

Тема 6. Кодирование Шеннона-Фано. Кодирование Хаффмена

Кодирование алгоритмом Шеннона-Фано производится следующим образом. Кодируемые знаки выписывают в таблицу в порядке убывания их вероятностей в сообщениях. Затем их разделяют на две группы так, чтобы значения сумм вероятностей в каждой группе были близкими. Все знаки одной из групп в соответствующем разряде кодируются, например, единицей, тогда знаки второй группы кодируются нулем. Каждую полученную в процессе деления группу подвергают вышеописанной операции до тех пор, пока в результате очередного деления в каждой группе не останется по одному знаку.

При использовании алгоритма Хаффмена, кодируемые знаки также располагают в порядке убывания их вероятностей. Далее на каждом этапе две последние позиции списка заменяются одной и ей приписывают вероятность, равную сумме вероятностей заменяемых позиций. После этого производится пересортировка списка по убыванию вероятностей, с сохранением информации о том, какие именно знаки объединялись на каждом этапе. Процесс продолжается до тех пор, пока не останется единственная позиция с вероятностью, равной 1.

После этого строится кодовое дерево. Корню дерева ставится в соответствие узел с вероятностью, равной 1. Далее каждому узлу приписываются два потомка с вероятностями, которые участвовали в формировании значения вероятности обрабатываемого узла. Так продолжают до достижения узлов, соответствующих вероятностям исходных знаков.

Процесс кодирования по кодовому дереву осуществляется следующим образом. Одной из ветвей, выходящей из каждого узла, например, с более высокой вероятностью, ставится в соответствие символ 1, а с меньшей – 0. Спуск от корня к нужному знаку дает код этого знака. Правило кодирования в случае равных вероятностей оговаривается особо.

Задача 6.1

Используя алгоритмы Шеннона-Фано и Хаффмена, провести эффективное кодирование ансамбля из восьми знаков z_i с заданными вероятностями (см. варианты).

Для построенного эффективного кода определить среднюю длину кодовой комбинации.

Варианты заданий:

$$1) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,26 & 0,24 & 0,15 & 0,12 & 0,09 & 0,07 & 0,05 & 0,02 \end{bmatrix}.$$

$$2) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,26 & 0,23 & 0,16 & 0,11 & 0,09 & 0,08 & 0,05 & 0,02 \end{bmatrix}.$$

$$3) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,21 & 0,19 & 0,15 & 0,13 & 0,12 & 0,09 & 0,06 & 0,05 \end{bmatrix}.$$

$$4) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,23 & 0,17 & 0,15 & 0,14 & 0,1 & 0,09 & 0,07 & 0,05 \end{bmatrix}.$$

$$5) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,26 & 0,19 & 0,14 & 0,11 & 0,1 & 0,08 & 0,07 & 0,05 \end{bmatrix}.$$

$$6) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,26 & 0,22 & 0,16 & 0,12 & 0,09 & 0,07 & 0,05 & 0,03 \end{bmatrix}.$$

$$7) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,2 & 0,18 & 0,14 & 0,12 & 0,12 & 0,1 & 0,08 & 0,06 \end{bmatrix}.$$

$$8) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,28 & 0,22 & 0,15 & 0,11 & 0,11 & 0,07 & 0,04 & 0,02 \end{bmatrix}.$$

$$9) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,22 & 0,18 & 0,15 & 0,13 & 0,13 & 0,09 & 0,05 & 0,05 \end{bmatrix}.$$

$$10) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,25 & 0,19 & 0,15 & 0,11 & 0,09 & 0,09 & 0,07 & 0,05 \end{bmatrix}.$$

$$11) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,24 & 0,17 & 0,16 & 0,14 & 0,09 & 0,08 & 0,07 & 0,05 \end{bmatrix}.$$

$$12) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,24 & 0,23 & 0,16 & 0,14 & 0,08 & 0,06 & 0,06 & 0,03 \end{bmatrix}.$$

$$13) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,21 & 0,17 & 0,14 & 0,13 & 0,12 & 0,1 & 0,07 & 0,06 \end{bmatrix}.$$

$$14) Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,22 & 0,18 & 0,16 & 0,12 & 0,12 & 0,1 & 0,05 & 0,05 \end{bmatrix}.$$

Решение задачи 6.2 (вариант 3)

$$Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ 0,26 & 0,24 & 0,15 & 0,12 & 0,09 & 0,07 & 0,05 & 0,02 \end{bmatrix}.$$

Выполним кодирование этого ансамбля по алгоритму Шеннона-Фано. Продемонстрируем первые три шага алгоритма после сортировки ансамбля по убыванию вероятностей:

