TRANSPORTA UN SAKARU INSTITŪTS

Методические указания к выполнению контрольной работы по курсу «Теория передачи и кодирования информации»

Составитель А. М. Скальский

Рига
2010

УДК 681.3

Transporta un sakaru institūts

Институт транспорта и связи

Методические указания к выполнению контрольной работы по курсу «Теория передачи и кодирования информации». Издание второе. Составитель А. М. Скальский. – Рига: ИТС, 2010. – 36 с.

Методические указания предназначены для студентов 3-го курса факультета электроники и компьютерных наук, обучающихся по программам «электроника» и «телекоммуникационные системы» всех форм обучения.

Оглавление

Требования к оформлению контрольной работы

Контрольная работа посвящена эффективному (энтропийному, статистическому, экономному) кодированию источников дискретных сообщений. Она выполняется как индивидуальное задание по одному из вариантов, указанному преподавателем.

В процессе выполнения контрольной работы должны быть выполнены все пункты задания, перечисленные в разделе «Требования к содержанию отчета».

Решение перечисленных пунктов задания необходимо выполнять в той последовательности, в которой они приведены в методических указаниях. Рекомендуется использовать стандартные обозначения, приведенные в Приложении 3.

Приступая к выполнению задания, предварительно следует проработать соответствующие разделы лекционного курса и практических занятий по данной теме.

Результаты контрольной работы представляются в виде пояснительной записки, которая должна содержать:

- титульный лист с указанием варианта задания;
- оглавление работы;
- вариант задания;
- ответы и расчетные соотношения по всем пунктам контрольной работы, включая расчетные промежуточные соотношения, итоговые таблицы а также необходимые графические материалы;
- список использованной литературы.

Используемые условные обозначения параметров должны быть расшифрованы. Полученным расчетным численным значениям характеристик кодов необходимо давать физическое толкование и приводить размерности величин.

Пояснительная записка оформляется с применением редактора Word.

Параметры страницы:

- размер бумаги Paper - A4;
- поле Top - 2 см;
- поле Button - 2 см;
- поле Left - 2 см;
- поле Right - 2 см;
- Header - 1,2 см;
- Fotter - 1,2 см.

Параметры шрифта:

- используемый шрифт – Times New Roman;
- начертание – Normal;
- размер шрифта – 12.

Параметры абзаца:

- выравнивание - Justify;
- красная строка – отступ 1,2 см;
- междустрочное расстояние – Single.

Все рисунки должны располагаться по центру. Номер и название рисунка располагаются под рисунком шрифтом 12 размера (Italic).

Рис. 1. *Название рисунка*

Номер и название таблиц всегда располагаются перед таблицей. Выравнивание налево. Размер шрифта 12. Оставляется одна пустая строка перед и после названия таблицы, а также после самой таблицы.

Таблица 1. Название таблицы

Параметр	**Значение**

Схемы, рисунки и таблицы должны иметь нумерацию по разделам пояснительной записки. В тексте пояснительной записки должны быть ссылки на рисунки и таблицы.

Ссылки на литературу в тексте оформляются в квадратные скобки, например [1]. Литература нумеруется в порядке упоминания ее в тексте.

Примеры оформления литературы:

Книга: автор, название (in Italics), местоположение издательства, издательство, год.

Статья: автор, название статьи, журнал (in Italics), выпуск и номер выпуска, год, страницы.

Электронные источники информации: автор, название статьи (in Italics), Интернет-ссылка.

Варианты заданий

№	***Сообщение источника***	*N*	*L*
1.	Архип осип и сопит, Осип охрип и хрипит	39	10
2.	поп Прокоп от вопросов отроков оторопел	39	12
3.	сколько колоковал столько переколоковал	39	12
4.	кол около колокола, колокол около кола	38	6
5.	а у Прокопа косоворотка то коротковата	38	11
6.	Турка курит трубку, курка клюет крупку	38	12
7.	автор аморального романа Роман новатор	38	12
8.	Перекресток перед переездом переполнен	38	14
9.	одного доброго слова довода достаточно	38	14
10.	Говорил про полковника и подполковника	38	12
11.	скороговорун скороговорил скороговорки	38	11
12.	принцип инициирования мимики или тика	37	14
13.	косой косит косой осоку под косогором	37	12
14.	там карета таракана катала как ракета	37	9
15.	сокол тот, кто самолетом смело летает	37	10
16.	кукушка купила кукушонку шапку-ушанку	37	11
17.	снова продано переносное оборудование	37	13
18.	скороговорки как караси на сковородке	37	13
19.	На горе Арарат рвала Варвара виноград	37	14
20.	У перепела и перепелки пять перепелят	37	12
21.	Протокол протоколом запротоколировали	37	12
22.	около катка трасса мотокросса коротка	37	9
23.	около ворот Прокоп прокопал пол окопа	37	10
24.	Нет ни критики ни риторики у истории	36	10
25.	Тот кроток с которого спрос короток	36	8
26.	Ткет ткач Тане ткани как чеканит чек	36	9
27.	золото не зло но много от него злого	36	9
28.	борона на боронованном новом огороде	36	11
29.	Иван в авангарде, да Варвара не рада	36	11
30.	перепел с перепелами перед перелетом	36	12
31.	Хохлатки-хохотушки хохотом хохотали	35	11
32.	жук жужжит еж лежит иже жить желают	35	11
33.	У сорока сороконожек ножек не сорок	35	10
34.	стойкий старик как критик и сатирик	35	9
35.	Теперь ревет тетерев перед деревней	35	10
36.	Наказание за заказ заказано законом	35	9
37.	При формировании или информировании	35	11
38.	они сами стали символами инициативы	35	12
39.	полководец полон планов по половцам	35	12
40.	в статистике ни риторики ни критики	35	11
41.	виток противостояния против истории	35	11
42.	Только он хоть кроток но и не робок	35	12

43.	Надо, надо дрова водворить до двора	35	11
44.	водовоз вез воды воз до водораздела	35	11
45.	уж с ужаса стал уже, ужу стало хуже	35	11
46.	набросок собкора бросок но короток	34	9
47.	молоковоз вез молоко около вокзала	34	9
48.	накануне аксакал скакал на скакуне	34	8
49.	они не исповедники и не посредники	34	11
50.	ваша слава лаваш, а наша шмат сала	34	10
51.	Сами заказали сказки как присказки	34	10
52.	разворот около ворот словно оборот	34	12
53.	до поры у норы, а в пору то в нору	34	12
54.	Стилистики стали считать и листики	34	10
55.	кукушка укутала кукушонка кушаком	33	10
56.	только он хоть кроток но не робок	33	11
57.	идеи электротехники и электроники	33	12
58.	ранние нравственные раны не равны	33	10
59.	Копна под попом, поп под колпаком	33	11
60.	Петер теперь то первее, то правее	33	11
61.	слава и сила слова стали на века	32	11
62.	толком толковать как ломом ломать	33	9
63.	если имеет место лесть или месть	32	9
64.	Копна под попом, поп под попоной	32	9
65.	скороговорок не перескороговориш	32	12
66.	пепелище не греет но еще теплеет	32	11
67.	около окна мокло огромное мочало	32	11
68.	колокол надо только подколоковать	32	11
69.	Водоросли росли около водоворота	32	12
70.	Саша по ошибке шапкой шишку сшиб	32	11
71.	волокли колокол не по колоковски	32	11
72.	Голодному да молодому долг долог	32	10
73.	около ворот колокол и коловорот	31	8
74.	у села ли села лиса или у леса?	31	8
75.	мороз заморозил розарий и озеро	31	10
76.	критик и историк Никита Никитин	31	10
77.	дровокол да дроворуб около дров	31	10
78.	водовоз воду вез от водопровода	31	11
79.	На дворе трава а на траве дрова	31	9
80.	около околотка кого-то колотили	31	9
81.	святее всех святых всего света	30	11
82.	ценны страницы истории станицы	30	11
83.	аппаратура авиационного ангара	30	12
84.	соло снова словно основа слова	30	7
85.	Колотил Клим молотом и клинком	30	8
86.	около колодца осколок колокола	30	8

87.	жужжит жужелица, жужжит у лица	30	10
88.	слово спонсора подобно доводам	30	12
89.	гора под орлом, орел под пером	30	11
90.	баран бараном, да рога - даром	30	10
91.	Клара украла у карла кораллы	28	8
92.	поп на копне, колпак на попе	28	9
93.	силой слова словно сон сняло	28	10
94.	хорошо то чего хорошо хочешь	28	10
95.	ну и инаугурации будут у гуру	29	11
96.	кот которого дорого продали	27	11
97.	“барабан барабанил на бал.’’	27	10
98.	Волки волоком волокли вола.	27	8
99.	Около околотка лоток и ток.	27	8
100.	наша Маша и Миша нашли шаль	27	8
101.	Королева Клара карала Карла	27	9
102.	слово сошло словно сто снов	27	8
103.	аппаратура огромного ангара	27	10
104.	Эта задача как удача за час	27	10
105.	палатка в заплатках за лат.	27	10
106.	Тот робот работал плоховато	27	10
107.	честь честью но есть нечего	27	10
108.	Дровокол говорил про волков	27	10
109.	Около ворот коловорот - вот	27	8
110.	Жутко жужжит жук около жижи	27	8
111.	Добр бобр до бобрят и ребят	27	9
112.	Колотил Клим молотом клинок	27	9
113.	в сокоскороварке сок кокоса	27	8
114.	в воде росли одни водоросли	27	9
115.	хорош шорох да горох дорог	27	8
116.	волки волоком волокли вола.	27	9
117.	перепела пели и еле летели	26	8
118.	Сорок сорок это сорок морок	26	9
119.	около окна огромное облако	26	11
120.	новое боковое окно огромно	26	10
121.	барабан барабанил на бал.	25	8
122.	ужа ужица ужалила на ужин	25	8
123.	коси коса споро пока роса	25	8
124.	только тот, кто кроток	22	8
125.	баран барабанил на бал	22	7
126.	козодой с косой козой	21	7
127.	У ворот сорок коровок	21	8
128.	до одного нового года	21	7
129.	хорошо то, что хорошо	21	8
130.	Варвара варила варево	21	8

Требования к содержанию отчета:

1. Структурная схема системы передачи дискретных сообщений и назначение ее элементов.
2. Модель заданного источника сообщений.
3. Информационные характеристики источника:
   - ✓ собственную информацию символов сообщения $I(a_i)$;
   - ✓ количество информации в заданном сообщении $I(A)$;
   - ✓ энтропия источника $H(A)$;
   - ✓ максимально возможная энтропия $H_{max}(A)$;
   - ✓ производительность источника $H'(A)$;
   - ✓ избыточность источника ρ_u .
4. Пропускная способность канала (при длительности битового интервала T_0 =1 ms) $C_к$.
5. Характеристики равновероятного двоичного кода источника:
   - ✓ длина кодового слова n_c ;
   - ✓ длина строчки записи сообщения $l_{стр}$;
   - ✓ энтропия кодера $H_c(B)$;
   - ✓ избыточность равномерного кода ρ_c ;
   - ✓ вероятности появления символов 0 и 1 в кодовых словах;
   - ✓ производительность кодера $H_c'(B)$;
   - ✓ эффективность использования канала $\eta_к$ и коэффициент сжатия кодовой строчки записи сообщения (информации);
6. Реализация алгоритма кодирования Шеннона-Фано для заданного источника.
7. Кодовое дерево Шеннона-Фано для заданного источника.
8. Характеристики полученного кода Шеннона–Фано:
   - ✓ средняя длина кодовых слов $\overline{n}_c$;
   - ✓ теоретически минимальная средняя длина кодовых слов $\overline{n}_c$;
   - ✓ длина строчки записи сообщения $l_{стр}$;
   - ✓ избыточность кода ρ_c ;
   - ✓ вероятности появления символов 0 и 1 в кодовых словах;
   - ✓ энтропия кодера $H_c(B)$;
   - ✓ производительность кодера $H_c'(B)$;
   - ✓ эффективность использования канала $\eta_к$ и коэффициент сжатия кодовой строчки записи сообщения (информации).
9. Реализация алгоритма Хаффмана для заданного источника и характеристики кода, аналогичные характеристикам кода Шеннона-Фано
10. Преобразование полученного кода Хаффмана в канонический код
11. Сравнительный анализ полученных результатов кодирования источника:
    - ✓ по энтропии кодеров;
    - ✓ по скорости передачи информации;
    - ✓ по избыточности кодирования;
    - ✓ по эффективности использования канала;
    - ✓ по сжатию кодовой строчки записи сообщения (информации).

