропии было введено К. Шенноном.

1.3. Свойства энтропии и информации

Утверждение 1.1. *Пусть* $A = \{A_1, A_2, \ldots, A_k\}$ — *разбиение, тогда* $H(A) \leq \log k$.

Доказательство. Рассмотрим функцию $f(x) = -x \log x$. $f(x)$ выпукла вверх при $x > 0$, так как $f''(x) = -\frac{1}{x \ln 2} < 0$. Тогда для функции f выполняется неравенство Йенсена: если $\sum_{i=1}^{k} \alpha_i = 1$ и $\alpha_i > 0$, то

$$\sum_{i=1}^{k} \alpha_i f(x_i) \leq f\left(\sum_{i=1}^{k} \alpha_i x_i\right).$$

Пусть $\alpha_i = 1/k$, а $x_i = p(A_i)$. Тогда

$$H(A) = k \sum_{i=1}^{k} \frac{1}{k} f(p(A_i)) \leq k f\left(\sum_{i=1}^{k} \frac{p(A_i)}{k}\right) =$$

$$= -k\left(\sum_{i=1}^{k} \frac{p(A_i)}{k}\right) \log\left(\sum_{i=1}^{k} \frac{p(A_i)}{k}\right) = \log k.$$

Утверждение доказано.

Пусть $B_j \in F$; определим *условную энтропию* $H(A|B_j)$ равенством

$$H(A|B_j) = -\sum_i p(A_i|B_j) \log p(A_i|B_j).$$

Пусть $A = \{A_1, A_2, \ldots, A_k\}$ и $B = \{B_1, B_2, \ldots, B_n\}$ — разбиения.

Энтропией A *при заданном* B называется

$$H(A|B) = \sum_j p(B_j) H(A|B_j).$$

Заметим, что $H(A|A) = 0$.

Через AB будем обозначать разбиение, состоящее из всевозможных событий вида $A_i B_j$, где $i = 1 \ldots k$, $j = 1 \ldots n$.

Утверждение 1.2. *Пусть* $A = \{A_1, \ldots, A_k\}$ *и* $B = \{B_1, \ldots, B_n\}$ — *разбиения, тогда* $H(AB) = H(B) + H(A|B)$.

Доказательство.

$$H(AB) = -\sum_{i,j} p(A_iB_j)\log p(A_iB_j) =$$

$$= -\sum_{i,j} p(A_iB_j)(\log p(A_i|B_j) + \log p(B_j)) =$$

$$= -\sum_j \left(p(B_j) \sum_i p(A_i|B_j)\log p(A_i|B_j) \right) - \sum_j \left(\log p(B_j) \sum_i p(A_iB_j) \right) =$$

$$= H(A|B) + H(B).$$

Последнее равенство следует из формулы полной вероятности

$$\sum_i p(A_iB_j) = \sum_i p(A_i)p(B_j\,|A_i) = p(B_j)$$

и определения $H(A|B)$. Утверждение доказано.

На множестве разбиений пространства Ω определим частичный порядок: $B \succeq A$ (B информативнее A), если $BA = B$.

Утверждение 1.3. *Пусть A, B, C — разбиения, $B \succeq C$. Тогда $H(A|B) \le H(A|C)$.*

Доказательство. По условию $BC = B$, тогда $C_mB_j = B_j$ или $C_mB_j = \emptyset$. Поскольку множества C_m не пересекаются, то для каждого B_j найдется единственное C_m такое, что $B_j = C_mB_j$. Множество чисел j, для которых $B_j = C_mB_j$, обозначим через $j(m)$. Введем обозначение $B_j^m = B_j$, если $j \in j(m)$. Тогда $C_m = \sum_{j(m)} B_j^m$ и справедливы соотношения

$$-\sum_{j(m)} p(A_iB_j^m)\log p(A_i|B_j^m) = -p(C_m)\sum_{j(m)} \frac{p(B_j^m)}{p(C_m)} p(A_i|B_j^m)\log p(A_i|B_j^m) \le$$

$$\le -\left(\sum_{j(m)} p(B_j^m)p(A_i|B_j^m) \right) \log \left(\frac{\sum_{j(m)} p(B_j^m)p(A_i|B_j^m)}{p(C_m)} \right) \le$$

$$\le -p(C_mA_i)\log p(A_i|C_m),$$

где первое неравенство следует из неравенства Йенсена для функции $-x\log x$ (см. утверждение 1.1), второе — по формуле полной вероятности.

Тогда

$$H(A|B) = -\sum_i \sum_m \sum_{j(m)} p(A_i B_j^m) \log p(A_i|B_j^m) \leq$$

$$\leq -\sum_i \sum_m p(C_m A_i) \log p(A_i|C_m) = H(A|C).$$

Утверждение доказано.

Следствие 1.1. *$H(A|B) \leq H(A)$.*

Доказательство. Рассмотрим $C = \{\Omega\}$. Нетрудно убедиться, что $H(A|C) = H(A)$ и для всех разбиений B выполнено $B \succeq C$. Тогда неравенство $H(A|B) \leq H(A)$ следует из предыдущего утверждения.

Информацией разбиения A относительно разбиения B называется

$$I(A,B) = H(A) - H(A|B).$$

Заметим, что $I(A,A) = H(A)$.

Следствие 1.2. *Если $B \succeq C$, то $I(A,B) \geq I(A,C)$.*

Доказательство следствия получается из определения величины $I(A,B)$ и утверждения 1.3.

1.4. Эмпирическая энтропия и число сочетаний

Пусть $A = \{a_1, a_2, \ldots, a_k\}$ — конечный алфавит. Пусть источник S порождает случайные последовательности букв алфавита A. Если вероятность появления буквы не зависит от предыдущих букв, то S называют *источником Бернулли*. Такой источник можно считать разбиением $S = \{A_1, A_2, \ldots, A_k\}$, где событие A_i состоит в появлении буквы a_i. Обычно букву a_i и событие A_i отождествляют и пишут $p(a_i)$, подразумевая $p(A_i)$. Определение источника общего вида будет дано в главе 3.

Пусть $x \in A^n$ — некоторое слово из n букв алфавита A, $r_i(x)$ — количество вхождений буквы a_i в слово x. Тогда $r_i(x)/n$ — частота

вхождений буквы a_i в слово x. Определим *эмпирическую энтропию* $F(x)$ слова x равенством

$$F(x) = -\sum_{i=1}^{k} \frac{r_i(x)}{n} \log \frac{r_i(x)}{n}.$$

Утверждение 1.4. *Пусть $A = \{a_1, a_2, \ldots, a_k\}$ — конечный алфавит, S — источник Бернулли, порождающий буквы алфавита A с вероятностями $p(a_i)$. Тогда*

$$|\sum_{x \in A^n} p(x)F(x) - H(S)| \leq \frac{k-1}{n \ln 2}.$$

Доказательство. Рассмотрим набор случайных величин $\xi_j^i(x)$:

$\xi_j^i(x) = \begin{cases} 1, \text{ если на } j\text{-м месте в слове } x \text{ находится } a_i; \\ 0 \text{ — в остальных случаях.} \end{cases}$

Тогда

$$M\xi_j^i(x) = \sum_{x \in A^n} p(x)\xi_j^i(x) = \sum_{x \in A^n} p(x_1)\ldots p(x_n)\xi_j^i(x) =$$

$$\sum_{x_j = a_i} p(x_1)\ldots p(x_{j-1})p(a_i)\ldots p(x_n) = p(a_i).$$

Поскольку $(\xi_j^i(x))^2 = \xi_j^i(x)$, справедливо равенство

$$D\xi_j^i(x) = M(\xi_j^i(x))^2 - (M\xi_j^i(x))^2 = p(a_i) - p(a_i)^2.$$

Кроме того, $r_i(x) = \sum_{j=1}^{n} \xi_j^i(x)$. По определению источника Бернулли S случайные величины $\xi_1^i, \xi_2^i, \ldots, \xi_n^i$ — независимы. Тогда

$$\sum_{x \in A^n} p(x)r_i(x) = Mr_i(x) = \sum_{j=1}^{n} M\xi_j^i(x) = np(a_i), \tag{1.1}$$

$$\sum_{x \in A^n} p(x)(r_i(x))^2 - \left(\sum_{x \in A^n} p(x)r_i(x)\right)^2 = Dr_i(x) =$$

$$= \sum_{j=1}^{n} D\xi_j^i(x) = n(p(a_i) - (p(a_i))^2).$$

Откуда

$$\sum_{x\in A^n} p(x)(r_i(x))^2 = Dr_i(x) + (Mr_i(x))^2 = n^2p(a_i)^2 + n(p(a_i) - (p(a_i))^2). \tag{1.2}$$

Из (1.1) следует, что

$$H(S) = -\sum_{i=1}^{k}\sum_{x\in A^n} p(x)\frac{r_i(x)}{n}\log p(a_i) = -\sum_{x\in A^n} p(x)\sum_{i=1}^{k}\frac{r_i(x)}{n}\log p(a_i).$$

Тогда

$$-\sum_{x\in A^n} p(x)F(x) + H(S) = \sum_{x\in A^n} p(x)\sum_{i=1}^{k}\frac{r_i(x)}{n}\log\frac{r_i(x)}{np(a_i)}. \tag{1.3}$$

Докажем неравенство

$$-\sum_{x\in A^n} p(x)F(x) + H(S) \le \frac{k-1}{n\ln 2}.$$

Из (1.1) следует, что

$$\sum_{x\in A^n} p(x)\sum_{i=1}^{k}\frac{r_i(x)}{n} = 1.$$

Функция $\log x$ выпукла вверх, тогда из неравенства Йенсена, (1.2) и (1.3) имеем

$$-\sum_{x\in A^n} p(x)F(x) + H(S) \le \log\left(\sum_{x\in A^n} p(x)\sum_{i=1}^{k}\frac{r_i^2(x)}{n^2p(a_i)}\right) =$$

$$= \log\left(\sum_{i=1}^{k}(p(a_i) + \frac{1-p(a_i)}{n})\right) = \log(1+\frac{k-1}{n}).$$

Применив неравенство $\ln(1+x) \le x$, получаем искомое неравенство.

Докажем, что $\sum_{x\in A^n} p(x)F(x) - H(S) \le 0$. Поскольку функция $-x\log x$ выпукла вверх и $\sum_{x\in A^n} p(x) = 1$, из (1.3), неравенства Йенсена и (1.1) получаем

$$\sum_{x\in A^n} p(x)F(x) - H(S) = -\sum_{x\in A^n} p(x)\sum_{i=1}^{k}\frac{r_i(x)}{n}\log\frac{r_i(x)}{np(a_i)} =$$

$$= \sum_{i=1}^{k} p(a_i) \left(\sum_{x \in A^n} p(x) \left(-\frac{r_i(x)}{np(a_i)} \log \frac{r_i(x)}{np(a_i)} \right) \right) \leq$$

$$\sum_{i=1}^{k} p(a_i) \left(-\left(\sum_{x \in A^n} \frac{p(x) r_i(x)}{np(a_i)} \right) \log \left(\sum_{x \in A^n} \frac{p(x) r_i(x)}{np(a_i)} \right) \right) =$$

$$= \sum_{i=1}^{k} p(a_i)(1 \log 1) = 0.$$

Утверждение доказано.

Предположим, что нам точно известны частоты вхождений букв $a_i \in A$ в слово $x \in A^n$. Оценим количество информации, которая необходима для восстановления слова, если известен его частотный состав. Пусть $T(x) = \{y \in A^n : r_i(y) = r_i(x) \quad \text{для всех} \quad i = 1 \ldots k\}$ — множество слов, имеющих тот же частотный состав, что и слово x. Тогда чтобы однозначно указать слово $y \in T$, нужно не менее $\log |T|$ двоичных знаков [3]. Иначе двум разным словам будет соответствовать один и тот же код (указатель). Оценим величину $\log |T|$.

Утверждение 1.5. *Пусть $x \in A^n$, тогда $F(x) - \frac{1}{n} \log |T(x)| = \alpha(n)$, где $\alpha(n) \geq 0, \alpha(n) = O\left(\frac{\log n}{n}\right)$.*

Доказательство. Как известно, число сочетаний из n элементов по $r_1(x), r_2(x), \ldots r_k(x)$ элементов k различных видов равняется

$$|T(x)| = \frac{n!}{r_1(x)! r_2(x)! \ldots r_k(x)!}.$$ [4]

Из формулы Стирлинга $n! = n^n e^{-n} \sqrt{2\pi n}(1 + o(1))$ получаем равенства

$$|T(x)| = \frac{n^n e^{r_1(x) + \ldots + r_k(x)}}{e^n r_1(x)^{r_1(x)} \ldots r_k(x)^{r_k(x)}} \frac{\sqrt{2\pi n}(1 + o(1))}{\sqrt{(2\pi)^k r_1(x) \ldots r_k(x)}(1 + o(1))^k} =$$

$$= \left(\frac{n}{r_1(x)} \right)^{r_1} \left(\frac{n}{r_2(x)} \right)^{r_2} \ldots \left(\frac{n}{r_k(x)} \right)^{r_k} \sqrt{\frac{2\pi n}{(2\pi)^k r_1(x) r_2(x) \ldots r_k(x)}} (1 + o(1)).$$

[3] Здесь и в дальнейшем через $|A|$ обозначается число элементов множества A.

[4] Здесь $0! = 1 = 1!$, поэтому в двух формулах ниже, если число $r_i = 0$ в знаменателе, то его следует заменить на 1.

Тогда

$$\frac{1}{n}\log|T| = \sum_{i=1}^{k}\frac{r_i(x)}{n}\log\frac{n}{r_i(x)} + \frac{1}{2n}\sum_{i=1}^{k}\log\frac{n}{r_i(x)} - \frac{k-1}{2n}\log n + O\left(\frac{1}{n}\right).$$

Утверждение доказано.

Следовательно, чтобы указать слово $x^n \in A^n$ известного частотного состава, необходимо затратить не менее $nF(x)$ битов. Таким образом, из утверждений 1.4, 1.5 мы получили второй способ определения энтропии $H(S) = \lim_{n\to\infty} MF(x^n) = \lim_{n\to\infty}\frac{1}{n}M\log|T(x^n)|$, где $x^n \in A^n$, т. е. $H(S)$ — минимальное среднее количество двоичных знаков, которые необходимо затратить для записи буквы в слове, порожденном источником S.

2. Побуквенное кодирование

2.1. Префиксные коды и неравенство Крафта

Пусть $E = \{0,1\}$, обозначим через $E^* = \cup_{i=1}^{\infty} E^i$ множество всех конечных наборов 0 и 1. Пусть $A = \{a_1, a_2, \ldots, a_i, \ldots\}$ — некоторый алфавит.

Кодированием (кодом)[5] называется инъективное отображение $f : A \to E^*$. $f(a_i)$ называется кодом буквы a_i, или *кодовым словом*. Через $|f(a_i)|$ будем обозначать длину кодового слова.

Кодирование f называется *дешифруемым*, если произвольная последовательность кодовых слов $f(a_{i_1})f(a_{i_2})\ldots f(a_{i_n})$, записанных слитно, однозначно разделяется на кодовые слова.

Кодирование f называется *префиксным*, если никакое кодовое слово $f(a_i)$ не является *префиксом* (началом) другого кодового слова $f(a_j)$.

Например, код, содержащий два кодовых слова 0 и 10, является префиксным; код, содержащий два кодовых слова 0 и 01, является дешифруемым, но не префиксным; а код, имеющий кодовые слова 0, 1 и 10, не является ни префиксным, ни дешифруемым.

Нетрудно заметить, что каждый префиксный код является дешифруемым, обратное неверно.

Рассмотрим произвольное бинарное дерево. Каждое левое ребро дерева пометим нулем, правое — единицей. Тогда каждой вершине

[5] Кодом обычно называют и множество кодовых слов.

дерева будет сопоставлен набор из 0 и 1 (код вершины), описывающий путь от корня до вершины, как показано на рис. 1.

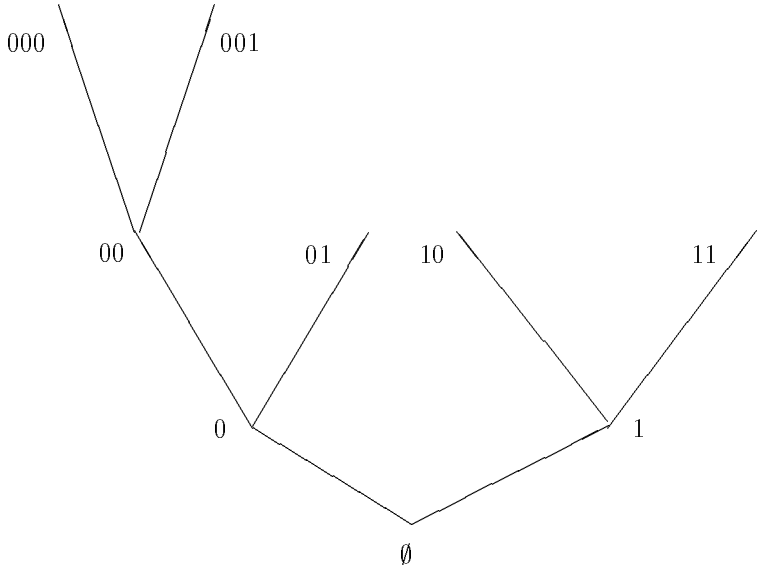


Рис. 1

Тупиковые вершины дерева называются листьями. Пусть x — некоторая вершина дерева, тогда на пути от корня дерева до x расположены те и только те вершины, коды которых являются префиксами кода x. Следовательно, код одной листовой вершины не может быть префиксом кода другой листовой вершины. Таким образом, множество кодов листьев дерева представляет собой некоторый префиксный код. Нетрудно заметить, что верно и обратное: произвольный префиксный код представляет собой подмножество кодов листьев некоторого дерева.

Теорема 2.1. *Пусть $l_1, l_2, \ldots, l_n$ — целые положительные числа, тогда для существования префиксного кода с длинами слов $l_1, l_2, \ldots, l_n$ необходимо и достаточно выполнения неравенства Крафта* $\sum_{j=1}^{n} 2^{-l_j} \leq 1$.

Доказательство. Перепишем неравенство Крафта в виде

$$\sum_{i=1}^{l} s_i 2^{-i} \leq 1, \tag{2.1}$$

где s_i — количество чисел $l_1, \ldots, l_n$, равных i, $l = \max l_j$.

Построим искомый набор кодовых слов по индукции. Из (2.1) следует, что $s_1 \leq 2$. Очевидно, что построить префиксный код, включающий не более двух кодовых слов длины 1, возможно. Пусть префиксный код, включающий s_i кодовых слов длины i, $i < k$, построен. Из неравенства (2.1) следует, что $\sum_{i=1}^{k} s_i 2^{-i} \leq 1$ и

$$s_k \leq 2^k - \sum_{i=1}^{k-1} s_i 2^{k-i}. \tag{2.2}$$

Рассмотрим E^k — множество всех двоичных слов длины k, $|E^k| = 2^k$. Каждое выбранное кодовое слово длины $i < k$ является префиксом 2^{k-i} кодовых слов длины k. Тогда все кодовые слова являются префиксами не более, чем $\sum_{i=1}^{k-1} s_i 2^{k-i}$ слов длины k. Следовательно, мы можем добавить во множество кодовых слов $2^k - \sum_{i=1}^{k-1} s_i 2^{k-i}$ слов длины k, сохранив свойство префиксности. Тогда существование кода, удовлетворяющего условию теоремы, следует из неравенства (2.2).

Пусть имеется некоторый префиксный код; покажем, что длины его кодовых слов удовлетворяют неравенству Крафта. Пусть l — максимальная длина кодового слова и s_i — количество кодовых слов длины i. Пусть $T_j \subset E^l$ — множество двоичных слов длины l, префиксом которых является j-ое кодовое слово длины l_j.

Докажем, что $T_{j_1} \cap T_{j_2} = \emptyset$. Рассмотрим x — произвольное двоичное слово длины l. Пусть два кодовых слова являются префиксами x. Очевидно, что более короткое из них является префиксом более длинного. А это противоречит тому, что код префиксный. Итак, $T_{j_1} \cap T_{j_2} = \emptyset$ и $T_j \subset E^l$. Тогда

$$\sum_{j=1}^{n} |T_j| \leq |E^l| = 2^l.$$

Поскольку $|T_j| = 2^{l-l_j}$, из последнего неравенства получаем неравенство

$$\sum_{j=1}^{n} 2^{l-l_j} \leq 2^l.$$

Теорема доказана.