Таблица 2

z_1	0,26		z_1	0,26	0	z_1	0,26	00	z_1	0,26	00
z_2	0,24		z_2	0,24	0	z_2	0,24	01	z_2	0,24	01
z_3	0,15		z_3	0,15	1	z_3	0,15	1	z_3	0,15	10
z_4	0,12		z_4	0,12	1	z_4	0,12	1	z_4	0,12	10
z_5	0,09		z_5	0,09	1	z_5	0,09	1	z_5	0,09	11
z_6	0,07		z_6	0,07	1	z_6	0,07	1	z_6	0,07	11
z_7	0,05		z_7	0,05	1	z_7	0,05	1	z_7	0,05	11
z_8	0,02		z_8	0,02	1	z_8	0,02	1	z_8	0,02	11

Выполняем эту процедуру до тех пор, пока в результате очередного деления в каждой группе не останется по одному знаку.

Таблица 3

z_1	0,26	00
z_2	0,24	01
z_3	0,15	100
z_4	0,12	101
z_5	0,09	110
z_6	0,07	1110
z_7	0,05	11110
z_8	0,02	11111

Средняя длина кодовой комбинации получившегося кода:

$$l_{aver} = 0,26 \cdot 2 + 0,24 \cdot 2 + 0,15 \cdot 3 + 0,12 \cdot 3 + \\ + 0,09 \cdot 3 + 0,07 \cdot 4 + 0,05 \cdot 5 + 0,02 \cdot 5 = 2.21.$$

Выполним кодирование этого ансамбля по алгоритму Хаффмена. Заполним таблицу после сортировки ансамбля по убыванию вероятностей. Будем отмечать новое получившееся значение вероятности (сумму) на каждом шаге подчёркиванием и полужирным шрифтом.

Таблица 4

z_1	0,26	0,26	0,26	0,26	<u>0,29</u>	<u>0,45</u>	<u>0,55</u>	<u>1</u>
z_2	0,24	0,24	0,24	0,24	0,26	0,29	0,45	
z_3	0,15	0,15	0,15	<u>0,21</u>	0,24	0,26		
z_4	0,12	0,12	<u>0,14</u>	0,15	0,21			
z_5	0,09	0,09	0,12	0,14				
z_6	0,07	0,07	0,09					
z_7	0,05	<u>0,07</u>						
z_8	0,02							

Построим дерево кодирования на основе полученной таблицы:

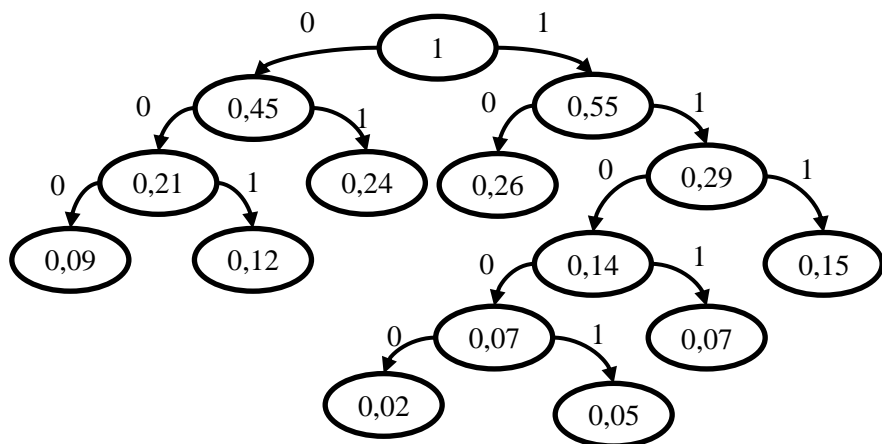


Рис. 2

На основе построенного дерева формируем коды:

Таблица 5

z_1	0,26	10
z_2	0,24	01
z_3	0,15	111
z_4	0,12	001
z_5	0,09	000
z_6	0,07	1101
z_7	0,05	11001
z_8	0,02	11000

Средняя длина кодовой комбинации получившегося кода:

$$l_{aver} = 0,26 \cdot 2 + 0,24 \cdot 2 + 0,15 \cdot 3 + 0,12 \cdot 3 + \\ + 0,09 \cdot 3 + 0,07 \cdot 4 + 0,05 \cdot 5 + 0,02 \cdot 5 = 2,21.$$

Задача 6.3

При помощи алгоритма Хаффмена построить эффективный код для ансамбля (см. варианты) с использованием блоков по два или три знака (см. варианты).

Вычислить среднюю длину кодовой комбинации.

