ЛЕКЦИЯ №3 ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ Ч.1

Преподаватель: Оцоков Шамиль Алиевич

Москва, 2021 г.

Неравномерный код

Если заданы четыре сообщения A1, A2, A3, A4 с вероятностями P(A1)=1/2, P(A2)=1/4, P(A3)=P(A4)= 1/8.

A_1	A_2	A_3	A_4
00	01	10	11

A_1	A_2	A_3	A_4
0	10	110	111

Идея метода Шеннона-Фано

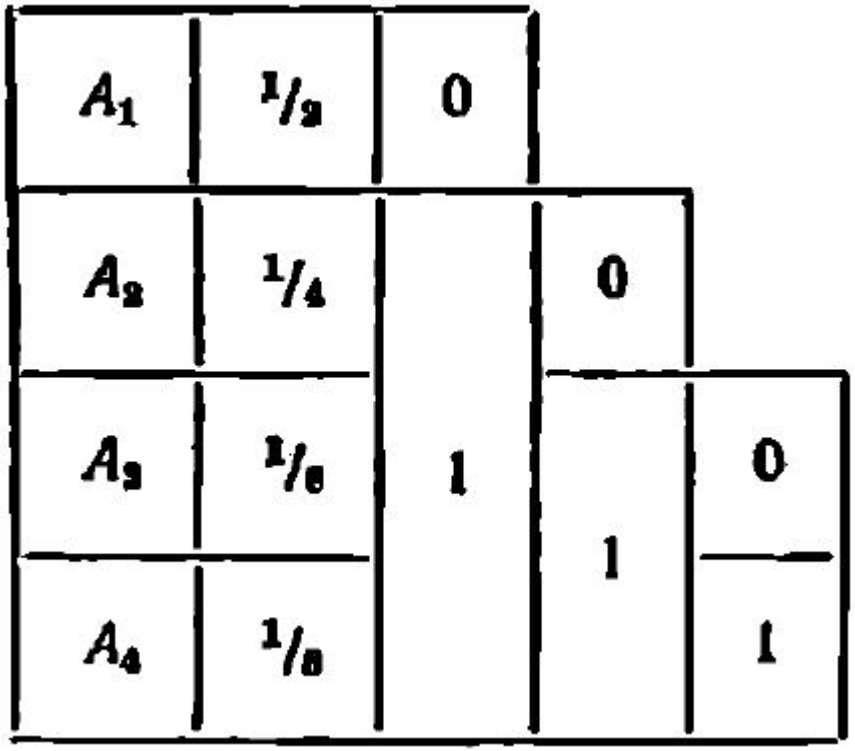
- Идея этого метода заключалась в том, чтобы заменить часто встречающиеся символы более короткими кодами, а редко встречающиеся последовательности более длинными кодами.
- Метод Шеннона-Фано относится к вероятностным методам сжатия

Схема метода Шеннона-Фано

Нетрудно описать общую схему метода Фано. Располагаем N сообщений в порядке убывания их вероятностей: $P(A_1) \geqslant P(A_2) \geqslant \ldots \geqslant P(A_N)$. Далее разбиваем множество сообщений на две группы так, чтобы суммарные вероятности сообщений каждой из групп были как можно более близки друг к другу. Сообщениям из одной группы в качестве первого символа кодового слова приписывается символ 0, сообщениям из другой — символ 1. По тому же принципу каждая из полученных групп снова разбивается на две части, и это разбиение определяет значение второго символа кодового слова. Процедура продолжается до тех пор, пока все множество не будет разбито на отдельные сообщения. В результате каждому из сообщений будет сопоставлено кодовое слово из нулей и единиц.

Идея метода Шеннона-Фано

Если заданы четыре сообщения A1, A2, A3, A4 с вероятностями P(A1)=1/2, P(A2)=1/4, P(A3)=P(A4)= 1/8.