Б. Макмиллан доказал в 1956 г., что неравенство Крафта справедливо для произвольного дешифруемого кода. Таким образом, для каждого дешифруемого кода найдется префиксный код с теми же длинами кодовых слов. Это позволяет нам ограничиться в дальнейшем исследованием только префиксных кодов.

2.2. Префиксные коды натурального ряда

Ясно, что обычная двоичная запись натуральных чисел (Bin) не является префиксным кодом. Например, $Bin(2) = 10$ является префиксом $Bin(5) = 101$. Простейшим способом преобразования обычной двоичной записи в префиксный код является добавление перед двоичной записью блока из нулей, равного по длине двоичной записи числа. Однако такой код увеличивает длину записи в два раза. В этом пункте мы рассмотрим примеры более сложных, но и более коротких префиксных кодов натуральных чисел. Введем несколько обозначений.
$D(n)$ — последовательность из n единиц с нулем в конце ($D(2) = 110$).
$B(n)$ — двоичная запись числа n без первого символа ($B(5) = 01$, $B(12) = 100$, $B(1) = B(0) = \emptyset$).
$B_d(n)$ — двоичная запись числа n, использующая ровно d символов. Например, $B_4(5) = 0101$.
Пусть $\lambda_d(n) = \lfloor \log n \rfloor - d$, в частности, $\lambda_0(n) = \lfloor \log n \rfloor = |B(n)|$ при $n \geq 1$.
Введем обозначение $\lambda_d^k(n) = \lambda_d(\lambda_d^{k-1}(n))$, где $\lambda_d^1(n) = \lambda_d(n)$.

Следующий код El натуральных чисел принадлежит П. Элайесу.
$El(0) = 10$, $El(1) = 11$ и при $n \geq 2$,
$El(n) = 0 \ldots 0 Bin(\lambda_0(n) + 1) B(n)$,
где количество нулей перед $Bin(\lambda_0(n) + 1)$ равняется $\lambda_0(\lambda_0(n) + 1)$. Например, $El(2) = 0\,10\,0$, $El(3) = 0\,10\,1$, $El(5) = 0\,11\,01$, $El(62) = 00\,110\,11110$. При кодировании чисел кодом Элайеса удобно пользоваться соотношениями: $\lambda_0(\lambda_0(n) + 1) = |Bin(\lambda_0(n) + 1)| - 1$ и $\lambda_0(n) = |B(n)|$.

Утверждение 2.1. *При* $n > 1$

$$|El(n)| = \lfloor \log n \rfloor + 2\lfloor \log(\lfloor \log n \rfloor + 1) \rfloor + 1 \leq \log n + 2 \log\log n + 3.$$

Доказательство. Из определений имеем $|B(n)| = \lfloor \log n \rfloor$,
$|Bin(\lambda_0(n)+1)| = \lfloor \log(\lfloor \log n \rfloor + 1) \rfloor + 1 \leq \log\log n + 2$ и $\lambda_0((\lambda_0(n)+1)) = \lfloor \log(\lfloor \log n \rfloor + 1) \rfloor \leq \log\log n + 1$. Утверждение доказано.

Если $n = 1 \ldots 255$, удобно использовать код $El'(n) = B_3(\lambda_0(n))B(n)$. Например, $El'(1) = 000$, $El'(5) = 010\,01$, $El'(62) = 101\,11110$.

Следующий код Lev был предложен В. И. Левенштейном. Если $\lambda_0^k(n) = 0$, то

$$Lev(n) = D(k)B(\lambda_0^{k-2}(n)) \ldots B(\lambda_0(n))B(n).$$

Например, $Lev(0) = 0$, $Lev(1) = 10$, $Lev(5) = 1110\,0\,01$, $Lev(62) = 11110\,0\,01\,11110$.

Преобразование двоичной записи числа в код Левенштейна осуществляется черезвычайно просто, поскольку $\lambda_0(n) = |B(n)|$, $\lambda_0^2(n) = |B(\lambda_0(n))|$ и так далее. Декодирование кода Левенштейна производится в обратном порядке, причем число итераций k равняется числу единиц до первого нуля.

Следующее семейство кодов предложил К. Стоут. Пусть $d \geq 2$ и $\lambda_d^k(n) < 2^d$. Тогда $St_d(n) = 0B_d(n)$, если $n < 2^d$, и
$St_d(n) = D(k)B_d(\lambda_d^k(n))B(\lambda_d^{k-1}(n)) \ldots B(\lambda_d(n))B(n)$, если $n \geq 2^d$.
Например, $St_2(0) = 0\,00$, $St_2(1) = 0\,01$, $St_2(5) = 10\,00\,01$, $St_2(62) = 10\,11\,11110$ и $St_3(0) = 0\,000$, $St_3(1) = 0\,001$, $St_3(5) = 0\,101$, $St_3(62) = 10\,010\,11110$.

Кодирование и декодирование чисел кодом Стоута подобно аналогичным процедурам кода Левенштейна. Нужно использовать, что $\lambda_d(n) = |B(n)| - d$, $\lambda_d^2(n) = |B(\lambda_d(n))| - d$ и т. д. Нетрудно показать, что при $n \to \infty$ $|Lev(n)| = \log n + (1 + o(1)) \log\log n$ и $|St_d(n)| = \log n + (1 + o(1)) \log\log n$.

Может оказаться полезным упрощенный код Стоута:
$St'_d(n) = 0B_d(n)$, если $n < 2^d$, и
$St'_d(n) = D(\lambda_d(n))B(n)$, если $n \geq 2^d$.
Например, $St'_2(5) = 10\,01$, $St'_2(10) = 110\,010$, $St'_2(62) = 1110\,11110$ и $St'_3(5) = 0\,101$, $St'_3(10) = 10\,010$, $St'_3(62) = 110\,11110$.

Утверждение 2.2. *Коды натурального ряда $El, El', Lev, St_d, St'_d$ — префиксные.*

Доказательство. Докажем утверждение для кода Левенштейна. Пусть $n_1, n_2 \in N$; покажем, что кодовое слово $Lev(n_1)$ не является префиксом кодового слова $Lev(n_2)$. Пусть $\lambda_0^{k_1}(n_1) = \lambda_0^{k_2}(n_2) = 0$. Если $k_1 \neq k_2$, тогда $Lev(n_1)$ не является префиксом $Lev(n_2)$, поскольку $D(k_1)$ — не префикс $D(k_2)$.

Пусть $k_1 = k_2$ и $\lambda_0^k(n_1) = \lambda_0^k(n_2), \ldots, \lambda_0^{i+1}(n_1) = \lambda_0^{i+1}(n_2)$, но $\lambda_0^i(n_1) \neq \lambda_0^i(n_2)$. Поскольку $\lambda_0^{i+1}(n) = |B(\lambda_0^i(n))|$, то $|B(\lambda_0^i(n_1))| = |B(\lambda_0^i(n_2))|$. Тогда $|D(k)B(\lambda_0^{k-2}(n_1))\ldots B(\lambda_0^i(n_1))| = |D(k)B(\lambda_0^{k-2}(n_2))\ldots B(\lambda_0^i(n_2))|$, но $D(k)B(\lambda_0^{k-2}(n_1))\ldots B(\lambda_0^i(n_1)) \neq D(k)B(\lambda_0^{k-2}(n_2))\ldots B(\lambda_0^i(n_2))$. То есть $D(k)B(\lambda_0^{k-2}(n_1))\ldots B(\lambda_0^i(n_1))$ не является префиксом кодового слова $Lev(n_2)$. И тем более $Lev(n_1)$ — не префикс $Lev(n_2)$. Таким образом, $Lev(n_1)$ может оказаться префиксом $Lev(n_2)$, только если $Lev(n_1) = Lev(n_2)$. Это невозможно, если n_1 и n_2 различны, так как $B(n_1) \neq B(n_2)$ при $n_1 > 1$ или $n_2 > 1$. Кроме того, $Lev(0) \neq Lev(1)$.

Мы доказали, что код Левенштейна префиксный. Для других перечисленных кодов утверждение можно доказать аналогично.

2.3. Нумерация двоичных слов заданного веса

Рассмотрим задачу нумерации элементов множества $S(k,n) \subset E^n$, состоящего из двоичных слов длины n, содержащих ровно по k единиц.

Упорядочим множество $S(k,n)$ лексикографически. Пусть $t(x_1x_2\ldots x_i)$ — количество слов из $S(k,n)$, имеющих префикс $x_1x_2\ldots x_i$. Ясно, что $t(x_1x_2\ldots x_{i-1}0) = C_{n-i}^m$, где $m = k - \sum_{j=1}^{i-1} x_j$ — количество несодержащихся в префиксе единиц. Обозначим через $L(x)$ номер слова $x \in S(k,n)$ в лексикографическом порядке. Заметим, что $L(x) = \sum_{i=1}^n x_i t(x_1x_2\ldots x_{i-1}0)$. Тогда $L(x) = \sum_{i=1}^n x_i C_{n-i}^{m_i}$, где $m_i = k - \sum_{j=1}^{i-1} x_j$. Поскольку $|S(k,n)| = C_n^k$, то для записи числа $L(x)$ достаточно $\lfloor \log C_n^k \rfloor + 1$ двоичных знаков.

2.4. Стоимость и избыточность кодирования. Теорема Шеннона

Пусть S — источник Бернулли с алфавитом $A = \{a_1, \ldots, a_k\}$ и вероятностями появления букв $p(a_1), \ldots, p(a_k)$. В этой главе речь пойдет только об источниках Бернулли, и говоря об источнике S, мы будем иметь в виду источник Бернулли S.

Энтропией источника Бернулли S называется величина $H(S) = -\sum_{i=1}^k p(a_i)\log p(a_i)$.

Стоимостью кодирования f источника S с алфавитом A называется величина $C(f,S) = \sum_{i=1}^k p(a_i)|f(a_i)|$.

Избыточностью кодирования f источника S с алфавитом A называется величина $R(f,S) = C(f,S) - H(S)$.

Докажем теорему кодирования Шеннона для источников Бернулли.

Теорема 2.2.

1) Для произвольного источника S и префиксного кода f избыточность кодирования неотрицательна, т. е. $R(f,S) \geq 0$.

2) Для каждого источника S найдется префиксный код f с избыточностью, не превышающей единицы, т. е. $R(f,S) \leq 1$.

Доказательство. Докажем первое утверждение теоремы:

$$-R(f,S) = H(S) - C(f,S) = \sum_{i=1}^{k} p(a_i) \log\left(\frac{1}{p(a_i)2^{|f(a_i)|}}\right).$$

Поскольку $\sum_{i=1}^{k} p(a_i) = 1$ и $\log$ — выпуклая вверх функция, то из неравенства Йенсена получаем

$$-R(f,S) \leq \log\left(\sum_{i=1}^{k} 2^{-|f(a_i)|}\right).$$

Из неравенства Крафта (теорема 2.1) следует, что $\sum_{i=1}^{k} 2^{-|f(a_i)|} \leq 1$. Тогда $-R(f,S) \leq 0$, и первое утверждение теоремы доказано.

Пусть $l_i = \lceil \log \frac{1}{p(a_i)} \rceil$. Тогда

$$\sum_{i=1}^{k} 2^{-l_i} \leq \sum_{i=1}^{k} 2^{-\log \frac{1}{p(a_i)}} = \sum_{i=1}^{k} p(a_i) = 1,$$

т. е. числа l_i удовлетворяют неравенству Крафта. Тогда из теоремы 2.1 следует, что найдется префиксное кодирование f такое, что $|f(a_i)| = l_i$. Оценим избыточность этого кодирования

$$R(f,S) = \sum_{i=1}^{k} p(a_i) \left(\lceil \log \frac{1}{p(a_i)} \rceil - \log \frac{1}{p(a_i)} \right) \leq \sum_{i=1}^{k} p(a_i) = 1. \qquad (2.3)$$

Теорема доказана.

2.5. Префиксные коды Шеннона, Гильберта–Мура, Шеннона–Фано

Построим код Шеннона, обладающий свойством $|f(a_i)| = \lceil \log \frac{1}{p(a_i)} \rceil$. Пронумеруем буквы алфавита так, чтобы $p(a_1) \geq p(a_2) \ldots \geq p(a_k) > 0$. Определим числа σ_i по индукции: $\sigma_1 = 0, \sigma_{i+1} = \sigma_i + p(a_i)$. Ясно, что

$0 \le \sigma_i < 1$, $1 \le i \le k$. В качестве кодового слова $f(a_i)$ возьмем первые после запятой $\lceil \log \frac{1}{p(a_i)} \rceil$ двоичных знаков числа σ_i.

Например, пусть $p(a_1) = 1/2, p(a_2) = 1/3, p(a_3) = 1/8, p(a_4) = 1/24$. Тогда в двоичной записи $\sigma_1 = 0,000.., \sigma_2 = 0,1000.., \sigma_3 = 0,1101.., \sigma_4 = 0,11110...$ По определению кода Шеннона получаем $f(a_1) = 0, f(a_2) = 10, f(a_3) = 110, f(a_4) = 11110$.

Утверждение 2.3. *Код Шеннона f — префиксный, $R(f,S) \le 1$.*

Доказательство. Первые после запятой n двоичных знаков чисел a и b $(1 > a > b > 0)$ совпадают, если и только если $a - b < 2^{-n}$.

Пусть $j > i$, тогда

$$\sigma_j - \sigma_i \ge p(a_i) = 2^{-\log(1/p(a_i))} \ge 2^{-\lceil \log(1/p(a_i)) \rceil} = 2^{-|f(a_i)|}.$$

Т. е. первые после запятой $|f(a_i)|$ двоичных знаков числа σ_j не совпадают с $f(a_i)$. Поскольку $p(a_j) \le p(a_i)$, то $|f(a_j)| \ge |f(a_i)|$, и префиксность кода Шеннона доказана. Неравенство $R(f,S) \le 1$ следует из (2.3). Утверждение доказано.

Говорят, что кодирование f сохраняет порядок, если из $i < j$ следует, что $f(a_i) < f(a_j)$ (имеется в виду лексикографический порядок на E^*). Рассмотрим префиксное кодирование Гильберта–Мура, которое сохраняет порядок и обладает небольшой избыточностью. Определим по индукции числа σ'_i: $\sigma'_1 = p(a_1)/2, \sigma'_{i+1} = \sigma'_i + (p(a_i) + p(a_{i+1}))/2$, $1 \le i \le k$. Ясно, что $0 < \sigma'_i < 1$. В качестве кодового слова $f(a_i)$ возьмем первые после запятой $\lceil \log \frac{1}{p(a_i)} \rceil + 1$ двоичных знаков числа σ'_i. Поскольку при $i < j$ имеем $\sigma'_i < \sigma'_j$, то и $f(a_i) < f(a_j)$.

Утверждение 2.4. *Кодирование Гильберта–Мура f — префиксное, $R(f,S) \le 2$.*

Доказательство этого утверждения аналогично доказательству утверждения 2.3.

Рассмотрим построение кода Шеннона–Фано. Пронумеруем буквы алфавита так, чтобы $p(a_1) \ge p(a_2) \ge \ldots \ge p(a_k) > 0$. Разделим алфавит на две части, имеющие наиболее близкие вероятности, не меняя порядка букв. Запишем 0 в качестве первого символа кода для всех букв первой половины алфавита и 1 в качестве первого символа кода для всех букв второй половины. Каждую из двух половин алфавита

делим опять на две части с возможно близкой вероятностью и приписываем к коду первых частей 0, к коду вторых частей — 1. Процесс деления продолжаем пока в каждой из частей не останется лишь по одной букве. Код Шеннона–Фано префиксный, поскольку кодовые слова оказываются кодами листовых вершин некоторого двоичного дерева.

Например, пусть $p(a_1) = 0,3, p(a_2) = 0,2, p(a_3) = p(a_4) = 0,15, p(a_5) = p(a_6) = 0,1$. Построим код Шеннона–Фано f для этого источника. Кодовое дерево изображено на рис. 2. Получаем кодовые слова: $f(a_1) = 00, f(a_2) = 01, f(a_3) = 100, f(a_4) = 101, f(a_5) = 110, f(a_6) = 111$.

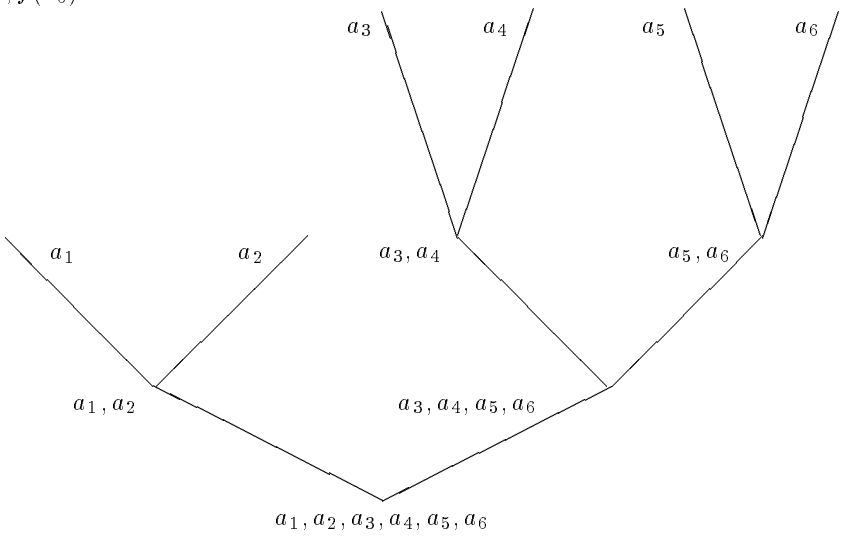


Рис. 2

Утверждение 2.5. *Для произвольного источника S с алфавитом $A = \{a_1, \ldots, a_k\}$, избыточность кодирования Шеннона–Фано не превышает избыточности кодирования Шеннона.*

Доказательство. Пусть f — кодирование Шеннона–Фано источника S. Заметим, что если все $p(a_i) = 2^{-m_i}$, где m_i — целые $i = 1 \ldots k$, то $|f(a_i)| = m_i$. Действительно, если $p(a_i) = 2^{-m_i}$, m_i — целые и вероятности букв a_i $i = 1 \ldots k$ упорядочены по убыванию, то алфавит A

можно, не меняя порядка букв, разделить на две части, вероятности которых равны. Очевидно, что для каждой из частей алфавита верно то же самое. Таким образом, мы получаем, что все выделенные на k-ой итерации части алфавита имеют одинаковые вероятности, равные 2^{-k}. Следовательно, буква a_i, имеющая вероятность $p(a_i) = 2^{-m_i}$, выделяется на m_i-ом шаге и $|f(a_i)| = m_i$. Т. е. если m_i целые, то утверждение верно.

Пусть числа m_i не целые. Введем обозначение $p'(a_i) = 2^{-\lceil m_i \rceil}$. Тогда $\sum_{i=1}^{k} p'(a_i) \leq 1$. Добавим к алфавиту A столько букв $a_{k+1}, a_{k+2}, \ldots, a_{k'}$ с вероятностями $p'(a_{k+j}) = p'(a_k)$, чтобы $\sum_{i=1}^{k'} p'(a_i) = 1$. Построим кодирование Шеннона–Фано f' для источника S' с алфавитом $A' = \{a_1, \ldots, a_{k'}\}$ и вероятностями букв $p'(a_i)$. По построению $k' \geq k$ и $p(a_i) \geq p'(a_i)$ при $i = 1 \ldots k$. Тогда после каждого деления на части алфавитов A и A' часть алфавита A будет содержать не больше букв, чем соответствующая часть алфавита A'. Следовательно,

$$|f(a_i)| \leq |f'(a_i)| \quad \text{при } i \leq k. \tag{2.4}$$

Но $|f'(a_i)| = \lceil \log \frac{1}{p(a_i)} \rceil$ при $i \leq k$, то есть $|f'(a_i)| = |g(a_i)|$ при $i \leq k$, где g — код Шеннона источника S. Тогда из неравенства (2.4) следует, что $C(f, S) \leq C(g, S)$, и утверждение доказано.

