

КОДИРОВАНИЕ СООБЩЕНИЙ,
КОДЫ ФАНО, ШЕННОНА И ХАФФМАНА

Под кодированием понимают преобразование алфавита источника сообщений $\mathbf{A}=\{a_i\}$, ($i = 1,2,\dots,K$) в алфавит некоторых кодовых символов $\mathbf{R}=\{x_j\}$, ($j = 1,2,\dots,N$).

Обычно размер алфавита кодовых символов $|\mathbf{R}|$ существенно меньше размера алфавита источника $|\mathbf{A}|$.

Последовательность кодовых символов, описывающих сообщение источника, называется *кодовым словом*.

Совокупность правил, в соответствии с которыми производятся операции преобразования алфавита источника сообщений в кодовые слова, называют *кодом*.

Буквы

Вероятность

A	0,3	1	1						11
B	0,2	1	2						10
C	0,1	2	1	1					011
D	0,1	2	1	2	1				0101
E	0,05	2	1	2	2				0100
F	0,03	2	2	1	1	1			00111
G	0,03	2	2	1	1	2			00110
H	0,03	2	2	1	2	1			00101
I	0,03	2	2	1	2	2			00100
J	0,03	2	2	2	1	1			00011
K	0,02	2	2	2	1	2	1		000101
L	0,02	2	2	2	1	2	2		000100
M	0,01	2	2	2	2	1	1		000011
N	0,01	2	2	2	2	1	2	1	0000101
O	0,01	2	2	2	2	1	2	2	0000100
P	0,01	2	2	2	2	2	1		000001
Q	0,01	2	2	2	2	2	2	1	0000001
R	0,01	2	2	2	2	2	2	2	0000000

Кодирование сообщений может преследовать различные цели.

Одна из них - представить сообщение в такой системе символов, которая обеспечивала бы **простоту и надежность аппаратной реализации** устройств и их эффективность.

Другая цель кодирования - обеспечить наилучшее согласование свойств источника сообщений со свойствами канала связи. При этом добиваются **выигрыша во времени передачи**

Еще одной целью кодирования является **засекречивание передаваемой информации**. При этом элементарным сообщениям

$$a_{k1} a_{k2} \dots a_{kn}$$

из алфавита Λ ставятся в соответствие последовательности, к примеру, цифр или букв из специальных кодовых таблиц, известных лишь отправителю и получателю информации.

Примером кодирования может служить преобразование дискретных сообщений $a_{k1} a_{k2} \dots a_{kn}$ из одних систем счисления в другие.

Кодирование в канале, или помехоустойчивое кодирование информации, используется для уменьшения количества ошибок, возникающих при передаче по каналу с помехами.

По условиям построения кодовых слов коды делятся на равномерные и неравномерные. Если кодовые слова имеют разную длину, то код называется *неравномерным*. Типичным представителем этой группы является код Морзе.

В *равномерных кодах* все сообщения передаются кодовыми словами с одинаковым числом элементов. Примером является телеграфный код с длиной кодового слова равной 5.

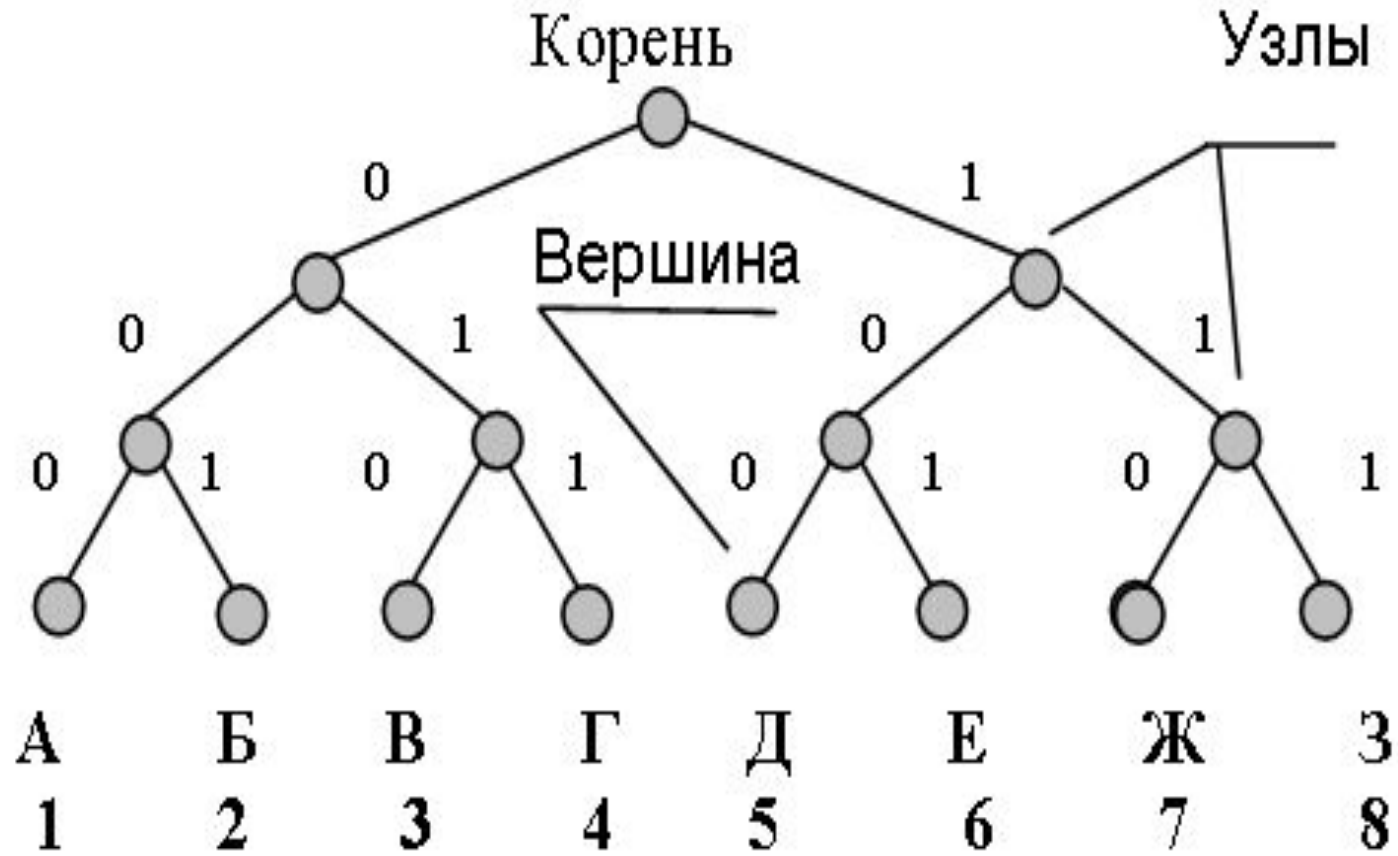
По размеру алфавита кодовых символов различают следующие виды кодов: единичные, двоичные, многопозиционные.