A_1	A_2	A_3	A_4
0	10	110	111

Идея метода Шеннона-Фано

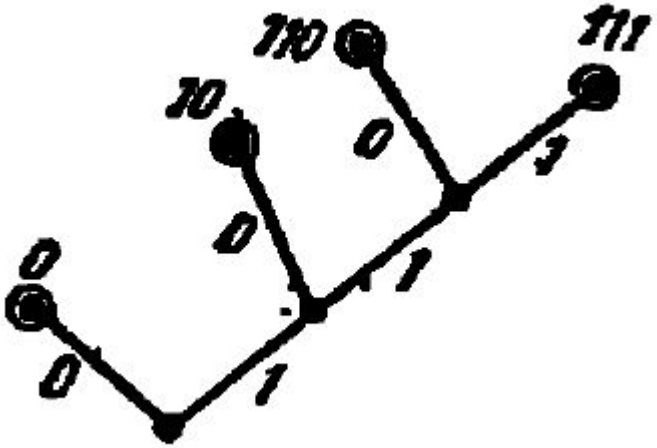
Уже этот пример показывает, что показателем эконо
ности или эффективности неравномерного кода являют
ие длины отдельных кодовых слов, а «средняя» их длина
определяемая равенством:

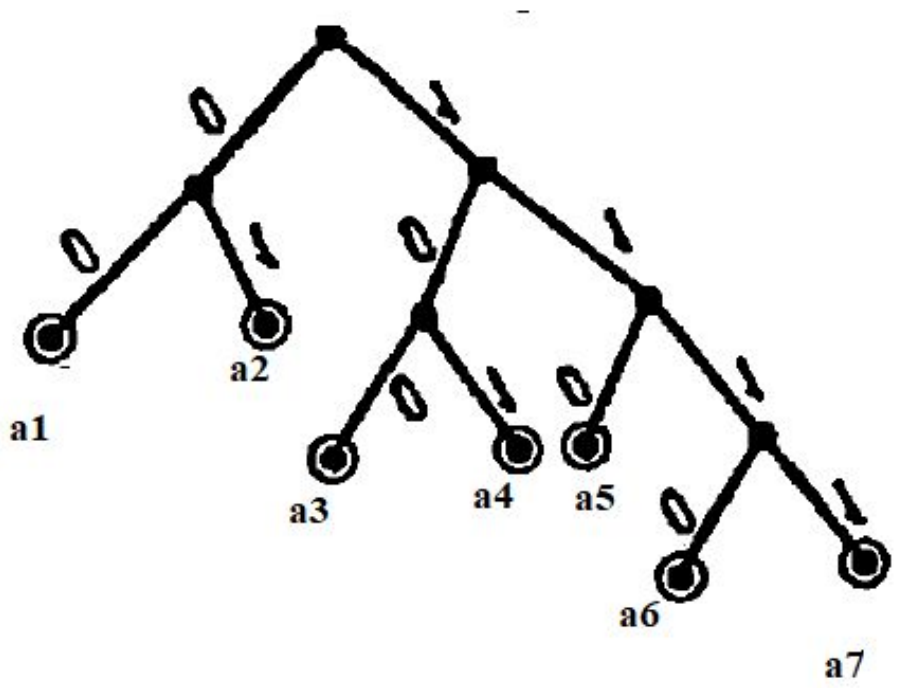
$$\bar{l} = \sum_{i=1}^{N} l_i P(A_i),$$

где l_i — длина кодового обозначения для сообщения A
$P(A_i)$ — вероятность сообщения A_i, N — общее число с
общений.

Кодовые деревья

Чем более вероятно сообщение, тем быстрее оно образует «самостоятельную» группу и тем более коротким словом оно будет закодировано. Эго обстоятельство обеспечивает высокую экономность кода Фано.





Кодовые деревья

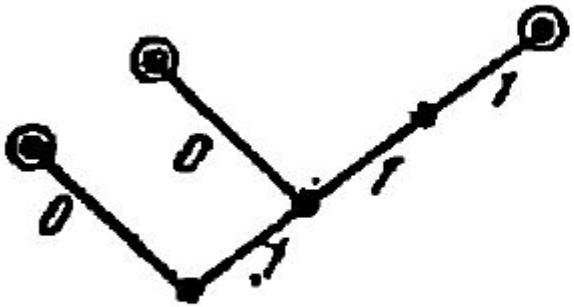
В любом кодовом тексте выделять отдельные кодовые слова без использования специальных разделительных знаков.
Чтобы код удовлетворял следующему требованию:
- всякая последовательность кодовых символов может быть единственным образом разбита на кодовые слова
Коды для которых последнее требование выполнено, называются **однозначно декодируемыми** (иногда их называют кодами без запятой).