2.6. Оптимальное кодирование, код Хаффмана

Префиксное кодирование f_0 называется *оптимальным* для источника S, если для каждого префиксного кодирования f источника S справедливо неравенство $R(f_0, S) \leq R(f, S)$.

Для каждого источника существует оптимальный код, поскольку множество префиксных кодов источника с избыточностью, меньшей либо равной 1, непусто и конечно. Один источник может иметь несколько оптимальных кодов с разными наборами длин кодовых слов.

Утверждение 2.6. *Пусть S — некоторый источник с алфавитом $A = \{a_1, a_2, \ldots, a_k\}$ и упорядоченными вероятностями появления букв: $p(a_1) \geq p(a_2) \geq \ldots \geq p(a_k) > 0$. Тогда найдется такой оптимальный код f_0 источника S, что $|f_0(a_1)| \leq |f_0(a_2)| \leq \ldots \leq |f(a_{k-1})| = |f_0(a_k)|$. Причем кодовые слова $f(a_{k-1})$ и $f(a_k)$ отличаются только последним символом.*

Доказательство. Пусть f — некоторое оптимальное кодирование источника S и $|f(a_i)| > |f(a_j)|$ для некоторых $i < j$. Если $p(a_i) > p(a_j)$,

то, поменяв местами кодовые слова $f(a_i)$ и $f(a_j)$, мы получим код, имеющий меньшую стоимость, что противоречит оптимальности f. Если $p(a_i) = p(a_j)$, то, поменяв местами кодовые слова $f(a_i)$ и $f(a_j)$, мы получим код той же стоимости. То есть существует оптимальный код f_0, для которого справедливы неравенства

$$|f_0(a_1)| \le |f_0(a_2)| \le \ldots \le |f_0(a_k)|.$$

Пусть не существует кодового слова $f_0(a_i)$, отличного от $f_0(a_k)$ только последним символом. Изменим кодовое слово $f_0(a_k)$, удалив последний символ. После этого преобразования все кодовые слова останутся различными, и код останется префиксным, так как $f_0(a_k)$ было самым длинным кодовым словом. Стоимость получившегося кодирования на $p(a_k)$ меньше стоимости исходного кодирования, что противоречит оптимальности кода f_0. Т. е. предположение было неверным и найдется кодовое слово $f_0(a_i)$, отличающиеся от $f_0(a_k)$ только последним символом. Тогда

$$|f_0(a_i)| = |f_0(a_{i+1})| = \ldots = |f_0(a_k)|,$$

и, поменяв местами кодовые слова $f_0(a_i)$ и $f_0(a_{k-1})$, мы не изменим стоимости кодирования. Утверждение доказано.

Определим процедуру сжатия источника S с алфавитом $A = \{a_1, a_2, \ldots, a_k\}$, состоящую в слиянии двух наименее вероятных букв.
1) Перенумеруем буквы a_i так, чтобы $p(a_1) \ge p(a_2) \ge \ldots \ge p(a_k)$.
2) Пусть $A' = \{a'_1, a'_2, \ldots, a'_{k-1}\}$, определим источник S' с алфавитом A' и вероятностями букв $p(a'_i) = p(a_i)$ при $i < k-1$ и $p(a'_{k-1}) = p(a_{k-1}) + p(a_k)$.

Введем обозначение $S^{(i)} = (S^{(i-1)})'$. По определению процедуры сжатия алфавит источника $S^{(i)}$ содержит $k-i$ букв. Построим код Хаффмана h по индукции. Алфавит источника $S^{(k-2)}$ состоит из двух букв a_1^{k-2} и a_2^{k-2}. Пусть

$$h^{(k-2)}(a_1^{k-2}) = 0, \quad h^{(k-2)}(a_2^{k-2}) = 1.$$

Пусть $h^{(i+1)}$ — префиксный код источника $S^{(i+1)}$ — уже известен. Поскольку $S^{(i+1)} = (S^{(i)})'$ и по построению найдется буква $a_j^{i+1} \in A^{i+1}$ такая, что $p(a_j^{i+1}) = p(a_{k-i-1}^i) + p(a_{k-i}^i)$, можно определить

$$h^{(i)}(a_{k-i-1}^i) = h^{(i+1)}(a_j^{i+1})0 \quad \text{и} \quad h^{(i)}(a_{k-i}^i) = h^{(i+1)}(a_j^{i+1})1.$$

Кодовые слова для остальных букв, вероятности которых не изменились, оставим прежними, т. е. $h^{(i)}(a_m^i) = h^{(i+1)}(a_m^{i+1})$. Ясно, что из префиксности кода $h^{(i+1)}$ следует префиксность кода h^i . Полученный по индукции код $h = h^{(0)}$ — искомый код Хаффмана для источника S.

Построим, например, код Хаффмана для источника с вероятностями появления букв $p(a_1) = 0,3, p(a_2) = 0,2, p(a_3) = p(a_4) = 0,15, p(a_5) = p(a_6) = 0,1$. Процесс построения кода изображен в виде дерева на рис. 3. Слева от вершины указана соответствующая ей буква, а справа — вероятность буквы. Кодовые слова $f(a_1) = 00, f(a_2) = 10, f(a_3) = 010, f(a_4) = 011, f(a_5) = 110, f(a_6) = 111$ оказываются кодами листовых вершин дерева, что подтверждает префиксность кода Хаффмана.

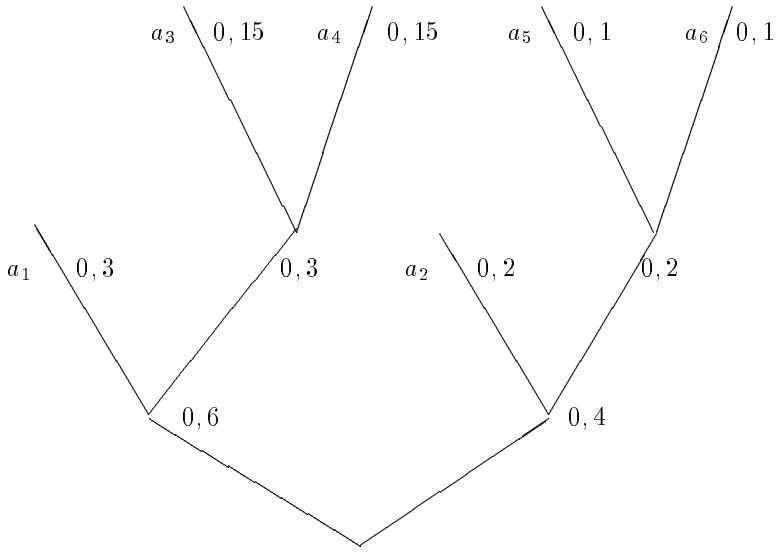


Рис. 3

Теорема 2.3. *Код Хаффмана является оптимальным.*

Доказательство. Пусть S — источник с алфавитом $A = \{a_1, a_2, \ldots, a_k\}$ и вероятностями $p(a_1), p(a_2), \ldots, p(a_k)$ появления букв.

Покажем по индукции, что коды $h^{(k-2)}, \ldots, h^{(i)}, \ldots, h^{(0)}$ являются оптимальными для источников $S^{(k-2)}, \ldots, S^{(i)}, \ldots, S$ соответственно.

Код $h^{(k-2)}$ является оптимальным, поскольку оба кодовых слова имеют минимальную длину 1.

Пусть код $h^{(i+1)}$ — оптимальный для источника $S^{(i+1)}$. Докажем от противного, что код $h^{(i)}$ — оптимальный для источника $S^{(i)}$. Пусть $\{a_1^i, a_2^i, \ldots, a_{k-i}^i\}$ — алфавит источника $S^{(i)}$ и $p(a_1^i) \geq p(a_2^i) \geq \ldots \geq p(a_{k-i}^i)$. Из утверждения 2.6 следует, что существует оптимальный код f источника $S^{(i)}$, для которого справедливы неравенства

$$|f(a_1^i)| \leq |f(a_2^i)| \leq \ldots \leq |f(a_{k-i-1}^i)| = |f(a_{k-i}^i)|,$$

причем кодовые слова $f(a_{k-i-1}^i)$ и $f(a_{k-i}^i)$ отличаются только последним символом. Рассмотрим источник $S^{(i+1)}$; по определению процедуры сжатия источника имеем $p(a_j^{i+1}) = p(a_j^i)$ при $j < k-i-1$ и $p(a_{k-i-1}^{i+1}) = p(a_{k-i-1}^i) + p(a_{k-i}^i)$. Определим код f' источника $S^{(i+1)}$ следующим образом:

$$f'(a_j^{i+1}) = f(a_j^i) \quad \text{при } j < k-i-1.$$

А кодовое слово $f'(a_{k-i-1}^{i+1})$ получим из кодового слова $f'(a_{k-i}^i)$ удалением последнего символа. Поскольку код f является префиксным и кодовое слово $f(a_{k-i}^i)$ имело наибольшую длину, то полученный код f' тоже является префиксным. По определению стоимости кодирования справедливы равенства

$$C(f', S^{(i+1)}) = \sum_{j=1}^{k-i} |f'(a_j^{i+1})| p(a_j^{i+1}) = C(f, S^{(i)}) + |f'(a_{k-i-1}^{i+1})| p(a_{k-i-1}^{i+1}) -$$

$$-|f(a_{k-i-1}^i)| p(a_{k-i-1}^i) - |f(a_{k-i}^i)| p(a_{k-i}^i) = C(f, S^{(i)}) - p(a_{k-i-1}^{i+1}).$$

Аналогично получаем, что

$$C(h^{(i+1)}, S^{(i+1)}) = C(h^{(i)}, S^{(i)}) - p(a_{k-i-1}^{i+1}).$$

Тогда из двух предыдущих равенств и предположения о неоптимальности $h^{(i)}$ имеем

$$C(h^{(i+1)}, S^{(i+1)}) - C(f', S^{(i+1)}) = C(h^{(i)}, S^{(i)}) - C(f, S^{(i)}) > 0.$$

Последнее неравенство противоречит оптимальности кода $h^{(i+1)}$. Теорема доказана.

3. Блочное и неблочное кодирование

3.1. Стационарные источники. Энтропия стационарного источника

Пусть $A = \{a_1, a_2, \ldots, a_k\}$ — конечный алфавит. Рассмотрим в качестве пространства событий Ω множество бесконечных (в обе стороны) последовательностей букв алфавита A, т. е. $\Omega = A^\infty$. Пусть $S_i^j \subset A^\infty$ состоит из последовательностей, имеющих на j-ом месте букву a_i. Ясно, что множество $S^j = \{S_i^j\}_{i=1\ldots k}$ является разбиением A^∞.

Источник S определяется как множество разбиений S^j с совокупностью всевозможных условных вероятностей элементов разбиений $p(S_{i_0}^{j_0}|S_{i_1}^{j_1}\ldots S_{i_n}^{j_n})$ и $p(S_i^j)$.

Источник S называется *стационарным*, если вероятности $p(S_{i_0}^{j_0}|S_{i_1}^{j_1}\ldots S_{i_n}^{j_n})$ и $p(S_i^j)$ независимы относительно сдвигов, т. е. справедливы равенства

$$p(S_{i_0}^{j_0}|S_{i_1}^{j_1}\ldots S_{i_n}^{j_n}) = p(S_{i_0}^{0}|S_{i_1}^{j_1-j_0}\ldots S_{i_n}^{j_n-j_0}), \qquad p(S_i^{j_0}) = p(S_i^0).$$

Если S — стационарный источник, то события $S_{i_1}^1 S_{i_2}^2 \ldots S_{i_n}^n$ обычно отождествляются с соответствующими наборами букв $a_{i_1}a_{i_2}\ldots a_{i_n}$ и вместо $p(S_{i_n}^{j+n}|S_{i_0}^j S_{i_1}^{j+1}\ldots S_{i_{n-1}}^{j+n-1})$ пишут $p(a_{i_n}|a_{i_0}a_{i_1}\ldots a_{i_{n-1}})$, подразумевая под этим вероятность появления буквы a_{i_n} после набора букв $a_{i_0}a_{i_1}\ldots a_{i_{n-1}}$.

Утверждение 3.1. *Пусть S — стационарный источник, тогда последовательность $\kappa_n = H(S^n|S^1S^2\ldots S^{n-1})$ монотонно убывает.*

Доказательство. Поскольку S — стационарный источник, то для всевозможных $i_1, \ldots, i_n$ справедливы равенства

$$p(S_{i_n}^{n+1}|S_{i_1}^2 S_{i_2}^3 \ldots S_{i_{n-1}}^n) = p(S_{i_n}^n|S_{i_1}^1 S_{i_2}^2 \ldots S_{i_{n-1}}^{n-1}),$$

$$p(S_{i_1}^2 S_{i_2}^3 \ldots S_{i_n}^{n+1}) = p(S_{i_1}^1 S_{i_2}^2 \ldots S_{i_n}^n).$$

Тогда

$$H(S^n|S^1S^2\ldots S^{n-1}) = H(S^{n+1}|S^2S^3\ldots S^n). \tag{3.1}$$

Поскольку $S^1S^2\ldots S^n \succeq S^2S^3\ldots S^n$, то из утверждения 1.3 следует, что

$$H(S^{n+1}|S^1S^2\ldots S^n) \le H(S^{n+1}|S^2S^3\ldots S^n).$$

Из (3.1) и последнего неравенства утверждение доказано.

Утверждение 3.2. *Пусть S — стационарный источник, тогда*

$$\lim_{n\to\infty}\frac{1}{n}H(S^1S^2\ldots S^n)=\lim_{n\to\infty}H(S^n|S^1S^2\ldots S^{n-1}).$$

Доказательство. Пусть $\kappa_n = H(S^n|S^1S^2\ldots S^{n-1})$. Из определения энтропии следует, что $\kappa_n \geq 0$. Из утверждения 3.1 следует, что последовательность κ_n монотонно убывает. Тогда существует предел $\lim_{n\to\infty}\kappa_n \geq 0$.

Покажем по индукции, что $H(S^1S^2\ldots S^n)=\sum_{i=1}^{n}\kappa_i$. По определению $H(S^1)=\kappa_1$. Пусть $H(S^1S^2\ldots S^{n-1})=\sum_{i=1}^{n-1}\kappa_i$, тогда из утверждения 1.2 имеем

$$H(S^1S^2\ldots S^n)=H(S^1S^2\ldots S^{n-1})+H(S^n|S^1S^2\ldots S^{n-1})=\kappa_n+\sum_{i=1}^{n-1}\kappa_i.$$

Известно, что предел среднего арифметического последовательности равен пределу последовательности, поэтому

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\kappa_i=\lim_{n\to\infty}\kappa_n.$$

Утверждение доказано.

Энтропией стационарного источника S называется величина

$$H(S)=\lim_{n\to\infty}H(S^n|S^1S^2\ldots S^{n-1}).$$

Стационарный источник S называется источником Бернулли, если $p(S_i^j|S_{i_1}^{j_1}S_{i_2}^{j_2}\ldots S_{i_n}^{j_n})=p(S_i^j)$. Другими словами, S — источник Бернулли, если вероятность появления буквы не зависит ни от места в последовательности, ни от предыдущих букв. Для источника Бернулли новое определение энтропии совпадает с определением, использовавшимся ранее:

$$H(S)=H(S^1)=-\sum_{i=1}^{k}p(S_i^1)\log p(S_i^1)=-\sum_{i=1}^{k}p(a_i)\log p(a_i).$$

Стационарный источник S называется *марковским источником r-го порядка*, если

$$p(S_i^j|S_{i_n}^{j-n}S_{i_{n-1}}^{j-n+1}\ldots S_{i_1}^{j-1})=p(S_i^j|S_{i_r}^{j-r}S_{i_{r-1}}^{j-r+1}\ldots S_{i_1}^{j-1})$$

при $r \le n$. Другими словами, вероятность появления следующей буквы зависит только от r предыдущих. Если S — марковский источник r-го порядка, то

$$H(S) = H(S^{r+1}|S^1S^2\dots S^r) =$$

$$= -\sum_{i_1\dots i_{r+1}} p(S^1_{i_1}S^2_{i_2}\dots S^r_{i_r}S^{r+1}_{i_{r+1}})\log p(S^{r+1}_{i_{r+1}}|S^1_{i_1}S^2_{i_2}\dots S^r_{i_r}) =$$

$$-\sum_{i_1\dots i_{r+1}} p(a_{i_1}\dots a_{i_{r+1}})\log p(a_{i_{r+1}}|a_{i_1}\dots a_{i_r}). \qquad (3.2)$$

Пронумеруем подряд все возможные слова из r букв $s_i = a_{i_1}\dots a_{i_r}$. Слова s_i называются *состояниями* марковского источника S r-го порядка. При появлении каждой новой буквы источник S переходит в новое состояние

$$s_i = a_{i_1}\dots a_{i_r} \to s_j = a_{i_2}\dots a_{i_{r+1}}.$$

Можно определить марковский источник общего вида, состояния которого не связаны с наборами из фиксированного числа букв.

Марковский источник называется *эргодическим*, если вероятность перехода через произвольное (большее некоторого фиксированного числа m) число шагов из каждого состояния s_i в произвольное состояние s_j больше нуля.

Если для некоторого эргодического марковского источника с n состояниями известны только вероятности перехода из одного состояния в другое, то вероятности его состояний можно получить из системы уравнений

$$\sum_{j=1}^{n} p(s_j)p(s_i|s_j) = p(s_i), \qquad \sum_{j=1}^{n} p(s_j) = 1. \qquad (3.3)$$

Пусть S — марковский источник первого порядка с алфавитом $\{a_1,\dots,a_k\}$, и вероятности $p(a_i|a_j)$ появления буквы a_i вслед за буквой a_j записаны в матрице $Q = \{q_{ij}\}$, где $q_{ij} = p(a_i|a_j)$. Тогда вероятности $p(a_i)$ можно вычислить из уравнения $Qp = p$, где $p = (p(a_1),\dots,p(a_k))$, учитывая, что $\sum_{i=1}^{k} p(a_i) = 1$.

Марковский источник можно полностью задать набором его состояний $s_1,\dots,s_n$, матрицей вероятностей перехода из одного состояния в другое $p(s_i|s_j)$ и набором вероятностей порождения букв в каждом из состояний $p(a_1|s_j),\dots,p(a_k|s_j)$ $1 \le j \le n$. Если считать, что последовательности, порождаемые источником, бесконечны только в одну

сторону, т. е. имеют начало, то для полного определения источника нужно задать еще начальное состояние.

Разделим бесконечную последовательность букв, порожденную марковским источником S с состояниями $s_1, \ldots, s_n$ на n подпоследовательностей, каждая из которых состоит из букв, порожденных в определенном состоянии s_j. Поскольку вероятность появления очередной буквы зависит только от состояния источника, то буквы j-й последовательности появляются независимо друг от друга с вероятностями $p(a_i|s_j)$. Имея в виду этот факт, говорят, что марковский источник разделяется на n источников Бернулли, каждый из которых определяется набором условных вероятностей $p(a_1|s_j), \ldots, p(a_k|s_j) \quad 1 \leq j \leq n$, и обладает энтропией

$$H_j(S) = -\sum_{i=1}^{k} p(a_i|s_j) \log p(a_i|s_j).$$

Из формулы (3.2) получаем формулу для энтропии марковского источника S с состояниями $s_1, \ldots, s_n$:

$$H(S) = -\sum_{j=1}^{n} p(s_j) \sum_{i=1}^{k} p(a_i|s_j) \log p(a_i|s_j) = -\sum_{j=1}^{n} p(s_j) H_j(S).$$

Таким образом, последовательность, порожденную марковским источником, можно эффективно кодировать, разделяя ее по числу состояний на n подпоследовательностей и кодируя каждую из них отдельно любым из побуквенных кодов, описанных в предыдущей главе.

3.2. Блочное кодирование и теорема кодирования Шеннона

Введем общее понятие кодирования. Пусть $A = \{a_1, a_2, \ldots, a_k\}$ и $E = \{0, 1\}$. $A^* = \cup_{i=1}^{\infty} A^i$ — множество конечных последовательностей букв алфавита A, и E^* — множество конечных двоичных последовательностей.