Наибольшее распространение получили двоичные коды, или коды с основанием 2, для которых размер алфавита кодовых символов $R=\{0,1\}$ равен двум

Самым простым способом задания кодов являются кодовые таблицы, ставящие в соответствие сообщениям a_i соответствующие им коды

Буква a_i	Число a_i	Код с основанием 10	Код с основанием 2
А	0	0	000
Б	1	1	001
В	2	2	010
Г	3	3	011
Д	4	4	100
Е	5	5	101
Ж	6	6	110
З	7	7	111

Другим наглядным способом описания кодов является их представление в виде кодового дерева



Для того, чтобы построить кодовое дерево для данного кода, начиная с некоторой точки - корня кодового дерева - проводятся ветви - 0 или 1. На вершинах кодового дерева находятся буквы алфавита источника, причем каждой букве соответствуют свой лист и свой путь от корня к листу. К примеру, букве А соответствует код 000, букве В – 010, букве Е – 101 и т.д.

Код, полученный с использованием кодового дерева, изображенного на рис. 1, является равномерным трехразрядным кодом.

Равномерные коды очень широко используются в силу своей простоты и удобства процедур кодирования-декодирования: каждой букве – одинаковое число бит; при получении заданного числа бит – в кодовой таблице ищется соответствующая буква

Если исходный алфавит содержит K символов, то для построения равномерного кода с использованием N кодовых символов необходимо обеспечить выполнение следующего условия:

$$K \leq N^q,$$

или

$$q \geq \frac{\log K}{\log N},$$

где q – количество элементов кодовой последовательности.

Например, при двоичном кодировании 32-х букв русского алфавита используется $q = \log_2 32 = 5$ разрядов, на чем и основывается телетайпный код.

Очевидно, что при различной вероятности появления букв исходного алфавита **равномерный код является избыточным**, так как энтропия, характеризующая информационную емкость сообщения максимальна при равновероятных буквах исходного текста.

Например, для телетайпного кода $N_0 = 5$ бит, а с учетом неравномерности появления различных букв исходного алфавита $H \approx 4,35$ бит. Устранение избыточности достигается применением **неравномерных кодов**, в которых буквы, имеющие наибольшую вероятность, кодируются наиболее короткими кодовыми последовательностями, а более длинные комбинации присваиваются редким буквам.

Если i -я буква, вероятность которой P_i , получает кодовую комбинацию длины q_i , то средняя длина комбинации

$$q_{cp} = \sum_{i=1}^K P_i q_i$$

Считая кодовые буквы равномерными, определяем наибольшую энтропию закодированного алфавита как $q_{cp} \log K$, которая не может быть меньше энтропии исходного алфавита H , т.е.

$$q_{cp} \log K \geq H.$$

Отсюда имеем $q_{cp} \geq (H / \log K)$.

Обычно $K=2$ и $q_{cp} \geq H$!!

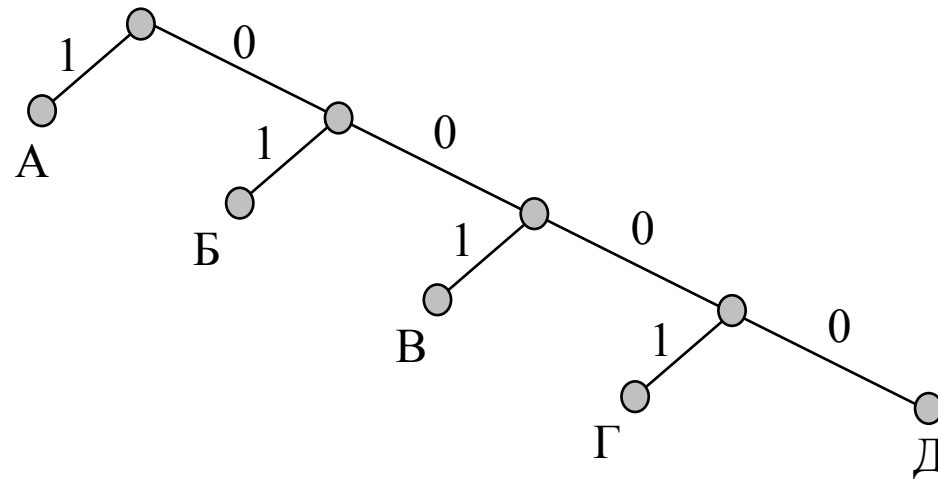
Чем ближе значение q_{cp} к энтропии H , тем более эффективно кодирование. В идеальном случае, когда $q_{cp} \approx H$, код называют *эффективным*.

Эффективное кодирование устраняет избыточность, приводит к сокращению длины сообщений, а значит, позволяет уменьшить время передачи сообщений или объем памяти, необходимой для их хранения.

При построении неравномерных кодов необходимо обеспечить возможность их однозначной расшифровки. В равномерных кодах такая проблема не возникает, т.к. при расшифровке достаточно кодовую последовательность разделить на группы, каждая из которых состоит из q элементов. В неравномерных кодах можно использовать разделительный символ между буквами алфавита (так поступают, например, при передаче сообщений с помощью азбуки Морзе).