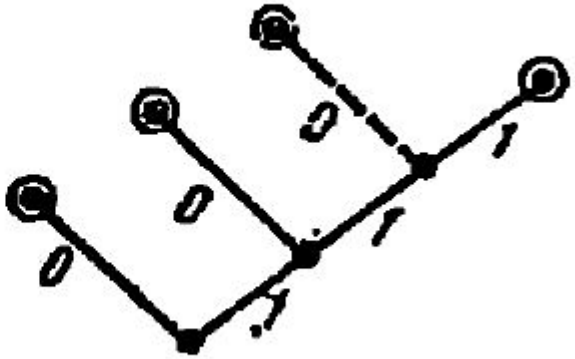
Наиболее простыми и употребимыми кодами без запятой являются так называемые **префиксные коды**, обладающие тем свойством. что никакое кодовое слово не является началом (префиксом) другого кодового слова. Если код префиксный, то, читая кодовую запись подряд от начала. мы всегда сможем
разобраться, где кончается одно кодовое слово и начинается следующее.

Префиксный код

Префиксный код называется полным если добавление к нему нового слова не нарушает свойство префиксности.

Подумать является ли двоичный код Фано является полным кодом.



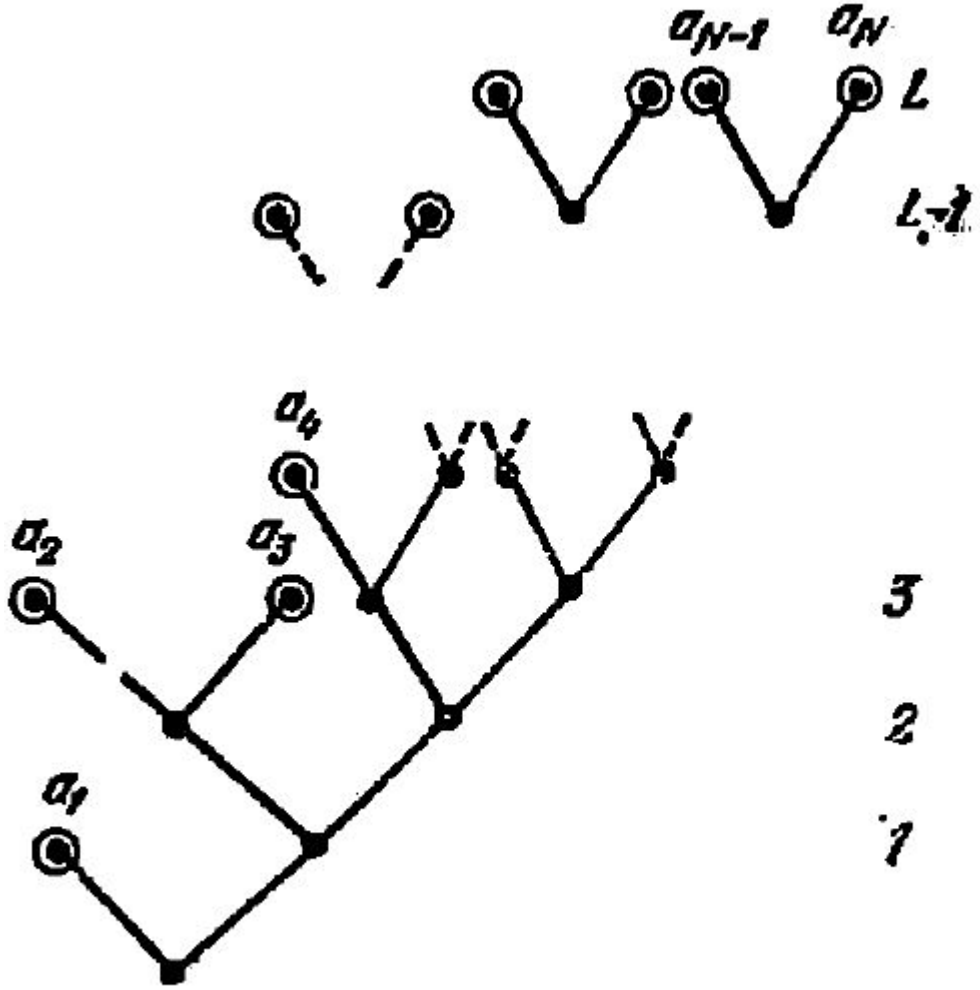


Однозначно декодируемый код, который не является префиксным

{1,10}, {01, 10, 011}

Не существует префиксного кода с длинами кодовых слов 1,1,2.
Существует ли префиксный код с заданными длинами слов?