1. Структурная схема системы передачи дискретных сообщений и назначение ее элементов

Структурная схема должна отражать рассматриваемую задачу – согласования дискретного источника и канала без шумов. В текстовой части необходимо привести назначение всех элементов схемы [1-3, 6].

2. Модель дискретного источника сообщений

В общем случае дискретный источник сообщений без памяти (ДИБП) характеризуется дискретным ансамблем A символов, т.е. полной совокупностью независимых его состояний с соответствующими вероятностями их появления, составляющими в сумме 1. Статистическую модель такого источника можно представить в виде:

$$A=\begin{Bmatrix} a_1, & a_2, & \ldots p(a_i) \ldots & p(a_L) \\ p(a_1), & p(a_2), & \ldots p(a_i) \ldots & p(a_L) \end{Bmatrix}, \quad \sum_{i=1}^{L} p(a_i)=1, \qquad (2.1)$$

где: $p(a_i)$ - априорные вероятности выбора источником состояний a_i;

L – объем алфавита источника.

При решении практических задач, в которых рассматриваются сообщения конечной длины ($N \neq \infty$), считая рассматриваемый источник эргодическим, правомерно использование не вероятностей символов сообщения, а абсолютных (F_i) или относительных (f_i) частот их появления в заданном сообщении. При этом сумма относительных частот символов сообщения будет равна 1, а сумма абсолютных частот будет равна числу символов в сообщении (N) - длине данного сообщения. Использование частот символов, а не вероятностей, правомочно при предположении, что рассматриваемый источник является эргодическим источником без памяти.

Так для сообщения *Двести кодеров и декодеров*, характеризующегося числом символов (букв) $N = 26$ (включая пробел, обозначаемый далее символом $\Leftrightarrow$) и $L = 11$, статистическую модель можно представить в виде[1]:

$$A=\begin{Bmatrix} Д & в & е & с & т & и & \Leftrightarrow & к & о & д & р \\ 1/26 & 3/26 & 4/26 & 1/26 & 1/26 & 2/26 & 3/26 & 2/26 & 4/26 & 3/26 & 2/26 \end{Bmatrix}, \quad \sum_{i=1}^{11} f_i = 1,$$

где $f_i = \frac{F_i}{N}$ - относительная частота появления i-ой буквы (символа) в сообщении при абсолютной частоте - F_i.

Так как N – для данного сообщения константа, то для компактности записи модели, возможно использование значений только абсолютных частот появления символов (букв):

$$A=\begin{Bmatrix} Д & в & е & с & т & и & \Leftrightarrow & к & о & д & р \\ 1 & 3 & 4 & 1 & 1 & 2 & 3 & 2 & 4 & 3 & 2 \end{Bmatrix}, \quad \sum_{i=1}^{11} F_i = 26 \qquad (2.2а)$$

или в более компактном виде:

[1] Сокращать правильные дроби или преобразовывать их в десятичные не рекомендуется, т.к. ее числитель – абсолютная частота появления символа, а знаменатель дроби – общее число символов в сообщении.

$$A = \{Д_1, в_3, е_4, с_1, т_1, и_2, \Leftrightarrow_3, к_2, о_4, д_3, р_2\},\ \sum_{i=1}^{11} F_i = 26\,. \qquad (2.2б)$$

3. Информационные характеристики источника

3. 1. Количественная мера информации

Вероятностная мера (по Шеннону) количества информации $I(a_i)$, содержащееся в одном отдельном символе a_i сообщения источника (или собственная информация символа, события) при вероятностью появления этого символа $p(a_i)$, определяется как [1 - 4]

$$I(a_i) = -\log_2 p(a_i)\,. \qquad (3.1)$$

Так как далее рассматривается только двоичное кодирование и количество информации измеряется в битах, то основание логарифма (равное 2) обычно не указывается.

Если источник A является дискретным источником без памяти (ДИБП), т.е. все символы его алфавита взаимно независимы, то, исходя из свойства аддитивности количества информации, количество информации, содержащееся во всем сообщении из N символов источника определяется как

$$I(A) = \sum_{i=1}^{N} I(a_i) = -\sum_{i=1}^{N} \log p(a_i) \quad [бит]. \qquad (3.2)$$

Если же в модели источника используются не вероятности символов, а относительная частота их появления в сообщении, то количество информации в сообщении определяется как

$$I(A) = -\sum_{i=1}^{N} \log f_i\ [бит]. \qquad (3.3)$$

Например, в сообщении *Двести кодеров и декодеров* источника, заданного моделью (2.2), разные символы (буквы) характеризуются различной частотой появления в сообщении и поэтому их собственная информация различна[2]:

$I(Д) = I(т) = I(с) = -\log f_i = -\log 1/26 = 4{,}7004$ бит;

$I(и) = I(к) = I(р) = -\log 2/26 = 3{,}7005$ бит;

$I(в) = I(\Leftrightarrow) = I(д) = -\log 3/26 = 3{,}1150$ бит;

$I(е) = I(о) = -\log 4/26 = 2{,}7005$ бит.

Эти результаты подтверждают теоретическое положение о том, что *количество информации отражает степень неопределенности события* - чем меньше вероятность символа (или их частота), тем больше его неопределенность и тем больше информации он содержит.

Количество информации в этом сообщении в соответствии с (3.3) составляет:

$$I(A) = -(3\log 1/26 + 3\cdot 2\log 2/26 + 3\cdot 3\log 3/26 + 2\cdot 4\log 4/26) = -(3\log 1 + 6\log 2 + 9\log 3 + 8\log 4 - 26\log 26) =$$

$$= -(6 + 9\cdot 1{,}585 + 8\cdot 2 - 26\cdot 4{,}7) = 85{,}935 \text{ [бит]}.$$

Полученное значение $I(A)$ соответствует значению (85,943 бит), которое можно получить простым арифметическим сложением собственной информации всех символов этого сообщения. Некоторая разность этих значений объясняется точностью вычислений логарифмов. Поэтому рекомендуется в дальнейшем вычисления проводить с точностью не менее чем до четвертого знака.

---

[2] При вычислении двоичных логарифмов можно воспользоваться формулой перехода к другому основанию: $\log_2 z = \lg z/\lg 2 = 3{,}322 \lg z$.

Для облегчения вычислений количества информации значения двоичных логарифмов целых чисел приведены в Приложении 1.

Если рассматривается источник равновероятных символов с объемом алфавита *L,* то количество собственной информации можно определять по формуле Хартли, являющейся частным случаем формулы (3.1):

$$I(a_i) = \log L \text{ [бит]} \qquad (3.4)$$

Физический смысл полученных значений количества собственной информации для каждого символа и общего количества информации в сообщении *I(A)* - это длина кодового слова для отдельных символов и, соответственно, длина строчки записи всего сообщения в двоичном коде.

3. 2. Энтропия источника *H(A)*

В теории информации для количественной оценки информации в сообщениях, а, значит, и оценки самого стационарного источника дискретных сообщений без памяти (ДИБП), используется *математическое ожидание количества информации источника (m_1),* получаемое путем усреднения $I(a_i)$ по всем L символам алфавита данного источника A:

$$H(A) = -m_1\{\log p(a_i)\} = -\sum_{i=1}^{L} p(a_i)\log p(a_i)\left[\frac{бит}{символ}\right] \qquad (3.5)$$

Данная величина *H(A)* получила название *энтропия источника* [1-4] и измеряется в бит/символ. Для краткости допускается запись ее размерности в битах.

Энтропия *H(A)* выражает среднюю неопределенность состояний источника сообщений и с точки зрения кодирования определяет *статистически среднее количество информации в битах, которое приходится на один символ алфавита источника*.

Например, энтропия источника *A* сообщения *Двести кодеров и декодеров,* исходя из модели (2.2), определяется по (3.5), в котором вероятности символов заменены на частоты их появления[3]:

$$H(A) = -\sum_{i=1}^{L} f_i \log f_i = -(3\cdot {}^{1}\!/_{26}\log {}^{1}\!/_{26} + 3\cdot {}^{2}\!/_{26}\log {}^{2}\!/_{26} + 3\cdot {}^{3}\!/_{26}\log {}^{3}\!/_{26} + 2\cdot {}^{4}\!/_{26}\log {}^{4}\!/_{26}) =$$
$$= -({}^{6}\!/_{26}\log 2 + {}^{9}\!/_{26}\log 3 + {}^{8}\!/_{26}\log 4 - \log 26) = -(\frac{6+9\cdot 1{,}585+8\cdot 2}{26} - 4{,}7005) = 3.3055\ [бит] \qquad (3.6)$$

Для удобства вычислений в Приложении 2 приведены значения энтропий – $p(a_i)\log p(a_i)$.

3. 3. Максимально возможная энтропия источника

Энтропия источника принимает максимальное значение при равновероятности (или равной частоте) всех символов *L,* составляющих алфавит источника [1-4]:

$$H_{\max}(A) = \log L \text{ [бит]}. \qquad (3.7)$$

Так максимально возможная энтропия источника сообщения *Двести кодеров и декодеров* составляет $H_{\max}(A) = \log L = \log 11 = 3{,}4594$ бит.

---

[3] Так как *H(A)* – это среднее количество информации, приходящейся на один символ сообщения, то должно выполняться соотношение $I(A) \approx N{\cdot}H(A)$. В приводимом примере это: 3,3055 ·26 = 85,943 (бит) ≈ I(A)=85,935 (бит)

3. 4. Производительность источника (ДИБП)

Под производительностью источника понимается *среднее количество информации, создаваемой источником в единицу времени.* При заданной и постоянной длительности символов сообщения производительность дискретного источника численно равна отношению энтропии источника к длительности символа T_0:

$$H'(A) = \frac{H(A)}{T_0}. \qquad (3.8)$$

Размерность величин производительности источника – бит/с.

3. 5. Избыточность источника (ДИБП)

Избыточностью источника дискретных сообщений с энтропией H(A) и объемом алфавита L называется относительная величина (или коэффициент)

$$\rho_u^p = \frac{H_{\max} - H(A)}{H_{\max}} = 1 - \frac{H(A)}{H_{\max}} = 1 - \frac{H(A)}{\log L}, \qquad (3.9)$$

где H_{max} - максимально возможное значение энтропии при данном объеме алфавита, которое возможно при независимых и равновероятных символах сообщения, определяемое в соответствии с (3.7).

Таким образом, ненулевая избыточность говорит о наличии либо не равновероятности символов сообщения, либо о наличии памяти источника, либо о том и другом. Относительная избыточность является неотрицательной величиной и измеряется в долях единицы или в %.

Исходя из (3.6) и (3.7), источник рассматриваемого сообщения (2.2) характеризуется

$$\rho_u = 1 - \frac{3{,}3055}{3{,}4594} = 0{,}0445 \text{ или } 4{,}45\%.$$

4. Пропускная способность канала

Пропускной способностью ($C_к$) канала называется наибольшая возможная скорость $v_к$ передачи информации по каналу при заданных ограничениях:

$$C_к = \max v_к = \lim \frac{H_{\max}(A)}{T_0} \text{ [бит/с].}$$

где T_0 – длительность двоичных символов (битовый интервал) на входе канала. Скорость передачи информации в общем случае зависит от статистических характеристик передаваемых сообщений и параметров канала. Количественно C_c выражается максимальным числом двоичных единиц информации, которые канал может передавать за 1 сек.