Кодированием (кодом) называется инъективное отображение $f : A^* \to E^*$.

Побуквенные дешифруемые коды $f : A \to E^*$ являются частным случаем кодирования, определенного выше, поскольку возможно доопределение $f(a_{i_1} a_{i_2} \ldots a_{i_n}) = f(a_{i_1}) f(a_{i_2}) \ldots f(a_{i_n})$.

Кодирование f называется *m-блочным* , если $f : A^m \to E^*$. В этом случае кодом слова произвольной длины является конкатенация кодов

блоков, составляющих данное слово. Если кодируемое слово не разделяется на целое число блоков длиной m, то его нужно дополнить произвольным образом и в конце кода слова приписать число добавленных букв.

Кодирование f называется *префиксным*, если для любых двух слов $x, y \in A^n$ одинаковой длины $f(x)$ не является префиксом $f(y)$.

Стоимостью кодирования f стационарного источника S с алфавитом A называется $C(f,S) = \limsup_{n\to\infty} C_n(f,S)$, где

$$C_n(f,S) = \frac{1}{n}\sum_{x\in A^n} p(x)|f(x)|.$$

В частности, для n-блочного кодирования f имеем $C(f,S) = C_n(f,S)$.

Избыточностью кодирования f стационарного источника S называется

$$R(f,S) = C(f,S) - H(S).$$

Мы также будем использовать обозначение $R_n(f,S) = C_n(f,S) - H(S)$.

Теорема 3.1.

1) Для каждого стационарного источника S с алфавитом A и произвольного префиксного кода f избыточность кодирования неотрицательна, т. е. $R(f,S) \geq 0$.

2) Для каждого стационарного источника S с алфавитом A найдется блочное префиксное кодирование со сколь угодно малой избыточностью. [6]

Доказательство. Рассмотрим алфавит $B = \{b_1, b_2, \ldots, b_{|A|^m}\}$, каждая буква которого является блоком из m букв алфавита A, т. е. $b_j = a_{i_1}a_{i_2}\ldots a_{i_m}$. Рассмотрим источник $Z(m)$ с алфавитом B, причем $Z^i(m) = S^{1+(i-1)m}\ldots S^{m+(i-1)m}$. Источник $Z(m)$ — стационарный, поскольку источник S — стационарный. Пусть $f : A^* \to E^*$ — префиксное кодирование, тогда f_m — сужение кода f на алфавит $B = A^m \subset A^*$ — является побуквенным префиксным кодом. Из теоремы 2.2 следует неравенство

$$0 \leq \sum_{x\in B} p(x)|f_m(x)| - H(Z^1(m)) = \sum_{x\in A^m} p(x)|f(x)| - H(S^1\ldots S^m).$$

[6] Теорема 3.1 является частным случаем теоремы кодирования Шеннона, которая будет сформулирована в главе 5.

Разделим последнюю формулу на m и перейдем к пределу при $m \to \infty$:

$$\limsup_{m\to\infty} \frac{1}{m} \sum_{x \in A^m} p(x)|f(x)| - \lim_{m\to\infty} \frac{1}{m} H(S^1 \dots S^m) \geq 0.$$

Тогда из утверждения 3.2 и определения стоимости кодирования следует, что $C(f,S) - H(S) \geq 0$. Первое утверждение теоремы доказано.

С другой стороны, из теоремы 2.2 следует, что для каждого источника найдется побуквенный префиксный код, избыточность которого не превышает 1. То есть существует такой код $f_m : B \to E^*$, что $R(f_m, Z^1(m)) \leq 1$. Тогда $\sum_{x \in A^m} p(x)|f_m(x)| - H(S^1 \dots S^m) \leq 1$ и

$$\frac{1}{m} \sum_{x \in A^m} p(x)|f(x)| - H(S) \leq \frac{1}{m} + \frac{1}{m} H(S^1 \dots S^m) - H(S). \qquad (3.4)$$

Из утверждения 3.2 следует, что величину $\frac{1}{m} H(S^1 \dots S^m) - H(S)$ можно сделать сколь угодно малой при $m \to \infty$. Заметим, что f_m — m-блочное кодирование по определению. Теорема доказана.

Следствие 3.1. *Пусть S — источник Бернулли, тогда найдется m-блочный код f_m такой, что $R(f,S) \leq \frac{1}{m}$.*

Доказательство. Для источника Бернулли S справедливо равенство

$$H(S^1 \dots S^m) = H(S^1) + H(S^2) + \dots + H(S^m) = mH(S),$$

тогда искомое утверждение следует из неравенства (3.4).

3.3. Неблочное кодирование, его энтропия и стоимость

В предыдущем пункте было рассмотрено кодирование, которое блоки из постоянного числа букв отображает в кодовые слова различной длины. В этом пункте мы рассмотрим кодирование, ставящее в соответствие словам различной длины кодовые слова одинаковой длины. Такое кодирование называют кодированием типа VB в отличие от блочного кодирования, имеющего тип BV. Можно также рассматривать кодирование типа VV, которое слова переменной длины отображает в слова переменной длины.

Пусть имеется источник S с алфавитом $A = \{a_1, \dots, a_k\}$. Пусть Δ — некоторое дерево, каждая вершина которого имеет k сыновей. Каждому из k направлений движения по дереву поставим в соответствие одну из букв алфавита. Каждую вершину дерева Δ отождествим

со словом алфавита A, описывающим путь из корня дерева в его вершину. Будем говорить, что вершина $x \in \Delta$ имеет вероятность $p(x)$, которая равна вероятности слова $x \in A^*$, порожденного источником S. Обозначим через Δ' множество листовых вершин дерева Δ и через $d(\Delta)$ — среднюю высоту дерева, т. е. $d(\Delta) = \sum_{x \in \Delta'} p(x)|x|$.

Утверждение 3.3. *Пусть S — источник Бернулли с алфавитом $A = \{a_1, \ldots, a_k\}$ и Δ — некоторое k-ичное дерево. Тогда справедливы равенства:*

1) $\sum_{x \in \Delta'} p(x) = 1$,

2) $d(\Delta) = \sum_{x \in \Delta} p(x)$,

3) $H(S) = -\frac{1}{d(\Delta)} \sum_{x \in \Delta'} p(x) \log p(x)$.

Доказательство. Докажем все три равенства индукцией по дереву. Пусть дерево Δ состоит только из корня и листовых вершин. Тогда равенства 1–3 справедливы, поскольку вероятность корня равна 0, а вероятности листовых вершин совпадают с вероятностями букв. Пусть равенства 1–3 справедливы для дерева Δ. Пусть Δ' — множество листов дерева Δ и $y \in \Delta'$. Добавим к дереву Δ листовые вершины $ya_1, ya_2, \ldots, ya_k$. Полученное дерево обозначим через Γ. Поскольку S — источник Бернулли, то $p(ya_i) = p(y)p(a_i)$ и

$$\sum_{x \in \Gamma'} p(x) = \sum_{x \in \Delta'} p(x) - p(y) + \sum_{i=1}^{k} p(ya_i) = 1,$$

т. е. первое равенство доказано. Имеем

$$\sum_{x \in \Gamma'} p(x)|x| - \sum_{x \in \Delta'} p(x)|x| = \sum_{i=1}^{k} p(ya_i)|ya_i| - p(y)|y| =$$

$$= \sum_{i=1}^{k} p(y)p(a_i)\,(|y| + 1) - p(y)|y| = p(y),$$

$$\sum_{x \in \Gamma} p(x) - \sum_{x \in \Delta} p(x) = \sum_{i=1}^{k} p(ya_i) = p(y). \tag{3.5}$$

Из двух последних равенств и предположения индукции следует, что

$$\sum_{x \in \Gamma} p(x) = \sum_{x \in \Gamma'} p(x)|x|.$$

Из (3.5) получаем равенство

$$H(S)\sum_{x\in\Gamma}p(x)-H(S)\sum_{x\in\Delta}p(x)=H(S)p(y).$$

По предположению индукции имеем

$$H(S)\sum_{x\in\Delta}p(x)=-\sum_{x\in\Delta'}p(x)\log p(x).$$

Тогда

$$H(S)\sum_{x\in\Gamma}p(x)=-\sum_{x\in\Delta'}p(x)\log p(x)+H(S)p(y)=$$

$$-\sum_{x\in\Delta'}p(x)\log p(x)-\sum_{i=1}^{k}p(ya_i)\log p(a_i)=-\sum_{x\in\Gamma'}p(x)\log p(x)+$$

$$+\sum_{i=1}^{k}(p(ya_i)\log p(ya_i)-p(ya_i)\log p(a_i))-p(y)\log p(y)=-\sum_{x\in\Gamma'}p(x)\log p(x).$$

Утверждение доказано.

Пусть Δ — k-ичное дерево, вершины которого соответствуют словам из букв алфавита $A=\{a_1,\ldots,a_k\}$. Поскольку $\sum_{x\in\Delta'}p(x)=1$, то множество Δ' можно считать некоторым алфавитом. Рассмотрим произвольное префиксное кодирование $f:\Delta'\to E^*$. Отображение f можно доопределить на всем множестве A^* так же, как это было сделано выше для блочного кода. Если все листы дерева Δ имеют одинаковую высоту, то f — блочное кодирование или кодирование типа BV. Если листовые вершины имеют различную высоту, а кодовые слова имеют одинаковую длину, то f — кодирование типа VB.

Утверждение 3.4. *Пусть S — источник Бернулли с алфавитом $A=\{a_1,\ldots,a_k\}$, и Δ — k-ичное дерево, $f:\Delta'\to E^*$ — префиксное кодирование* [7]*. Тогда*

$$C(f,S)=\frac{1}{d(\Delta)}\sum_{x\in\Delta'}p(x)|f(x)|. \tag{3.6}$$

[7] Утверждение 3.4 справедливо для произвольного марковского источника.

Доказательство. Процедуру кодирования порождаемой источником S последовательности определяет случайное движение по дереву Δ, начиная с корня. Если a_i — первая буква последовательности, то переходим к i-тому сыну корня и т. д., пока не достигнем листовой вершины, от которой вернемся к корню и начнем сначала. Этот процесс описывается некоторым источником S_Δ с алфавитом Δ. Если S — источник Бернулли, то S_Δ — марковский источник первого порядка, так как вероятность перехода к следующей вершине дерева зависит только от текущей вершины. Причем справедливы равенства:

$p_\Delta(x|y) = p(a_i)$, если $x = ya_i$ и $y \notin \Delta'$,

$p_\Delta(x|y) = 0$, если $x \neq ya_i$ и $y \notin \Delta'$,

$p_\Delta(x|y) = p(a_i)$, если $x = a_i$ и $y \in \Delta'$,

$p_\Delta(x|y) = 0$, если $x \neq a_i$ и $y \in \Delta'$,

где $p_\Delta(x|y)$ — условные вероятности порождения букв источником S_Δ. Числа $p_\Delta(x)$, $x \in \Delta$ должны удовлетворять системе уравнений (3.3). Тогда справедливы равенства

$$\begin{cases} p_\Delta(xa_i) = \sum_{y\in\Delta} p_\Delta(xa_i|y)p_\Delta(y) = p(a_i)p_\Delta(x), \\ p_\Delta(a_i) = \sum_{y\in\Delta} p_\Delta(a_i|y)p_\Delta(y) = \sum_{y\in\Delta'} p(a_i)p_\Delta(y), \\ \sum_{x\in\Delta} p_\Delta(x) = 1. \end{cases}$$

Из утверждения 3.3 следует, что $\sum_{x\in\Delta'} p(x) = 1$ и $\sum_{x\in\Delta} p(x) = d(\Delta)$. Тогда решение системы уравнений —

$$p_\Delta(x) = \frac{1}{d(\Delta)} p(x). \tag{3.7}$$

Пусть $f : \Delta' \to E^*$ — префиксное кодирование источника S. В то же время f можно рассматривать как отображение из Δ в $E^* \cup \emptyset$, определив $f(x) = \emptyset$ при $x \in \Delta\backslash\Delta'$. Тогда отображение f можно считать побуквенным кодированием источника S_Δ, порождающего те же последовательности, что и источник S. Тогда

$$C(f,S) = C(f,S_\Delta) = \sum_{x\in\Delta} p_\Delta(x)|f(x)| = \sum_{x\in\Delta'} p_\Delta(x)|f(x)|.$$

Из (3.7) и последнего равенства следует равенство (3.6). Утверждение доказано.

3.4. Кодирование Ходака

Пусть S — стационарный источник с алфавитом $A = \{a_1, \ldots, a_k\}$ и $m > 0$ — целое. Построим по индукции k-ичное дерево Δ, начиная с дерева, состоящего только из корня. Пусть построено некоторое k-ичное дерево Γ, вероятности вершин которого определяются источником S. Выберем лист $x \in \Gamma'$ с наибольшей вероятностью и добавим всех его сыновей к дереву Γ. Будем повторять эту процедуру, пока число листов дерева не превышает 2^m. Всем листам полученного дерева Δ припишем в качестве кодового слова различные двоичные слова длины m. Таким образом мы получим кодирование $f : \Delta' \to E^m$ типа VB, предложенное Г. Л. Ходаком в 1969 году [8]. Построим, например, кодирование Ходака с длиной кодовых слов $m = 3$ для источника Бернулли с вероятностями букв $p(a_1) = 2/5$, $p(a_2) = 3/5$. На рис. 4 изображено искомое дерево Δ, буква a_1 соответствует левому направлению, a_2 — правому. Рядом с каждой вершиной дерева указана ее вероятность.

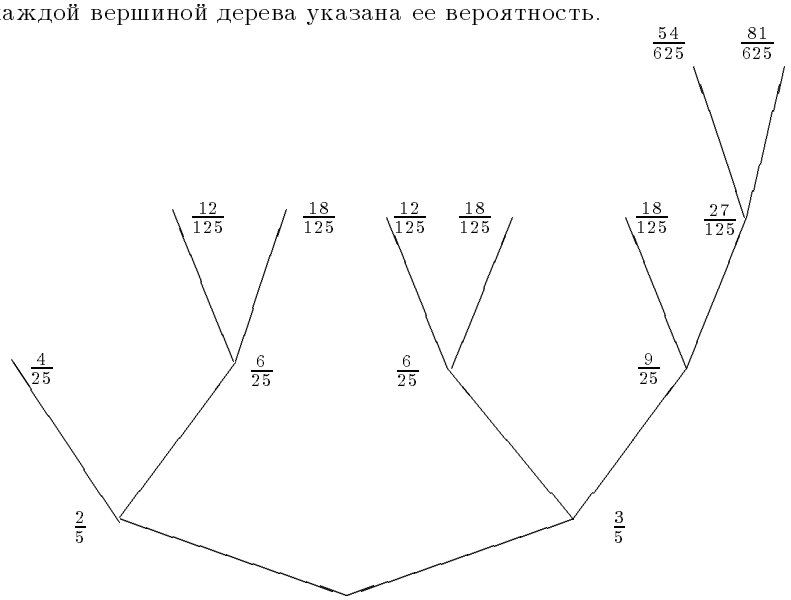


Рис. 4

[8] Аналогичный метод кодирования был предложен Б. П. Танстэлом в 1967 г.

Мы получили следующий код: $f(a_1a_1) = 000, f(a_1a_2a_1) = 001, f(a_1a_2a_2) = 010, f(a_2a_1a_1) = 011, f(a_2a_1a_2) = 100, f(a_2a_2a_1) = 101, f(a_2a_2a_2a_1) = 110, f(a_2a_2a_2a_2) = 111$.

Утверждение 3.5. *Пусть S — источник Бернулли с алфавитом $A = \{a_1, \ldots, a_k\}$ и $p(a_k) = \min p(a_i) > 0$. Пусть f — кодирование Ходака и Δ — дерево, соответствующее коду. Тогда*

$$R(f, S) \leq -\frac{\log p(a_k)}{d(\Delta)}.$$

Доказательство. Пусть $x \in \Delta'$. Докажем от противного, что

$$2^m p(x) \leq 1/p(a_k). \tag{3.8}$$

Пусть $p(xa_k) = p(x)p(a_k) > 1/2^m$. По построению если $y \in \Delta'$, то $y = za_i$ и $p(z) \geq p(x)$. Тогда $p(y) = p(z)p(a_i) \geq p(xa_k)$. Обозначим через $\bar{\Delta}'$ объединение множества Δ' и сыновей вершины x. По определению кодирования Ходака имеем $|\bar{\Delta}'| > 2^m$. Тогда

$$\sum_{y \in \bar{\Delta}'} p(y) > 2^m p(xa_k) > 1,$$

что противоречит утверждению 3.3. Т. е. формула (3.8) доказана. Из утверждений 3.3, 3.4 и формулы (3.8) получаем

$$R(f, S) = C(f, S) - H(S) = \frac{1}{d(\Delta)} \sum_{x \in \Delta'} p(x)|f(x)| + \frac{1}{d(\Delta)} \sum_{x \in \Delta'} p(x) \log p(x) =$$

$$= \frac{1}{d(\Delta)} \sum_{x \in \Delta'} p(x) \log(2^m p(x)) \leq \frac{1}{d(\Delta)} \log \frac{1}{p(a_k)} \sum_{x \in \Delta'} p(x) = -\frac{\log p(a_k)}{d(\Delta)}.$$

Утверждение доказано.

3.5. Арифметическое кодирование

Основная идея арифметического кодирования состоит в следующем наблюдении. Пусть S — марковский источник с алфавитом $A = \{a_1, \ldots, a_k\}$. Упорядочим все наборы из n букв алфавита A лексикографически, т. е. $a_{i_1}, a_{i_2}, \ldots, a_{i_n} \prec a_{j_1}, a_{j_2}, \ldots, a_{j_n}$, если $i_k = j_k$

при $k < l$ и $i_l \le j_l$. Для каждого слова $x \in A^n$ определим величины $L(x) = \sum\limits_{y \prec x,\, y \in A^n} p(y)$ и $R(x) = L(x) + p(x)$.

В каждом полуинтервале длины I, $I \le 1$ найдется двоично-рациональное число со знаменателем $2^{\lceil -\log I \rceil}$, поскольку разность между двумя любыми ближайшими числами с таким знаменателем не превосходит I ($2^{-\lceil -\log I \rceil} \le 2^{\log I} = I$). Поставим в соответствие каждому слову $x \in A^n$ двоично-рациональное число $q(x) \in [L(x), R(x))$ со знаменателем $2^{\lceil -\log p(x) \rceil}$. В качестве кода $f(x)$ рассмотрим двоичную запись числителя числа $q(x)$ с использованием $\lceil -\log p(x) \rceil$ двоичных символов, т. е.

$$|f(x)| = \lceil -\log p(x) \rceil. \tag{3.9}$$

Поскольку полуинтервалы $[L(x), R(x))$, соответствующие различным последовательностям x одинаковой длины, не пересекаются, то $|f(xa_i)| > |f(x)|$ и все числа $q(x)$, $x \in A^n$ попарно различны, т. е. отображение $f : A^n \to E^*$ инъективно. Если $p(a_i|s_j) < 1/2$ для всех $a_i \in A$ и s_j — состояний источника S, то $|f(xa_i)| > |f(x)|$ и отображение $f : A^* \to E^*$, оказывается инъективным.[9] Оценим избыточность кодирования f. Из (3.9) имеем

$$C_n(f, S) = \frac{1}{n} \sum_{x \in A^n} p(x) \lceil -\log p(x) \rceil.$$

Тогда аналогично неравенству (3.4) из последнего неравенства получаем, что

$$R(f, S) = \lim_{n \to \infty} \frac{1}{n} \sum_{x \in A^n} p(x)(\log p(x) + \lceil -\log p(x) \rceil) +$$

$$+ \lim_{n \to \infty} \left(\frac{1}{n} H(S^1 \ldots S^n) - H(S)\right) = 0.$$

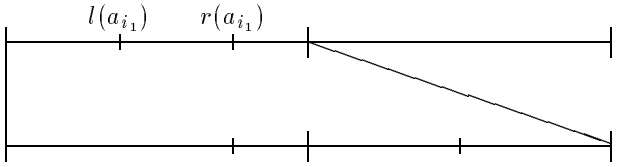
К сожалению, в таком виде арифметическое кодирование не может применяться на практике, поскольку требуемая точность вычислений быстро растет с увеличением длины кодируемой последовательности. Ниже будет описана наиболее известная практическая реализация арифметического кодирования, предложеная Й. Риссаненом в 1976 г.