Пусть $V=\{a_1, a_2, \ldots, a_N\}$ — префиксный двоичный код, дерево которого схематически изображено



Пусть n_k — число кодовых слов длины k (n_k совпадает с числом концевых вершин k-го этажа). Конечно, справедливо неравенство

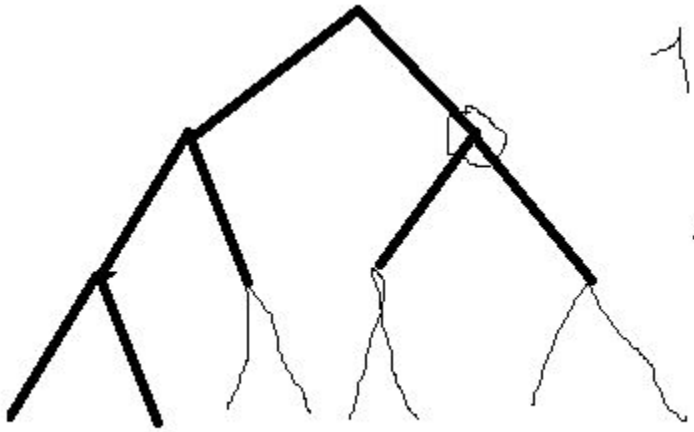
Неравенство Крафта

$$n_k \leqslant 2^k,$$

так как 2^k — максимально возможное число вершин на k-м этаже двоичного дерева. Однако в случае префиксного кода для n_k можно получить гораздо более точную оценку, чем (1). В самом деле, если $n_1, n_2, \ldots, n_{k-1}$ — число конечных вершин 1; 2; ... ; k—1 этажей дерева, то число всех вершин k-го этажа кодового дерева равно

$$2^k - 2^{k-1}n_1 - 2^{k-2}n_2 - \ldots - 2n_{k-1},$$

Неравенство Крафта



$$2^k - 1 \cdot 2^{k-1}$$

Неравенство Крафта

$$2^k - 2^{k-1}n_1 - 2^{k-2}n_2 - \ldots - 2n_{k-1},$$

и потому

$$n_k \leqslant 2^k - 2^{k-1}n_1 - 2^{k-2}n_2 - \ldots - 2n_{k-1} \qquad (2)$$

или иначе

$$2^{k-1}n_1 + 2^{k-2}n_2 + \ldots + 2n_{k-1} + n_k \leqslant 2^k.$$

Деля обе части последнего неравенства на 2^k, получаем:

$$\sum_{i=1}^{k} n_i 2^{-i} \leqslant 1. \qquad (3)$$

Неравенство Крафта

Неравенство (3) верно для любого $k \leqslant L$ (L — максимальная длина кодовых слов), в частности

$$\sum_{i=1}^{L} n_i 2^{-i} \leqslant 1. \qquad (4)$$

Если $l_1, l_2, \ldots, l_N$ — длины кодовых слов $a_1, a_2, \ldots, a_N$, то неравенство (4) запишется в таком виде:

$$2^{-l_1} + 2^{-l_2} + \ldots + 2^{-l_N} \leqslant 1. \qquad (5)$$

Это и есть то условие, которому обязаны удовлетворять длины кодовых слов двоичного префиксного кода.

Неравенство Крафта

Это и есть то условие, которому обязаны удовлетворять длины кодовых слов двоичного префиксного кода.

Оказывается, что неравенство (5), называемое в теории кодирования неравенством Крафта, является также достаточным условием того, чтобы существовал префиксный код с длинами кодовых слов $l_1, l_2, \ldots, l_N$.

Рассуждаем так. Если среди чисел $l_1, l_2, \ldots, l_N$ имеется ровно n_i чисел, равных i, то неравенство (5) можно переписать в виде (4), где L — максимальное из данных чисел. Из