Варианты заданий:

1) $Z = \begin{bmatrix} z_1 & z_2 \\ 0,9 & 0,1 \end{bmatrix}$, блоками по три знака.

2) $Z = \begin{bmatrix} z_1 & z_2 \\ 0,8 & 0,2 \end{bmatrix}$, блоками по три знака.

3) $Z = \begin{bmatrix} z_1 & z_2 \\ 0,7 & 0,3 \end{bmatrix}$, блоками по три знака.

4) $Z = \begin{bmatrix} z_1 & z_2 \\ 0,6 & 0,4 \end{bmatrix}$, блоками по три знака.

5) $Z = \begin{bmatrix} z_1 & z_2 & z_3 \\ 0,7 & 0,2 & 0,1 \end{bmatrix}$, блоками по три знака.

6) $Z = \begin{bmatrix} z_1 & z_2 & z_3 \\ 0,6 & 0,3 & 0,1 \end{bmatrix}$, блоками по три знака.

7) $Z = \begin{bmatrix} z_1 & z_2 & z_3 \\ 0,5 & 0,2 & 0,3 \end{bmatrix}$, блоками по три знака.

Арифметическое кодирование.

Задача 5.1: Выполните сжатие фразы (табл. 5.1) с помощью метода арифметического кодирования. Представьте полученный результат в виде двоичной дроби. Рассчитайте количество информации, передаваемой при равномерном кодировании кодом *ASCII* (8 бит на символ) и арифметическом кодировании. Сравните полученные значения.

Таблица 5.1

Вариант	Фраза
1	carry
2	holla
3	apple
4	dandy
5	comma
6	malty
7	moody
8	salty
9	funny
10	patty

Решение:

Арифметическое сжатие представляет собой очень интересный метод, в котором весь передаваемый текст представляется в виде дроби. Для удобства кодирования будем рассматривать построение дроби на интервале $[0, 1)$, где 0 – включается в интервал, а 1 – нет. Весь интервал или отрезок разбивается на подынтервалы с длинами, равными вероятностям появления символов текста.

Для примера рассмотрим сжатие текста «КАТАКАНА». Рассчитаем вероятности появления каждого символа в этом тексте (табл. 5.2).

Таблица 5.2

Вероятности появления символов и границы подынтервалов каждого символа

Символ	Частота	Вероятность	Диапазон
А	4	0.5	$[0.0, 0.5)$
К	2	0.25	$[0.5, 0.75)$
Т	1	0.125	$[0.75, 0.875)$
Н	1	0.125	$[0.875, 1)$

Процесс кодирования заключается в постепенном уменьшении длительности интервала. При кодировании первого символа в качестве рабочего интервала выбирается отрезок $[0, 1)$. В дальнейшем он разбивается

на подынтервалы в соответствии с вероятностями появления символов текста, а в качестве следующего рабочего интервала берется диапазон, соответствующий текущему кодируемому символу «К», в данном случае это – $[0.5, 0.75)$. Длина данного интервала пропорциональна вероятности появления символа «К».

После этого та же самая процедура повторяется для второго символа – «А». За начальный интервал в данном случае принимается промежуток $[0.5, 0.75)$, полученный на предыдущем шаге. И вновь данный промежуток разбивается на подынтервалы в соответствии с вероятностями появления символов и вновь выбирается участок, соответствующий очередной передаваемой букве – «А». В данном случае границы нового интервала приведены на рис. 5.1.