Если же отказаться от разделительных символов, то следует запретить такие кодовые комбинации, начальные части которых уже использованы в качестве самостоятельной комбинации. Например, если 101 означает код какой-то буквы, то нельзя использовать комбинации 1, 10 или 10101

Для построения неравномерных кодов, допускающих однозначное декодирование необходимо, чтобы всем буквам алфавита соответствовали лишь листья кодового дерева (Рисунок). Здесь ни одна кодовая комбинация не является началом другой, более длинной, поэтому неоднозначности декодирования не будет. Такие неравномерные коды называются *префиксными*.



Прием и декодирование неравномерных кодов - процедура гораздо более сложная, чем для равномерных. Возникает вопрос, **зачем же используются неравномерные коды?**

Рассмотрим пример кодирования сообщений a_i из алфавита объемом $K = 8$ с помощью произвольного q -разрядного двоичного кода.

Пусть источник сообщений выдает некоторый текст с алфавитом от А до З и одинаковой вероятностью букв $P(a_i) = 1/8$.

Кодирующее устройство кодирует эти буквы равномерным трехразрядным кодом (см. таблицу).

Определим основные информационные характеристики источника с таким алфавитом:

-энтропия источника

$$H = -\sum_{i=1}^K P_i \log P_i \quad H_0 = \log K = 3$$

-избыточность источника

$$r_{И} = \frac{H_0 - H}{H_0} = 0$$

-среднее число символов в коде

$$q_{cp} = \sum_{i=1}^K q_i \cdot P_i = \sum_{i=1}^8 3 \cdot \frac{1}{8} = 3$$

- избыточность кода

$$r_k = \frac{q_{cp} - H}{q_{cp}} = 0$$

Таким образом, при кодировании сообщений с равновероятными буквами избыточность выбранного (равномерного) кода оказалась равной нулю.

Пусть теперь вероятности появления в тексте различных букв будут разными :

Буква	А	Б	В	Г	Д	Е	Ж	З
вероятность	0.6	0.2	0.1	0.04	0.025	0.015	0.01	0.01

$$H = -\sum_{i=1}^K P_i \log P_i \quad H=1,781$$

Среднее число символов на одно сообщение при использовании того же равномерного трехразрядного кода,

$$q_{cp} = \sum_{i=1}^K q_i \cdot P_i = \sum_{i=1}^8 3 \cdot \frac{1}{8} = 3$$

а избыточность $r_k = \frac{q_{cp} - H}{q_{cp}} = 1 - \frac{1.781}{3} \approx 0.41$

или довольно значительной величиной - в среднем 4 символа из 10 не несут никакой информации !!!

В связи с тем, что при кодировании неравновероятных сообщений равномерные коды обладают большой избыточностью, было предложено использовать неравномерные коды, размер кодовых комбинаций которых был бы согласован с вероятностью появления различных букв.

Методы построения префиксных кодов

Первый метод (Метод Р. Фано). Алгоритм сводится к последовательному выполнению следующих ПЯТИ шагов:

1. Буквы алфавита \mathbf{A} упорядочиваем по убыванию вероятностей:

$$P(a_1) \geq P(a_2) \geq \dots \geq P(a_K).$$

2. Множество упорядоченных букв разбивается на два подмножества $\mathbf{A}^{(0)}$, $\mathbf{A}^{(1)}$ с помощью некоторого порогового целого числа $1 \leq k^{(1)} \leq K - 1$ так, чтобы величина

$$J^{(1)} = \left| \sum_{i=1}^{k^{(1)}} P(a_i) - \sum_{i=k^{(1)}+1}^K P(a_i) \right|$$

достигала наименьшего возможного значения.

Буквам из подмножества $\mathbf{A}^{(0)}$ приписываем 0, буквам подмножества $\mathbf{A}^{(1)}$ приписываем 1.

3. Если подгруппы $\mathbf{A}^{(0)}$, $\mathbf{A}^{(1)}$ состоят более чем из двух букв, то разбиваем множество букв каждой из подгрупп на две подгруппы $\mathbf{A}^{(00)}$, $\mathbf{A}^{(01)}$ и $\mathbf{A}^{(10)}$, $\mathbf{A}^{(11)}$, соответственно, с помощью пороговых целых чисел ,

$$1 \leq \mathbf{k}^{(11)} \leq k^{(1)} - 1,$$

$$k^{(1)} \leq \mathbf{k}^{(12)} \leq K - 1$$

$$J^{(21)} = \left| \sum_{i=1}^{k^{(11)}} P(a_i) - \sum_{i=k^{(11)}+1}^{k^1} P(a_i) \right|$$

так, чтобы величины

$$J^{(22)} = \left| \sum_{i=k^{(1)}+1}^{k^{(12)}} P(a_i) - \sum_{i=k^{(12)}+1}^K P(a_i) \right|$$

достигали наименьших возможных значений. Буквам из подгрупп $\mathbf{A}^{(00)}$, $\mathbf{A}^{(10)}$ с нулевыми последними индексами приписываем 0, буквам из подгрупп $\mathbf{A}^{(01)}$, $\mathbf{A}^{(11)}$ приписываем 1.

Если подгруппа $\mathbf{A}^{(ij)}$ $i, j = 0, 1$ состоит более чем из одной буквы, то переходим к шагу 4. Если все подгруппы состоят из одной буквы, то переходим к шагу 5.

4. Если есть подгруппы, состоящие более чем из одной буквы, то разбиваем каждую из них на две подгруппы, исходя из соотношения, введенного на шаге 2. Буквам подгрупп с нулевыми последними индексами приписываем нуль, буквам подгрупп с единичными последними индексами приписываем единицу.

Если все образовавшиеся подгруппы состоят из одной буквы, то переходим к шагу 5. Если есть подгруппы, состоящие более чем из одной буквы, то повторяем шаг 4.