Неравенство Крафта

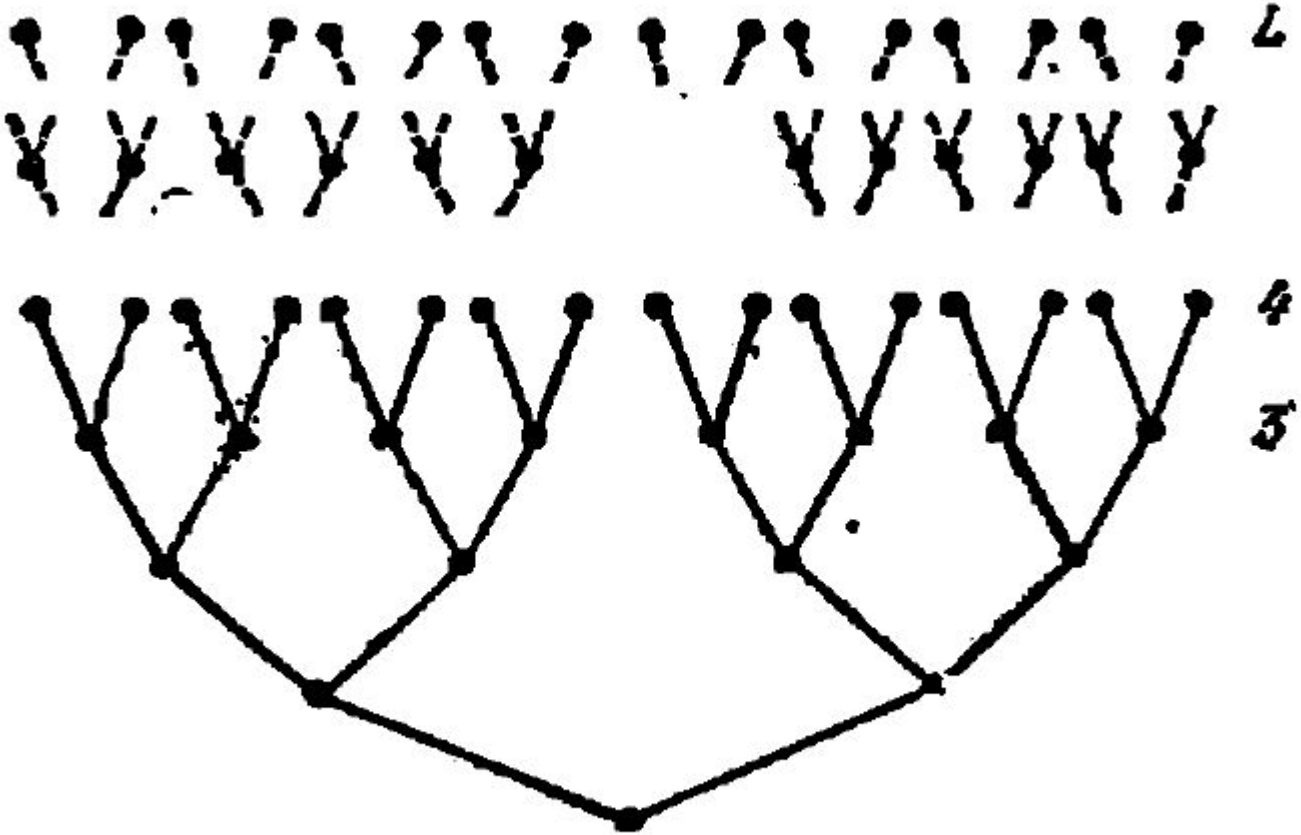


Рис. 11.

Неравенство Крафта

справедливости (4) подавно следует, что верны неравенства (3) для всех $k \leqslant L$, а, следовательно, и неравенство (2).

Обратимся к рис. 11, на котором изображено дерево «высоты» L, имеющее наибольшее число вершин и ветвей (ребер). Все концевые вершины (их 2^L) такого дерева находятся на последнем L-ом этаже, а из каждой вершины промежуточного этажа исходят ровно две ветви.

Для построения нужного префиксного кода мы должны подходящим образом выбрать n_1 слов длины 1, n_2 слов длины 2, вообще n_k слов длины k ($1 \leqslant k \leqslant L$) или, иными словами, n_1 концевых вершин на первом, n_2 — на втором, ..., n_k — на k-ом этаже.

Неравенство Крафта

Из неравенства (2) при $k=1$ получаем $n_1 \leqslant 2$, т. е. требуемое число не превосходит общего числа вершин первого этажа. Значит, на этом этаже можно выбрать какие-то n_1 вершин в качестве концевых (n_1 равно 0, 1 или 2). Если это сделано, то из общего числа вершин второго этажа (их $2^2=4$) для построения кода можно использовать лишь $4-2n_1$ (почему?). Однако нам хватит и этого числа вершин, так как из неравенства (2) при $k=2$ вытекает

$$n_2 \leqslant 4-2n_1.$$

Аналогично, при $k=3$ имеем неравенство:

$$n_3 \leqslant 2^3-4n_1-2n_2.$$

Неравенство Крафта

Правая часть его вновь совпадает с допустимым для построения префиксного кода числом вершин третьего этажа, если на первых двух этажах уже выбраны n_1 и n_2 концевых вершин. Значит, снова можно выбрать n_2 концевых вершин на третьем этаже. Продолжая этот процесс вплоть до $k=L$, мы и получим требуемый код.