[9] Кодирование f можно сделать префиксным, удлинив каждое кодовое слово на один символ, как это сделано в коде Гильберта–Мура.

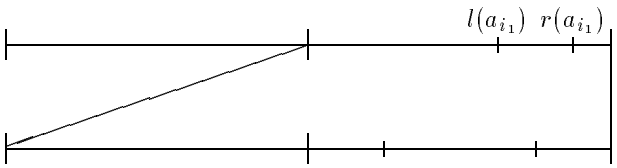
Этот метод использует арифметические вычисления с числами, двоичная запись которых содержит не более t двоичных знаков. Целое число $t > 0$ является параметром кодирования. Для упрощения вычислений положим, что S — источник Бернулли и $p(a_i) \le 1/4$ для всех $1 \le i \le k$. Определим величины $\sigma_1 = 0$, $\sigma_i = \sigma_{i-1} + p(a_{i-1})$. Разделим полуинтервал $[0,1)$ на полуинтервалы $i(a_i) = [l(a_i), r(a_i))$, где $l(a_i) = \lfloor \sigma_i 2^t \rfloor / 2^t$ и $r(a_i) = \lfloor \sigma_{i+1} 2^t \rfloor / 2^t$. Каждой букве $a_i \in A$ соответствует i-ый полуинтервал, длина которого приблизительно равна $p(a_i)$. Пусть a_{i_1} — первая буква кодируемого слова $x \in A^n$. Поскольку $p(a_i) \le 1/4$, возможны три случая:
1) $i(a_{i_1}) \subset [0, 1/2)$;
2) $i(a_{i_1}) \subset [1/2, 1)$;
3) $i(a_{i_1}) \subset [1/4, 3/4)$.

Пусть $I(x)$ — полуинтервал, соответствующий слову $x \in A^n$; в первом случае имеем $I(x) \subset i(a_{i_1}) \subset [0, 1/2)$. Тогда все числа $q \in I(x)$ имеют 0 первым символом после запятой. Поэтому 0 — первый символ кода $f(x)$. Во втором случае первым символом кода $f(x)$ является 1. В третьем случае первый символ пока неизвестен.

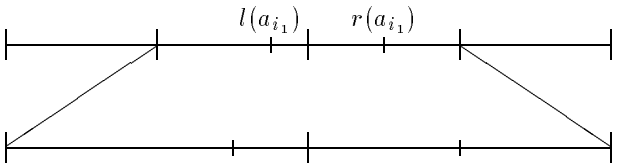
Проведем операцию изменения масштаба (растяжения) полуинтервала. В первом случае новый полуинтервал — $[2l(a_{i_1}), 2r(a_{i_1}))$.



Во втором случае — $[2l(a_{i_1}) - 1, 2r(a_{i_1}) - 1)$.



В третьем случае — $[2l(a_{i_1}) - 1/2, 2r(a_{i_1}) - 1/2)$.



В каждом случае новый полуинтервал длиннее полуинтервала $i(a_{i_1})$ в два раза. Будем продолжать процедуру изменения масштаба до тех пор, пока растянутый полуинтервал укладывается целиком в правую, левую или среднюю половину полуинтервала $[0, 1)$. При растяжении левой половины полуинтервала $[0, 1)$ будем приписывать 0 в качесте следующего символа кода $f(x)$, при растяжении правой половины $[0, 1)$ будем приписывать 1. Если перед растяжением левой (правой) половины $[0, 1)$ текущий полуинтервал оказывался h раз в средней половине, то после нуля (единицы) нужно приписать h единиц (нулей), т. е. в целом $01\ldots1\,(10\ldots0)$.

В итоге получим полуинтервал $[l/2^t, r/2^t)$, где l и r — целые, $0 \le l < r \le 2^t$ и $r - l > 2^{t-2}$. Из $[l/2^t, r/2^t)$ выделим полуинтервал, соответствующий a_{i_2} — второй букве слова x: $i(a_{i_1}a_{i_2}) = [l(a_{i_1}a_{i_2}), r(a_{i_1}a_{i_2}))$, где $l(a_{i_1}a_{i_2}) = (l + \lfloor (r-l)\sigma_{i_2} \rfloor)/2^t$ и $r(a_{i_1}a_{i_2}) = (l + \lfloor (r-l)\sigma_{1+i_2} \rfloor)/2^t$. Проделаем с полуинтервалом $i(a_{i_1}a_{i_2})$ те же операции изменения масштаба, что и с полуинтервалом $i(a_{i_1})$. В результате получим начало кода $f(x)$, соответствующее двум начальным буквам. Выполнив такую же процедуру для букв $a_{i_3}, a_{i_4}, \ldots a_{i_n}$, получим код $f(x)$. Если соответствующий последней букве полуинтервал после растяжения не совпадает с $[0, 1)$, то к коду нужно приписать еще два символа, указывающих, в какой четверти $[0, 1)$ находится последнее число $l/2^t$.

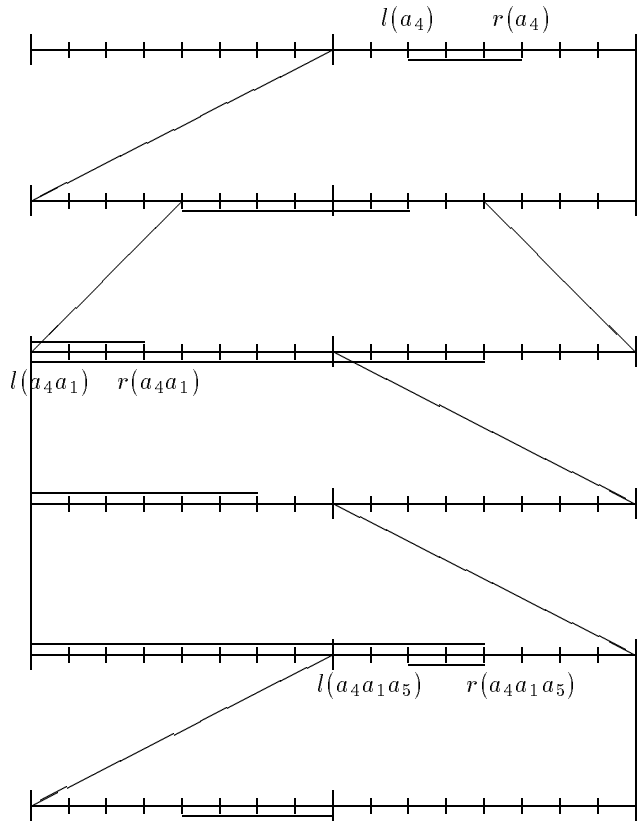


Рис. 5

Из процедуры арифметического кодирования непосредственно вытекает следующее замечание.

Замечание 3.1. *Полуинтервалы, соответствующие словам одинаковой длины, не пересекаются. Соответствующий слову x полуинтервал целиком содержится в полуинтервалах, соответствующих произвольному началу слова x.*

Рассмотрим пример арифметического кодирования ($t = 4$) слова $a_4a_1a_2$, порожденного источником Бернулли с алфавитом $A =$

$\{a_1, a_2, a_3, a_4, a_5\}$ и вероятностями появления букв $p(a_1) = p(a_3) = 1/4$, $p(a_2) = p(a_4) = p(a_5) = 1/6$.

Получаем равенства:

$l(a_4) = \frac{\lfloor 16 \cdot 2/3 \rfloor}{16} = \frac{10}{16}$, $r(a_4) = \frac{\lfloor 16 \cdot 5/6 \rfloor}{16} = \frac{13}{16}$,

$l(a_4a_1) = \frac{\lfloor 12 \cdot 0 \rfloor}{16} = 0$, $r(a_4a_1) = \frac{\lfloor 12/4 \rfloor}{16} = \frac{3}{16}$,

$l(a_4a_1a_5) = \frac{\lfloor 12 \cdot 5/6 \rfloor}{16} = \frac{10}{16}$, $r(a_4a_1a_5) = \frac{\lfloor 12 \cdot 1 \rfloor}{16} = \frac{12}{16}$.

Процесс построения арифметического кода изображен на рис. 5. Получаем $f(a_4a_1a_5) = 1010101$.

Чтобы декодировать арифметический код, нужно построить слово, имеющее заданный код. Начнем декодировать кодовое слово 1010101. Первый символ кодового слова 1, следовательно, первая буква закодированного слова a_3, a_4 или a_5. Следующий символ 0, поэтому первая буква закодированного слова a_3 или a_4. Так как третий символ 1, мы заключаем, что первая буква — a_4. Процесс декодирования изображен на рис. 6.

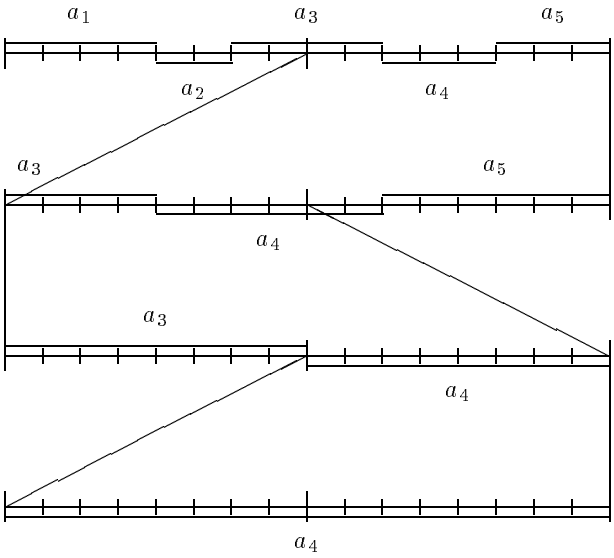


Рис. 6

После повторения процедуры кодирования символа a_4 (см. пример выше) разделим полученный полуинтервал на части, соответствующие

буквам a_1, a_2, a_3, a_4 , a_5, и определим вторую букву закодированного слова аналогичным образом.

3.6. Избыточность арифметического кодирования

Запишем формально рекуррентные формулы, определяющие процедуру арифметического кодирования. Пусть x — некоторое слово в алфавите A. Обозначим через xa_i слово, полученное из x добавлением буквы a_i. Пусть $d(x)$ — количество изменений масштаба при кодировании слова x; $L(x)$ и $R(x)$ — левая и правая граница полуинтервала, соответствующего последовательности x в первоначальном масштабе ($L(x) \neq l(x)!$). Тогда арифметическое кодирование определяется следующими формулами: $R(\emptyset) = 1$, $L(\emptyset) = 0$, $d(\emptyset) = 0$,

1) $I(x) = R(x) - L(x)$,

2) $I(xa_i) = (\lfloor \sigma_{i+1} I(x) 2^{d(x)+t} \rfloor - \lfloor \sigma_i I(x) 2^{d(x)+t} \rfloor)/2^{d(x)+t}$,

3) $1/4 < I(x)2^{d(x)} \leq 1$,

4) $L(xa_i) = L(x) + \lfloor \sigma_i I(x) 2^{d(x)+t} \rfloor / 2^{d(x)+t}$.

Арифметическим кодом слова является запись числа $\lfloor L(x) 2^{d(x)+2} \rfloor$, использующая ровно $d(x) + 2$ двоичных знаков, т. е.

5) $f(x) = B_{d(x)+2} \lfloor L(x) 2^{d(x)+2} \rfloor$.

Утверждение 3.6. *Пусть $A = \{a_1, \ldots, a_k\}$ — алфавит, и для всех $1 \leq i \leq k$ выполнены неравенства $1/2^{t-2} \leq p(a_i) \leq 1/4$. Тогда отображение f, определяемое формулами 1–5, является префиксным кодированием.*

Доказательство. Пусть x — произвольное слово в алфавите A. Из 3 получаем неравенство

$$L(x) + 2^{-d(x)-2} \leq L(x) + I(x) = R(x). \tag{3.10}$$

Пусть $a, b \in E^*$ и a — префикс b, тогда двоичные числа $u = 0, a$ и $v = 0, b$ удовлетворяют неравенству

$$u \leq v \leq u + 2^{-|a|}. \tag{3.11}$$

Докажем от противного, что отображение f обладает свойством префиксности. Пусть $x, y \in A^*$ — некоторые слова и $f(x)$ — префикс $f(y)$. Поскольку $|f(x)| = d(x) + 2$, то из (3.10) и (3.11) получаем

$$L(x) \leq L(y) \leq L(x) + 2^{-d(x)-2} \leq R(x),$$

т. е. полуинтервалы $[L(x), R(x))$ и $[L(y), R(y))$ пересекаются. Из замечания 3.1 вытекает, что слово x является префиксом слова y (противоположное противоречит тому, что $f(x)$ — префикс $f(y)$). Таким образом, если $x \neq y$ и $|x| = |y|$, то $f(x)$ — не префикс $f(y)$. Если $[L(x), R(x))$ и $[L(y), R(y))$ пересекаются, то из замечания 3.1 следует, что $[L(y), R(y)) \subset [L(x), R(x))$, но $d(x) < d(y)$, так как $p(a_i) \leq 1/4$. Таким образом, отображение f инъективно.

Теорема 3.2. *Пусть S — источник Бернулли с алфавитом $A = \{a_1, \ldots, a_k\}$. Пусть для всех i, $1 \leq i \leq k$ выполнены неравенства $1/2^{t-3} \leq p(a_i) \leq 1/4$* [10]*. Тогда*

$$R(f, S) \leq \frac{k - \sum_{i=1}^{k} \log p(a_i)}{2^{t-2}},$$

где f — арифметическое кодирование с параметром t.

Доказательство. Пусть $x = a_{i_1} \ldots a_{i_n}$. Введем обозначение $x_1^m = a_{i_1} \ldots a_{i_m}$ при $m \leq n$. Докажем методом индукции, что

$$I(x_1^n) > \prod_{j=1}^{n} (p(a_{i_j}) - 1/2^{t-2}). \tag{3.12}$$

Из равенства 2 имеем

$$I(a_{i_1}) = \frac{1}{2^t}(\lfloor \sigma_{i_1+1} 2^t \rfloor - \lfloor \sigma_{i_1} 2^t \rfloor) > p(a_{i_1}) - 1/2^t.$$

Пусть неравенство (3.12) верно для $n-1$, тогда из 2 и 3 получаем

$$I(x_1^n) = I(x_1^{n-1} a_{i_n}) = \frac{1}{2^{d(x_1^{n-1})+t}}(\lfloor I(x_1^{n-1}) \sigma_{i_n+1} 2^{d(x_1^{n-1})+t} \rfloor -$$

$$-\lfloor I(x_1^{n-1}) \sigma_{i_n} 2^{d(x_1^{n-1})+t} \rfloor) > I(x_1^{n-1}) \left(p(a_{i_n}) - \frac{1}{I(x_1^{n-1}) 2^{d(x_1^{n-1})+t}} \right) \geq$$

$$\geq I(x_1^{n-1})(p(a_{i_n}) - \frac{1}{2^{t-2}}),$$

[10] Ограничение $p(a_i) \leq 1/4$ несущественно для метода кодирования, но удобно при доказательстве теоремы.

т. е. неравенство (3.12) доказано. Из 3 и (3.12) следует, что

$$1 > 2^{d(x_1^n)} \prod_{j=1}^{n} (p(a_{i_j}) - 1/2^{t-2}),$$

откуда

$$d(x_1^n) < -\sum_{j=1}^{n} \log(p(a_{i_j}) - 1/2^{t-2}).$$

На пространстве событий A^n определим случайные величины $\xi_j : A^n \to R$, $1 \le j \le n$ равенствами

$$\xi_j(x_1^n) = -\log(p(a_{i_j}) - 1/2^{t-2}).$$

Поскольку S — источник Бернулли, то ξ_j независимы и одинаково распределены. Тогда из свойств математического ожидания получаем

$$\frac{1}{n} M d(x_1^n) < \frac{1}{n} M \sum_{j=1}^{n} \xi_j(x_1^n) = M\xi_1(x_1^n) = -\sum_{i=1}^{k} p(a_i) \log(p(a_i) - \frac{1}{2^{t-2}}) =$$

$$-\sum_{i=1}^{k} (p(a_i) - \frac{1}{2^{t-2}}) \log(p(a_i) - \frac{1}{2^{t-2}}) - \frac{1}{2^{t-2}} \sum_{i=1}^{k} \log(p(a_i) - \frac{1}{2^{t-2}}). \quad (3.13)$$

Функция $-x \log x$ возрастает при $0 < x < e^{-1}$. Из условия имеем $p(a_i) \le 1/4 < e^{-1}$, тогда

$$H(S) = -\sum_{i=1}^{k} p(a_i) \log p(a_i) > -\sum_{i=1}^{k} (p(a_i) - \frac{1}{2^{t-2}}) \log(p(a_i) - \frac{1}{2^{t-2}}). \quad (3.14)$$

Пусть f — арифметическое кодирование с параметром t; из формулы 5 имеем $|f(x_1^n)| = d(x_1^n) + 2$. Оценим избыточность кодирования f, используя формулы (3.13) и (3.14):

$$R(f, S) = \lim_{n \to \infty} \frac{1}{n} \sum_{x \in A^n} p(x)|f(x)| - H(S) =$$

$$= \lim_{n \to \infty} (\frac{1}{n} M d(x_1^n) + \frac{2}{n}) - H(S) < -\frac{1}{2^{t-2}} \sum_{i=1}^{k} \log(p(a_i) - \frac{1}{2^{t-2}}). \quad (3.15)$$

Из условия имеем $p(a_i) \ge 1/2^{t-3}$, тогда $p(a_i) - 1/2^{t-2} \ge p(a_i)/2$ и $-\log(p(a_i) - 1/2^{t-2}) \le 1 - \log p(a_i)$. Из (3.15) и последнего равенства следует заключение теоремы.

4. Универсальное кодирование

4.1. Оптимальное универсальное кодирование

В предыдущих главах было рассмотрено кодирование последовательностей, порожденных известным источником. Теперь нас будет интересовать более сложная задача — кодирование последовательностей, порожденных источником S, про который известно лишь то, что он принадлежит некоторому множеству $\mathcal{S}$. Будем считать, что все источники $S \in \mathcal{S}$ имеют одинаковый алфавит $A = \{a_1, \ldots, a_k\}$.

Избыточностью кодирования f на множестве источников $\mathcal{S}$ будем называть величину $R(f, \mathcal{S}) = \sup_{S \in \mathcal{S}} R(f, S)$.

Введем обозначение

$$R_n(f, \mathcal{S}) = \sup_{S \in \mathcal{S}} R_n(f, S) = \sup_{S \in \mathcal{S}} \left(\frac{1}{n}(M|f(x)|) - H(S) \right).$$

Тогда $R(f, \mathcal{S}) = \limsup_{n \to \infty} R_n(f, \mathcal{S})$.

Оптимальным универсальным на $\mathcal{S}$ кодированием будем называть префиксное кодирование f_0, если для всех целых $n > 0$ и произвольного префиксного кодирования f верно неравенство $R_n(f_0, \mathcal{S}) \le R_n(f, \mathcal{S})$.

В дальнейшем мы ограничимся рассмотрением в качестве $\mathcal{S}$ множества всех источников Бернулли с алфавитом A.

Теорема 4.1. *Пусть $\mathcal{S}$ — множество источников Бернулли с алфавитом $A = \{a_1, \ldots, a_k\}$. Тогда для оптимального универсального на $\mathcal{S}$ кодирования f_0 справедливо равенство*

$$R(f_0(x), \mathcal{S}) = \frac{(k-1)\log n}{2n} + O(\frac{1}{n})$$

при $n \to \infty$ [11].

Доказательство. Нам понадобится интеграл Дирихле, который нетрудно вычислить методом индукции по k. Пусть $r_i > -1$ и $1 \le i \le k$, тогда

$$\int_U \cdots \int \varrho_1^{r_1} \ldots \varrho_{k-1}^{r_{k-1}} (1 - \varrho_1 - \ldots - \varrho_{k-1})^{r_k} \, d\varrho = \frac{\Gamma(r_1 + 1) \ldots \Gamma(r_k + 1)}{\Gamma(\sum_{i=1}^{k} r_i + k)}, \tag{4.1}$$

[11] Теорема 4.1 доказана Р. Е. Кричевским в 1970 г. Аналогичная теорема для марковских источников доказана В. К. Трофимовым в 1974 г.