В рассматриваемой задаче предполагается идеальный дискретный двоичный канал без памяти. Для такого канала

$$C_к = 1/T_0,$$

Для повышения эффективности использования канала при передаче информации должна быть решена задача согласования источника и канала. Эта задача решается за счет используется статистического (эффективного) кодирование источника. Количественно степень согласования производительности источника сообщений и пропускной способности канала при

заданном канале и использовании того или иного кодирования оценивается коэффициентом эффективности кодирования:

$$\eta_c = \frac{\boldsymbol{H'(B)}}{\boldsymbol{C_к}}. \qquad (4.1)$$

Одновременно коэффициент эффективности кодирования определяет насколько эффективно при данном кодировании будет использоваться канал. С этой точки зрения η_c равносильно называть коэффициентом эффективности использования канала ($\eta_к = \eta_c$). Значения этого коэффициента удовлетворяют неравенству

$$\mathbf{0} \le \eta_к \le \mathbf{1}.$$

4.1. Эффективное кодирование дискретного источника без памяти

До настоящего времени не существует единого метода эффективного кодирования источника. Существует большое число частных методов кодирования источника (Шеннона, Шеннона-Фано, Хаффмана, Гильберта-Мура, Бабкина-Фитингофа, арифметическое кодирование, интервальное кодирование, LZW и много других). Задачей любого их этих методов является *согласование производительности источника сообщений с пропускной способностью канала*. При заданной пропускной способности канала эта задача сводится к увеличению производительности источника путем изменения статистических характеристик закодированного сообщения и поэтому для нее часто используется другое название – *задача кодирования источника:*

- если ко входу канала подключен кодер с производительностью равной пропускной способности канала ($\boldsymbol{H'(B) = C_к}$), то источник согласован с каналом;
- если $\boldsymbol{H'(B) < C_к}$, то такой источник не согласован с каналом связи и последний используется неэффективно.

Теория кодирования дискретных источников основывается на двух теоремах Шеннона о кодировании ДИБП [1-4, 6], в которых утверждается существование *M*-ичных префиксных равномерных и неравномерных кодов с определенной длиной кодовых слов n_c и, соответственно, $\overline{n}_c$. В этих теоремах утверждается, что длина кодового слова n_c(для равномерного кода) или средняя длина кодового слова $\overline{n}_c$ (для неравномерного кода) не может быть меньше энтропии кодируемого источника. Термин *префиксный код* означает то, что никакое кодовое слово не может быть началом другого кодового слова.

Само существование однозначно декодируемого кода *M*-ичного кода, содержащего *L* кодовых слов с длинами n_i, определяется неравенством Крафта [1, 6]:

$$\sum_{i=1}^{L} \boldsymbol{M}^{-n_i} \le \mathbf{1} \qquad (4.2)$$

Далее предполагается только двоичное кодирование, т.е. *M*=2.

5. Равномерный двоичный код

Предположим, что источник $\boldsymbol{A}$ сообщения *Двести кодеров и декодеров* порождает множество равновероятных и независимых символов-букв (любой из них обозначим как a_i) и при *L*=11 каждый из них характеризуется $p(a_i)$=1/11.

Следовательно, собственная информация каждой буквы сообщения составляет $I(a_i) = -\log p(a_i) = -\log 1/11 = 3{,}4594$ бит и по теореме Шеннона минимальное значение принимается $\boldsymbol{n_i = 4}$.

Необходимым и достаточным условием возможности создания однозначно декодируемого кода для такого источника является выполнение неравенства Крафта (4.2). В данном случае: $\sum_{i=1}^{L} M^{-n_{ik}} = \frac{11}{2^4} \le 1$. Выполнение неравенства Крафта подтверждает возможность однозначного кодирования и декодирования данного источника при равномерном двоичном кодировании.

Кодирование символов алфавита производится обычной записью натуральных двоичных (M=2) чисел соответствующих номерам символов (в табл. 5.1 они выделены жирным шрифтом). Но при этом получаем неравномерный и не префиксный код, который не может быть однозначно декодируемый. Поэтому для трансформации такого кода в равномерный префиксный код с $\boldsymbol{n_c \ge \log L = 3{,}4594}$ впереди добавляются соответствующие блоки нулей до минимального целого числа $\boldsymbol{n_c = 4}$. В результате полученный код является простым двоичным кодом, каждая кодовая комбинация которого отличается от соседней не менее чем на единицу, т.е., его минимальное кодовое расстояние d_0=1.

Таблица 5.1. Равномерный префиксный код

символы сообщения	кодовые комбинации
a_1	000**1**
a_2	00**10**
a_3	00**11**
a_4	0**100**
a_5	0**101**
a_6	0**110**
a_7	0**111**
a_8	**1000**
a_9	**1001**
a_{10}	**1010**
a_{11}	**1011**

Длина строчки записи сообщения *Двести кодеров и декодеров* при кодировании равномерным кодом составит: $\boldsymbol{l_{стр} = n_c \cdot N = 4 \cdot 26 = 104}$ бит.

1. Энтропия кодера $\boldsymbol{H(B) = \frac{H(A)}{n_c \log M} = \frac{3{,}4594}{4} = 0{,}86485}$ бит

Следовательно, каждый символов (0 и 1) такого равномерного кода в среднем содержит информации меньше чем максимально возможное $H_{\max}(B) = \log M = 1$ и такой код не является безизбыточным.

2. Избыточность такого равномерного кода $\rho_c = 1 - \frac{H(B)}{H_{\max}(B)} = 1 - 0{,}86485 =$ 0,13515 или (13,515%)
3. Исходя из табл. 5.1, вероятности появления символов 0 и 1 в кодовых словах:

$$p(0) = \frac{\sum_{i=1}^{L} n_i(0)\,p(a_i)}{n_c} = \frac{4 \cdot 3/11 + 5 \cdot 2/11 + 2 \cdot 1/11}{4} = 6/11;$$

$$p(1) = \frac{\sum_{i=1}^{L} n_i(1)\,p(a_i)}{n_c} = \frac{2 \cdot 3/11 + 5 \cdot 2/11 + 4 \cdot 1/11}{4} = 5/11,$$

где $n_i(0)$ и $n_i(1)$ - количество 0 и 1 в i- ом кодовом слове.

Полученные значения $p(0) > p(1)$ свидетельствуют о неравновероятности символов этого кода, что обуславливает наличие в нем избыточности.

4. Производительность равномерного кодера $H_c'(B) = \frac{H_c(B)}{T_0} = 864{,}85$ *бит / с* - меньше максимально возможной производительности кодера и пропускной способности канала (1000 бит/с), что снижает эффективность использования канала;
5. Эффективность использования канала $\eta_c = \frac{H_к(B)}{C_c} = \frac{H_к/T_0}{1/T_0} = 0{,}86485$ (86,485%).

Следовательно, равномерное кодирование не обеспечивает полного согласования источника и канала, что объясняется значительной избыточностью равномерного кода (13,515%), которая приводит к уменьшению производительности кодера и к не эффективному использованию канала.

6. Метод Шеннона-Фано построения эффективного кода.

Достоинство этого метода кодирования состоит в том, что по одной вероятности символа $p(a_i)$ можно непосредственно определить длину n_i соответствующего кодового слова.

Пусть задан источник $A = \begin{Bmatrix} a_1 & a_2 & \dots & a_L \\ p(a_1) & p(a_2) & \dots & p(a_L) \end{Bmatrix}$

Длины кодовых слов при двоичном кодировании, естественно, определяются собственной информацией каждого из них, т.е., $-\log p(a_i)$ с учетом того, что n_i – должно быть целым числом. Для каждой $p(a_i)$ найдется такое целое число n_i, что

$$-\log p(a_i) \le n_i < -\log p(a_i) + 1.$$

Умножив данное выражение на $p(a_i)$ и просуммировав по всему ансамблю i, получим:

$$-\sum_{i=1}^{L} p(a_i)\log p(a_i) \le \sum_{i=1}^{L} p(a_i) n_i < -\sum_{i=1}^{L} p(a_i)\log p(a_i) + 1.$$

В принятых обозначениях это выражение записывается как

$$H(A) \le \overline{n}_к < H(A) + 1 \qquad (6.1)$$

Отсюда следует, что как утверждается в основной теореме кодирования источника [1-4,6] *для неравномерного кода Шеннона-Фано энтропия источника определяет нижнюю границу средней длины кодовых слов*. Более того, представляется возможным определение длин кодовых слов по их вероятности как количество информации в каждом из них.

Пример 6.1: Определить возможность построения однозначно декодируемого кода для источника $A = \begin{Bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\ 1/4 & 1/4 & 1/8 & 1/8 & 1/8 & 1/16 & 1/16 \end{Bmatrix}$. (6.2)

При этом $I(a_1)= I(a_2)= -\log ¼ = 2$ *бит;* $I(a_3)= I(a_4)= I(a_5)=3$ *бит;* $I(a_6)= I(a_7)=4$ *бит*.

Тогда $n_1 = n_2 = 2, \quad n_3 = n_4 = n_5 = 3, \quad n_6 = n_7 = 4.$

Проверяем неравенство Крафта $\sum_{i=1}^{L} M^{-n_i} \le 1$, где M=2, L=7:

$$\frac{2}{2^2} + \frac{3}{2^3} + \frac{2}{2^4} = \frac{8+6+2}{16} = 1.$$

Следовательно, такой однозначно декодируемый код можно построить. После этого строим сами кодовые слова заданных длин префиксного двоичного кода:

$a_1 \to 00, a_2 \to 01, a_3 \to 100, a_4 \to 101, a_5 \to 110, a_6 \to 1110, a_7 \to 1111.$

Такое множество кодовых слов, как будет показано в разделе 6.2, соответствует оптимальному коду такого источника.

Общие правила большинства методов статистического кодирования для формирования кодовых слов со средней длиной, близкой к её нижней границе, исходя из алгоритма Шеннона, следующие [1]:

Правило 1. *В каждой позиции кодового слова различные символы алфавита должны использоваться с равными вероятностями*.

Правило 2. *Вероятности появления кодовых символов в каждой позиции кодового слова должны быть взаимно независимыми.*

Выполнение этих правил обеспечивает минимизацию избыточности кода, что увеличение среднего количества информации в каждом кодовом символе, т.е. его $H(B)$, и, следовательно, минимизацию средней длины кодовых слов и минимизацию длины строчки записи сообщения в данном коде.

Рассматриваемый алгоритм формирования кода Шеннона-Фано пошаговый:

Шаг 1. Символы исходного алфавита источника ранжируются по не возрастанию вероятностей с записью в виде столбца таблицы. В результате получаем множество символов алфавита источника с новым порядком следования символов a_i, при котором $p(a_i) \ge p(a_{i+1})$.

Шаг 2. Рассматриваемое множество из L символов алфавита разбиваем на M=2 подмножеств, не меняя порядка следования символов, так чтобы сумма вероятностей символов a_i в каждом подмножестве были примерно одинаковы и максимально близки к 1/2. Получаем два подмножества G_0 и G_1:

$G_0 = (a_{i=1}, a_{i=2}, \ldots, a_{i=\kappa})$,

$G_1 = (a_{i=k+1}, a_{i=k+2}, \ldots, a_{i=L})$.

Шаг 3. Формируем первый символ кодовых слов. Для этого всем символам из подмножества G_0 приписываем кодовый символ 0, а G_1 - символ 1.

Шаг 4. Просматриваем подмножества. Если одно из них состоит из единственного символа, то для этого a_i процесс построения кодового слова считается законченным.
Шаг 5. Для второго подмножества, содержащего два и более символов, выполняем действия, соответствующие Шагу 2 и Шагу 3. В результате чего получаем последующие символы кодовых слов.

Процесс работы алгоритма заканчивается тогда, когда все подмножества G_j будут содержать ровно по одному символу исходного алфавита.