Если кодовый алфавит содержит d символов, то подобным же образом доказывается, что необходимым и достаточным условием для существования префиксного кода с длинами слов $l_1, l_2, \ldots, l_N$ является выполнение неравенства

$$d^{-l_1}+d^{-l_2}+\ldots+d^{-l_N}\leqslant 1. \qquad (6)$$

Неравенство Крафта

2. Доказать, что префиксный код является полным тогда тогда, когда в неравенстве Крафта достигается равенство:

$$2^{-l_1}+2^{-l_2}+\ldots+2^{-l_N}=1.$$

Рассмотрим дерево, соответствующее полному префиксному коду. Представим себе, что на это дерево взбирается обезьяна. Начав с корня, она наугад выбирает любую из 2 исходящих из него ветвей; вероятность такого выбора равна $1/2$. Добравшись до очередной развилки, обезьяна снова наугад выбирает некоторую ветвь с вероятностью $1/2$ (напомним, что из каждой промежуточной вершины дерева полного кода исходит ровно 2 ветвей). Тогда вероятность того, что обезьяна достигнет какой-то определенной концевой вершины, находящейся на высоте k, равна $(1/2)^k$. Если таких вершин n_k, то с вероятностью $n_k 2^{-k}$ обезьяна остановится на высоте k. На какой-то высоте от 1 до L обезьяне придется остановиться (вероятность этого равна 1). Поэтому $\sum_{k=1}^{L} n_k 2^{-k}=1$.

(В приведенном рассуждении мы использовали правила сложения и умножения вероятностей.)

Метод Шеннона построения префиксного кода с заданными длинами слов

Пусть числа $l_1, l_2, \ldots, l_N$ удовлетворяют неравенству

$$\sum_{i=1}^{N} 2^{-l_i} \leqslant 1.$$

Можно считать, что $l_1 \leqslant l_2 \leqslant \ldots \leqslant l_N$. Рассмотрим последовательность чисел

$$q_1 = 0;\ q_2 = 2^{-l_1};\ \ldots;\ q_j = \sum_{i=1}^{j-1} 2^{-l_i},\ \ldots,\ q_N = \sum_{i=1}^{N-1} 2^{-l_i}. \qquad (7)$$

Метод Шеннона построения префиксного кода с заданными длинами слов

Заметим, что все эти числа заключены в пределах $0 \leqslant q_j < 1$, поэтому каждое из них может быть представлено двоичной дробью вида $\sum_{k=1}^{n} \alpha_k 2^{-k}$, где каждое α_k есть 0 или 1. При этом из (7) можно заключить, что все эти дроби конечны, и двоичная запись для q_j имеет не более l_j значащих цифр. Таким образом, любое число q_j однозначно представимо в виде:

$$q_j = \sum_{i=1}^{l_j} c_{ij} 2^{-i},$$

Метод Шеннона построения префиксного кода с заданными длинами слов

где всякое c_{ij} есть 0 или 1. Следовательно, каждому q_j однозначно отвечает слово $v_j = c_{1j}c_{2j} \ldots c_{l_j j}$ длины l_j. Рассмотрим код $V = \{v_1, v_2, \ldots, v_N\}$. Покажем, что он обладает свойством префикса. Пусть v_j и v_k два слова ($k > j$). Тогда, согласно (7), $q_k - q_j \geqslant 2^{-l_j}$, а это означает, что l_j-й символ слова v_j не совпадает с l_j-м символом слова v_k. Следовательно, v_j не является началом v_k, откуда и вытекает префиксность кода V.

Метод Шеннона построения префиксного кода с заданными длинами слов

$l_1 = 1,\ l_2 = l_3 = 3,\ l_4 = 4.$

q1 = 0,
q2 = 2^(-1) = 1/2
q3 = 2^(-1) + 2^(-3) = 1/2 + 1/8 = 5/8
q4 = 2^(-1) + 2^(-3) + 2^(-3) = 3/4
v1 = 0
v2 = 100
v3 = 101
v4 = 1100

$$K > j \qquad q_K = \sum_{i=1}^{K-1} 2^{-l_i}$$

$$q_j = \sum_{i=1}^{j-1} 2^{-l_i}$$

$$\Delta = q_K - q_j = \sum_{i=j}^{K-1} 2^{-l_i} > 2^{-l_j} \rightarrow$$

Свойства оптимальный код

Пусть сообщения $A_1, A_2, \ldots, A_N$ имеют вероятности $p_1, p_2, \ldots, p_N$ $(p_1 \geqslant p_2 \geqslant \ldots \geqslant p_N)$ и кодируются двоичными словами $a_1, a_2, \ldots, a_N$, имеющими длины $l_1, l_2, \ldots, l_N$. Постараемся выяснить, какими свойствами должен обладать двоичный код, если он оптимален.