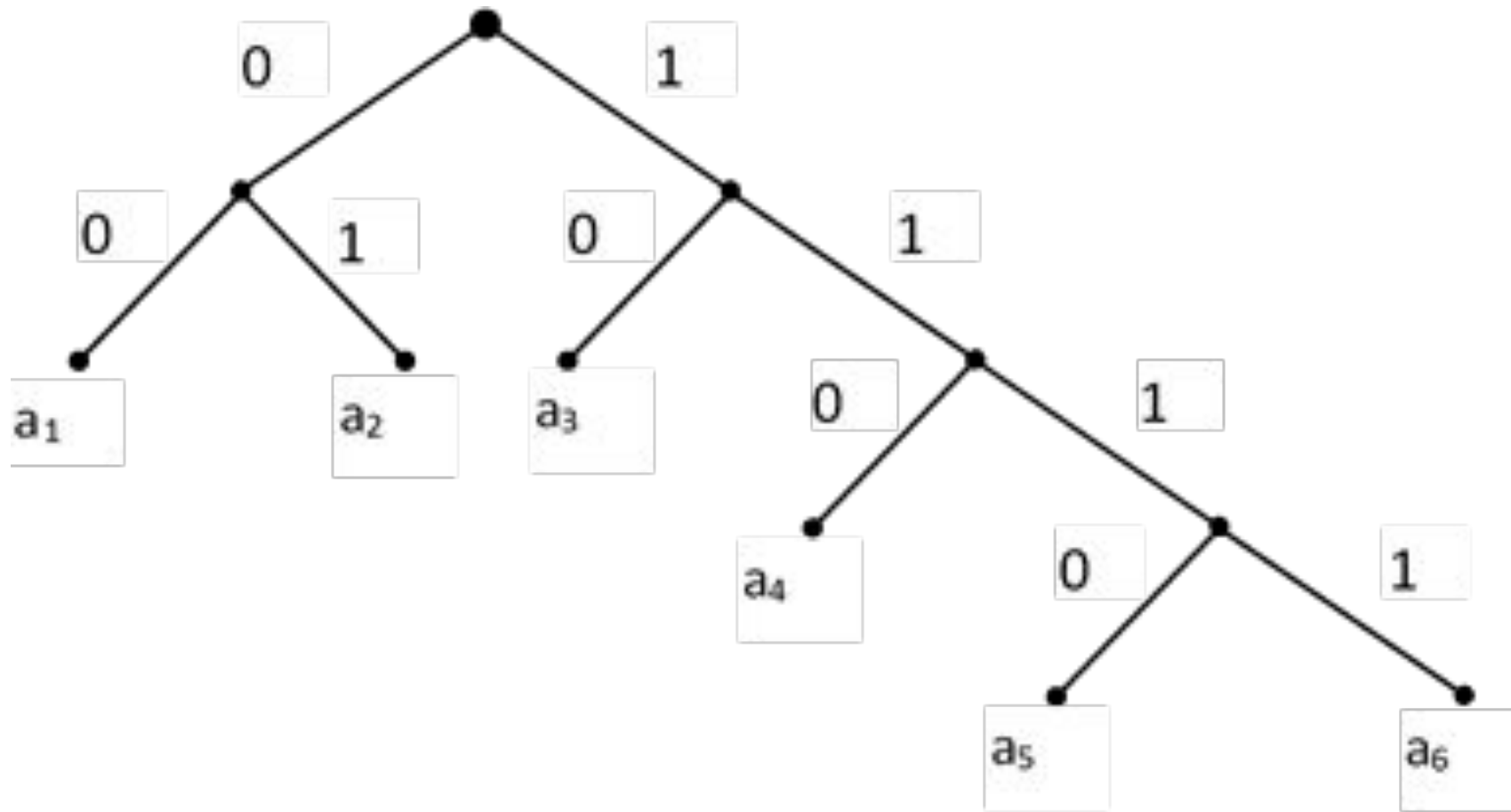
5. Если образовавшиеся подгруппы состоят из одной буквы, то последовательно, начиная с первой метки, выписываем нули и единицы, относящиеся к каждой букве алфавита **A**.

В итоге получается двоичный префиксный код для заданного источника.

Рассмотрим следующий пример:

Буква	$P(a_i)$	I	II	III	IV	Код	$q_i \cdot P_i$				
a_1	0.36	$A^{(0)}$	0	$A^{(00)}$	0			00	0.72		
a_2	0.18		0	$A^{(01)}$	1			01	0.36		
a_3	0.18	$A^{(1)}$	1	$A^{(10)}$	0			10	0.36		
a_4	0.12		1	$A^{(11)}$	1	$A^{(110)}$	0		110	0.36	
a_5	0.09		1		1	$A^{(111)}$	1	$A^{(1110)}$	0	1110	0.36
a_6	0.07		1		1		1	$A^{(1111)}$	1	1111	0.28

Средняя длина для построенного кода $q_{cp} = 2,44$. Соответствующее кодовое дерево имеет вид:



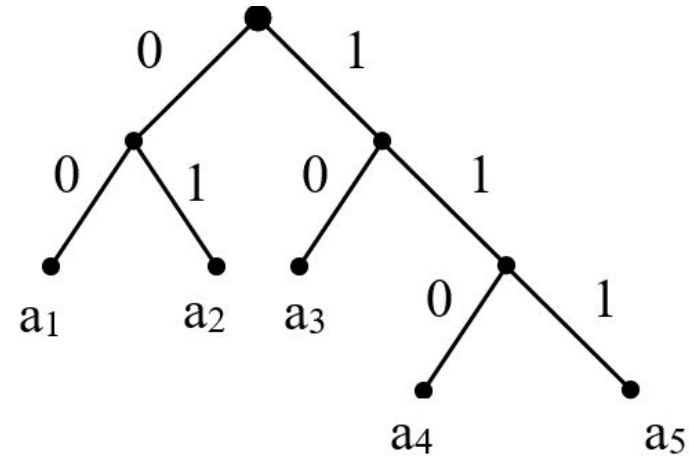
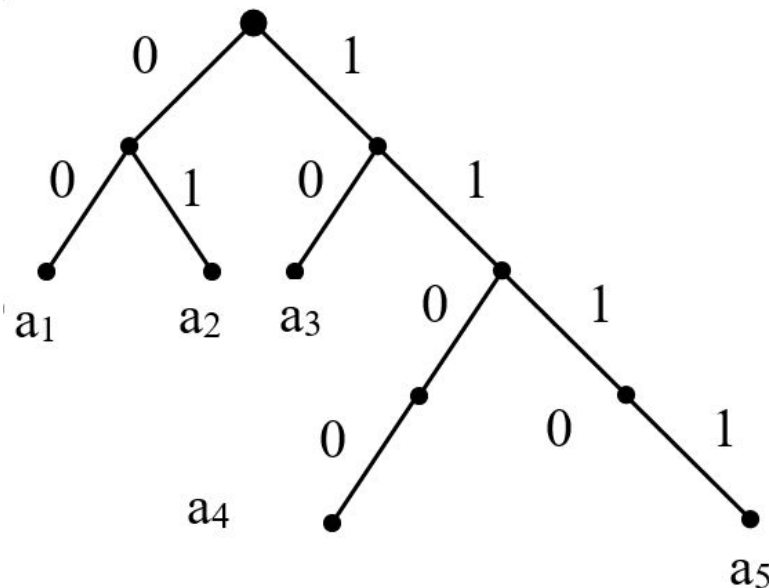
Полученный код называют еще **префиксным множеством**.

