$-0{,}11\log 0{,}11 \approx 0{,}5$ дв. зн./букву, а равномерный код $A \to 1$, $Б \to 0$ (равносильный применению кода Шеннона — Фано к совокупности двух имеющихся букв) требует затраты одного двоичного знака на каждую букву — в два раза больше. Нетрудно проверить, однако, что применение кода Шеннона — Фано к всевозможным двухбуквенным комбинациям здесь приводит к коду, в котором на каждую букву приходится в среднем 0,66 двоичных знаков; применение того же кода к блокам из трех букв позволяет понизить среднее число двоичных знаков, приходящихся на одну букву, до 0,55; наконец, кодирование по методу Шеннона — Фано всевозможных четырехбуквенных блоков требует затраты на каждую букву в среднем 0,52 двоичных знаков — всего на 4% больше минимального значения 0,50 дв. зн./букву.

Близок к коду Шеннона — Фано, но еще выгодней, чем этот последний, так называемый **код Хафмана** (см. [63]), к описанию которого мы сейчас и перейдем. Построение этого кода опирается на простое преобразование того алфавита, на котором записываются передаваемые по линии связи сообщения, называемое *сжатием* алфавита. Путь мы имеем алфавит А, содержащий буквы $a_1, a_2, \ldots, a_n$, вероятности появления которых в сообщении соответственно равны $p_1, p_2, \ldots, p_n$; при этом мы считаем буквы расположенными в порядке убывания их вероятностей (или частот), т. е. полагаем, что

$$p_1 \geqslant p_2 \geqslant p_3 \geqslant \ldots \geqslant p_{n-1} \geqslant p_n.$$

Условимся теперь *не различать между собой две наименее вероятные буквы нашего алфавита*, т. е. будем считать, что a_{n-1} и a_n — это одна и та же буква b нового алфавита $А_1$, содержащего, очевидно, буквы $a_1, a_2, \ldots, a_{n-2}$ и b (т. е. a_{n-1} или a_n), вероятности появления которых в сообщении соответственно равны $p_1, p_2, \ldots, p_{n-2}$ и $p_{n-1} + p_n$. Алфавит $А_1$ и называется *полученным из алфавита* А *с помощью сжатия* (или *однократного сжатия*).

Прилагательное «однократное» в скобках в конце последней фразы имеет следующий смысл. Расположим буквы нового алфавита $А_1$ в порядке убывания их вероятностей и подвергнем сжатию алфавит $А_1$; при этом мы придем к алфавиту $А_2$, про который естественно сказать, что он получается из первоначального алфавита А с помощью *двукратного сжатия* (а из алфавита $А_1$ — с помощью простого или однократного сжатия). Ясно, что алфавит $А_2$ будет содержать уже всего $n - 2$ буквы. Продолжая эту процедуру, мы будем приходить ко все более коротким алфавитам; после $(n-2)$-кратного сжатия мы придем к алфавиту $А_{n-2}$, содержащему уже всего д в е буквы.

Вот, например, как преобразуются с помощью последовательных сжатий рассмотренный выше алфавит, содержащий 6 букв, вероятности которых равны 0,4, 0,2, 0,2, 0,1, 0,05 и 0,05:

№ буквы	Вероятности				
	исходный алфавит А	сжатые алфавиты			
		$А_1$	$А_2$	$А_3$	$А_4$
1	0,4	0,4	0,4	0,4	0,6
2	0,2	0,2	0,2	0,4	0,4
3	0,2	0,2	0,2	0,2	
4	0,1	0,1	0,2		
5	0,05	0,1			
6	0,05				

Условимся теперь приписывать двум буквам последнего алфавита $А_{n-2}$ кодовые обозначения 1 и 0. Далее, если кодовые обозначения уже приписаны всем буквам алфавита $А_j$, то буквам «предыдущего» алфавита $А_{j-1}$ (где, разумеется, $А_{1-1} = А_0$ — это исходный алфавит А), сохранившимся и в алфавите $А_j$, мы припишем т е ж е кодовые обозначения, которые они имели в алфавите $А_{i-1}$; двум же буквам a' и a'' алфавита $А_j$, «слившимся» в одну букву b алфавита $А_{j-1}$, мы припишем обозначения, получающиеся из кодового обозначения буквы b добавлением цифр 1 и 0 в конце (см. таблицу на следующей странице).