Необходимо обратить внимание на ряд условий, которые нужно иметь в виду при практическом применении этого метода:
Первое. При разбиении на подмножества не разрешается переставлять ранжированные символы с целью выравнивания сумм вероятностей. Порядок следования символов, полученный на Шаге 1, сохраняется в течение всего времени работы алгоритма.
Второе. Разбиение на подмножества не всегда выполняется однозначным образом. Это связано с тем, что иногда некоторый "пограничный" символ a_i может быть присоединён к любому из каких-то двух подмножеств G_j или G_{j+1} равноценным образом. Если a_i присоединяется последним символом в G_j, то в ней суммарная вероятность станет на $p(a_i)$ больше, чем в группе G_{j+1} . Если же a_i включить первым символом в G_{j+1}, то большая сумма вероятностей будет в G_{j+1}. Подобная ситуация при разбиении на подмножества может возникнуть не один раз. Следовательно, результат кодирования однозначно не определяется. Можно получить несколько равноценных между собой кодов для одного и того же алфавита. С целью получения однозначности результатов оговаривают дополнительные условия. Например, можно потребовать, чтобы большее значение суммы вероятностей было в группе с меньшим номером. Однако именно такое требование не является обязательным и используемый способ разбиения оговариваются самим исполнителем при выполнение задания.
Третье. Следует на всех шагах придерживаться одинаковой последовательности приписывания символов 0 или 1 подмножествам символов исходного алфавита. Это условие исполнителю следует оговорить до начала работы алгоритма. Обычно придерживаются следующего правила: символам из подмножества с номером j приписывают 0-й символ кодового алфавита, а подмножеству с номером $j+1$, соответственно, приписывают 1. В противном случае будет получен инверсный код с той же статистикой.

Пример 6.2. Рассмотрим алгоритм Шеннона-Фано при кодировании двоичным кодом символов источника (табл. 6.1) с проранжированным на Шаге 1 множеством из L=5 не равновероятных символов источника $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{Bmatrix} \boldsymbol{a_1},\ \boldsymbol{a_2},\ \boldsymbol{a_3},\ \boldsymbol{a_4},\ \boldsymbol{a_5} \\ \mathbf{0{,}4;0{,}35;0{,}1;\ 0{,}1;0{,}05} \end{Bmatrix} \qquad (6.3)$$

При этом примем условие, что символы с меньшими суммарными вероятностями включаются в подмножества с меньшим номером. Тогда на Шаге 2 получаем подмножество $G_0 = (a_1)$ с вероятностью 0,4 и $G_1=(a_2,a_3,a_4,a_5)$ с суммарной вероятностью 0,6. Принимаем второе условие: на Шаге 3 кодовый символ 0

приписываем подмножеству с меньшим номером, т.е., символу a_1 приписываем кодовый символ 0, а символам подмножества G_1 в качестве первого разряда - 1. Так как подмножество G_1 состоит из четырех символов, то разбиваем его на новые два подмножества G_2=(a_2) с вероятностью 0,35 и G_3=(a_3, a_4, a_5) с суммарной вероятностью 0,25. Далее аналогично производится формирование подмножеств G_4, G_5, G_6 и G_7.

Таблица 6.1. Код Шеннона-Фано

a_j	$p(a_j)$	Подмножества G и кодовые символы				кодовые слова	$n_j\, p(a_j)$
a_1	0,4	$G_0 \to 0$				0	0,4
a_2	0,35	$G_1 \to 1$	$G_2 \to 0$			10	0,7
a_3	0,1		$G_3 \to 1$	$G_4 \to 0$		110	0,3
a_4	0,1			$G_5 \to 1$	$G_6 \to 0$	1110	0,4
a_5	0,05				$G_7 \to 1$	1111	0,2

Получен неравномерный код с $\bar{n}_c = \sum_{j=1}^{5} n_j p(a_j) = 2$.

7. Кодовое дерево [1]

Кодовое дерево или дерево решений наглядно отображает процессы при мгновенном декодировании кодовых слов.

Приведем кодовое дерево (рис. 7.1) рассмотренного в примере 6.2 кода Шеннона-Фано. Корень дерева отображает все множество G символов источника a_1, a_2, a_3, a_4, a_5. Так кодирование двоичное, то это множество символов разбивается на два ($M=2$) максимально равновероятных подмножества: G_0, состоящее из одного символа a_1 с вероятностью 0,4, и G_1, состоящее из остальных символов $a_2 \ldots a_5$ с суммарной вероятностью 0,6. Две ветки, идущие от корня дерева к узлам первого порядка, соответствуют выбору между нулем и единицей в качестве первого символа кодовых слов - левая ветвь (G_0) соответствует выбору *0*, правая (G_1) - *1*. Левая ветка заканчивается концевым узлом первого порядка – листиком, отображающим кодовое слово символа $a_1 \to 0$. Далее две ветви (G_2 и G_3) , идущие от родительского узла G_1 первого порядка, соответствуют выбору второго символа кодовых слов, левая ветвь G_2 снова обозначена *0*, правая G_3 *-1* и т.д. Последовательные символы каждого кодового слова образуются продвижением от корня дерева до концевых узлов (листьев) соответствующих символов. Множество листьев образует множество кодовых слов данного кода: пяти символам алфавита источника соответствуют пять кодовых слов с длиной, определяемой *высотой кодового дерева* или количеством уровней узлов (в рассматриваемом примере n= 4). Кодовое дерево показывает как "расщепляется" ранжированное множество символов сообщения по принципу равновероятности его подмножеств – ветвей кодового дерева.

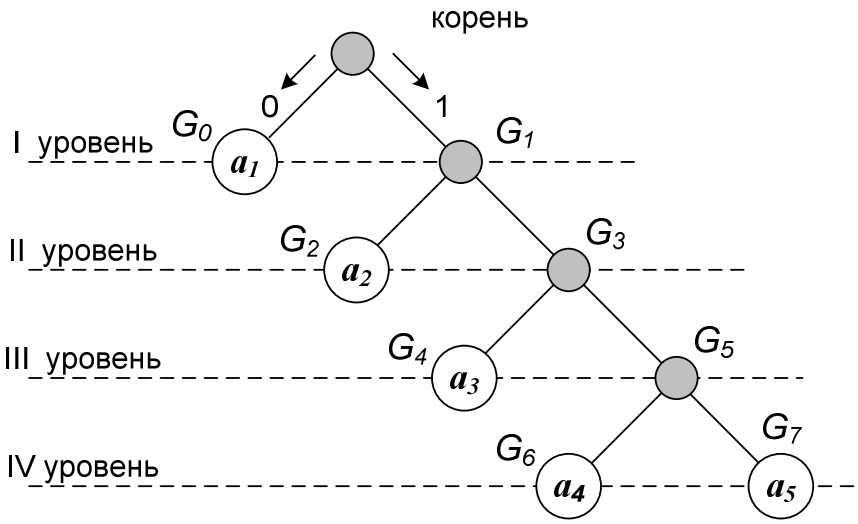


Рис. 7.1. *Кодовое дерево источника (пример 6.2)*

Можно заметить, что сумма совместной информации в последовательных родительских узлах, ведущих к некоторым концевым узлам, соответствует суммам собственной информации символов источника, соответствующих данным узлам. Таким образом, чтобы достичь малого значения $\overline{n}_K$, следует выбирать кодовые слова таким образом, чтобы все ребра, идущие из узла, были по возможности равновероятны. В вышеприведенном дереве каждая ветвь выбирается как почти равновероятная.

7.1. Условие оптимальности кодов источника

При построении кода Шеннона-Фано разбиение ранжированного множества символов алфавита источника на два равновероятных подмножества (а при построении кодового дерева – на равновероятные ветви) возможно только тогда, когда вероятности символов сообщения (или их относительные частоты) и длины их кодовых слов будут соответствовать равенству:

$$p(a_i) = M^{-n_i}$$

Это равенство является необходимым условием для получения любого оптимального неравномерного кода. Критерием оптимальности кода в данном случае является минимизация средней длины кодовых слов или минимизация высоты кодового дерева.

Источник с именно такой статистикой рассматривался в примере 6.1. У такого источника вероятности символов $p(a_i) = \frac{1}{2^{n_i}}$ (где n_i =2, 3, 4) а энтропия

$$H(A) = -\sum_{i=1}^{7} p(a_i)\log p(a_i) = -(2\cdot\frac{1}{4}\log\frac{1}{4} + 3\cdot\frac{1}{8}\log\frac{1}{8} + 2\cdot\frac{1}{16}\log\frac{1}{16}) = 2\frac{5}{8}\ бит$$

и средняя длина кодовых слов $\overline{n}_c = \mathbf{2\frac{5}{8}}$ бит.

Следовательно, энтропия кодера $\boldsymbol{H(B) = \frac{H(A)}{\overline{n}_c \log M} = 1}$ - и при таком распределении вероятностей символов источника кодирование является оптимальным: при этом неравенство (6.1) преобразуется в равенство $\boldsymbol{H(A) = \overline{n}_c}$.

8. Характеристики кода Шеннона-Фано

Для подтверждения того факта, что кодирование Шеннона-Фано для источника с произвольной статистикой является достаточно хорошим, определим числовые характеристики источника из примера 6.2 и характеристики его равномерного и неравномерного кодов.

8.1. Характеристики источника, представленного моделью (6.2):

1. $H(A) = -\sum_{i=1}^{5} p(a_i) \cdot \log p(a_i) = 0{,}5288 + 0{,}5301 + 0{,}3322 + 0{,}3322 + 0{,}2161 = 1{,}94$ бит
2. $H_{\max}(A) = \log L = \log 5 = 2{,}322$ бит
3. $\rho_u^p = 1 - \frac{H(A)}{H_{\max}(A)} = 1 - \frac{1{,}94}{2{,}32} = 0{,}164$ - избыточность ДИБП, которую нужно уменьшить.

8.2. Характеристики возможного равномерного кодирования такого источника:

1. Число кодовых комбинаций кода при основании M должно удовлетворять неравенству $M^{n_к} \geq L$.

Отсюда $n_c \geq \frac{\log L}{\log M} = \frac{\log 5}{\log 2}$; Выбираем ближайшее целое $n_к$=3.

2. Энтропия кодера $H(B) = \frac{H(A)}{n_c} = \frac{1{,}94}{3} = 0{,}647$ бит

3. Тогда избыточность такого кодера: $\rho_c = 1 - \frac{H(A)}{n_c \cdot \log M} = 1 - \frac{1{,}94}{3} = 0{,}353$ при коэффициенте сжатия $K_{сж} = 0{,}647$.

4. Производительность кодера $H'(B) = \nu_к \cdot H(B) = 0{,}647 \cdot \nu_к$ (бит/с)

5. Пропускная способность двоичного канала: $C_к = \nu_к \cdot \log M = \nu_к$ (бит/с)

6. Эффективность использования канала: $\eta_к = \frac{H'(B)}{C_к} = \frac{0.647 \cdot \nu_к}{\nu_к} = 0{,}647$.

Полученные результаты показывают, что при равномерном кодировании избыточность кода по сравнению с избыточностью самого источника увеличилась и эффективность использования канала является низкой. Следовательно, равномерное кодирование источника нельзя считать эффективным.

8.3. Характеристики кода Шеннона-Фано для такого источника:

1. Средняя длина кодовых слов $\bar{n}_c = \sum_{i=1}^{5} n_i \cdot p(a_i) = 0{,}4 + 0{,}7 + 0{,}3 + 0{,}4 + 0{,}2 = 2$ бит.
2. Нижняя граница для средней длины кодовых слов, определяемая энтропией источника $\bar{n}_c = H(A) = 1{,}94$ бит.
3. Энтропия кодера $H(B) = \frac{H(A)}{\bar{n}_c \log M} = \frac{1{,}94}{2} = 0{,}97$ бит
4. Производительность кодера $H'(B) = \nu_к H(B) = 0{,}97\nu_к$ бит/с
5. Избыточность кодера:

$$\rho_c = 1 - \frac{H(A)}{\bar{n}_c} = 1 - \frac{H(B)}{\log M} = 1 - 0{,}97 = 0{,}03; \quad K_{сж} = \frac{H(B)}{\log M} = 0{,}97.$$

5. Эффективность использования канала:

$$\eta_к = \frac{H'(B)}{C_к} = \frac{\nu_к H(B)}{\nu_к} = 0{,}97$$

Сравним результаты, полученные для равномерного и неравномерного эффективного кодирования данного источника:

- Теоретически возможная минимальная средняя длина кодовых слов составляет: $\min \bar{n}_c = H(A) = 1{,}94$.
- Реально получена средняя длина кодовых слов неравномерного кода Шеннона-Фано $\bar{n}_c = 2$ и фиксированная длина кодовых слов при равномерном кодировании n_c=3.
- Энтропия эффективного кодера и производительность кодера при неравномерном кодировании выше (*0,97>0,647*), чем при равномерном;
- Избыточность неравномерного кода ниже (*0,03<0,353*);
- Эффективность использования канала, соответственно, выше (0,97) и весьма близка к единице.