1. В оптимальном коде менее вероятное сообщение не может кодироваться более коротким словом, т. е. если $p_i < p_j$, то $l_i \geqslant l_j$.

Свойства оптимальный код

1. В оптимальном коде менее вероятное сообщение не может кодироваться более коротким словом, т. е. если $p_i < p_j$, то $l_i \geqslant l_j$.

Действительно, в противном случае поменяем ролями кодовые обозначения для A_i и A_j. При этом средняя длина кодовых слов изменится на величину

$$p_i l_i + p_j l_j - p_i l_j - p_j l_i = (p_i - p_j)(l_i - l_j) > 0,$$

т. е. уменьшится, что противоречит определению оптимального кода.

Свойства оптимальный код

2. Если код оптимален, то всегда можно так перенумеровать сообщения и соответствующие им кодовые слова, что $p_1 \geqslant p_2 \geqslant \ldots \geqslant p_N$ и при этом

$$l_1 \leqslant l_2 \leqslant \ldots \leqslant l_N. \qquad (1)$$

В самом деле, если $p_i > p_{i+1}$, то из свойства 1 следует, что $l_i \leqslant l_{i+1}$. Если же $p_i = p_{i+1}$, но $l_i > l_{i+1}$, то переставим сообщения A_i и A_{i+1} и соответствующие им кодовые слова. Повторяя эту процедуру нужное число раз, мы и получим требуемую нумерацию.

Из неравенств (1) следует, что сообщение A_N кодируется словом a_N наибольшей длины l_N.

Свойства оптимальный код

3. В оптимальном двоичном коде всегда найдется, по крайней мере, два слова наибольшей длины, равной l_N, и таких, что они отличаются друг от друга лишь в последнем символе.

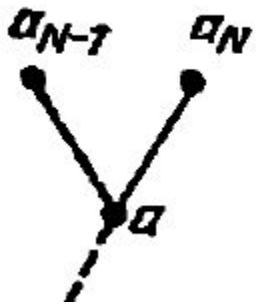
Действительно, если бы это было не так, то можно было бы просто откинуть последний символ кодового слова a_N, не нарушая свойства префиксности кода. При этом мы, очевидно, уменьшили бы среднюю длину кодового слова.

Пусть слово a_t имеет ту же длину, что и a_N и отличается от него лишь в последнем знаке. Согласно свойствам 1 и 2 можно считать, что $l_t = l_{t+1} = \ldots = l_N$. Если $t \neq N-1$, то можно поменять ролями кодовые обозначения a_t и a_{N-1}, не нарушая при этом неравенств (1).

Свойства оптимальный код

Итак, всегда существует такой оптимальный код, в котором кодовые обозначения двух (наименее вероятных) сообщений A_{N-1} и A_N отличаются лишь в последнем символе.

Отмеченное обстоятельство позволяет для решения задачи рассматривать только такие двоичные коды, у которых кодовые обозначения a_{N-1} и a_N для двух наименее вероятных сообщений A_{N-1} и A_N имеют наибольшую длину, отличаясь лишь в последнем символе. Это значит, что концевые вершины a_{N-1} и a_N кодового дерева искомого кода должны быть соединены с одной и той же вершиной a предыдущего «этажа» (см. рис. 12).



Свойства оптимальный код

Рассмотрим новое множество сообщений $A^{(1)}=\{A_1, A_2, \ldots, A_{N-2}, A\}$ с вероятностями $p_1, p_2, \ldots, p_{N-2}, p=p_{N-1}+p_N$. Оно получается из множества $\{A_1, A_2, \ldots, A_{N-2}, A_{N-1}, A_N\}$ объединением двух наименее вероятных сообщений A_{N-1}, A_N в одно сообщение A. Будем говорить, что $A^{(1)}$ получается *сжатием* из $\{A_1, A_2, \ldots, A_{N-2}, A_{N-1}, A_N\}$.