Легко видеть, что из самого построения получаемого таким образом *кода Хафмана* вытекает, что он удовлетворяет указанному на стр. 188 общему условию: никакое кодовое обозначение не является здесь началом другого, более длинного кодового обозначения. Заметим еще, что кодирование некоторого алфавита по методу Хафмана (так же, впрочем, как и по методу Шеннона — Фано) не

№ буквы	вероятности и кодовые обозначения				
	исходный алфавит A	сжатые алфавиты			
		A_1	A_2	A_3	A_4
1	0,4 0	0,4 0	0,4 0	0,4 0	0,6 1
2	0,2 10	0,2 10	0,2 10	0,4 11	0,4 0
3	0,2 111	0,2 111	0,2 111	0,2 10	
4	0,1 1101	0,1 1101	0,2 110		
5	0,05 11001	0,1 1100			
6	0,05 11000				

является однозначно определенной процедурой. Так, например, на любом этапе построения кода можно, разумеется, заменить цифру 1 на цифру 0 и наоборот; при этом мы получим два разных кода (отличающихся, правда, весьма несущественно друг от друга и имеющих те же длины всех кодовых обозначений). Но помимо того в некоторых случаях можно построить и несколько существенно различающихся кодов Хафмана; так, например, в разобранном выше примере можно строить код и в соответствии со следующей таблицей:

№ буквы	вероятности и кодовые обозначения				
	исходный алфавит A	сжатые алфавиты			
		A_1	A_2	A_3	A_4
1	0,4 11	0,4 11	0,4 11	0,4 0	0,6 1
2	0,2 01	0,2 01	0,2 10	0,4 11	0,4 0
3	0,2 00	0,2 00	0,2 01	0,2 10	
4	0,1 100	0,1 101	0,2 00		
5	0,05 1011	0,1 100			
6	0,05 1010				

Получаемый при этом новый код также является кодом Хафмана, но длины имеющихся кодовых обозначений теперь уже оказываются совсем другими. Отметим, однако, что *среднее число* элементарных сигналов, приходящихся на одну букву, для обоих построенных кодов Хафмана оказывается точно одинаковым: в первом случае оно равно

$$1\cdot 0{,}4 + 2\cdot 0{,}2 + 3\cdot 0{,}2 + 4\cdot 0{,}1 + 5\cdot (0{,}05 + 0{,}05) = 2{,}3,$$

а во втором — равно

$$2\cdot(0{,}4+0{,}2+0{,}2)+3\cdot 0{,}1+4\cdot(0{,}05+0{,}05)=2{,}3.$$

Далее, оба кода явно относятся к числу весьма экономных (в данном конкретном случае средняя длина кодового обозначения здесь совпадает с той, которая получилась выше при использовании кода Шеннона — Фано). Более того, можно показать, что код Хафмана всегда является самым экономным из всех возможных в том смысле, что *ни для какого другого метода кодирования букв некоторого алфавита среднее число элементарных сигналов, приходящихся на одну букву, не может быть меньше того, какое получается при кодировании по методу Хафмана* (отсюда, разумеется, сразу вытекает и то, что для любых двух кодов Хафмана средняя длина кодового обозначения должна быть точно одинаковой — ведь оба они являются наиболее экономными).