Так как максимальное количество информации, которое может перености двоичный сигнал равно *1* биту, то видно, что энтропия кодера Шеннона-Фано (*0,97*) весьма близка к максимальной – это объясняется тем, что символы (*0* и *1*) кодовых слов почти равновероятны[4]:

$$p(0) = \frac{0{,}4 + 0{,}35 + 0{,}1 + 0{,}1}{2} = 0{,}475$$

$$p(1) = \frac{0{,}35 + 2 \cdot 0{,}1 + 3 \cdot 0{,}1 + 4 \cdot 0{,}05}{2} = 0{,}525$$

Таким образом, кодирование по алгоритму Шеннона-Фано обеспечивает формирование префиксного кода, является в общем случае достаточно эффективным и хорошим, т.к., кодер преобразовал неравновероятные независимые элементы сообщения источника сообщений в почти равновероятные независимые кодовые символы, значительно уменьшает среднюю длину кодовых слов и избыточность источника, что позволяет достаточно эффективно использовать пропускную способность канала.

Для ансамбля равновероятных символов равномерный код является оптимальным и если число символов алфавита источника равно целой степени 2, то всегда $H(A) = \bar{n}_c$.

К достоинствам кода Шеннона-Фано необходимо отнести простоту реализации а, значит, в высокую скорость кодирования-декодирования. Недостаток кода – его не оптимальность в общем случае.

9. Классический алгоритм Хаффмана

Алгоритм эффективного кодирования Хаффмана [1-4, 6] всегда приводит к получению минимально-избыточного множества кодовых слов в том смысле, что никакие другие множества не имеют меньшего среднего числа символов на

[4] Так как для двоичного кода 0 и 1 представляют полную группу событий, то *p(0)+p(1)*=1

сообщение и поэтому код Хаффмана является минимально-избыточным. Он, как и алгоритм Шеннона – Фано, обеспечивает формирование префиксного кода и также основывается на блочном кодировании с переменной длиной, что обеспечивает:
- устранение памяти за счет блочного кодирования,
- устранение избыточности за счет переменной длины: высоковероятные символы сообщения кодируются короткими кодовыми комбинациями, а низковероятные – длинными.

***Определение**: Если алфавит источника* $A=\{a_1,a_2...a_L\}$ *состоит из L различных символов с вероятностями, соответственно, p(*a_1*), p(*a_2*),…p(*a_L*), то множество кодовых слов* $B=\{b_1,b_2...b_L\}$ *является кодом Хаффмана или минимально-избыточным кодом, удовлетворяющим следующим условиям:*

1 - b_i *не является префиксом для* b_j *при* $i \neq j$ *;*

2 – средняя длина кодовых слов $\sum_{i=1}^{L} p(b_i)n_i$ *минимальна.*

Первое из этих условий является условием префиксности кода, а второе – условием его минимальной избыточности.

Алгоритм кодирования источника по Хаффману пошаговый и приводит к получению множества кодовых слов с минимальной избыточностью при использовании следующих процедур:

Шаг 1: Производим ранжировку исходного множества символов источника - все множество *L* символов представляют списком в порядке не возрастания их вероятностей;

Шаг 2: Так как дальнейшее кодирование двоичное, то формируем подмножество из двух символов, имеющих наименьшие вероятности в ранжированном списке, и вычисляем общую вероятность этого подмножества, рассматриваемого далее как мнимый символ с суммарной вероятностью группы входящих в него символов;

Шаг 3: Добавляем этот мнимый символ в список и вновь производим ранжировку символов в порядке не возрастания вероятностей;

Шаг 4: Снова образуем подмножество из двух символов с наименьшими вероятностями и вычисляем их общую вероятность для нового мнимого символа.

Шаг 5: Вновь, рассматривая это подмножество как новый символ с суммарной вероятностью, вносим его в список и ранжируем все символы в порядке убывания их вероятностей, как показано для ниже рассматриваемого примера.

Повторяя последовательно подобные шаги, производим формирования подмножеств, пока в ансамбле не останется единственный мнимый символ с суммарной вероятностью единица.

Завершающий шаг. Проводя линии, соединяющие мнимые и реальные символы, образующие последовательные подмножества, получаем «свернутое» кодовое дерево, в котором каждый промежуточный (родительский) узел имеет вероятность равную сумме вероятностей всех узлов, находящихся ниже, а концевые узлы являются листиками. Соответствующие им кодовые слова можно построить, приписывая различные символы из заданного алфавита *M* ветвям, исходящим из промежуточных узлов.

Как видно, алгоритм Хаффмана в отличие от алгоритма Шеннона-Фано является алгоритмом кодирования «снизу-вверх»

Так как при таком алгоритме кодовым словам соответствуют только концевые узлы, то полученный код является префиксным.

Полученное таким образом множество кодовых слов оптимально (в том смысле, что при данном распределении вероятности символов источника не существует множества кодовых слов с меньшей средней длиной кодового слова). Полученный при этом код характеризуется минимально возможной при всех других алгоритмах кодирования источника избыточностью.

Пример 9.1.: пусть источник сообщений A с $L=6$ характеризуется распределением вероятностей $p(a_1)=0,25;\ p(a_2)=0,25;\ (a_3)=0,2;\ p(a_4)=0,15;\ p(a_5)=0,1;\ p(a_6)=0,05$. Кодер источника использует алфавит B с $M=2$ (двоичный кодер).

Пошагово выполняя все процедуры кодирования, представленные на рис. 9.1, получаем код Хаффмана для данного источника, представленный в табл. соответствия 9.1.

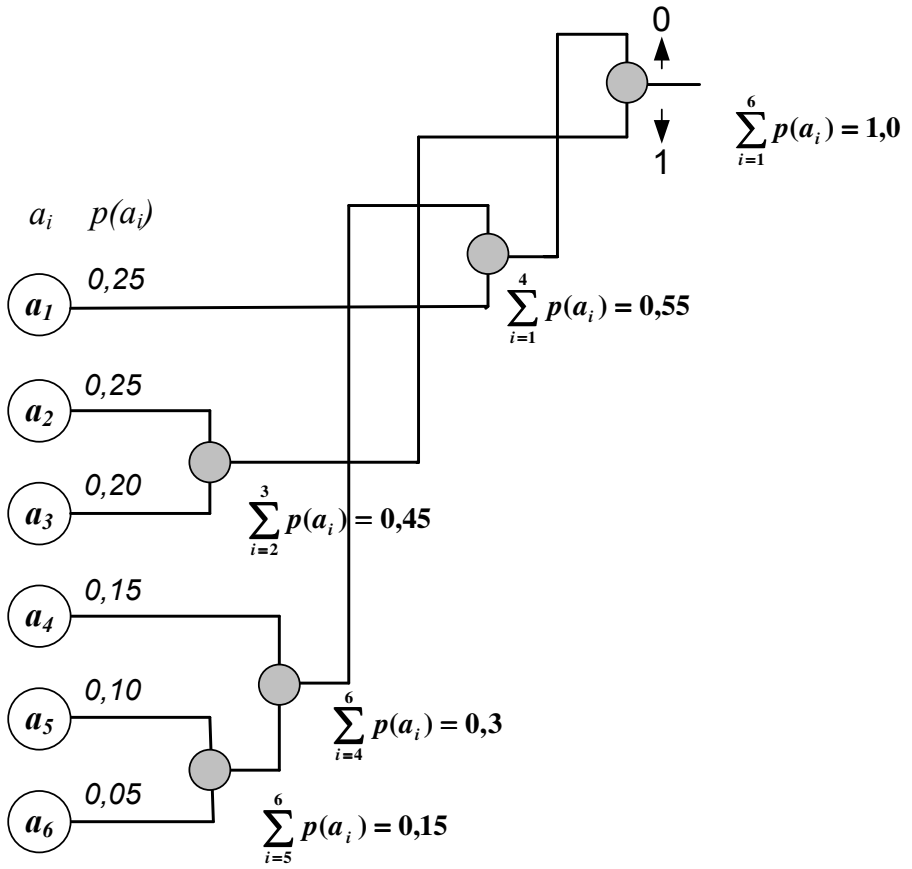


***Рис.9.1**. «Свернутое» кодовое дерево Хаффмана*

9.1. Характеристики полученного кода Хаффмана:

1. Энтропия источника $H(A) = -\sum_{i=1}^{6} p(a_i)\log p(a_i) = 2,422$ бит.
2. Средняя длина кодовых слов

$\overline{n}_c = \sum_{i=1}^{L} p(a_i)n_i = 4\cdot 0,05+4\cdot 0,1+3\cdot 0,15+2\cdot 0,2+2\cdot 0,25+2\cdot 0,25=2,45$ бит.

3. Энтропия кодера при этом:

$$\boldsymbol{H(B) = \frac{H(A)}{\overline{n}_c \log M} = -\sum_{i=1}^{6} \frac{p(a_i)\log p(a_i)}{\overline{n}_c} = \frac{2{,}422}{2{,}45} = 0{,}988}\ \text{бит.}$$

4. Вероятности кодовых символов в полученном коде:

$$p(0) = \frac{1{,}1}{2{,}45} = 0{,}449; \quad p(1) = \frac{1{,}35}{2{,}45} = 0{,}551.$$

5. Избыточность неравномерного кода

$$\boldsymbol{\rho_c = 1 - \frac{H(B)}{\log M} = 1 - 0{,}988 = 0{,}012}$$

6. Производительность кодера $\boldsymbol{H'(B) = \nu_к H(B) = 0{,}988\,\nu_к}$ бит/с
7. Эффективность использования канала $\eta_к = 0{,}988$.

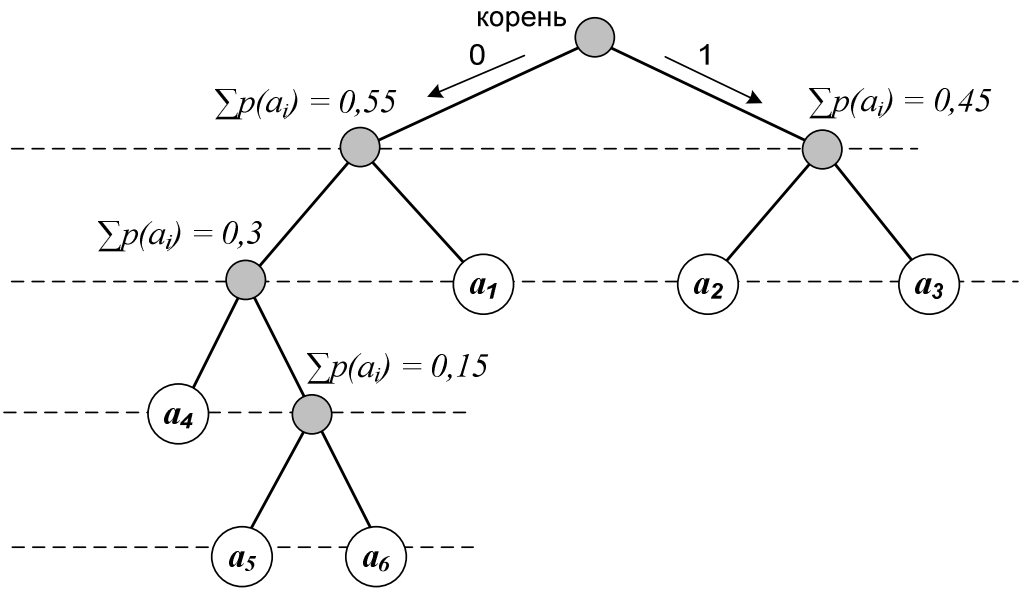


Таблица 9.1.
Код Хаффмана

a_i	b_i
a_1	01
a_2	10
a_3	11
a_4	000
a_5	0010
a_6	0011

Рис.9.2. *Кодовое дерево кода Хаффмана для примера 9.1.*

9.2. Варианты построения кодов Хаффмана

На практике возможны варианты реализации классического алгоритма Хаффмана. Рассмотрим их на следующем примере.