В нашем случае это множество следующее:

$$S = \{00, 01, 10, 110, 1110, 1111\}.$$

ОПЕРАЦИЯ УСЕЧЕНИЯ

Рассмотрим кодовое дерево, максимальный порядок концевых вершин которого равен n . Предположим, что вершине a n -го порядка предшествует вершина b $(n-1)$ -го порядка, из которой выходит единственное ребро, ведущее в a . Других рёбер из вершины b не выходит. Операцию удаления этого ребра и превращения промежуточной вершины b $(n-1)$ -го порядка в концевую вершину будем называть *операцией усечения*. Средняя кодовая длина после операции усечения становится меньше. Пример, приведен ниже.



Вектор КРАФТА

Если $\mathcal{S} = \{w_1, w_2, \dots, w_K\}$ – префиксное множество, то можно определить некоторый вектор $\nu(\mathcal{S}) = (q_1, q_2, \dots, q_K)$, состоящий из чисел, являющихся длинами соответствующих префиксных последовательностей т.е. q_i – это длина w_i .

Вектор (q_1, q_2, \dots, q_k) , состоящий из неубывающих положительных целых чисел, удовлетворяющих неравенству Крафта

$$2^{-q_1} + 2^{-q_2} + \dots + 2^{-q_k} \leq 1$$

называется *вектором Крафта*.

Для него справедливо следующее утверждение: если \mathcal{S} – какое-либо префиксное множество, то $\nu(\mathcal{S})$ – вектор Крафта.

Иными словами, *длины двоичных последовательностей в префиксном множестве удовлетворяют неравенству Крафта.*

Для него справедливо следующее утверждение: если \mathcal{S} – какое-либо префиксное множество, то $\nu(\mathcal{S})$ – вектор Крафта.

Иными словами, *длины двоичных последовательностей в префиксном множестве удовлетворяют неравенству Крафта.*

Неравенство Крафта формулируется в виде следующей теоремы:

Теорема. Пусть q_1, q_2, \dots, q_K набор положительных целых чисел. Для того чтобы существовал префиксный код с длинами кодовых слов q_1, q_2, \dots, q_K , необходимо и достаточно, чтобы выполнялось неравенство

$$2^{-q_1} + 2^{-q_2} + \dots + 2^{-q_K} \leq 1$$

Теорема не утверждает, что любой код с длинами кодовых слов, удовлетворяющими неравенству Крафта, является префиксным. Например, множество двоичных кодовых слов $\{0,01,11\}$ удовлетворяет неравенству, но не обладает свойством префикса

Примеры простейших префиксных множеств и соответствующие им векторы Крафта:

$$S_1 = \{0, 10, 11\} \text{ и } v(S_1) = (1, 2, 2);$$

$$S_2 = \{0, 10, 110, 111\} \text{ и } v(S_2) = (1, 2, 3, 3);$$

$$S_3 = \{0, 10, 110, 1110, 1111\} \text{ и } v(S_3) = (1, 2, 3, 4, 4);$$

$$S_4 = \{0, 10, 1100, 1101, 1110, 1111\} \text{ и } v(S_4) = (1, 2, 4, 4, 4, 4);$$

Второй метод построения двоичных префиксных кодов (Метод К. Шеннона).

1. Буквы алфавита Λ упорядочиваем по убыванию вероятностей:

$$P(a_1) \geq P(a_2) \geq \dots \geq P(a_K).$$

2. Находим числа $q_i, i=1, \dots, K$, исходя из условия:

$$\frac{1}{2^{q_i}} \leq P(a_i) < \frac{1}{2^{q_i-1}}$$

3. Подсчитываем накопленные суммы:

$$P_1 = 0, P_2 = P(a_1), P_3 = P(a_1) + P(a_2), \dots \quad P_n = \sum_{i=1}^K P(a_i)$$