Доказательство этого свойства оптимальности кодов Хафмана совсем несложно. Рассмотрим снова какой-то n-буквенный алфавит (обозначим его, например, через В), содержащий буквы $b_1, b_2, \ldots, b_{n-1}, b_n$, вероятности которых равны $q_1, q_2, \ldots, q_{n-1}, q_n$, где

$$q_1 \geqslant q_2 \geqslant \ldots \geqslant q_{n-1} \geqslant q_n, \qquad (*)$$

и получающийся из него сжатием $(n-1)$-буквенный алфавит (алфавит $В_1$), содержащий буквы $b_1, b_2, \ldots, b_{n-2}, c$, вероятности появления которых соответственно равны $q_1, q_2, \ldots, q_{n-2}, q_{n-1}+q_n=q$. Предположим теперь, что мы имеем какую-то систему кодовых обозначений для букв алфавита $В_1$; эту систему кодовых обозначений мы перенесем затем и в алфавит В, сохранив обозначения всех букв, входящих одновременно в оба алфавита, а буквам b_{n-1} и b_n приписав обозначения, получающиеся из обозначения буквы c с прибавлением в конце соответственно цифр 1 и 0. Покажем теперь, что *если код для алфавита* $В_1$ *был оптимальным, то и полученный таким путем код для алфавита* В *будет оптимальным.*

Выделенное курсивом утверждение мы будем доказывать от противного. А именно, мы предположим, что полученный код для В не является оптимальным, и покажем, что в таком случае *не мог быть оптимальным* и исходный *код для* $В_1$. В самом деле, обозначим среднюю длину кодового обозначения буквы (т. е. среднее число приходящихся на одну букву элементарных сигналов) для рассматриваемых кодов, отвечающих алфавитам $В_1$ и В, через L_1 и L; при этом, очевидно,

$$L = L_1 + q \qquad (**)$$

Действительно, алфавиты $В_1$ и В отличаются лишь тем, что имеющая вероятность q буква c алфавита $В_1$ заменяется в алфавите В двумя буквами b_{n-1} и b_n с той же самой общей вероятностью появления q $(= q_{n-1} + q_n)$; отвечающие же этим алфавитам длины кодовых обозначений отличаются лишь увеличением на единицу длин, отвечающих буквам b_{n-1} и b_n, по сравнению с длиной, отвечающей букве c алфавита B_1. Отсюда и из определения средней длины кодового обозначения сразу следует соотношение (**).

Мы предположили, что отвечающий алфавиту В код н е о п т и м а л е н; другими словами — что существует отличный от рассматриваемого код, сопоставляющий буквам $b_1, b_2, \ldots, b_{n-1}, b_n$ кодовые обозначения длин (в элементарных сигналах) $k_1, k_2, \ldots, k_{n-1}, k_n$, такой, что для него средняя длина кодового обозначения одной буквы

$$L' = k_1 \cdot q_1 + k_2 \cdot q_2 + \ldots + k_{n-1} \cdot q_{n-1} + k_n \cdot q_n$$

меньше L. При этом мы можем считать, что

$$k_1 \leqslant k_2 \leqslant \ldots \leqslant k_{n-1} \leqslant k_n. \qquad (***)$$

В самом деле, если b_i и b_j (где i и j — какие-то два из номеров 1, 2, ..., n) — такие буквы, что $q_i < q_j$ (откуда в силу (*) следует неравенство $i > j$), а $k_i < k_j$, то мы просто поменяем кодовые обозначения букв b_i и b_j, после чего средняя длина кодового обозначения буквы еще уменьшится; поэтому если $q_i > q_j$, то обязательно $k_i \leqslant k_j$. Ну а в пределах группы букв $b_u, b_{u+1}, \ldots, b_v$ (где $1 \leqslant u < v \leqslant n$) такой, что $q_u = q_{u+1} = \ldots = q_v$, мы всегда можем расположить буквы в таком порядке, что $k_u \leqslant k_{u+1} \leqslant \ldots \leqslant k_v$.