Пример 9.2: Пусть задан источник $\boldsymbol{A}$**:** $A = \begin{Bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\ 0{,}4 & 0{,}2 & 0{,}2 & 0{,}1 & 0{,}05 & 0{,}05 \end{Bmatrix}$

Построить для такого источника варианты кода Хаффмана.

Вариант первый. В процессе ранжировки будем размещать формируемые подмножества G_i (мнимые символы) как можно ниже - под реальные символы с равными вероятностями. Назовем такой вариант кодирования «под» (табл. 9.2).

Таблица 9.2. Варианты кода Хаффмана

	Вероятности:					
	исходного алфавита	пошаговых промежуточных списков (подмножеств)				
		G_1	G_2	G_3	G_4	G_5
a_1	0,4	0,4	0,4	0,4	0,6	1,0
a_2	0,2	0,2	0,2	0,4	0,4	
a_3	0,2	0,2	0,2	0,2		
a_4	0,1	0,1	0,2			
a_5	0,05	0,1				
a_6	0,05					

При таком варианте ранжирования промежуточный списков формируется следующий код Хаффмана: $a_1=1$, $a_2=01$, $a_3=000$, $a_4=0010$, $a_5=00110$, $a_6=0011$ со средней длиной кодовых слов

$\bar{n}_c = 0{,}4 + 0.4 + 0{,}6 + 0{,}4 + 0{,}25 + 0{,}25 = 2{,}3$ *бит*. Энтропия такого кодера составляет:

$H(B) = \frac{H(A)}{\bar{n}_c} = -\sum_{i=1}^{6} \frac{p(a_i)\log p(a_i)}{\bar{n}_c} = \frac{2{,}2219}{2{,}3} = 0{,}966$ *бит* , что определяет наличие

избыточности в таком коде: $\rho_c = 1 - \frac{H(B)}{\log M} = 1 - \frac{0{,}966}{1} = 0{,}034$

Следовательно, символы двоичного кода не будут равновероятны:

$p(0) = \frac{1{,}35}{2{,}3} = 0{,}587$ *и* $p(1) = \frac{0{,}95}{2{,}3} = 0{,}413$.

Неоднозначность кодирования Хаффмана наглядно иллюстрируется на примере кодирования этого же источника, но с изменением порядка ранжирования промежуточных списков.

Вариант второй: в процессе ранжировки будем размещать объединенные в одно подмножество символы как можно выше - над равновероятными. И назовем этот вариант кодирования «над».

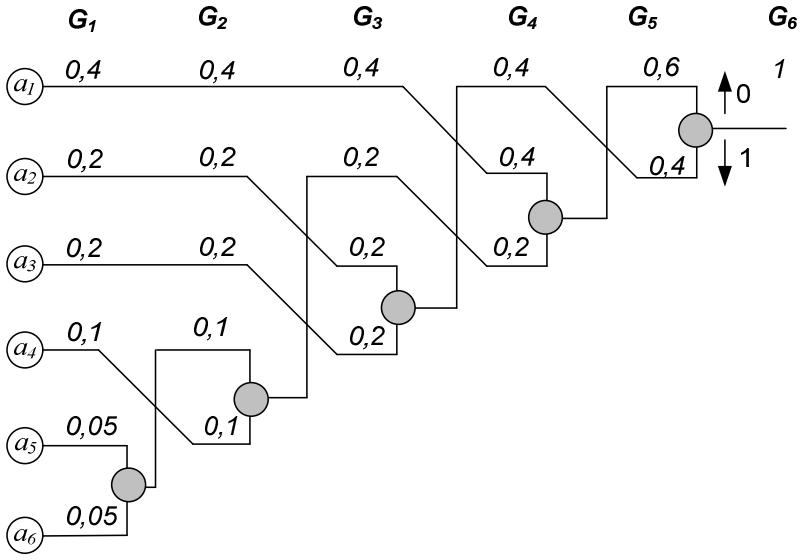


Рис.9.3. *Процедура кодирования Хаффмана по варианту «над».*

В результате получим следующий код Хаффмана для того же источника (процедура кодирования представлена на рис. 9.3):
a_1=00, a_2=10, a_3=11, a_4=011, a_5=0100, a_6=0101 с той же средней длиной кодовых слов $\overline{n}_c = 0{,}8 + 0.4 + 0{,}4 + 0{,}3 + 0{,}2 + 0{,}2 = 2{,}3$ *бит*, с тем же распределением двоичных символов и такой же избыточностью кода. Это еще раз подтверждает, что эффективность кодирования не зависит от процедуры кодирования, а полностью определяется только методом кодирования.

Но при практической реализации кодера Хаффмана полученные разные наборы длин кодовых слов требуют разных аппаратных затрат. Поэтому возникает вопрос, какому из этих двух кодов отдать предпочтение? Более разумным выбором [6] является тот, при котором длина кодовых слов меньше изменяется по ансамблю кодовых слов. Последнее характеризуется дисперсией длины кодовых слов для каждого из вариантов вышеприведенных процедур кодирования:

$$\sigma^2_{под}(n) = p(n_i)(n_i - \overline{n}_к)^2 = 0{,}4(1-2{,}3)^2 + 0{,}2(2-2{,}3)^2 + 0{,}2(3-2{,}3)^2 + 0{,}1(4-2{,}3)^2 + 2\cdot 0{,}05(5-2{,}3)^2 = 1{,}81;$$

$$\sigma^2_{над}(n) = 0{,}4(2-2{,}3)^2 + 2\cdot 0{,}2(2-2{,}3)^2 + 0{,}1(3-2{,}3)^2 + 2\cdot 0{,}05(4-2{,}3)^2 = 0{,}41.$$

Таким образом, для данного источника при использовании для кодирования сообщений конечной длины процедура «над» обеспечивает существенно меньшую дисперсию длин кодовых слов и поэтому является более предпочтительной при ее реализации.

Оба эти варианты кода Хаффмана будут одинаково эффективны при соблюдении условий *(1)* и *(2)* построения такого кода. Эффективность кодирования по алгоритму Хаффмана определяется только статистическими характеристиками кодируемого источника.

9.3. Построчное представление кодов Хаффмана

Примеры 9.1 и 9.2 иллюстрировались двумя разными графическими представлениями процедур кодирования, которые отличаются только компактностью. Построчное представление процедур [5] не требует графических иллюстраций. Но оно не менее наглядно отображает процедуры формирования пошаговых подмножеств кодируемых символов. Проиллюстрируем этот метод на примере кодирования сообщения *Двести кодеров и декодеров* с использованием модели источника (2.3). Естественно, что процедура кодирования остается пошаговая. Далее реализуется вариант кодирования «под».

Построчные процедуры формирования кодового дерева будем представлять в следующем виде:
- операция формирования родительского узла обозначим знаком +;
- дочерние узлы объединяются в родительский узел круглыми скобками;
- частоты родительских узлов равняются сумме частот дочерних узлов.

Шаг 1. Ранжировка символов источника в порядке не убывания их частот:
1. $е_4$, $о_4$, $в_3$, ⇔$_3$, $д_3$, $и_2$, $к_2$, $р_2$, $Д_1$, $с_1$, $т_1$

Шаг 2. Из списка выбираются два символа с наименьшими частотами *($с_1$* и *$т_1$)*, которые становятся листовыми узлами кодового дерева и объединяются в один родительский

узел с суммарной частотой - $(с_1+ т_1)_2$. Сформированный узел добавляется к новому списку:

2. $е_4, о_4, в_3, \Leftrightarrow_3, д_3, и_2, к_2, р_2, (с_1+ т_1)_2, Д_1$

Шаг 3. Формируется новый родительский узел из дочерних узлов списка с оставшимися наименьшими частотами $((с_1+ т_1)_2+, Д_1)_3$ и добавляется к новому списку

3. $е_4, о_4, в_3, \Leftrightarrow_3, д_3, ((с_1+ т_1)_2+, Д_1)_3, и_2, к_2, р_2$

Шаги 2-3 повторяются до тех пор, пока на шаге 11 в списке не останется единственный узел, называемый корнем дерева Хаффмана с суммарной частотой $\sum_{i=1}^{11} F_i = 26$.

4. $е_4, о_4, (к_2+ р_2)_4, в_3, \Leftrightarrow_3, д_3, ((с_1+ т_1)_2+ Д_1)_3, и_2$

5. $(((с_1+ т_1)_2+ Д_1)_3+ и_2)_5, е_4, о_4, (к_2+ р_2)_4, в_3, \Leftrightarrow_3, д_3$

6. $(\Leftrightarrow_3 + д_3)_6, (((с_1+ т_1)_2+, Д_1)_3+ и_2)_5, е_4, о_4, (к_2+ р_2)_4, в_3$

7. $((к_2+ р_2)_4 +в_3)_7, (\Leftrightarrow_3 +д_3)_6, (((с_1+ т_1)_2+ Д_1)_3+ и_2)_5, е_4, о_4$

8. $(е_4+ о_4)_8, ((к_2+ р_2)_4 +в_3)_7, (\Leftrightarrow_3 +д_3)_6, (((с_1+ т_1)_2+ Д_1)_3+ и_2)_5$

9. $((\Leftrightarrow_3 +д_3)_6+ (((с_1+ т_1)_2+ Д_1)_3+ и_2)_5)_{11}, (е_4+ о_4)_8, ((к_2+ р_2)_4 +в_3)_7$

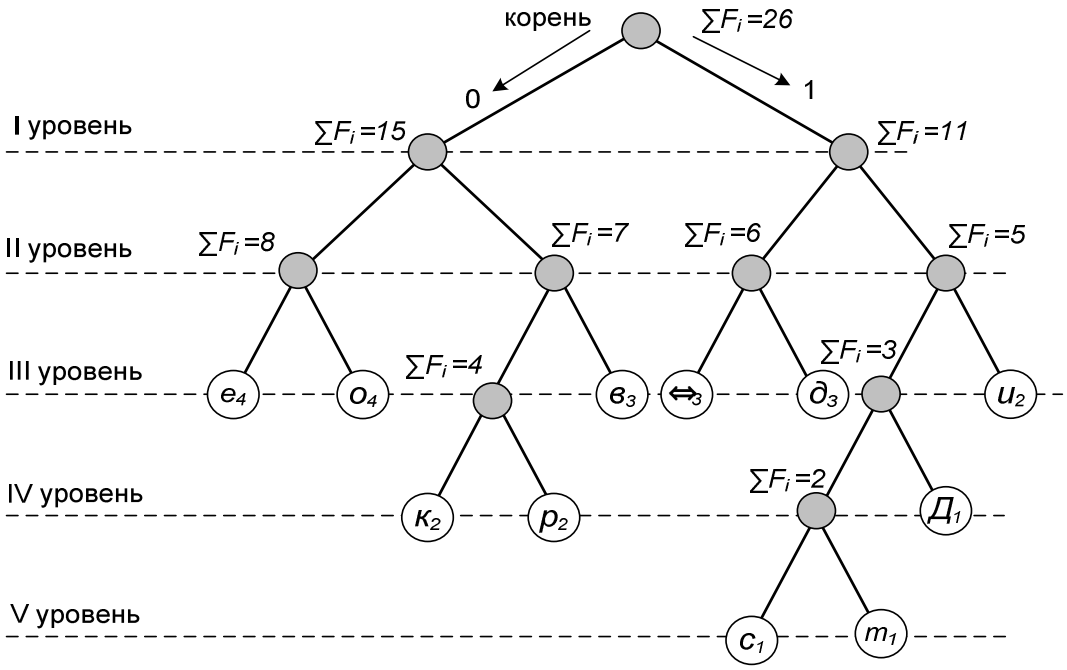
10. $((е_4+ о_4)_8+ ((к_2+ р_2)_4 +в_3)_7)_{15}, ((\Leftrightarrow_3 +д_3)_6+ (((с_1+ т_1)_2+ Д_1)_3+ и_2)_5)_{11}$