Пусть для $A^{(1)}$ построена некоторая система кодовых обозначений $K^{(1)}=\{a_1, a_2, \ldots, a_{N-2}, a\}$, иными словами, указано некоторое кодовое дерево с концевыми вершинами $a_1, a_2, \ldots, a_{N-2}, a$. Этой системе можно сопоставить код $K=\{a_1, a_2, \ldots, a_{N-2}, a_{N-1}, a_N\}$ для исходного множества сообщений, в котором слова a_{N-1} и a_N получаются из слова

Свойства оптимальный код

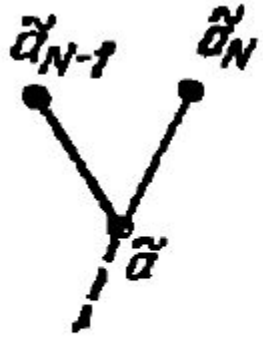


Рис. 13.

a добавлением соответственно 0 и 1. Процедуру перехода от $K^{(1)}$ к K назовем *расщеплением*.

Справедливо следующее утверждение, открывающее путь для построения оптимального кода:

Если код $K^{(1)}$ для множества сообщений $A^{(1)}$ является оптимальным, то оптимален также и код K для исходного множества сообщений.

Для доказательства установим связь между средними длинами $\bar{l}$ и $\bar{l}'$ слов кодов K и $K^{(1)}$. Она, очевидно, такова:

$$\bar{l} = \bar{l}' + p. \qquad (2)$$

Свойства оптимальный код

Предположим, что код K не является оптимальным, т. е. существует код K_1 со средней длиной $\bar{l}_1 < \bar{l}$. Как отмечалось, можно считать, что концевые вершины $\tilde{a}_{N-1}$ и $\tilde{a}_N$ его кодового дерева (см. рис. 13) соответствуют кодовым обозначениям для наименее вероятных сообщений A_{N-1} и A_N. Тогда эти обозначения отличаются лишь в последнем символе. Рассмотрим код $K_1^{(1)} = \{\tilde{a}_1, \ldots, \tilde{a}_{N-2}, \tilde{a}\}$, в котором слово $\tilde{a}$ получается из $\tilde{a}_{N-1}$ и $\tilde{a}_N$ отбрасыванием последнего символа. Средние длины $\bar{l}_1$ и $\bar{l}'_1$ связаны соотношением, аналогичным (2):

$$\bar{l}_1 = \bar{l}'_1 + p.$$

Алгоритм Хаффмана

Из неравенства $\bar{l}_1 < \bar{l}$ следует $\bar{l}'_1 < \bar{l}'$, что противоречит оптимальности кода $K^{(1)}$. Утверждение доказано.

Теперь ясно, что для построения оптимального кода можно использовать последовательные сжатия исходного множества сообщений.

Проиллюстрируем процесс последовательных сжатий и расщеплений на примере множества из пяти сообщений с вероятностями $p_1 = 0{,}4$; $p_2 = p_3 = p_4 = p_5 = 0{,}15$. Процесс этот отражен в следующей таблице:

Алгоритм Хаффмана

Сообщения	Вероятности и кодовые обозначения									
	Исходное множество		Сжатые множества							
			$A^{(1)}$		$A^{(2)}$		$A^{(3)}$			
A_1	0,4	1	0,4	1	0,4	1	→ 0,6	0		
A_2	0,15	010	→ 0,3	00	0,3	00	0,4	1		
A_3	0,15	011	0,15	010	→ 0,3	01				
A_4	0,15	000	0,15	011						
A_5	0,15	001								

Алгоритм Хаффмана

Каждое из множеств $A^{(1)}$, $A^{(2)}$, $A^{(3)}$ получается сжатием предыдущего множества. Множество $A^{(3)}$ состоит из двух сообщений, поэтому оптимальный код $K^{(3)}$ содержит два кодовых обозначения — 0 и 1. Последовательное расщепление $K^{(3)}$ дает оптимальный код для исходной системы сообщений.

Средняя длина $\bar{l}$ кодовых слов, равная $0{,}4+4\times3\times0{,}15=2{,}2$, является, как это следует из предыдущего, минимально возможной для данного множества сообщений.

Описанный метод кодирования был предложен в 1952 г. американским математиком Д. А. Хаффменом и называется его именем. Сравним теперь оптимальный код из таблицы 12 с кодом Фано для того же множества сообщений, который строится ниже.

Алгоритм Хаффмана

1. Доказать, что всякий оптимальный код является полным.