Из неравенств (***), в частности, следует, что букве b_n отвечает кодовое обозначение, имеющее с а м у ю б о л ь ш у ю длину k_n. Далее, мы можем быть уверены в существовании такой буквы b_l алфавита В, кодовое обозначение которой *получается из кодового обозначения буквы* b_n *заменой последнего элементарного сигнала* — 1 на 0 или 0 на 1. В самом деле, если бы такое кодовое обозначение вовсе отсутствовало, то мы могли бы просто откинуть последний элементарный сигнал в кодовом обозначении буквы b_n, не придя при этом в противоречие с основным условием, определяющим коды без разделительного знака (напомним, что букв, имеющих более длинные, чем b_n, кодовые обозначения, у нас нет). Но при этом мы снова уменьшили бы среднюю длину кодового обозначения одной буквы, что противоречит предположению об оптимальности рассматриваемого кода.

Но из неравенств (***) и равенства $k_l = k_n$ следует, что неизбежно $k_l = k_{n-1}$ (но при этом не обязательно $l = n - 1$). Поменяем теперь кодовые обозначения букв b_l и b_{n-1}, если $l \neq n - 1$ (если $l = n - 1$, то этот этап рассуждения является лишним); при этом величина L', очевидно, не изменится. А теперь перейдем от рассматриваемого кода для алфавита В к коду для алфавита B_1, сохранив кодовые обозначения всех букв $b_1, b_2, \ldots, b_{n-2}$, а букве c приписав кодовое обозначение, получающееся из кодовых обозначений букв b_{n-1} и b_n отбрасыванием последней цифры (которой эти кодовые обозначения лишь и отличаются). Очевидно, что средняя длина L_1' полученного таким путем кода для алфавита B_1 связана со средней длиной L' кода для B аналогичным (**) соотношением

$$L' = L_1' + q,$$

откуда, в силу неравенства $L' < L$, следует, что

$$L_1' < L_1.$$

Но это и доказывает, что исходный код для B_1 *не был оптимальным.*

Мы, по существу, уже завершили доказательство оптимальности кодов Хафмана. Действительно, ясно, что принятый нами код для последнего алфавита A_{n-2}, приписывающий двум буквам, из которых этот алфавит состоит, кодовые обозначения 1 и 0, является оптимальным: отвечающая ему средняя длина 1 кодового обозначения буквы никак не может быть уменьшена. Но отсюда, в силу только что доказанного, следует, что и код для алфавита A_{n-3} является оптимальным, откуда, в свою очередь, вытекает оптимальность кода для алфавита A_{n-4} и т. д.— и так до последнего кода (кода Хафмана), отвечающего исходному алфавиту $A_{1-1} = A_0$, т. е. алфавиту A.

Достигнутая в рассмотренных выше примерах степень близости среднего числа двоичных знаков, приходящихся на одну букву сообщения, к значению H может быть еще сколь угодно увеличена при помощи перехода к кодированию все более и более длинных блоков. Это вытекает из следующего общего утверждения, которое мы будем в дальнейшем называть основной теоремой о кодировании[1]): *при кодировании сообщения, разбитого на N-буквенные блоки, можно, выбрав N достаточно большим, добиться того, чтобы среднее число двоичных элементарных сигналов, приходящихся на одну букву исходного сообщения, было сколь угодно близко к H* (другими словами — сколь угодно близко к отношению количества H информации, содержащейся в одной букве сообщения, к 1 биту, т. е. к наибольшему количеству информации, могущему содержаться в одном элементарном сигнале). Иначе это можно сформулировать еще так: *очень длинное сообщение из M букв может быть закодировано при помощи сколь угодно близкого к MH* (но, разумеется, ни в каком случае не меньшего!) *числа элементарных сигналов, если только предварительно разбить это сообщение на*

---

[1]) Точнее следовало бы сказать: основной теоремой о кодировании при отсутствии помех. Обобщение этого результата на случай наиболее выгодного кодирования, учитывающего влияние помех, будет рассмотрено в § 4.