11.$(((е_4+ о_4)_8+ ((к_2+ р_2)_4 +в_3)_7)_{15} + ((\Leftrightarrow_3 +д_3)_6+ (((с_1+ т_1)_2+ Д_1)_3+ и_2)_5)_{11})_{26}$

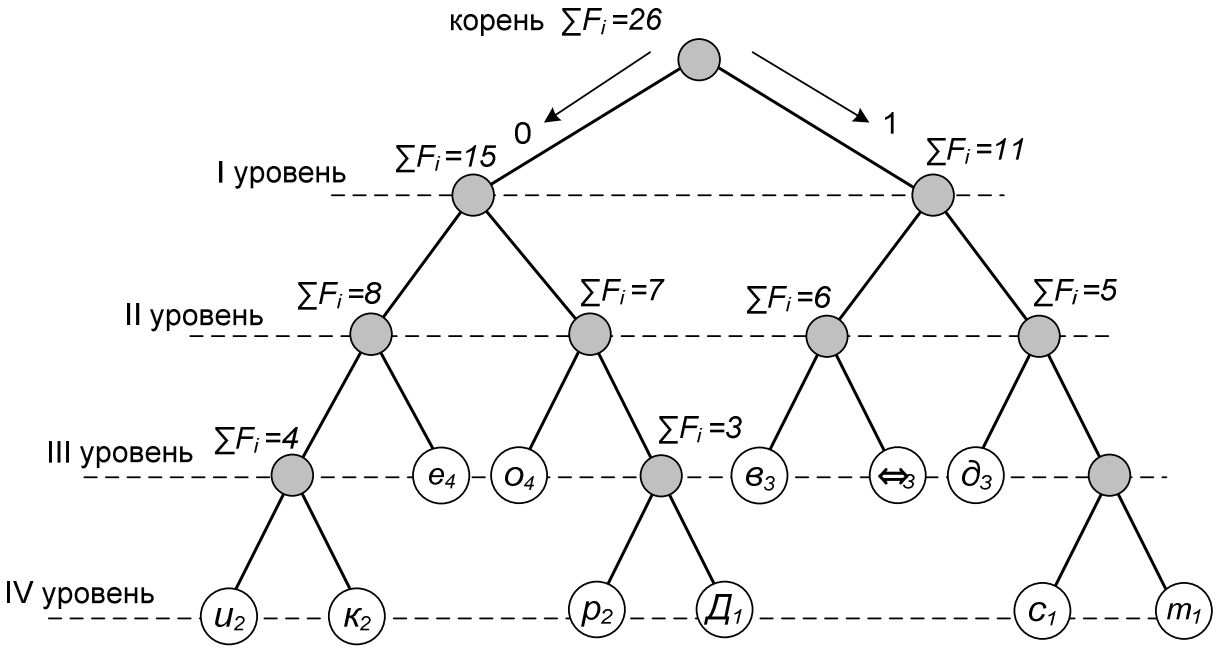
По последнему списку кодируемых символов прослеживаются следующие закономерности построения соответствующему ему кодового дерева:

- количество знаков + в списке соответствует числу родительских узлов кодового дерева включая корень: в общем случае это L-1;
- общее количество скобок (как открывающих так и закрывающих) соответствует общему количеству ребер кодового дерева: в общем случае это 2(L-1);
- для любого символа в списке (т.е. листика кодового дерева) количество закрывающих скобок с большими частотами определяет число ребер, соединяющих его с корнем кодового дерева, т.е., определяет уровень узла этого символа на кодовом дереве. Представление в графическом виде последнего списка кодируемых символов является кодовым деревом формируемого кода. Для этого необходимо выполнить операции обратные пошаговому составления последнего списка (*11*) кодируемых символов: идти от суммарной частоты в меньшим по их ранжировке. При этом необходимо соблюдать принятое условие: большие частоты дают направление ребра влево (в кодовом слове это означает значение «0» в разряде), в меньшие частоты – вправо («1» в данном разряде кодового слова). Сформированное таким образом кодовое дерево для сообщения *Двести кодеров и декодеров* представлено на рис. 9.4.а.

Процедуры кодирования по варианту «над» аналогичны выше рассмотренному варианту «под». Единственное отличие состоит только в том, что каждый вновь образованный родительский узел должен размещаться в новой строчке впереди символа или родительского узла с такой же абсолютной частотой. Соответствующее этому варианту кодирование того же сообщения кодовое дерево приведено на рис. 9.4 (б).



а)



б)

Рис. 9.4. *Дерево кода Хаффмана по варианту кодирования «под» (а) и «над» (б)*

\

Таблица 9.1. Варианты кодов Хаффмана

a_i	Кодовое слово вариант «под»	Кодовое слово вариант «над»	Канонический код Хаффмана
$е_4$	000	001	010
$о_4$	001	010	011
$в_3$	011	100	100
$\Leftrightarrow_3$	100	101	101
$д_3$	101	110	110
$и_2$	111	0000	111
$к_2$	0100	0001	0001
$р_2$	0101	0110	0010
$Д_1$	1101	0111	0011
$с_1$	11000	1110	00000
$т_1$	11001	1111	00001

Табл. 9.1 является таблицей соответствия для кода, где во второй и третьей колонках представлены результаты кодирования данного сообщения в вариантах «под» и «над».

9.3.1.Характеристики сформированного кода Хаффмана по варианту «под»:

1. Средняя длина кодовых слов

   $\overline{\boldsymbol{n}}_c = \sum_{i=1}^{L} \boldsymbol{f}_i \boldsymbol{n}_i$ =2·4/26·3+3·3/26·3+2/26·3+2·2/26·4+1/26·4+2·1/26·5=

   $\frac{\mathbf{87}}{\mathbf{26}} = \mathbf{3}\frac{\mathbf{3}}{\mathbf{26}}$ бит.

2. Длина строчки записи сообщения составляет

   $\boldsymbol{l}_{стр} = \overline{\boldsymbol{n}}_c \cdot \boldsymbol{N} = \frac{\mathbf{87}}{\mathbf{26}} \cdot \mathbf{26} = \mathbf{87}$ бит.

3. Энтропия кодера с учетом (3.6) при этом составляет:

   $H(B) = \frac{H(A)}{\overline{n}_к \log M} = \frac{3{,}3055}{3\,3/26} = 0{,}9878$ бит.

Полученному значению *H(B)* можно дать физическое толкование – он определяет коэффициент сжатия сообщения = 0,9878. Кроме того, коэффициент сжатия для данного кода можно определить и по соотношению строчек записи сообщения – теоретически, определяемой *I(A)*=85,935 бит и при кодировании по алгоритму Хаффмана – 87бит: $\frac{85{,}935}{87} = 0{,}9878$.

4. Вероятности кодовых символов в полученном коде:

   $p(0) = \frac{48}{87}; \qquad p(1) = \frac{39}{87}.$

5. Избыточность кода

$$\rho_c = 1 - \frac{H(B)}{\log M} = 1 - 0{,}9878 = 0{,}0122$$ или 1,22%

6. Производительность кодера $H'(B) = \nu_к H(B) = 0{,}9878\,\nu_к$ бит/с
7. Эффективность использования канала $\eta_к$ = 0,9878.

9.3.2. Характеристики кода Хаффмана по варианту «над»

Кодовое дерево Хаффмана для этого варианта кодирования представлено на рис. 9.3.б, а соответствующие кодовые слова приведены во второй колонке таблицы соответствия (табл. 9.1). Как видно, сами кодовые слова символов сообщения отличаются от результатов кодирования по варианту «под». Но статистические характеристики сформированного кода (кроме статистики 0 и 1) при этом не изменились. Читателю предлагается в этом убедиться самостоятельно.

10. Канонический код Хаффмана [5]

Код Хаффмана $B = \{b_1, b_2, ..., b_L\}$ *называется каноническим, если:*

1. *Короткие кодовые слова которого (при дополнении их нулями справа) имеют численное значение (т.е. в десятичной системе) больше длинных;*
2. *Коды одинаковой длины численно возрастают при увеличении объема алфавита.*

Канонический код может быть построен на основе любого варианта кодирования Хаффмана при выполнении двух условий: код должен оставаться префиксным и длины кодовых слов не должны меняться. При этом канонический код обладает важным свойством (его называют числовым свойством): *порядковый номер любого листового узла на занимаемом им уровне численно (в десятичной системе) равен двоичному коду соответствующего ему символа.*

В качестве примера трансформируем код Хаффмана, рассматриваемый в 9.1 (вариант «под»), в канонический.

Исходя из условия, что длины кодовых слов не должны меняться, число листовых узлов по всем пяти уровням не должно меняться. Так на уровне V кодового дерева их было 2. При этом все они, естественно, листовые с десятичными номерами 0 и 1. Исходя из основного свойства канонического кода этим узлам должны соответствовать следующие кодовые слова (двоичного кода):

$с \rightarrow 0_{dec} = 00000_{bin}$, $т \rightarrow 00001$.

На уровне IV имеется один родительский узел (номер 0) и три листовых. Поэтому первый листовой узел должен иметь порядковый номер 1, второй -2, третий -3. Т.е., их канонические коды этого уровня будет:

$к \rightarrow 0001$, $р \rightarrow 0010$, $Д \rightarrow 0011$.

На уровне III после двух родительских узлов должны размещаться листовые узлы следующих символов:

$е \rightarrow 010$, $о \rightarrow 011$, $в \rightarrow 100$, $\Leftrightarrow \rightarrow 101$, $д \rightarrow 110$, $и \rightarrow 111$.

В кодовом дереве Хаффмана на уровне II и I листовые узлы отсутствуют. Поэтому их не может быть и в каноническом коде.

Полученный канонический код представлен в табл. соответствия 9.3 и графически – на рис.10.1.

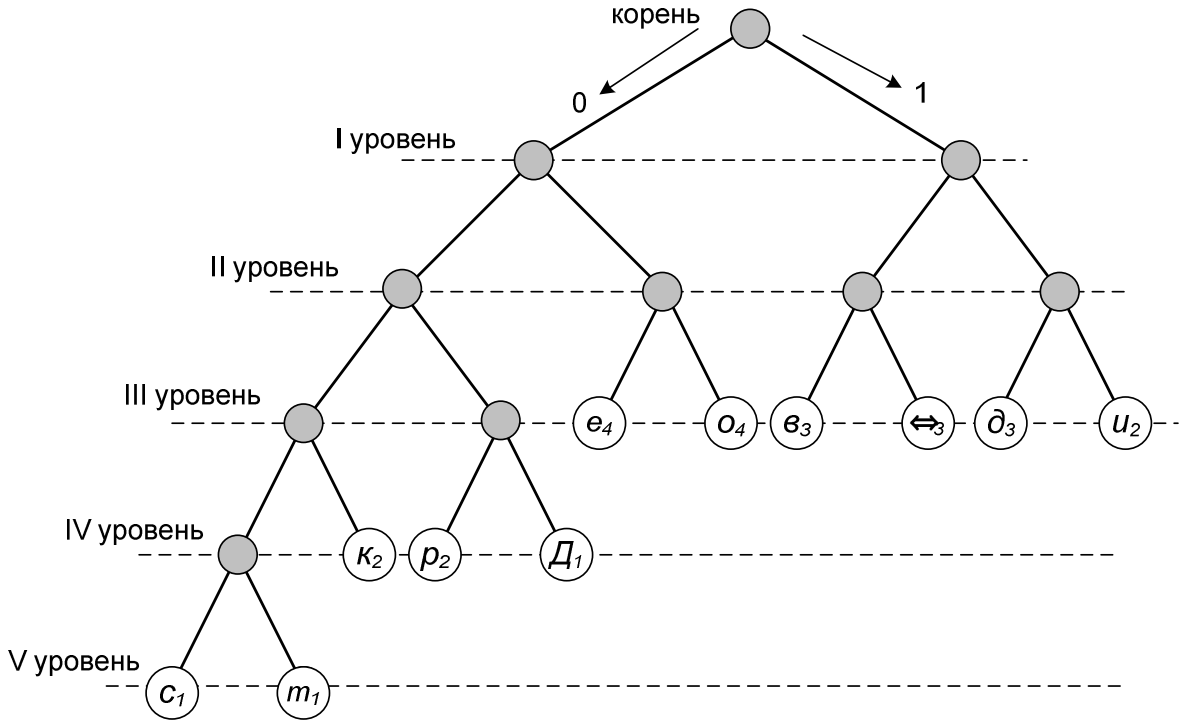


Рис.10.1. *Каноническое дерево Хаффмана*

Так как кодовые слова кода соответствуют только листовым узлам, то этим выполняется свойство префиксности кода. Легко убедиться, что и другие свойства канонического кода выполняются. Так, например, если дополнить символ $e \rightarrow 010$ справа до максимальной длины одним нулем, то его численное значение (в десятичной системе счисления) будет больше любого из значений кодов на IV уровне, т.к. $0100 > 0011$ ($4_{\text{dec}} > 3_{\text{dec}}$). Так же выполняется числовое свойство для кодов любого уровня (например, $a_1 \rightarrow 0100 > a_4 \rightarrow 001$). Все это позволяет сделать важный для практического применения этих кодов в алгоритмах кодирования источника (а в технических системах при сжатии информации) вывод: *канонические коды определяются своими длинами.* Так, например, кодовое слово 00011 должно соответствовать третьему листовому узлу на пятом уровне кодового дерева.