где $U = \{\varrho \in R^{k-1} : \varrho_i > 0; \sum_{i=1}^{k-1} \varrho_i < 1\}$. Здесь $\Gamma(r)$ — гамма-функция Эйлера, $\Gamma(r+1) = r!$, если r — натуральное число. Каждому источнику Бернулли $S \in \mathcal{S}$ взаимно-однозначно соответствует вектор $\varrho \in U$, где $\varrho_i = p_S(a_i)$ — вероятность порождения источником S буквы a_i. В дальнейшем через $p(x, \varrho)$, где через $x \in A^n$ и $\varrho \in U$ будем обозначать вероятность слова x, порожденного источником Бернулли с вероятностями появления букв $\varrho_1, \varrho_2, \ldots, \varrho_{k-1}, 1 - \sum_{i=1}^{k-1} \varrho_i$. Тогда

$$p(x, \varrho) = \varrho_1^{r_1} \ldots \varrho_{k-1}^{r_{k-1}} (1 - \varrho_1 - \ldots - \varrho_{k-1})^{r_k}, \tag{4.2}$$

где r_i — количество вхождений буквы a_i в слово x. Обозначим через $R_n(f, \varrho)$ избыточность кодирования f на источнике Бернулли с распределением вероятностей ϱ, а через $H(\varrho)$ — энтропию этого источника. Тогда

$$R_n(f, \varrho) = \frac{1}{n} \sum_{x \in A^n} p(x, \varrho) |f(x)| - H(\varrho). \tag{4.3}$$

Определим среднюю избыточность кодирования f на множестве источников $\mathcal{S}$ следующим образом:

$$\bar{R}_n(f, \mathcal{S}) = \frac{1}{\mu(U)} \int_U \ldots \int R_n(f, \varrho)\, d\varrho, \quad \text{где} \quad \mu(U) = \int_U \ldots \int 1\, d\varrho.$$

Из монотонности операции интегрирования следуют неравенства

$$R_n(f, \mathcal{S}) = \sup_{\varrho \in U} R_n(f, \varrho) = \frac{1}{\mu(U)} \int_U \ldots \int \sup_{\varrho \in U} R_n(f, \varrho)\, d\varrho \geq$$

$$\geq \frac{1}{\mu(U)} \int_U \ldots \int R_n(f, \varrho)\, d\varrho \geq \frac{1}{\mu(U)} \int_U \ldots \int \inf_{\varrho \in U} R_n(f, \varrho)\, d\varrho = \inf_{\varrho \in U} R_n(f, \varrho).$$

Т. е. для произвольного префиксного кода f справедливы неравенства

$$R_n(f, \mathcal{S}) \geq \bar{R}_n(f, S) \geq \inf_{S \in \mathcal{S}} R_n(f, S). \tag{4.4}$$

Из (4.3) имеем

$$n\bar{R}_n(f, \mathcal{S}) = \frac{1}{\mu(U)} \int_U \ldots \int \Big(\sum_{x \in A^n} p(x, \varrho) |f(x)| - nH(\varrho) \Big)\, d\varrho =$$

$$= \sum_{x \in A^n} |f(x)| \left(\frac{1}{\mu(U)} \int_U \ldots \int p(x, \varrho)\, d\varrho \right) - \frac{n}{\mu(U)} \int_U \ldots \int H(\varrho)\, d\varrho. \tag{4.5}$$

Введем обозначение

$$\hat{p}(x) = \frac{1}{\mu(U)} \int_U \dots \int p(x, \varrho)\, d\varrho, \tag{4.6}$$

тогда

$$\sum_{x \in A^n} \hat{p}(x) = \frac{1}{\mu(U)} \int_U \dots \int \sum_{x \in A^n} p(x, \varrho)\, d\varrho = 1.$$

Таким образом, величины $\hat{p}(x)$ можно считать вероятностями порождения букв x некоторым источником Бернулли $\hat{S}$ с алфавитом A^n. Введем обозначение

$$C(n) = H(\hat{S}) - \frac{n}{\mu(U)} \int_U \dots \int H(\varrho)\, d\varrho,$$

тогда из равенства (4.5) получаем

$$n\bar{R}_n(f, \mathcal{S}) = \sum_{x \in A^n} |f(x)| \hat{p}(x) - H(\hat{S}) + C(n). \tag{4.7}$$

Из теоремы 2.2 известно, что

$$\sum_{x \in A^n} |f(x)| \hat{p}(x) - H(\hat{S}) \geq 0.$$

С другой стороны, из утверждения 2.3 для кодирования Шеннона $\hat{f}$ источника $\hat{S}$ имеем

$$|\hat{f}(x)| = \lceil -\log \hat{p}(x) \rceil, \tag{4.8}$$

$$\sum_{x \in A^n} |\hat{f}(x)| \hat{p}(x) - H(\hat{S}) \leq 1.$$

Тогда из (4.7) получаем неравенства

$$\bar{R}_n(f, \mathcal{S}) \geq C(n)/n \quad \text{и} \quad \bar{R}_n(\hat{f}, \mathcal{S}) \leq (1 + C(n))/n, \tag{4.9}$$

где f — произвольное префиксное кодирование, а $\hat{f}$ — кодирование Шеннона.

Оценим величину $\inf_{S \in \mathcal{S}} R_n(\hat{f}, \mathcal{S})$. Из формулы Дирихле (4.1) при $r_i = 0$, $1 \leq i \leq k$ получаем равенство

$$\int_U \dots \int 1\, d\varrho = \frac{1}{(k-1)!}.$$

Из формул (4.1) и (4.2) имеем

$$\int_U \dots \int p(x,\varrho)\,d\varrho = \frac{r_1!r_2!\dots r_k!}{(\sum_{i=1}^k r_i + k - 1)!},$$

где r_i — число вхождений буквы a_i в слово $x \in A^n$. Из формул (4.6) и (4.8) получаем

$$|\hat{f}(x)| = \lceil -\log \hat{p}(x) \rceil \geq \log\left(\frac{\int_U \dots \int 1\,d\varrho}{\int_U \dots \int p(x,\varrho)\,d\varrho}\right) = \log\left(\frac{(n+k-1)!}{r_1!r_2!\dots r_k!(k-1)!}\right).$$

Из утверждения 1.5 следует, что

$$\log\left(\frac{(n+k-1)!}{r_1!r_2!\dots r_k!(k-1)!}\right) = \log\left(\frac{n!}{r_1!r_2!\dots r_k!}\right) +$$

$$+ \log\left(\frac{(n+k-1)!}{(k-1)!n!}\right) \geq nF(x) + \frac{1}{2}\sum_{i=1}^k \log\frac{n}{r_i} - \frac{k-1}{2}\log n +$$

$$+ (k-1)\log\frac{n}{k-1} + c \geq nF(x) + \frac{k-1}{2}\log n + c',$$

где c, c' — некоторые постоянные [12]. Тогда из утверждения 1.4 и последнего неравенства для произвольного $S \in \mathcal{S}$ при $n \to \infty$ имеем

$$R_n(\hat{f}, S) = \frac{1}{n}\sum_{x \in A^n} |\hat{f}(x)|p(x) - H(S) \geq \frac{k-1}{2n}\log n + \frac{c'}{n} +$$

$$+ \sum_{x \in A^n} F(x)p(x) - H(S) \geq \frac{k-1}{2n}\log n + \frac{c'}{n},$$

т. е. $\inf_{S \in \mathcal{S}} R_n(\hat{f}, \mathcal{S}) \geq \frac{k-1}{2n}\log n + O(1/n)$. Пусть f_0 — оптимальное универсальное на $\mathcal{S}$ кодирование, тогда из формул (4.4), (4.9) и последнего неравенства получаем

$$R_n(f_0, \mathcal{S}) \geq \bar{R}_n(f_0, \mathcal{S}) \geq \frac{C(n)}{n} \geq \bar{R}_n(\hat{f}, \mathcal{S}) - \frac{1}{n} \geq$$

$$\geq \inf_{S \in \mathcal{S}} R_n(\hat{f}, \mathcal{S}) - \frac{1}{n} \geq \frac{k-1}{2n}\log n + O(1/n).$$

[12] Как и в утверждении 1.5, $r_i = 0$ в знаменателе следует считать единицей.

Докажем теперь, что существует такое префиксное кодирование f_1, что $R_n(f_1, S) \le \frac{k-1}{2n} \log n + O(1/n)$. Сначала покажем, что найдутся такие постоянная C и префиксное кодирование f_1, что

$$|f_1(x)| = nF(x) + \frac{k-1}{2} \log n + C. \tag{4.10}$$

По теореме 2.1 для этого достаточно проверить справедливость неравенства Крафта

$$\sum_{x \in A^n} 2^{-|f_1(x)|} \le 1. \tag{4.11}$$

Пусть $r = (r_1 \dots r_k)$, $r_i \ge 0$ — целые и $\sum_{i=1}^k r_i = n$. Обозначим через $G(r)$ множество слов $x \in A^n$, содержащих по r_i букв a_i. Тогда из утверждения 1.5 имеем равенство

$$\log |G(r)| = \log \left(\frac{n!}{r_1! \dots r_k!} \right) = nF(x) + \frac{1}{2} \sum_{i=1}^k \log \frac{n}{r_i} - \frac{k-1}{2} \log n + C'(r),$$

где $x \in G(r)$ и $C'(r) \le C'$, $C' > 0$— постоянная. Кроме того, $F(x)$, и, следовательно, $|f_1(x)|$ одинаковы для всех $x \in G(x)$. Тогда из (4.10) получаем равенства

$$\sum_{x \in A^n} 2^{-|f_1(x)|} = \sum_{r_1 + \dots + r_k = n, r_i \ge 0} |G(r)| 2^{-|f_1(x)|} =$$

$$= \sum_{r_1 + \dots + r_k = n, r_i \ge 0} 2^{\frac{1}{2} \sum_{i=1}^k \log(n/r_i) - (k-1) \log n + C'(r) - C} =$$

$$= \sum_{r_1 + \dots + r_k = n, r_i \ge 0} 2^{C'(r) - C} \frac{1}{n^{k-1}} \sqrt{\frac{n}{r_1} \frac{n}{r_2} \dots \frac{n}{r_k}}. \tag{4.12}$$

Функция $g(\varrho_1 \varrho_2 \dots \varrho_{k-1}) = \frac{1}{\sqrt{\varrho_1 \varrho_2 \dots \varrho_{k-1} (1 - \varrho_1 - \dots - \varrho_{k-1})}}$ монотонна по каждой из переменных в окрестности 0, кроме того, из (4.1) следует, что несобственный интеграл Римана функции $g(\varrho_1 \varrho_2 \dots \varrho_{k-1})$ сходится в области $U = \{\varrho \in R^{k-1} : \varrho_i > 0; \sum_{i=1}^{k-1} \varrho_i < 1\}$. Поэтому суммы Римана (4.12) функции $g(\varrho_1 \varrho_2 \dots \varrho_{k-1})$ сходятся к интегралу от этой функции по области U, т. е.

$$\sum_{r_1 + \dots + r_k = n, r_i \ge 0} \frac{1}{n^{k-1}} \sqrt{\frac{n}{r_1} \frac{n}{r_2} \dots \frac{n}{r_k}} =$$

$$= \sum_{\frac{r_1}{n}+\ldots+\frac{r_{k-1}}{n}\le 1, r_i\ge 0} \frac{1}{n^{k-1}} \frac{1}{\sqrt{\frac{r_1}{n}\frac{r_2}{n}\ldots\frac{r_{k-1}}{n}(1-\frac{r_1}{n}-\ldots-\frac{r_{k-1}}{n})}} =$$

$$= \int_U \ldots \int \frac{1}{\sqrt{\varrho_1\varrho_2\ldots\varrho_{k-1}(1-\varrho_1-\ldots-\varrho_{k-1})}}\, d\varrho + \alpha(n) = \frac{(\sqrt{\pi})^k}{\Gamma(k/2)} + \alpha(n),$$

где $\alpha(n) \to 0$. Тогда, выбирая $C \ge 1 + C'(r) + \log\left(\frac{(\sqrt{\pi})^k}{\Gamma(k/2)}\right)$, получаем неравенство

$$\lim_{n\to\infty} \sum_{x\in A^n} 2^{-|f_1(x)|} \le \frac{1}{2},$$

т. е. кодирование f_1 удовлетворяет неравенству Крафта (4.11) при $n \to \infty$. Из (4.10) и утверждения 1.4 для произвольного источника $S \in \mathcal{S}$ получаем неравенство

$$R_n(f_1, S) \le \frac{k-1}{2n} \log n + O(1/n).$$

Поскольку f_0 — оптимальное универсальное на $\mathcal{S}$ кодирование, то $R_n(f_0, S) \le R_n(f_1, S)$ и теорема доказана.

Заметим, что для каждого источника Бернулли S найдется такое n-блочное кодирование f_S, что $R_n(f_S, S) \le \frac{1}{n}$ (теорема 3.1). Таким образом, оптимальное универсальное на $\mathcal{S}$ кодирование f_0 не является оптимальным для каждого источника $S \in \mathcal{S}$ в отдельности несмотря на то что $R(f_0, \mathcal{S}) = 0$.

4.2. Кодирование Бабкина–Фитингофа

В теореме 4.1 установлена избыточность оптимального универсального кодирования, но не предложено способа построения этого кодирования. Ниже будет описан предложенный Б. М. Фитингофом в 1966 г. способ построения универсального кодирования, близкого к оптимальному. В. Ф. Бабкин в 1971 г. предложил обладающий полиномиальной трудоемкостью алгоритм вычисления кодовых слов.

Рассмотрим алфавит $A = \{a_0, a_1\}$, состоящий из двух букв. Пусть $x \in A^n$; через $r(x)$ обозначим количество букв a_1 в слове x. Пусть $S(n, r) = \{x \in A^n, r(x) = r\}$ — множество слов длины n, содержащих ровно по r букв a_1. В п. 2.3 предложена нумерация множества $S(n, r)$, т. е. взаимнооднозначное отображение $L(x) : S(n, r) \to [1 \ldots C_n^r]$. Для каждого $x \in A^n$ n-блочный код Бабкина–Фитингофа $f_n(x)$ определяется равенством

$$f_n(x) = B_{d_1}(r) B_{d_2}(L(x)),$$

где $d_1 = \lfloor \log n \rfloor + 1$ и $d_2 = \lfloor \log C_n^r \rfloor + 1$, $B_d(n)$ — двоичная запись числа n, использующая ровно d символов.

Утверждение 4.1. *Пусть $\mathcal{S}$ — множество источников Бернулли с алфавитом $A = \{a_0, a_1\}$ и f_n — кодирование Бабкина-Фитингофа, тогда*

$$R_n(f, \mathcal{S}) \le \frac{C + \log n}{n},$$

где C — некоторая постоянная.

Доказательство. Из определения кодирования f получаем

$$|f_n(x)| \le \log C_n^{r(x)} + \log n + 2.$$

Из утверждения 1.5 следует, что $F(x) \ge \frac{1}{n} \log C_n^{r(x)}$; тогда для произвольного источника $S \in \mathcal{S}$ из двух последних неравенств имеем

$$R_n(f_n, S) = \frac{1}{n} \sum_{x \in A^n} p(x)|f(x)| - H(S) \le \sum_{x \in A^n} p(x)F(x) + \frac{\log n + 2}{n} - H(S) \tag{4.13}$$

В утверждении 1.4 для произвольного источника Бернулли S доказано равенство $|\sum_{x \in A^n} p(x)F(x) - H(S)| = O(1/n)$. Таким образом, утверждение следует из последнего равенства и (4.13).

4.3. Интервальное кодирование

В этом пункте мы рассмотрим универсальное кодирование, которое называется интервальным. Оно отличается простотой реализации и оказывается эффективным при кодировании источников с большим алфавитом или с часто повторяющимися сериями одинаковых букв. В интервальном кодировании каждая буква исходной последовательности заменяется на число, равное количеству букв до предыдущего включения той же буквы. Например, слово

$$(a_1a_2a_3)a_3a_3a_3a_2a_2a_2a_1a_1a_1a_3 \tag{4.14}$$

будет преобразовано в последовательность чисел (...)0004008006. Известны две модификации этого метода, позволяющие уменьшить стоимость кодирования. Первая из них была предложена Б. Я. Рябко в 1980 г. и названа им методом "стопки книг". Метод "стопки книг" отличается от интервального кодирования тем, что вместо числа всех букв

между двумя одинаковыми указывается число различных букв между ними. Так, слово (4.14) будет преобразовано в (...)0001002002. Другая модификация (IFC) заключается в том, что теперь буква в исходном слове заменяется числом букв с большими номерами, разделяющими текущее и предыдущее включение буквы. Например, слово (4.14) будет преобразовано в три последовательности, соответствующие трём различным буквам: a_3 : (...)0000, a_2 : (...)400, a_1 : (...)800. Декодирование нужно начинать с первой буквы алфавита, оставляя для других букв соответствующее количество пустых мест.

Для кодирования последовательности чисел, которая получается из исходной последовательности после замены букв числами, можно использовать произвольный префиксный код натуральных чисел. В частности, можно применить код Элайеса (см. п. 2.2). Тогда интервальный код слова $x = a_{i_1} \dots a_{i_n}$ определяется равенством

$$f(x) = El(k_1)El(k_2) \dots El(k_n),$$

где $k_j = \min\limits_{l<j, i_j=i_l} (j - l - 1)$ — количество букв между a_{i_j} и ближайшей такой же буквой.

Для устранения неопределенности при кодированиии первого появления буквы в слове можно, например, в начало каждого слова добавлять список букв алфавита как в (4.14).

Теорема 4.2. *Пусть $\mathcal{S}$ — множество источников Бернулли с алфавитом $A = \{a_1, \dots, a_k\}$, а f — интервальное кодирование* [13]. *Тогда*

$$R(f, \mathcal{S}) \le 2 \log\log k + 3.$$

Доказательство. Из утверждения 2.1 при $n > 1$ следует неравенство

$$|El(n)| \le \log n + 2 \log\log n + 3$$

и $|El(0)| = |El(1)| = 2$. Пусть $x \in A^n$ и буква a_i встречается в слове x r_i раз на $t_1, t_2, \dots, t_{r_i}$ местах. Тогда количество битов, затраченное на кодирование всех, за исключением первого, вхождений буквы a_i, не превышает

$$\sum_{j=1}^{r_i - 1} (\log(t_{j+1} - t_j) + 2\overline{\log\log}(t_{j+1} - t_j) + 3),$$

[13] Теорема 4.2 доказана Б. Я. Рябко в 1980 г.

где $\overline{\log\log}1 = 0$ и $\overline{\log\log}n = \log\log n$ при $n > 1$. Тогда из (4.14) и неравенства Йенсена для выпуклых вверх функций $\log x$ и $\log\log x$ получаем, что количество битов, затраченных на кодирование всего слова x, за исключением первых вхождений каждой буквы не превышает

$$\sum_{i=1}^{k} r_i \sum_{j=1}^{r_i-1} \frac{1}{r_i}(\log(t_{j+1}-t_j) + 2\overline{\log\log}(t_{j+1}-t_j)) + 3n \leq$$

$$3n + n\sum_{i=1}^{k} \frac{r_i}{n} \log \frac{n}{r_i} + 2n \sum_{i=1}^{k} \frac{r_i}{n} \log\log \frac{n}{r_i} \leq n(3 + 2\log\log k) + n \sum_{i=1}^{k} \frac{r_i}{n} \log \frac{n}{r_i}.$$

Тогда из предыдущего неравенства получаем

$$\frac{1}{n}|f(x)| - F(x) \leq 2\log\log k + 3 + \frac{C'(k)}{n},$$

где $C'(k)$ — стоимость кодирования первых вхождений k букв алфавита A. Из предыдущего неравенства и утверждения 1.4 для произвольного $S \in \mathcal{S}$ имеем

$$R_n(f,S) = C_n(f,S) - H(S) \leq \frac{1}{n} \sum_{x \in A^n} p(x)|f(x)| - \sum_{x \in A^n} p(x)F(x) + \frac{k-1}{n \ln 2} \leq$$

$$2\log\log k + 3 + \frac{C'(k) + (k-1)\log e}{n}.$$

Тогда $R(f, \mathcal{S}) = \limsup_{n\to\infty} R_n(f, \mathcal{S}) \leq 2\log\log k + 3$. Теорема доказана.