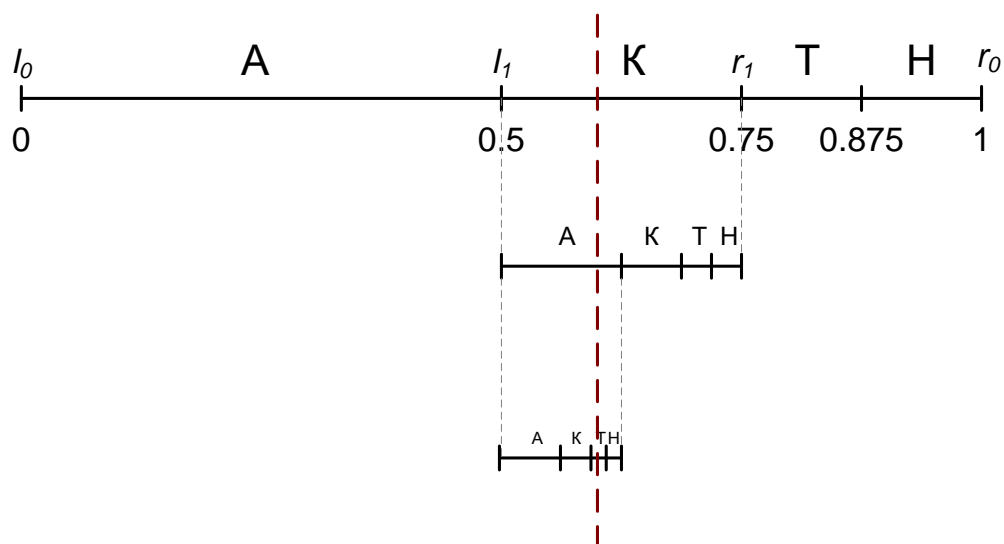


Рис. 5.1. Принцип арифметического сжатия на примере интервала $[0, 1)$

Таким образом, окончательная длина промежутка будет равна произведению вероятностей всех встретившихся символов, а его начало зависит от порядка следования символов в потоке. Алгоритм арифметического сжатия приведен на рис. 5.2.

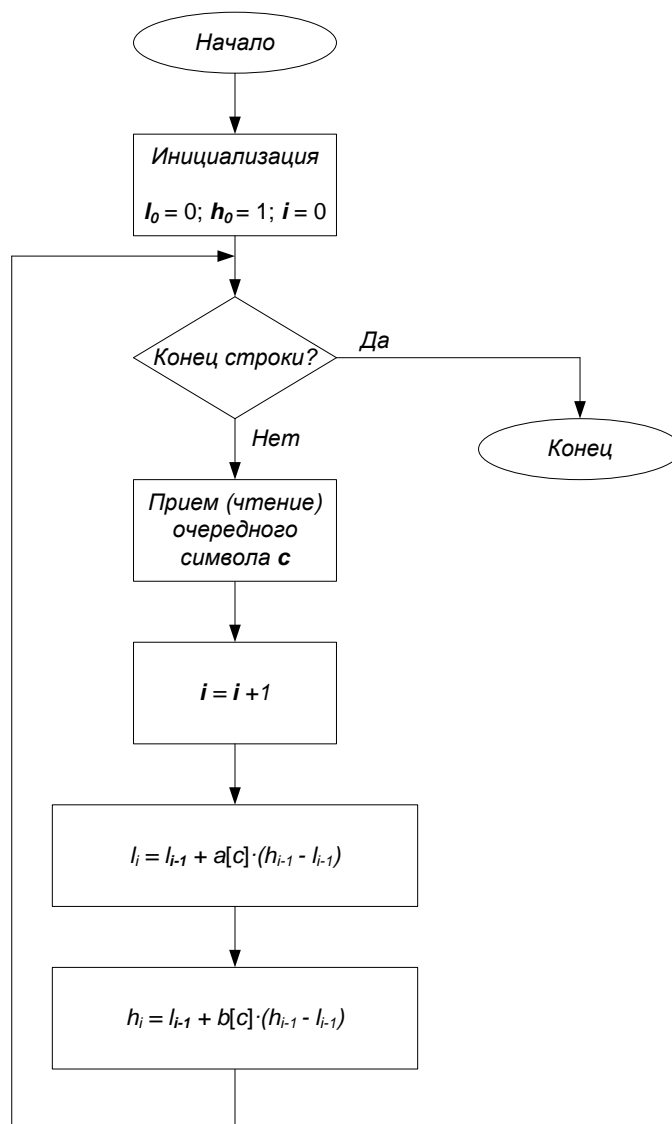


Рис. 5.2. Алгоритм арифметического сжатия

Поскольку конечным результатом работы алгоритма является пара чисел – левая и правая граница интервала, то для восстановления текста на противоположную сторону достаточно будет передать лишь одно число, лежащее в данном интервале. Для нашего случая достаточно будет передать число 0,234.

Процесс декодирования выглядит достаточно просто – по указанному числу мы можем сразу же определить первый символ текста, затем, разбив выбранный интервал на подынтервалы – второй, затем аналогично третий и т.д.

Заметим также, что передача очень длинного дробного числа может оказаться неэффективной, если будет использоваться обычная форма записи числа с плавающей точкой. Гораздо более компактной формы записи дробного числа можно добиться с помощью **двоичной дроби** вида $0.a_1a_2a_3...a_i = a_1 \cdot 1/2 + a_2 \cdot 1/4 + a_3 \cdot 1/8 + ... + a_i \cdot 1/2^i$. В этом случае при сжатии требуется дописывать в дробь дополнительные знаки (1 или 0) до тех пор, пока получившееся число не попадет в требуемый интервал. Для примера,

число 0.625 можно представить двоичной дробью 0.101 (а передать нужно лишь 101).