Характеристики канонического кода остаются такими же как и у классического кода Хаффмана, как алгоритм кодирования остается прежним. Но так как сами кодовые слова несколько другие, то меняется статистика 0 и 1:

$$p(0) = \frac{46}{87}, \quad p(1) = \frac{41}{87}.$$

Канонические коды Хаффмана широко применяются в алгоритмах сжатия информации современных систем передачи информации.

11. Заключение

В заключении отчета необходимо в соответствии с пунктом 11 (см. требования к содержанию задания) привести сравнительный анализ полученных характеристик кодирования источника по всем реализованным алгоритмам кодирования и дать рекомендации по выбору эффективного кодирования заданного источника.

Литература:

Основная литература (из фондов библиотеки ИТС)

1. Р. Фано. *Передача информации. Статистическая теория связи*. М.: Мир, 1965. - 435 с.
2. В.П. Цымбал. *Теория информации и кодирования*. Киев, Выща школа, 1977. - 288 с.
3. Дмитриев В. И. *Прикладная теория информации*. М.: Высшая школа, 1989. - 320 с.
4. М. Вернер. *Основы кодирования*. Учебник для ВУЗов. М.: Техносфера, 2004. -288 с.
5. Симаков Александр. *Код Хаффмана.* Сыктывкарский Государственный Университет. http://www.webcenter.ru/~xander/HuffmanCode/huffcode.html

Дополнительная литература

6. Р. В. Хэмминг. *Теория кодирования и теория информации.* М.: Радио и связь, 1983. -176 с.
7. Духин А.А. *Теория информации.* М. : Гелиос АРВ, 2007. - 248с.
8. Колесник В.Д., Полтырев Г.Ш. *Курс теории информации* . М.: Наука, 1982. - 416с.
9. Костров Б.В. *Основы цифровой передачи и кодирования информации.* М.: ТехБук, 2007. - 192с.
10. Скальский А.М. *Толковый словарь основных терминов по теории передачи информации и основам кодирования. Рига :* Институт транспорта и связи, 2005. - 76с.
11. Игнатов В.А. *Теория информации и передачи сигналов*. М.: Совю радио, 1991. - 280 с.
12. Котоусов А.С. *Теория информации*. М.: Радио и связь, 2003. -152 с.
13. Кудряшов Б.Д. *Теория информации*. С-Пб.: Питер, 2009. -320 с.
14. *Теория информации и кодирование* (Самсонов Б.Б.,Плохов Е.М.,Филоненков А.И.,Кречет Т.В.) - Ростов : Феникс, 2002. – 288 с.

Приложение 1

Фрагмент таблицы значений двоичных логарифмов целых чисел

x	$log_2 x$	x	$log_2 x$	x	$log_2 x$
1	0,0000	**47**	5,5545	**93**	6,5391
2	1,0000	**48**	5,5800	**94**	6,5545
3	1,5850	**49**	5,6146	**95**	6,5698
4	2,0000	**50**	5,6438	**96**	6,5849
5	2,3219	**51**	5,6724	**97**	6,6000
6	2,5850	**52**	5,7004	**98**	6,6146
7	2,8074	**53**	5,7279	**99**	6,6293
8	3,0000	**54**	5,7548	**100**	6,6438
9	3,1699	**55**	5,7813	**101**	6,6582
10	3,3219	**56**	5,8073	**102**	6,6724
11	3,4594	**57**	5,8328	**103**	6,6864
12	3,5850	**58**	5,8579	**104**	6,7003
13	3,7004	**59**	5.8826	**105**	6,7142
14	3,8074	**60**	5,9068	**106**	6,7278
15	3,9069	**61**	5,9307	**107**	6,7414
16	4,0000	**62**	5,9541	**108**	6,7548
17	4,0875	**63**	5,9772	**109**	6,7681
18	4,1699	**64**	6,0000	**110**	6,7813
19	4,2479	**65**	6,0223	**111**	6,7944
20	4,3219	**66**	6,0443	**112**	6,8073
21	4,3923	**67**	6,0660	**113**	6,8201
22	4,4594	**68**	6,0874	**114**	6,8328
23	4,5236	**69**	6,1085	**115**	6,8454
24	4,5850	**70**	6,1292	**116**	6,8579
25	4,6438	**71**	6,1497	**117**	6,8703
26	4,7004	**72**	6,1699	**118**	6,8826
27	4,7548	**73**	6,1898	**119**	6,8948
28	4,8073	**74**	6,2094	**120**	6,9068
29	4,8579	**75**	6,2287	**121**	6,9188
30	4,9068	**76**	6,2479	**122**	6,9306
31	4,9541	**77**	6.2667	**123**	6,9424
32	5,0000	**78**	6,2853	**124**	6,9541
33	5,0443	**79**	6,3037	**125**	6,9657
34	5,0874	**80**	6,3219	**126**	6.9772
35	5,1292	**81**	6,3398	**127**	6,9886
36	5,1699	**82**	6,3575	**128**	7,0000
37	5,2094	**83**	6,3750	**200**	7,6438
38	5,2479	**84**	6,3922	**256**	8,0000
39	5,2853	**85**	6,4093	**300**	8,2287
40	5,3219	**86**	6,4262	**400**	8,6438
41	5,3575	**87**	6,4429	**500**	8,9657
42	5,3922	**88**	6,4594	**600**	9,2287
43	5,4262	**89**	6,4757	**700**	9,4511
44	5,4594	**90**	6,4918	**800**	9,6437
45	5,4918	**91**	6,5077	**900**	9,8136
46	5,5235	**92**	6,5235	**1000**	9,9657

Приложение 2

Фрагмент таблицы значений функции $-p(x)\log_2 p(x)$

p(x)	*-p(x)logp(x)*	*p(x)*	*-p(x)logp(x)*	*p(x)*	*-p(x)logp(x)*
0,001	0,0099	**0,055**	0,2301	**0,400**	0,5288
0,002	0,0179	**0,060**	0,2435	**0,405**	0,5281
0,003	0,0251	**0,065**	0,2563	**0,410**	0,5274
0,004	0,0319	**0,070**	0,2686	**0,420**	0,5256
0,005	0,0382	**0,075**	0,2803	**0,425**	0,5246
0,006	0,0443	**0,080**	0,2915	**0,430**	0,5236
0,007	0,0501	**0,085**	0,3023	**0,440**	0,5210
0,008	0,0557	**0,090**	0,3127	**0,450**	0,5184
0,009	0,0612	**0,095**	0,3226	**0,460**	0,5153
0,010	0,0664	**0,100**	0,3322	**0,470**	0,5119
0,011	0,0716	**0,110**	0,3503	**0,475**	0,5102
0,012	0,0766	**0,120**	0,3671	**0,480**	0,5083
0,013	0,0814	**0,125**	0,3750	**0,490**	0,5043
0,014	0,0862	**0,130**	0,3826	**0.500**	0,5000
0,015	0,0909	**0,140**	0,3971	**0,505**	0,4977
0,016	0,0954	**0,150**	0,4105	**0,510**	0,4954
0,017	0,0999	**0,160**	0,4230	**0,525**	0,4880
0,018	0,1043	**0,170**	0,4346	**0,550**	0,4744
0,019	0,1086	**0,175**	0,4400	**0,575**	0,4591
0,020	0,1129	**0,180**	0,4453	**0,600**	0.4432
0,021	0,1170	**0,190**	0,4552	**0,610**	0,4350
0,022	0,1211	**0,200**	0,4644	**0,625**	0,4238
0,023	0,1252	**0,210**	0,4728	**0,650**	0,4040
0,024	0,1291	**0,220**	0,4806	**0,675**	0,3828
0,025	0,1330	**0,225**	0,4842	**0,700**	0.3602
0,026	0,1369	**0,230**	0.4877	**0,710**	0,3508
0.027	0,1407	**0,240**	0,4941	**0,725**	0,3364
0,028	0,1444	**0,250**	0,5000	**0,750**	0,3113
0,029	0,1481	**0,260**	0,5053	**0,775**	0,2850
0,030	0,1518	**0,270**	0,5100	**0,800**	0,2575
0,031	0,1554	**0.275**	0,5122	**0,810**	0,2462
0,032	0,1589	**0.280**	0,5142	**0,825**	0,2290
0,033	0,1642	**0,290**	0,5179	**0,850**	0,1993
0,034	0,1658	**0,300**	0,5211	**0,875**	0,1810
0,035	0,1693	**0,310**	0,5238	**0,900**	0,1368
0,036	0,1727	**0,320**	0,5260	**0,910**	0.1238
0,037	0,1760	**0,325**	0,5270	**0,925**	0,1040
0,038	0,1793	**0,330**	0,5278	**0,950**	0,0703
0,039	0,1825	**0,340**	0,5292	**0,960**	0,0565
0,040	0,1858	**0,350**	0,5301	**0,975**	0,0356
0,041	0,1889	**0,360**	0,5306	**0,980**	0,0286
0,042	0,1941	**0,370**	0,5307	**0,990**	0,0143
0,045	0,2013	**0,375**	0,5306	**0,995**	0,0072
0,047	0,2073	**0,380**	0,5305	**0,997**	0,0043
0,050	0,2161	**0,390**	0,5298	**0,999**	0,0014

Приложение 3

Указатель используемых обозначений

$\boldsymbol{A}$ – ансамбль (множество) символов или состояний источника

a_i **–** символ алфавита источника

$\boldsymbol{B}$ **–** ансамбль двоичных символов кодера

b_j – выходной (внутренний) символ кодера

$C_к$ – пропускная способность канала

dec – запись в десятичной системе

F_i – абсолютная частота символов источника

f_i – относительная частота символов источника

G_j – подмножество символов

$H(A)$ – энтропия источника

$H_к(B)$ – энтропия кодера-

$H'(A)$ – производительность источника

η – эффективность

$I(A)$ – количество информации в сообщении

$I(a_i)$ **–** собственная информация символа сообщения

$K_{сж}$ – коэффициент сжатия информации

L – объем алфавита источника сообщений

$l_{стр}$ – длина кодовой строчки записи сообщения

M – объем алфавита кодера

$m_1\{x\}$ – математическое ожидание случайной величины x

n_c – длина кодового слова

$\bar{n}_c$ - средняя длина кодового слова

N – количество символов в сообщении источника

$p(a_i)$ – априорная вероятность символа a_i

$v_к$ - скорость канальная

T_0 – длительность кодового символа

ρ_u^p - относительная избыточность (коэффициент избыточности) источника, обусловленная не равновероятностью его состояний

ρ_c - избыточность кода

$\sigma^2(n)$ - дисперсия длин кодовых слов