4.4. Схема кодирования Лемпела–Зива

В 1977 и 1978 г. А. Лемпелом и Я. Зивом были предложены два метода универсального кодирования. В настоящее время известно множество модификаций этих алгоритмов. Ниже будут описаны два исходных метода и модификация, предложенная Т. А. Велчем в 1984 г.

Пусть $A = \{a_1, \ldots, a_k\}$ — некоторый алфавит и $x \in A^n$. Через x_l^r обозначим слово, состоящее из букв слова $x = a_{i_1} \ldots a_{i_n}$, начиная с l-ой и заканчивая r-ой, т. е. $x_l^r = a_{i_l} \ldots a_{i_r}$. Схема кодирования, предложенная А. Лемпелом и Я. Зивом в 1977 г. (в дальнейшем именуемая LZ77) состоит в разделении кодируемого слова $x_1^n \in A^n$ на подслова σ_i, $i = 1 \ldots m$ по следующему правилу. Пусть начало слова x_1^n уже разделено на подслова, то есть представляет собой конкатенацию подслов $\sigma_1\sigma_2 \ldots \sigma_i$ и $x_1^n = \sigma_1 \ldots \sigma_i x_{l_i}^n$. Выберем следующее подслово $\sigma_{i+1} = x_{l_i}^{r_i}$

как наиболее длинное начало остатка $x_{l_i}^n$, которое уже встречалась в $x_1^{r_i-1}$, то есть

$$\sigma_{i+1} = x_{l_i}^{r_i} = x_{n_i}^{n_i+r_i-l_i},$$

где $n_i < l_i$. Кодом каждого подслова σ_{i+1} будет пара чисел $(n_i, r_i - l_i + 1)$. Например, слово $(a_1a_2)a_2a_1a_2a_1a_1a_2a_1a_2a_1a_2$ разделяется на подслова $a_2, a_1a_2, a_1, a_1a_2a_1, a_2a_1a_2$ и кодируется последовательностью пар чисел $(2,1),(1,2),(1,1),(4,3),(3,3)$. Первое число в каждой паре целесообразно записывать в двоичном виде с использованием ровно $\lfloor \log l_i \rfloor + 1$ битов, второе можно кодировать произвольным префиксным кодом чисел натурального ряда.

Схема кодирования, предложенная А. Лемпелом и Я. Зивом в 1978 году (в дальнейшем именуемая LZ78) отличается от описанной выше тем, что на каждом шаге выбирается наиболее длинное начало остатка $x_{l_i}^n$, которое совпадает с некоторым уже выделенным подсловом σ_j, $j < i$, и к нему добавляется еще одна буква, то есть $\sigma_{i+1} = \sigma_j a_{p_i}$. Кодом подслова σ_{i+1} будет пара чисел (j, p_i). Например, слово $a_2a_1a_2a_1a_1a_2a_1a_2a_1$ разделяется на подслова $a_2, a_1, a_2a_1, a_1a_2, a_1a_2a_1$ и кодируется последовательностью пар чисел $(0,2),(0,1),(1,1),(2,2),(4,1)$.

Кодирование f, построенное по схеме LZ78, определим как последовательность пар чисел (j_i, p_i), причем первое число пары записано в двоичном виде с использованием $d_i = \lfloor \log i \rfloor + 1$ битов, а второе — $d = \lfloor \log k \rfloor + 1$ битов. Т. е. если $x = \sigma_1 \dots \sigma_m$, то кодирование, построенное по схеме LZ78, определяется равенством

$$f(x) = B_{d_1}(j_1)B_d(p_1)B_{d_2}(j_2)B_d(p_2)\dots B_{d_m}(j_m)B_d(p_m).$$

Тогда

$$|f(x)| \le \sum_{i=1}^{m}(\log i + \log k + 2) \le m(\log m + \log k + 2). \qquad (4.15)$$

Модификация Велча — LZW отличается от LZ78 тем, что на каждом шаге выбирается σ_{i+1} так, чтобы $\sigma_{i+1} = \sigma_j a_{p_i}$, $j < i$, где a_{p_i} — первая буква подслова σ_{j+1}, то есть a_{p_i} непосредственно следует за σ_j в слове x_1^n . Кодом подслова σ_{i+1} будет число j. Например, слово $(a_2)(a_1)a_2a_1a_2a_1a_1a_2a_1a_2a_1$ разделяется на подслова $a_2a_1, a_2a_1, a_1a_2, a_1a_2a_1$ и кодируется последовательностью чисел $1,1,2,5$.

Оценим избыточность кодирования по схеме LZ78.

Теорема 4.3. *Пусть $\mathcal{S}$ — множество источников Бернулли с алфавитом $A = \{a_1, \ldots, a_k\}$ и f — кодирование, построенное по схеме LZ78* [14]. *Тогда $R(f, \mathcal{S}) = 0$.*

Доказательство. Оценим эмпирическую энтропию слова $x \in A^n$. Пусть $a_0 \neq a_i$, где $i = 1 \ldots k$. Рассмотрим слово $\hat{x} = \hat{\sigma}_1 \ldots \hat{\sigma}_m$, где $\hat{\sigma}_i = \sigma_i a_0$. Слово $\hat{x}$ отличается от x тем, что подслова σ_i разделены в нем буквами a_0, которые играют роль запятой. Рассмотрим множество $T = \{\tau : [1 \ldots m] \to [1 \ldots m] - \text{взаимнооднозначно}\}$ перестановок длины m. Все слова $y(\tau) = \hat{\sigma}_{\tau(1)} \ldots \hat{\sigma}_{\tau(m)}$ различны, поскольку в соответствии со схемой LZ78 все слова σ_i получаются различными и не содержат добавочной буквы a_0.

Количество различных перестановок $y(\tau)$ не превосходит числа всевозможных различных перестановок букв в слове $\hat{x}$, то есть

$$m! \leq \frac{(n + r_0)!}{r_0! r_1! r_2! \ldots r_k!},$$

где r_i — количество включений буквы a_i, $i = 0 \ldots k$, в слове $\hat{x}$. Учитывая, что $r_0 = m$ и количество включений буквы a_i, $i = 1 \ldots k$, в словах $\hat{x}$ и x совпадает, из предыдущего неравенства и утверждения 1.5 имеем

$$F(x) \geq \frac{1}{n} \log \frac{n!}{r_1! r_2! \ldots r_k!} \geq \frac{1}{n} \log \frac{n!(m!)^2}{(m+n)!}. \tag{4.16}$$

Из формулы Стирлинга следует, что

$$\log \frac{n!(m!)^2}{(m+n)!} \geq m \log m - m \log \frac{n}{m} - C'm,$$

где $C' > 0$ — некоторая постоянная. Тогда из (4.15), (4.16) и предыдущего неравенства имеем

$$\frac{1}{n}|f(x)| - F(x) \leq \frac{m}{n} \log \frac{n}{m} + (C' + 2 + \log k) \frac{m}{n}.$$

Для произвольного источника $S \in \mathcal{S}$ из последнего неравенства и утверждения 1.4 получаем неравенства

$$R_n(f, S) = C_n(f, S) - H(S) \leq \frac{1}{n} \sum_{x \in A^n} p(x)|f(x)| -$$

[14] Теорема 4.3 верна также для схем кодирования LZ77, LZW и марковских источников.

$$-\sum_{x\in A^n} p(x)F(x) + \frac{k-1}{n\ln 2} \le \frac{m}{n}\log\frac{n}{m} + C\frac{m}{n},$$

где $C > 0$ — некоторая постоянная.

Поскольку длины подслов $|\sigma_i| \to \infty$ при $i \to \infty$, то $\frac{m}{n} \to 0$ при $n \to \infty$. Тогда из последнего неравенства заключаем, что $R_n(f, S) \to 0$ при $n \to \infty$. Поскольку источник $S \in \mathcal{S}$ произвольный, то $R(f, \mathcal{S}) = 0$. Теорема доказана.

5. Передача сообщений по каналам связи, допускающим ошибки

5.1. Канал связи и его пропускная способность

В этой главе мы рассмотрим проблему передачи сообщений, т. е. слов, порожденных некоторым источником, по допускающему ошибки каналу связи. Наиболее простым способом борьбы с ошибками при передаче сообщений является многократное повторение переданной буквы. Например, вместо 01101 можно передавать 000111111000111. Такой метод неэффективен, поскольку в несколько раз увеличивает длину сообщения. Нашей задачей является построение кодирования, с одной стороны, уменьшающего возможность искажения сообщения при передаче, а с другой стороны, не слишком удлиняющего сообщение. Эффективные методы кодирования будут предложены ниже, а вначале мы рассмотрим математическую модель искажения сообщений.

Дискретным каналом (связи) L называется пара из алфавита $A = \{a_1, \ldots, a_k\}$ на входе канала и алфавита $B = \{b_1, \ldots, b_m\}$ на выходе канала с набором вероятностей $p(y|x)$ получения сообщения $y \in B^n$ при передаче сообщения $x \in A^n$.

Если сообщения на входе канала порождаются некоторым источником S_A с алфавитом A, то сообщения на выходе канала можно считать порожденными источником $S_B = L(S_A)$, где вероятность порождения источником S_B сообщения y определяется по формуле: $p(y) = \sum_{x\in A^*} p(y|x)p(x)$.

Дискретный канал называется *каналом без памяти*, если

$$p(b_{i_1}\ldots b_{i_n}|a_{j_1}\ldots a_{j_n}) = \prod_{t=1}^{n} p(b_{i_t}|a_{j_t})$$

для произвольных слов $b_{i_1}\ldots b_{i_n}$ и $a_{j_1}\ldots a_{j_n}$.

Канал без памяти полностью определяется алфавитами A на входе, B на выходе канала и матрицей P переходных вероятностей $p_{ij} = p(b_i|a_j)$. Если $m = k$ и P — единичная матрица, то канал не искажает сообщения. Заметим, что если L — канал без памяти и S_A — источник Бернулли, то и $S_B = L(S_A)$ является источником Бернулли.

Пропускной способностью канала без памяти L называется величина

$$c(L) = \sup I(S_A, S_B),$$

где sup берется по всем источникам Бернулли S_A с алфавитом A, а $S_B = L(S_A)$.

В дальнейшем мы ограничимся рассмотрением каналов без памяти с алфавитами из двух букв на входе и выходе канала. Двоичный канал без памяти называется *симметричным*, если матрица P переходных вероятностей симметрична.

Утверждение 5.1. *Пусть L — симметричный канал без памяти с алфавитами $A = \{a_0, a_1\}$ на входе и $B = \{b_0, b_1\}$ на выходе канала. Пусть $p(b_0|a_0) = p(b_1|a_1) = p$, тогда $c(L) = 1 + p\log p + (1-p)\log(1-p)$.*

Доказательство. Рассмотрим произвольный источник Бернулли S_A с алфавитом $A = \{a_0, a_1\}$ и $S_B = L(S_A)$. Пусть $p(a_1) = q$ и $p(a_0) = 1-q$, тогда

$$p(b_1) = p(a_1)p(b_1|a_1) + p(a_0)p(b_1|a_0) = pq + (1-p)(1-q),$$

$$p(b_0) = 1 - p(b_1) = p + q - 2pq.$$

Из определения условной энтропии имеем

$$H(S_B|a_0) = H(S_B|a_1) = -p\log p - (1-p)\log(1-p)$$

и

$$H(S_B|S_A) = p(a_0)H(S_B|a_0) + p(a_1)H(S_B|a_1) = -p\log p - (1-p)\log(1-p). \tag{5.1}$$

По определению информации имеем $I(S_A, S_B) = H(S_B) - H(S_B|S_A)$. А из утверждения 1.1 следует, что $\max H(S_B) = 1$ достигается при $p(b_0) = p(b_1) = 1/2$. Тогда из (5.1) и определения информации получаем

$$c(L) = \max I(S_A, S_B) = 1 + p\log p + (1-p)\log(1-p).$$

Утверждение доказано.

5.2. Теорема кодирования Шеннона

Пусть S — стационарный источник с алфавитом $A = \{a_0, a_1\}$, а L — канал без памяти с алфавитами $A = \{a_0, a_1\}$ на входе и $B = \{b_0, b_1\}$ на выходе канала. Ошибкой передачи будем называть получение буквы b_1 при передаче буквы a_0 и получение b_0 при передаче буквы a_1. Обозначим через $q(L, S)$ среднюю вероятность ошибки при передаче по каналу L сообщений источника S:

$$q(L, S) = p(a_0)p(b_1|a_0) + p(a_1)p(b_0|a_1). \tag{5.2}$$

Замечание 5.1. *Без ограничения общности можно считать, что* $q(L, S) \leq 1/2$.

В противном случае можно поменять местами буквы b_0 и b_1, после чего средняя вероятность ошибки передачи станет равной $1 - q(L, S)$.

Теорема 5.1. *(Обращение теоремы кодирования)* [15]. *Пусть L — канал без памяти с алфавитами $A = \{a_0, a_1\}$ на входе и $B = \{b_0, b_1\}$ на выходе канала и $1 \geq c_1 > c(L)$. Тогда найдется $\varepsilon > 0$ такое, что для каждого стационарного источника S_A с алфавитом A и энтропией $H(S_A) \geq c_1$ справедливо неравенство $q(L, S_A) > \varepsilon$.*

Доказательство. Пусть S'_A — источник Бернулли с алфавитом A, порождающий буквы a_0 и a_1 с теми же вероятностями, что и источник S_A. Из формулы (5.2) видно, что $q(L, S'_A) = q(L, S_A)$. Введем обозначения $p_1 = p(a_0|b_1)$ и $p_0 = p(a_1|b_0)$. Тогда

$$q(L, S_A) = p(a_0)p(b_1|a_0) + p(a_1)p(b_0|a_1) =$$

$$= p(a_0b_1) + p(a_1b_0) = p(b_1)p_1 + p(b_0)p_0.$$

Пусть $h(x) = -x \log x - (1 - x) \log(1 - x)$. Из определения условной энтропии имеем

$$H(S'_A|b_0) = -p_0 \log p_0 - (1 - p_0) \log(1 - p_0) = h(p_0),$$

$$H(S'_A|b_1) = -p_1 \log p_1 - (1 - p_1) \log(1 - p_1) = h(p_1),$$

[15] Теорема 5.1 была доказана Р. М. Фано в 1952 г. Аналогичная теорема справедлива для источников с конечным алфавитом и для широкого класса дискретных каналов (см. [3]).

$$H(S'_A|S_B) = p(b_0)h(p_0) + p(b_1)h(p_1),$$

где $S_B = L(S'_A)$.

Функция $h(x)$ выпукла вверх на интервале $(0,1)$, так как $h''(x) = -\frac{1}{x\ln 2} + \frac{1}{(x-1)\ln 2} < 0$. Тогда из неравенства Йенсена следует, что

$$H(S'_A|S_B) \le h(p(b_0)p_0 + p(b_1)p_1) = h(q(L,S_A)).$$

Из определения пропускной способности канала и следствия 1.1 получим неравенства

$$c(L) \ge I(S'_A, S_B) = H(S'_A) - H(S'_A|S_B) \ge H(S_A) - h(q(L,S_A)).$$

Тогда

$$h(q(L,S_A)) \ge c_1 - c(L),$$

причем $0 < c_1 - c(L) \le 1$ из условия теоремы и $0 \le q(L,S_A) \le 1/2$ из замечания 5.1. Функция $h(x)$ монотонно возрастает на интервале $(0,1/2)$ и принимает всевозможные значения в интервале $(0,1)$. Тогда для обратной функции $h^{-1}(y)$ получаем неравенство $q(L,S_A) \ge h^{-1}(c_1 - c(L)) = \varepsilon > 0$. Теорема доказана.

Будем называть равноблочным кодированием инъективное отображение $f : D^m \to E^n$. Пусть S_D — некоторый стационарный источник, порождающий слова $x \in D^*$. Множество $f(x) \in E^*$ двоичных слов с вероятностями $p(f(x)|f(y)) = p(x|y)$ определяет стационарный источник $\hat{S} = f(S_D)$ с алфавитом E. Поскольку f — инъективная функция, то $f(x) \in E^n$ можно однозначно определить по $x \in D^m$ и наоборот. Тогда справедливо равенство

$$H(S_D^1 \ldots S_D^m|\hat{S}^1 \ldots \hat{S}^n) = H(\hat{S}^1 \ldots \hat{S}^n|S_D^1 \ldots S_D^m) = 0.$$

Из утверждения 1.2 и последнего равенства получаем, что $H(\hat{S}^1 \ldots \hat{S}^n) = H(S_D^1 \ldots S_D^m)$ и $H(\hat{S}^1 \ldots \hat{S}^{kn}) = H(S_D^1 \ldots S_D^{km})$ для целых $k > 0$. Тогда из определения энтропии источника и утверждения 3.2 имеем

$$H(S_D) = \lim_{k\to\infty} \frac{1}{km} H(S_D^1 \ldots S_D^{km}) = \lim_{k\to\infty} \frac{n}{kmn} H(\hat{S}^1 \ldots \hat{S}^{kn}) = \frac{n}{m} H(\hat{S}),$$

т. е.

$$H(S_D) = C(f,S)H(\hat{S}).^{16} \qquad (5.3)$$

[16] Формула (5.3) аналогична формуле 3 из утверждения 3.3 и справедлива для произвольного блочного кодирования.

Предположим, что мы кодируем равноблочным кодированием f сообщения, порожденные стационарным источником S, и передаем кодовые слова по каналу без памяти L. Тогда из теоремы 5.1 и равенства (5.3) получаем следствие.

Следствие 5.1. *Пусть $1 \geq c_1 > c(L)$, тогда найдется такое $\varepsilon > 0$, что для каждого источника S, энтропия которого удовлетворяет неравенству $C(f,S) \leq H(S)/c_1$, средняя вероятность ошибки при передаче кодового символа не меньше ε.*

Однако ошибки при передаче кодовых символов еще не означают невозможности правильного декодирования сообщений. Например, если равноблочное кодирование $f : E \to E^3$ просто утраивает все символы ($f(0) = 000, f(1) = 111$), то одиночные ошибки при передаче кодовых слов могут быть исправлены. Рассмотрим произвольное равноблочное кодирование $f : D^m \to E^n$. Обозначим через $F(x) \subset E^n$ множество всех двоичных слов длины n, которые декодируются как x. Средняя вероятность $q(f,L,S)$ ошибки, которая будет допущена при передаче по каналу L слов, порожденных источником S и закодированных кодом f, определяется равенством

$$q(f,L,S) = \sum_{x \in D^m} p(x) \left(\sum_{y \notin F(x)} p(y|f(x)) \right).$$

Следующая теорема утверждает, что возможна надежная передача сообщений источника S по допускающему ошибки каналу L, если сообщения источника S предварительно специальным образом закодированы.

Теорема 5.2. *(Теорема кодирования Шеннона)* [17]. *Пусть L — канал без памяти, $0 < c_1 < c(L)$, S — стационарный источник. Тогда для произвольного $\varepsilon > 0$ найдется равноблочное кодирование f такое, что $C(f,S) \leq H(S)/c_1$ и $q(f,L,S) < \varepsilon$.*

Подробное доказательство этой теоремы можно найти, например, в [3].

[17] Теорема 5.2 была сформулирована К. Шенноном в 1948 г., первое строгое доказательство этой теоремы было дано А. Файнстейном в 1954 г.

5.3. Коды, исправляющие ошибки

Теорема кодирования Шеннона доказывает существование кодирования, обеспечивающего надежную передачу сообщений, но не предлагает конструктивного способа построения такого кодирования. Поскольку сообщения, порождаемые источником S с произвольным алфавитом, можно закодировать двоичными словами, то достаточно ограничиться рассмотрением равноблочного кодирования вида $f : E^m \to E^n$, где $m \leq n$. Такое кодирование называют (n, m)-кодированием [18]. Коды, обеспечивающие надежную передачу сообщений по каналу связи, называют *кодами, исправляющими ошибки* . Известны коды, исправляющие ошибки различного типа, но мы ограничимся рассмотрением (n, m)-кодов, исправляющих одиночные или кратные ошибки в кодовых словах.