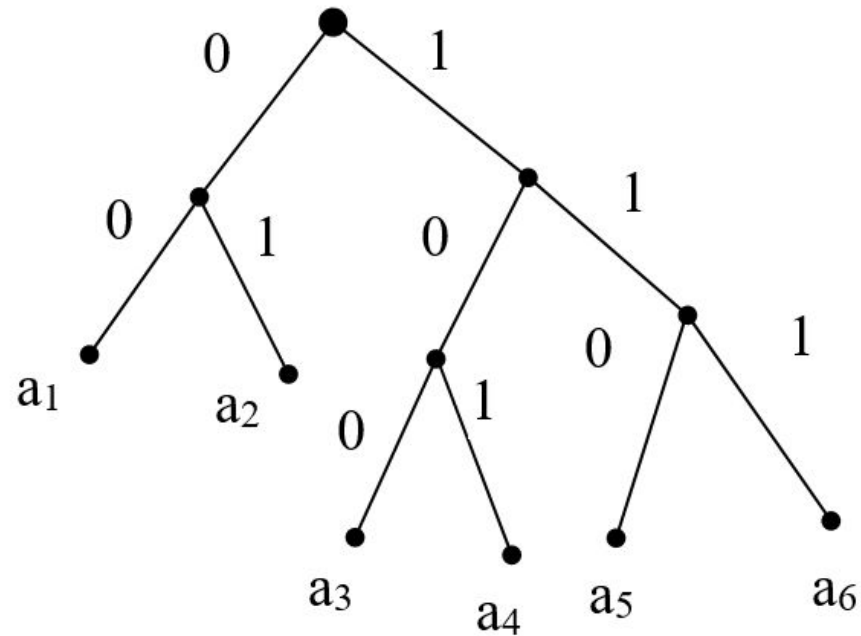
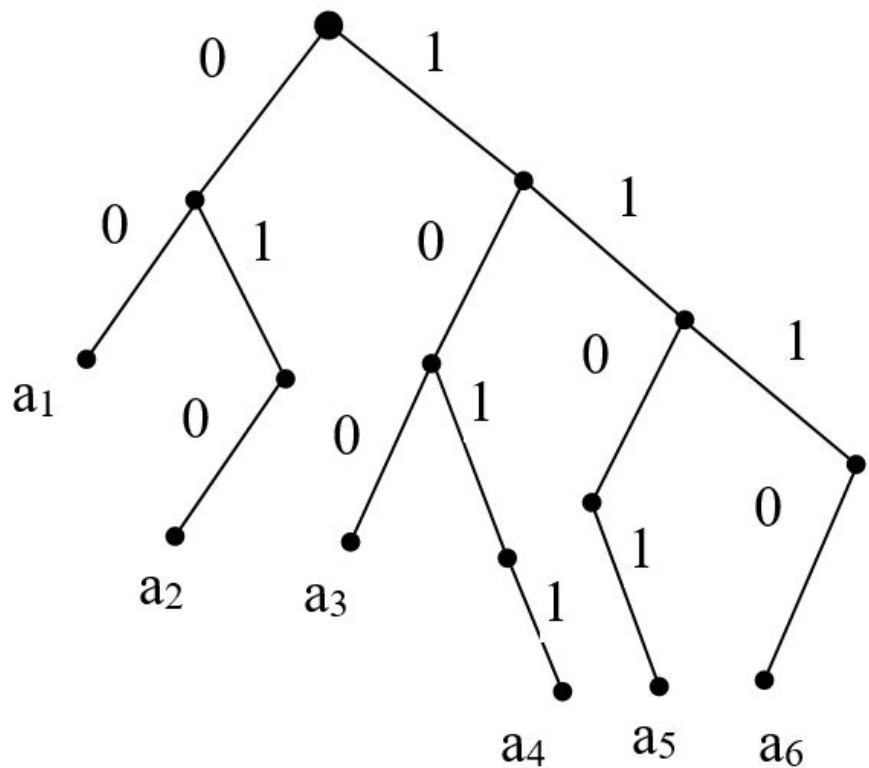
4. Находим первые после запятой q_i знаков в разложении числа P_i в двоичную дробь: $i=1, \dots, K$. Цифры этого разложения, стоящие после запятой, являются кодовым словом, соответствующим букве a_i .

5. Если необходимо, производим операцию усечения.

Для сравнения методов рассмотрим тот же самый источник сообщений и проделаем последовательно шаги алгоритма:

	1	2	3	4	5	
Буква	$P(a_i)$	q_i	Накопленная P_i	код до	код после	$q_i \cdot P_i$
a_1	0.36	2	0,00	00	00	0,72
a_2	0.18	3	0,36 = 0,01011...	010	01	0,36
a_3	0.18	3	0,54 = 0,10001...	100	100	0,54
a_4	0.12	4	0,72 = 0,10111...	1011	101	0,36
a_5	0.09	4	0,84 = 0,11010...	1101	110	0,27
a_6	0.07	4	0,93 = 0,11101...	1110	111	0,21

Средняя длина для построенного кода $q_{cp} = 2,46$. Соответствующее кодовое дерево до и после усечения имеет вид:



Алгоритм Шеннона-Фано применим и при основании кода больше двух ($K > 2$). В этом случае алфавит разбивается на K частей примерно одинаковой суммарной вероятности

Методика кодирования Хаффмана (Хаффмэна)

Рассмотренные выше алгоритмы кодирования не всегда приводят к хорошему результату, вследствие отсутствия четких рекомендаций относительно того, как делить множество кодируемых знаков на подгруппы. Рассмотрим методику кодирования Хаффмана, которая свободна от этого недостатка.

1. Кодируемые знаки, также как при использовании метода Шеннона и Фано, располагают в порядке убывания их вероятностей (таблица).
2. Далее на каждом этапе две последние позиции списка заменяются одной и ей приписывают вероятность, равную сумме вероятностей заменяемых позиций.
3. После этого производится пересортировка списка по убыванию вероятностей, с сохранением информации о том, какие именно знаки объединялись на каждом этапе.
4. Процесс продолжается до тех пор, пока не останется единственная позиция с вероятностью, равной 1.

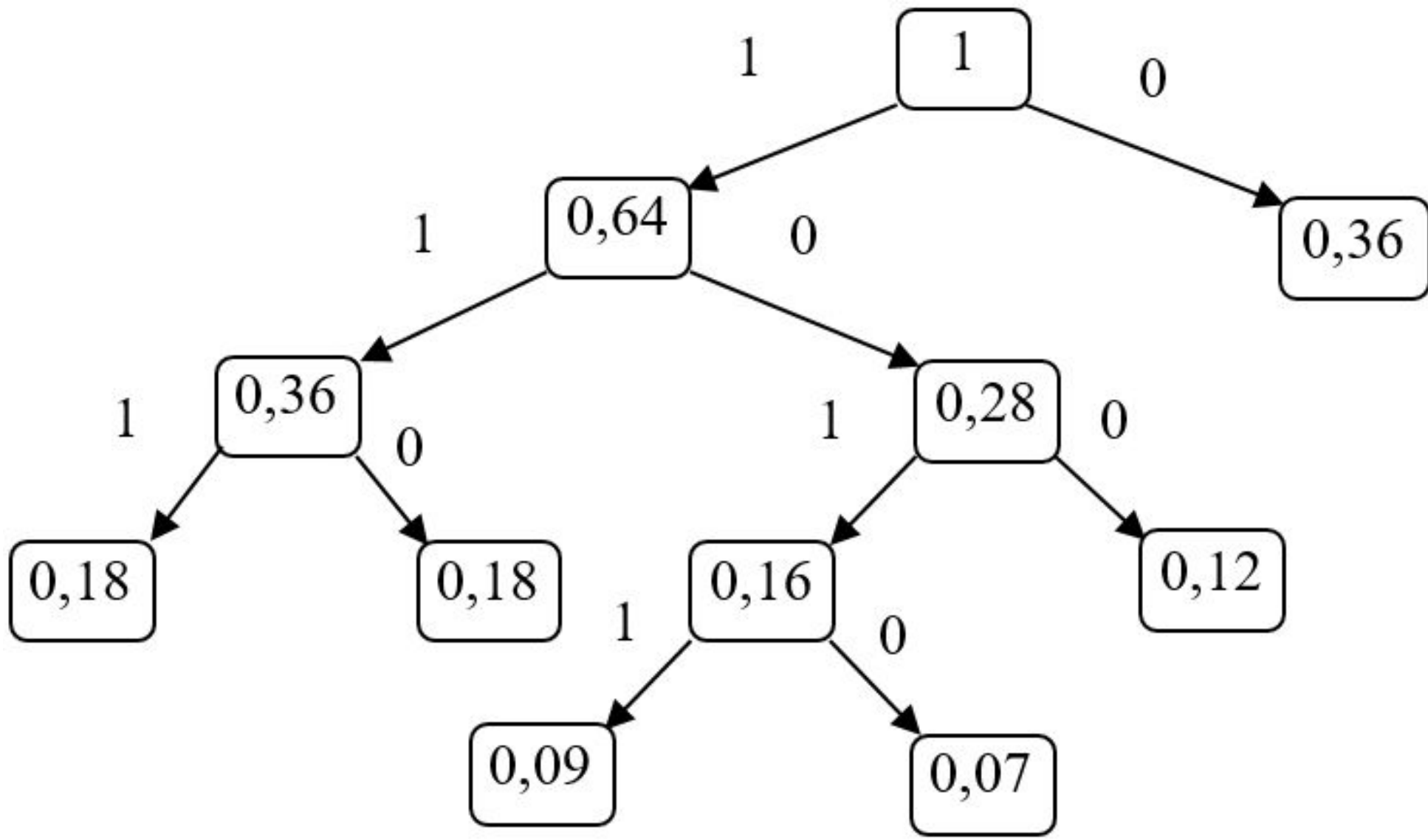
Буква	$P(a_i)$	Вспомогательные группировки				
		1	2	3	4	5
a_1	0.36	0.36	0.36	0.36	<u>0.64</u>	1
a_2	0.18	0.18	<u>0.28</u>	<u>0.36</u>	0.36	
a_3	0.18	0.18	0.18	<u>0.28</u>		
a_4	0.12	<u>0.16</u>	0.18			
a_5	0.09	0.12				
a_6	0.07					

После этого строится кодовое дерево.

а) Корню дерева ставится в соответствие узел с вероятностью, равной 1.

б) Далее каждому узлу приписываются два потомка с вероятностями, которые участвовали в формировании значения вероятности обрабатываемого узла.

в) 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.

Жирным шрифтом в таблице выделены объединяемые позиции, подчеркиванием – получаемые при объединении позиции.

Буква	$P(a_i)$	q_i	код	$q_i \cdot P_i$
a_1	0.36	1	0	0,36
a_2	0.18	3	111	0,54
a_3	0.18	3	110	0,54
a_4	0.12	3	100	0,36
a_5	0.09	4	1011	0,36
a_6	0.07	4	1010	0,28

Средняя длина для построенного кода $q_{cp} = 2,44$,

ЭНТРОПИЯ ИСТОЧНИКА В ЭТОМ ПРИМЕРЕ:

$$H(Z) = 2,37$$

БЛОЧНОЕ КОДИРОВАНИЕ

Можно видеть, что как для кода Хаффмана, так и для кодов Фано и Шеннона среднее количество двоичных символов, приходящееся на символ источника, приближается к энтропии источника, но не равно ей.

Данный результат представляет собой следствие теоремы кодирования без шума для источника (**первой теоремы Шеннона**), которая утверждает:

1. При любой производительности источника сообщений, меньшей пропускной способности канала, т. е. при условии

$$\bar{I}(Z) = C_{\delta} - \varepsilon,$$

где ε — сколь угодно малая, положительная величина, существует способ кодирования, позволяющий передавать по каналу все сообщения, вырабатываемые источником.

2. Не существует способа кодирования, обеспечивающего передачу сообщений без их неограниченного накопления, если $\bar{I}(Z) > C_{\delta}$.

Рассматриваемая теорема Шеннона часто приводится и в другой формулировке:

сообщения источника с энтропией $H(Z)$ всегда можно закодировать последовательностями символов с объемом алфавита m так, что среднее число символов на знак сообщения q_{cp} будет сколь угодно близко к величине $H(Z)/\log m$,

но не менее ее.

Данное утверждение обосновывается также указанием на возможную процедуру кодирования, при которой обеспечивается равновероятное и независимое поступление символов на вход канала, а следовательно, и максимальное количество переносимой каждым из них информации, равное $\log m$. Как мы установили ранее, в общем случае это возможно, если кодировать сообщения **длинными блоками**. Указанная граница достигается асимптотически при **безграничном увеличении длины кодируемых блоков**.

Итак, из теоремы Шеннона следует, что избыточность в последовательностях символов можно устранить, если перейти к кодированию достаточно большими блоками.

Попробуем на примере алфавита из 2 букв – А и Б.