Говорят, что (n, m)-кодирование f исправляет l ошибок, если существует алгоритм, позволяющий определить переданное кодовое слово $f(x) \in E^n$, когда в нем не более l двоичных знаков изменены на противоположные.

Наиболее простыми примерами кодов, исправляющих ошибки, являются коды с проверкой на четность. Рассмотрим $(u_1, u_2, u_3) \in E^3$; определим числа $u_4, u_5, u_6 \in E$ так, чтобы суммы $s_1 = u_1 + u_2 + u_4$, $s_2 = u_1 + u_3 + u_5$ и $s_3 = u_2 + u_3 + u_6$ были четными. Убедимся, что кодирование $f : E^3 \to E^6$, сопоставляющее тройкам (u_1, u_2, u_3) шестерки $(u_1, u_2, u_3, u_4, u_5, u_6)$ по указанному правилу, исправляет одиночные ошибки. Действительно, если произойдет ошибка в символе u_1, то суммы s_1 и s_2 станут нечетными, если в символе u_2 — то s_1 и s_3 станут нечетными, в u_3 — s_2 и s_3, в u_4 — s_1, в u_5 — s_2, в u_6 — s_3. Т. е. все одиночные ошибки приводят к разным изменениям четности сумм s_1, s_2 и s_3 и, значит, могут быть определены и исправлены. Ошибку сразу в двух символах уже не удается исправить. Например, ошибку в символах u_4 и u_5 нельзя отличить от ошибки в u_1. В арифметике по модулю 2 кодирование f записывается проще: $u_4 = u_1 + u_2$, $u_5 = u_1 + u_3$ и $u_6 = u_2 + u_3$. В дальнейшем подразумевается, что все операции с элементами E^n проводятся по модулю 2.

Вообще (n, m)-*кодирование с проверкой на четность* определяется системой равенств $v = f(u)$:

$$v_i = u_i \quad \text{при} \quad 1 \leq i \leq m,$$

$$v_{i+m} = b_{i1}u_1 + b_{i2}u_2 + \ldots + b_{im}u_m \tag{5.4}$$

[18] (n, m)-кодом обычно называют множество кодовых слов.

при $1 \le i \le n-m$ и $b_{ij} \in E$. Т. е. каждое кодовое слово $v \in f(E^m)$ удовлетворяет системе уравнений [19]

$$b_{i1}v_1 + b_{i2}v_2 + \ldots + b_{im}v_m + v_{i+m} = 0,$$

где $1 \le i \le n-m$ или $Bv = 0$, где B — матрица размерности $(n-m) \times n$,

$$B = \begin{pmatrix} b_{11} & b_{12} & \ldots & b_{1m} & 1 & 0 & \ldots & 0 \\ b_{21} & b_{22} & \ldots & b_{2m} & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ b_{n-m1} & b_{22} & \ldots & b_{n-mm} & 0 & 0 & \ldots & 1 \end{pmatrix}.$$

Матрица B называется *проверочной матрицей* кодирования f, первые m символов кода — $v_1 \ldots v_m$ называются — *информационными*, а остальные $n-m$ символов — $v_{m+1} \ldots v_n$ — называются *проверочными*.

Пусть слово $v' \in E^n$ отличается от кодового слова $v \in E^n$ i-ым двоичным знаком, тогда $v' + v = e_i$, где e_i — вектор из нулей и единицы на i-ом месте. Из линейности операции умножения матриц получаем равенство

$$Bv' = Bv' + Bv = Be_i = b_i,$$

где b_i — i-ый столбец матрицы B. Если все столбцы b_i, $i = 1 \ldots n$ матрицы B различны между собой и отличны от нулевого столбца, то мы всегда можем определить, в каком символе кодового слова допущена ошибка, и исправить ее.

Утверждение 5.2. *Если $n - m \ge \log(n+1)$, то найдется (n,m)-код, исправляющий одиночные ошибки.*

Доказательство. Из сказаного выше ясно, что (n,m)-код f, определенный системой уравнений (5.4), исправляет одиночные ошибки, если все столбцы его проверочной матрицы различные и ненулевые. Таким образом, достаточно доказать существование матрицы размерности $(n-m) \times n$, все столбцы которой ненулевые и различные. Очевидно, $|E^{n-m}| = 2^{n-m}$, т. е. имеется $2^{n-m} - 1$ различных ненулевых столбцов длины $n-m$. Поскольку неравенство $2^{n-m} - 1 \ge n$ эквивалентно неравенству $n - m \ge \log(n+1)$, то утверждение доказано.

В качестве столбца b_i проверочной матрицы B удобно выбрать $B_{n-m}(i)$ — двоичную запись числа i, использующую ровно $n-m$ битов.

[19] Операции + и - в арифметике по модулю 2 совпадают.

Если v — кодовое слово и $v'+v = e_i$, то, умножая матрицу B на столбец v', сразу получаем номер позиции, в которой произошла ошибка:

$$Bv' = B(v' + v) = Be_i = B_{n-m}(i).$$

При таком определении матрицы B проверочные и информационные символы расположены в беспорядке, но это не усложняет кодирование. Например, проверочной для $(7, 4)$-кода является матрица

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

1-й, 2-й и 4-й символы — проверочные, а остальные — информационные. Например, кодом слова 1011 является слово $v = \underline{0}\underline{1}1\underline{0}011$ (проверочные символы подчеркнуты). Пусть получено сообщение $v' = 0010011$, тогда $Bv' = 010$, т. е. ошибка во втором символе.

Коды с проверкой на четность, проверочная матрица которых содержит все возможные ненулевые столбцы (как в примере выше), называются кодами Хэмминга.

5.4. Границы Хэмминга и Варшамова–Гильберта

В этом пункте мы выясним, при каких значениях m, n и l могут существовать (n, m)-коды, исправляющие l ошибок.

Пусть $v, u \in E^n$, тогда *расстоянием Хэмминга* между v и u называется величина $d(u, v)$, равная числу позиций, в которых слова u и v отличаются. Заметим, что кодирование, исправляющее l ошибок, декодирует слово $v \in E^n$ так же, как и кодовое слово $u \in E^n$, если $d(u, v) \le l$.

Теорема 5.3. *(граница Хэмминга). Для каждого (n, m)-кода, исправляющего l ошибок, справедливо неравенство*

$$n - m \ge \log(\sum_{i=0}^{l} C_n^i).$$

Доказательство. Пусть f — некоторое (n, m)-кодирование и $x \in E^m$. Обозначим через $F(x) \subset E^n$ множество слов, которые декодируются как x. Обозначим через $F^i(x) \subset E^n$ множество слов длины n, отличающихся от слова $f(x)$ в i позициях, т. е. $F^i(x) = \{y \in E^n : d(f(x), y) = i\}$. Поскольку f исправляет l ошибок, то $F^i(x) \subset F(x)$ при $0 \le i \le l$. Кроме

того, $F^i(x) \cap F^j(x) = \emptyset$ при $i \neq j$ и $|F^i(x)| = C_n^i$. Тогда

$$|F(x)| \geq \sum_{i=0}^{l} |F^i(x)| = \sum_{i=0}^{l} C_n^i. \tag{5.5}$$

Ясно, что если $x \neq x'$, то $F(x) \cap F(x') = \emptyset$, и из неравенства (5.5) получаем

$$2^n \geq \sum_{x \in E^m} |F(x)| \geq 2^m \sum_{i=0}^{l} C_n^i.$$

Отсюда следует искомое неравенство

$$n - m \geq \log(\sum_{i=0}^{l} C_n^i).$$

Теорема доказана.

Из теоремы 5.3 и утверждения 5.2 следует, что (n, m)-код, исправляющий одиночные ошибки, существует тогда и только тогда, когда $n - m \geq \log(1 + n)$.

Теорема 5.4. *(граница Варшамова–Гильберта)* [20]*. Если*

$$n - m \geq \log(\sum_{i=0}^{2l-1} C_n^i),$$

то найдется (n, m)-код, исправляющий l ошибок.

Доказательство. Докажем существование (n, m)-кода с проверкой на четность, исправляющего l ошибок. Для этого достаточно построить матрицу B размерности $(n-m) \times n$, обладающую следующим свойством: если $u \in E^n$ — кодовое слово и $v \neq v'$ таковы, что $d(u, v) \leq l$ и $d(u, v') \leq l$, то $Bv \neq Bv'$. Это позволит определить произвольные $l' \leq l$ ошибок в произвольном кодовом слове.

Ясно, что $Bu' = \sum_{i=1}^{l'} b_{i_j}$, где b_{i_j} — столбцы матрицы B и i_j — номера позиций, в которых слово u' и кодовое слово u отличаются. Следовательно достаточно построить множество $U_n \subset E^{n-m}$ столбцов матрицы B, удовлетворяющее свойствам:

[20] Теорема 5.4 доказана Р. Р. Варшамовым в 1957 г.

1) $|U_n| = n$;
2) $b_i \neq 0$, $b_i \neq b_j$ при $i \neq j$, где $b_i \in U_n$;
3) $\sum_{j=1}^{l_1} b_{i_j} \neq \sum_{k=1}^{l_2} b_{i_k}$, где $l_1, l_2 \leq l$, $b_i \in U_n$ и наборы индексов $\{i_j\}$ и $\{i_k\}$ различны.

Пусть $U \subset E^n$, обозначим через $L(U)$ множество всевозможных сумм из не более чем $2l-1$ элементов множества U, т. е.

$$L(U) = \{\sum_{k=1}^{l'} b_{i_k} : b_{i_k} \in U, \quad 0 \leq l' \leq 2l-1\}.$$

Тогда

$$|L(U)| \leq \sum_{i=0}^{2l-1} C_{|U|}^i. \tag{5.6}$$

Будем строить множества U_k, $|U_k| = k$, $k \leq n$, обладающие свойствами 2 и 3, по индукции. Положим $U_1 = \{e_1\}$. Пусть построено множество U_k, $k < n$. Тогда из неравенств (5.6) и $2^{n-m} \geq \sum_{i=0}^{2l-1} C_n^i$ и из условия теоремы получаем неравенство

$$|L(U_k)| \leq \sum_{i=0}^{2l-1} C_k^i \leq \sum_{i=0}^{2l-1} C_n^i \leq 2^{n-m}.$$

Таким образом, найдется вектор $b_{k+1} \in E^{n-m} \backslash L(U_k)$. Очевидно, множество $U_{k+1} = U_k \cup \{b_{k+1}\}$ удовлетворяет условию 2. Из определения $L(U_k)$ следует, что $b_{k+1} \neq \sum_{k=1}^{l'} b_{i_k}$ при $0 \leq l' \leq 2l-1$. Тогда

$$b_{k+1} + \sum_{j=1}^{l_1} b_{i_j} \neq \sum_{k=1}^{l_2} b_{i_k},$$

где $l_1 \leq l-1$ и $l_2 \leq l$, т. е. множество U_{k+1} удовлетворяет условию 3. При $k = n-1$ $U_{k+1} = U_n$ — искомое множество. Теорема доказана.

Задачи

1. Пусть A и B — разбиения; доказать, что:
   а) если A и B — независимы, то $H(A|B) = H(A)$ и $I(A,B) = 0$;
   б) если $A \preceq B$, то $H(B) \geq H(A)$, $H(A|B) = 0$ и $I(A,B) = H(A)$.
2. Пусть A, B и C — разбиения; доказать утверждения:
   а) $H(AB|C) = H(A|C) + H(B|AC)$;
   б) $H(BA|CA) \leq H(B|C)$;
   в) $I(BC,A) + I(B,C) = I(AC,B) + I(A,C)$;
   г) из $I(A,BC) = I(A,B)$ следует, что $I(C,AB) = I(C,B)$;
   д) если A и B независимы, то $I(AB,C) \geq I(A,C) + I(B,C)$.
3. Пусть A, B, C, D — разбиения, причем AD и BC независимы, C и D независимы, тогда $H(AB|CD) = H(A|D) + H(B|C)$.
4. Пусть $A = \{A_1, A_2\}$ и $B = \{B_1, B_2\}$ — разбиения, вычислить $I(A,B)$ если
   а) $p(A_1B_1) = 1/8, p(A_1B_2) = 1/16, p(A_2B_1) = 3/16, p(A_2B_2) = 10/16$;
   б) $p(B_1) = 2/3, p(B_2) = 1/3, p(A_1|B_1) = 1/4, p(A_1|B_2) = 3/4,$
   $p(A_2|B_1) = 3/4, p(A_2|B_2) = 1/4$;
   в) $p(B_1) = 2/3, p(B_2) = 1/3, p(A_1|B_1) = 1/2, p(A_1|B_2) = 1,$
   $p(A_2|B_1) = 1/2, p(A_2|B_2) = 0$.
5. Найти коды El, Lev, St_2 и St_3 чисел
   а) 7, б) 12, в) 74, г) 69, е) 134.
6. Найти число x, если
   а)$El(x) = 001010001$; б)$El(x) = 0011000101$;
   в)$Lev(x) = 111100001101$; г)$Lev(x) = 11101110$;
   д)$St_2(x) = 101100111$; е)$St_2(x) = 110001000000100$;
   ж)$St_3(x) = 1001010011$; з)$St_3(x) = 1010100001110$.
7. Найти набор натуральных чисел $x_1, x_2, \ldots$, если
   а) $El(x) = 01100001011001001000000010100100110$;
   б) $Lev(x) = 11100001111000010011110100011101$.
8. Построить код Шеннона и вычислить стоимость кодирования для источника Бернулли с вероятностями букв
   а) 0,4; 0,2; 0,2; 0,1; 0,1;
   б) 0,3; 0,2; 0,15; 0,15; 0,1; 0,1 .
9. Построить коды Шеннона–Фано и Хаффмана и вычислить стоимости кодирования для источника Бернулли с вероятностями букв
   а) 0,4; 0,1; 0,1; 0,1; 0,1; 0,1; 0,1; б) 0,3; 0,3; 0,15; 0,15; 0,1;
   в) 0,3; 0,2; 0,2; 0,15; 0,15; г) 0,4; 0,25; 0,1; 0,1; 0,1; 0,05 .

10. Построить код Гильберта–Мура и вычислить стоимость кодирования для источника Бернулли с вероятностями букв (буквы упорядочены по возрастанию номеров)

а) 0,1; 0,4; 0,2; 0,1; 0,2;

б) 0,1; 0,25; 0,3; 0,25; 0,1.

11. Построить коды Шеннона и Хаффмана и вычислить избыточность кодирования для источника Бернулли с вероятностями букв

а) 1/4; 1/4; 1/8; 1/8; 1/16; 1/16; 1/16; 1/16;

б) 7/16; 5/16; 3/16; 1/16.

12. Доказать, что для кодов Хаффмана и Шеннона–Фано неравенство Крафта превращается в равенство.

13. Привести пример источника Бернулли, имеющего коды Хаффмана с различными наборами длин кодовых слов.

14. Вычислить энтропию марковского источника первого порядка, если задана матрица P переходных вероятностей ($p_{ij} = p(a_i|a_j)$):

а) $P = \begin{pmatrix} 1/2 & 1 & 1/4 \\ 1/4 & 0 & 0 \\ 1/4 & 0 & 3/4 \end{pmatrix}$; б) $P = \begin{pmatrix} 2/3 & 0 & 1/2 \\ 1/6 & 2/3 & 1/2 \\ 1/6 & 1/3 & 0 \end{pmatrix}$;

в) $P = \begin{pmatrix} 1/3 & 1/6 & 1/2 \\ 2/3 & 1/3 & 0 \\ 0 & 1/2 & 1/2 \end{pmatrix}$; г) $P = \begin{pmatrix} 1/2 & 1/3 & 1/2 \\ 1/4 & 1/3 & 1/2 \\ 1/4 & 1/3 & 0 \end{pmatrix}$.

15. Построить блочный код Хаффмана с блоками длиной 3 и вычислить его избыточность для источников Бернулли с вероятностями букв

а) $p(a_1) = 8/9$, $p(a_2) = 1/9$;

б) $p(a_1) = 6/7$, $p(a_2) = 1/7$.

16. Построить блочные коды Хаффмана с блоками длиной 2 для источников из задачи 14, вычислить стоимость кодирования.

17. Построить блочный код Хаффмана с блоками длиной 3 для марковского источника первого порядка с матрицей переходных вероятностей $P = \begin{pmatrix} 1/3 & 3/4 \\ 2/3 & 1/4 \end{pmatrix}$. Вычислить избыточность кодирования.

18. Построить кодирование Ходака с длиной кодового слова $m = 2$ и вычислить стоимость кодирования для источника Бернулли с вероятностями букв

а) $p(a_1) = 3/10$, $p(a_2) = 7/10$;

б) $p(a_1) = 2/5$, $p(a_2) = 3/5$.

19. Построить кодирование Ходака с длиной кодового слова $m = 3$ и

вычислить стоимость кодирования для источника Бернулли с вероятностями букв

а) $p(a_1) = 1/3$, $p(a_2) = 2/3$;

б) $p(a_1) = 1/4$, $p(a_2) = 3/4$;

в) $p(a_1) = 1/8$, $p(a_2) = 7/8$.

20. Найти арифметический код ($t = 4$) слова x, порожденного источником Бернулли, с вероятностями букв

а) $p(a_1) = p(a_3) = 1/4, p(a_2) = p(a_4) = p(a_5) = 1/6$, $x = a_5a_4a_4$;

б) $p(a_1) = p(a_3) = 1/4, p(a_2) = p(a_4) = p(a_5) = 1/6$, $x = a_3a_2a_1$;

в) $p(a_1) = p(a_2) = 1/6, p(a_3) = p(a_4) = 1/3$, $x = a_1a_2a_4a_4$;

г) $p(a_1) = p(a_2) = 1/6, p(a_3) = p(a_4) = 1/3$, $x = a_3a_1a_1a_4$.

21. Определить слово, порожденное источником из задачи $20, а)$, если арифметический код ($t = 4$) слова — 01010001.

22. Определить слово, порожденное источником из задачи $20, в)$, если арифметический код ($t = 4$) слова — 11100101.

23. Закодировать (в виде последовательности целых чисел) интервальным кодированием, кодированием "стопка книг" и кодированием IFC слово

а) (ABCD)AABCADABBC; б) (ABCD)BDAAACCACDB;

в) (ABCD)BACACCDADA; г) (ABCD)DADCCACCABD.

24. Декодировать закодированное "стопкой книг" слово в алфавите из четырех букв:

а) 0102231313; б) 1320332000.

25. С помощью схемы кодирования LZ77 представить в виде последовательности пар целых чисел слово

а) (01)00110010110; б) (01)11010101010101.

26. С помощью схемы кодирования LZ78 представить в виде последовательности пар целых чисел слово

а) 00110001011; б) 11010101010.

27. С помощью схемы кодирования LZW представить в виде последовательности целых чисел слово

а) (0,0,1,1)010110011; б) (1,1,0,0)111010100.

28. Найти пропускную способность канала без памяти с матрицей P переходных вероятностей

а) $P = \begin{pmatrix} 1 & 1/3 \\ 0 & 2/3 \end{pmatrix}$; б) $P = \begin{pmatrix} 1-p-q & q \\ p & p \\ q & 1-p-q \end{pmatrix}$;

в) $P = \begin{pmatrix} 1-p & p & 0 \\ 0 & 1-p & p \\ p & 0 & 1-p \end{pmatrix}$.

29. Пусть (7,4)-код имеет проверочную матрицу

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

При условии, что при передаче было допущено не более одной ошибки, определить кодовое слово, если получено

а) 1101011; б) 1011010;

в) 1001011; г) 0100011.

30. Построить проверочную матрицу (6,3)-кода, исправляющего одиночные ошибки.

31. Построить проверочную матрицу (9,5)-кода, исправляющего одиночные ошибки.

32. Построить проверочную матрицу (8,2)-кода, исправляющего двойные ошибки.

33. Построить проверочную матрицу (10,3)-кода, исправляющего двойные ошибки.

34. Доказать, что для сколь угодно малого $\varepsilon > 0$ найдется код f, исправляющий l ошибок, со стоимостью кодирования не превышающей $1 + \varepsilon$.

Содержание