Пусть $p(A)=0.7$ и $p(B)=0.3$

$H=-0.7*\log(0.7)-0.3*\log(0.3)=0.881\dots$

Используем метод Шеннона-Фано

Буква	Вероятность	код
А	0.7	1
Б	0.3	0

$1*0.7+1*0.3=1$ двоичн. знаков на букву.

$1-0.881=0.12$. Стало быть, на 12% больше минимального по Шеннону значения 0.881.

Буква	Вероятность	код
АА	$0.7*0.7=0,49$	1
АБ	$0.7*0.3=0,21$	01
БА	$0.3*0.7=0,21$	001
ББ	$0.3*0.3=0,21$	000

$$1*0.49+2*0.21+3*0.3=1,81$$

На одну букву приходится $1,81/2=0,905$ на 3% больше 0,881 !!!!

Если применить метод Ш-Ф для 3-х буквенных комбинаций ААА, ААБ итд получится 0,895 двоичных знаков – на 1.5% больше.

Недостатки методов эффективного кодирования

1. Различия в длине кодовых комбинаций. Обычно знаки на вход устройства кодирования поступают через равные промежутки времени. Если им соответствуют комбинации различной длины, то для обеспечения полной загрузки канала при передаче без потерь необходимо предусмотреть **буферное устройство**, как на передающей, так и на приемной стороне.

2. Задержка в передаче информации. Достоинства эффективного кодирования проявляются в полной мере при кодировании длинными блоками. Для этого необходимо накапливать знаки, как при кодировании, так и при декодировании, что **приводит к значительным задержкам**.

3. Низкая помехозащищенность. Даже одиночная ошибка, возникшая в процессе передачи, может **нарушить свойство префиксности кода** и повлечь за собой неправильное декодирование ряда последующих комбинаций. Это явление называют треком ошибки.

4. Сложность технической реализации. Использование буферных устройств, для обеспечения равномерной загрузки канала, при разной длине кодовых комбинаций и организация кодирования блоками для повышения эффективности приводят **к усложнению реализации систем эффективного кодирования**.

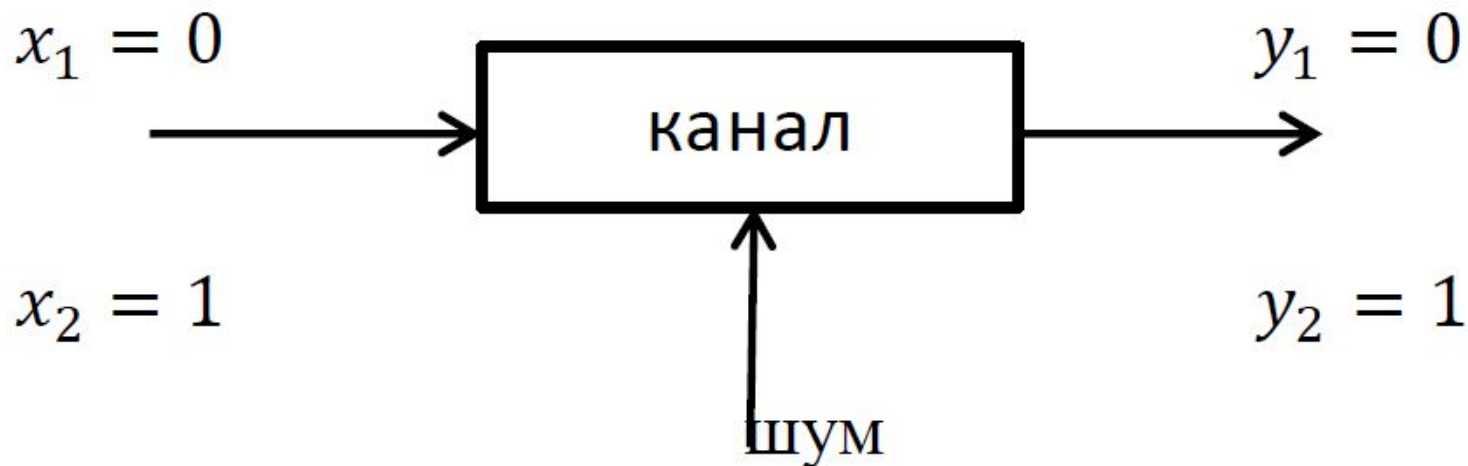
Задача 1. Для передачи сообщений используется код, состоящий из трех символов, вероятности появления которых равны 0.6, 0.3 и 0.1. Корреляция между символами отсутствует. Определить избыточность кода.

Задача

На вход дискретного симметричного канала без памяти поступают двоичные символы $x_1 = 0$ и $x_2 = 1$ с априорными вероятностями $p(x_1) = 0,85$ и $p(x_2) = 0,15$. Переходные вероятности $p(y_j/x_i)$ в таком канале задаются соотношением

$$p(y_j/x_i) = \begin{cases} p, & j \neq i \\ 1 - p, & j = i \end{cases}$$

где $p=0.05$ - вероятность ошибки. Определить все апостериорные вероятности.



В таком канале каждый кодовый символ может быть принят с ошибочной вероятностью

$$p(y_1/x_2) = p(y_2/x_1) = p = 0,05.$$

А может быть передан правильно с вероятностью

$$p(y_1/x_1) = p(y_2/x_2) = q = 0,95$$

$$P(y_j | x_i) = \frac{P(x_i, y_j)}{P(x_i)},$$

$$P(x_i, y_j) = P(y_j)P(x_i | y_j).$$

$$P(x_i | y_j) = \frac{P(x_i, y_j)}{P(y_j)}$$

$$p(x_1/y_2) = \frac{p(x_1)p(y_2/x_1)}{p(x_1)p(y_2/x_1) + p(x_2)p(y_2/x_2)} = \frac{0,85 \cdot 0,05}{0,85 \cdot 0,05 + 0,15 \cdot 0,95} =$$

$$= 0,23,$$

$$p(x_2/y_2) = \frac{p(x_2)p(y_2/x_2)}{p(x_2)p(y_2/x_2) + p(x_1)p(y_2/x_1)} = \frac{0,15 \cdot 0,95}{0,15 \cdot 0,95 + 0,85 \cdot 0,05} =$$

$$= 0,77.$$